Araştırma Makalesi
Bayburt Üniversitesi Fen Bilimleri Dergisi
Cilt 2, Sayı 2

Geliş Tarihi: 30.11.2019
Kabul Tarihi: 11.12.2019

dergipark.gov.tr/bufbd

A Research on Cyber Security: Software Security

Mustafa OF*

Kocaeli Üniversitesi, Kocaeli Meslek Yüksekokulu, Bilgisayar Teknolojisi, Kocaeli, Türkiye

Keywords:

Software Security, Internet Communication Protocol, Software Assurance Maturity Models

Abstract

The foundation of the Internet was developed by the U. S. A. in the late 1960s. The core is the ARPANET network. This network was developed by ARPA (Advensed Research Project Agency). Its purpose is to have a durable and secure military network. In the 1980s, the budget was allocated for the development of this network, and each day it grew to form the current Internet network. Because the Internet is a computer network, a network protocol is needed for interconnected devices to communicate. The name of this protocol is TCP / IP (Transmission Control Protocol / Internet Protocol). In 1983 it became available. It has several protocols. Network protocols are similar to foreign languages used people. Devices connected to the network must have the same network protocol. This protocol, which is the basis of the Internet, has many security weakness. The number of Internet users in 2018 exceeded 7 billion (Hootsuite We Are Social, 2019). Such a large number of users have exposed a growing vulnerability. Since the day smartphones entered our lives, the vulnerabilities have grown even bigger. As a result of the weakness of security of the Internet protocol and users have little security information, the Internet has become an insecure network. It can be attacked at any time in a public network. Necessary measures should be taken to protect against cyber attacks.

In this study, items that need to be considered in software security for software developers and users will be explained. Software is a passing application that allows users to meet their wishes. This application needs to be developed and presented more safely. How can cyberattacks resulting from the use of software be tackled? The answer to this question will be investigated.

Siber Güvenlik Üzerine Bir Araştırma: Yazılım Güvenliği

Anahtar Kelimeler:

Yazılım Güvenliği, İnternet İletişim Protokolü, Yazılım Güvenliği Olgunluk Modelleri

Özet

İnternetin temeli 1960 yılının sonlarına doğru Amerika Birleşik Devletleri tarafından atılmıştır. Çekirdeği ARPANET ağıdır. Bu ağ, ARPA (Advensed Research Project Agency) adlı ajans tarafından geliştirilmiştir. Amacı, dayanıklı ve güvenli bir askeri ağ olmasıdır. 1980'li yıllarda bu ağın gelişmesi için bütçe ayrılmış ve her geçen gün büyüyerek şimdiki İnternet ağını meydana getirmiştir. İnternetin bir bilgisayar ağı olmasından dolayı birbirlerine bağlı cihazların iletişime girebilmeleri için bir ağ iletişim kuralına ihtiyaç vardır. Bu iletişim kuralının adı TCP/IP'dir (Transmission Control Protocol/Internet Protocol). 1983 yılında kullanılabilir hale gelmiştir. Çeşitli protokollere sahiptir. Ağ iletişim kuralları, insanlar arasında kullanılan yabancı lisanlara benzemektedir. Ağa bağlı aygıtların da aynı ağ iletişim kuralına sahip olmaları gerekmektedir. İnternetin temeli olan bu iletişim kuralı birçok güvenlik açığını da beraberinde getirmiştir. 2018 yılındaki İnternet kullanıcı sayısı 7 milyarı aşmıştır (Hootsuite We Are Social, 2019). Bu kadar büyük bir kullanıcı sayısı her geçen gün büyüyen bir güvenlik açığını ortaya çıkartmıştır. Akıllı telefonların hayatımıza girdiği günden bu yana güvenlik açıkları daha da büyümüştür. İnternet iletişim kuralının çok güvenli olmaması ve kullanıcıların da güvenlik konusunda gerekli dikkate sahip bulunmamaları sonucunda İnternet, güvensiz bir ağ haline gelmiştir. Herkese açık bir ağda her an

*e-Posta: mustafaof@kocaeli.edu.tr

saldırıya maruz kalınabilir. Siber saldırılardan korunmak için temel tedbirlerin alınması gereklidir.
Bu çalışmada, yazılım geliştiricilerine ve kullanıcılara hitaben yazılım güvenliğinde dikkat edilmesi gerekli maddeler açıklanacaktır. Yazılımlar, kullanıcıların istekleri ile buluşmasını sağlayan bir aracı uygulamadır. Bu aracın daha güvenli olarak geliştirilmesi ve sunulması gerekmektedir. Yazılımların kullanımlarından ortaya çıkan siber saldırılarla nasıl mücadele edilebilir? Sorusunun cevabı aranacaktır.

1 GİRİŞ

Bilişim dünyası oldukça hızlı gelişmekte ve hepimizi çok geniş bir alanda kontrolü altına almaktadır. Büyük veri (Big Data) adıyla anılan veri topluluğu gün geçtikçe büyümektedir. Bu veriler, bize ait bilgilerdir. Verilerin kötü niyetli kişi veya kurumların eline geçmesiyle tehlikenin boyutu ciddi derecede artmaktadır. Bilişim dünyasındaki tüm kontrolün bir ya da birkaç kurumun kontrolüne geçmesi ile üstünlük savaşları gittikçe kızışmaktadır. Bu savaşlar, ülkeler arası stratejileri de etkilemektedir. Bulut teknolojisi adı altında verilerin, onların sunucularında tutulması sayesinde dünyadaki birçok kişinin projesi veya önemli bilgileri tereyağından kıl çeker gibi ellerine geçmiş durumdadır. Big Data içerisindeki bilgiler içerisinden anlamlı olanları veya işe yarayanları çekip alabilecek yapay zekâ destekli yazılımların gece gündüz çalıştırılması da dikkate şayan bir durum olarak karşımıza çıkmaktadır. Bulut teknolojisi öncüleri için Google Drive, Microsoft OneDrive örneklerini sayabiliriz.

Bilginin ne denli önemli olduğuna dair hatırlatmadan sonra bilginin güvenliği mefhumunu irdelemenin önemini belirtmemiz gerekmektedir. Önemli bilgilerin rızamız dışında birilerinin eline geçmesi oldukça ciddi bir durumdur. 2006 yılında kurduğu Wikileaks adlı web sitesi üzerinden gizli bilgileri yayınlayan Julian Assange adlı Avustralya'lı bilgisayar programcısı, ülkeler arasındaki dengeleri değiştirecek adımlar atmıştır. Şu an Londra'da tutulan Assange, değerli verilerin korunması kavramının ne derece önemli olduğunu bizlere anlatmaya yarayan örneklerden sadece biridir.

A.B.D. merkezli bir uluslararası market araştırma şirketi olan International Data Corporation'a (http://www.idc.com) göre dünya genelinde 2016 yılında yaklaşık olarak 74 milyar dolar siber güvenlik harcaması gerçekleştirilmiş. Bu oranın 2020 yılında %40'a yakın bir artışa ulaşacağı tahmin edilmektedir. Siber saldırıların en çok yapıldığı 10 ülkeden biri de Türkiye'dir. Özellikle enerji sektörü, etkiledikleri ortam bakımından ilk sıraları almaktadır. Bu açığı gören sigortacılık hizmeti veren firmalar, siber güvenlik paketlerini müşterilerine sunmaya başlamışlardır. Çünkü siber güvenlik sorunları, büyük yıkım yapabilecek riskler arasındadır.

Bu çalışmada, bilgilerinin kontrolünü sağlayan yazılımlar ile veri alışverişinin güvenliği üzerinde durulacaktır. Gerekli olan tedbirlerin neler olduğu anlatılmaya çalışılacaktır. Yazılım güvenliği seviyesinin arttırmak için hangi tedbirlere başvurulacağı izah edilecektir. Önü alınamaz bir şekilde bilgilerin dijital ortama aktarılması aşamasında bilinmesi gereken çok önemli adımlar açıklanacaktır.

2 MATERYAL VE METOD

Güvenlik açığını meydana gelmesine sebep olabilecek açıkların öncelikle bilinmesi gereklidir. Şu ana kadar belirlenmiş güvenlik zafiyetleri ortaya çıkartılmıştır. Risk seviyesine göre sıralanmıştır. Bu sıralamayı bilgi güvenliği şirketleri veya gönüllü örgütler yapmaktadır. Özellikle OWASP (Open Web Application Security Project) bu örgütlerden biridir. OWASP, web üzerinden çalışabilen uygulamaların daha güvenli bir şekilde geliştirilmesini standart hale getirmeye çalışan kâr amacı gütmeyen bir kuruluştur.

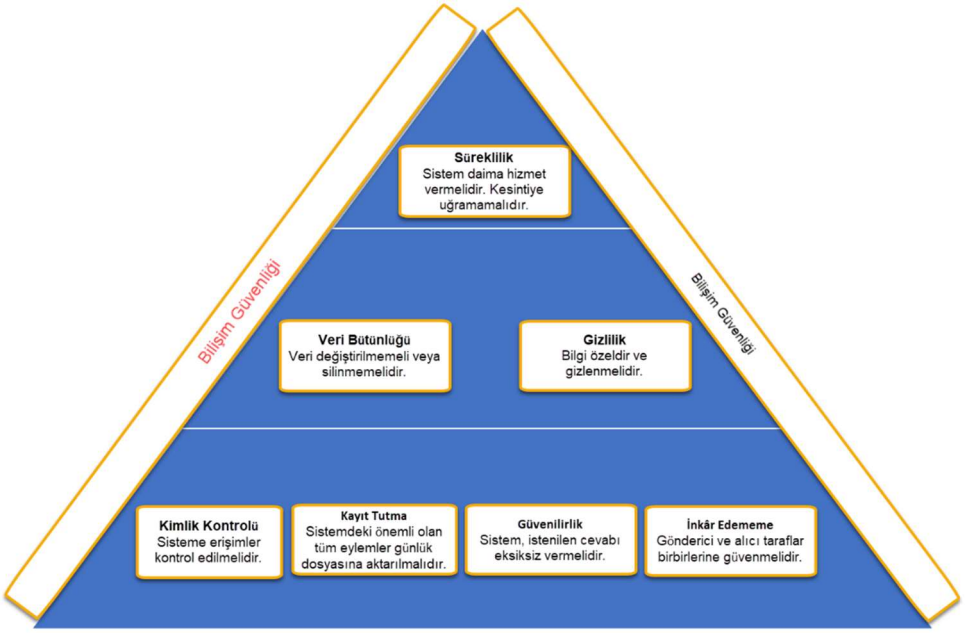
Yazılımların büyük bir çoğunluğu web üzerinden veya İnternet üzerinden çalışmaktadır. Her yerden erişim, paylaşma vb. artıları, uygulamaların masaüstü uygulaması formundan İnternet formuna geçmesini sağlamıştır. Kolaylıkları ile birlikte zararlarını da beraberinde getirmiştir. Yazılım geliştiricilerinin güvenlik zafiyetlerine göstermiş oldukları düşük ilgi yüzünden geliştiricilerin uygulamaların yaptıkları işlere daha çok zaman harcamasını ve bu çalışmalara yönelik geliştirmesini doğurmuştur. Yazılımlara daha az güvenlik kontrolleri eklenmesinin sonucunda uygulamaların güvenlik açıkları artmıştır. Büyüyen bir yazılım geliştirmek oldukça zahmetli bir iştir. 4 satırdan binlerce satıra ulaşan kodlar, kontrol edilmesi zor ve zahmetli bir duruma geleceklerdir. Bu yüzden yazılımların geliştirilmesinde özellikle güvenlik durumunu kontrol edilebilecek bir yapıya kavuşturacak modeller kullanılmalıdır. Yazılım güvenliği olgunluk modeli bu modellerden biridir. Bununla beraber geliştiricilere ve kullanıcılara yönelik belli başlı temel uyarılar ortaya çıkartılmıştır.

2.1 Yazılım Güvenliği

Yazılım güvenliği, yazılımın kendinden kaynaklı veya farklı kullanımı esnasında her kullanıcının karşısına çıkabileceği haberli veya habersiz ataklara karşı durabilmek için sağlanan önlemlerin bütününe atfedilen isimdir. Güvenlik kavramı ile ilgili başvurulan her türlü tedbirde uygulanan kurallar topluluğudur.

Bilişim güvenliği, aşağıdaki temel standartlardan oluşmaktadır;

- Süreklilik
- Veri Bütünlüğü
- Gizlilik



Şekil 1. Bilişim güvenliğinin temel prensipleri

Bahsi geçen güvenlik aşamalarına tam olarak uyan yazılımların geliştirilmesi gereklidir. Fakat her yazılım, geliştiricisi tarafından tam anlamıyla bu standartlara uymamaktadır. Tahmin edilmesi gereken durumun çeşitliliği yazılım geliştiricilerini en çok zorlayan süreçlerden birisidir. Bu yüzden yazılımlar, belirli dönemlerde güncelleme yaparak eksik taraflarını kapatmaktadırlar. İşletim sistemleri, bu konuda en fazla açıkları olan yazılımlar topluluğudur. Sebebi, hüküm sürdükleri alanın çok geniş olmasından kaynaklanmaktadır. Yazılımın hüküm sürdüğü alan ne kadar büyürse o derecede güvensizliği ve zafiyeti artmaktadır. Bir yazılımda kontrolü geliştiricisinin elinde olmayan birbirinden bağımsız farklı yazılımlar bulunmaktadır. Bunlardan biri de veri tabanı yönetim sistemleri yazılımlarıdır. Güvenlik zafiyetleri oldukça fazladır. Verilerin merkezi olarak yönetilmesini sağlayan veri tabanı yönetim sistemleri yazılımların birçoğunda bulunmaktadır. Bunun gibi daha birçok farklı yardımcı yazılımlar örnek verilebilir. Tamamen saf ve her aşaması geliştiricisi tarafından ortaya çıkartılan yazılım sayısı oldukça azdır.

2.2 Yazılım Güvenliği Olgunluk Modelleri

İnternet'in yaygınlaşması ile yazılım güvenliği sorunlarının artmaya başlamıştır. Bu sorunların giderilmesi için bir yol haritasının kullanılması zorunluluğu ortaya çıkmıştır. Sonuç olarak yazılım olgunluk modeli adı verilen bir model geliştirilmiştir.

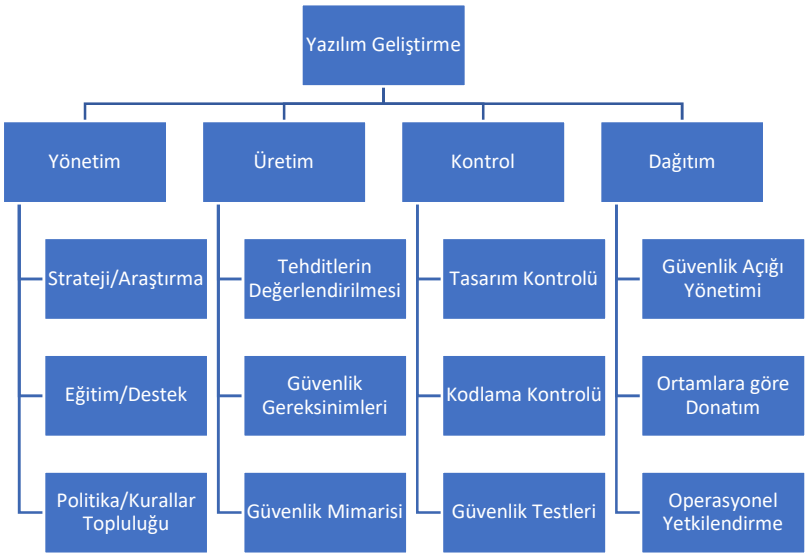
Yazılım güvenliği olgunluk modelleri, 2008 yılında bir yol haritasına ile sahneye çıkmıştır. Bu modeller; BSIMM (Building Security In Maturity Model) ve OpenSAMM (Open Software Assurance Maturity Model) adı altında geliştirilmiştir. Daha detaylı bilgi, https://www.opensamm.org adresinde bulunmaktadır. Yazılım güvenliğine yeni başlamış, farklı boyutlardaki işletmeler bu standartlar yardımı ile yazılım geliştirme aşamalarını daha güvenli hale getirebilir. Emekleme aşamasında olan bu modeller zamanla daha sağlam bir şekilde yerlerine oturacaklardır. OpenSAMM, bir OWASP (Open Web Application Security Project) projesidir.

Yazılım olgunluk modeli ile işletmeler;

- Geliştirdikleri yazılımların, modelde hangi aşamaya denk geldiğini görecekler ve düzeltmelerini yapacaklardır.
- Gözden kaçan veya hiç akla gelmeyecek eksiklikleri bu yol haritası sayesinde görebileceklerdir.
- Belirlenen eksikliklerin giderilmesine yönelik tedbirlerin alınma aşamaları belirlenecektir.

Bu sayede, daha güvenli yazılımlar geliştirilebilecektir.
SAMM modeli temel olarak dört aşamadan oluşmaktadır.

1. Yönetim
2. Üretim
3. Kontrol
4. Dağıtım
5.



Şekil 2. SAMM uygulama aşamaları

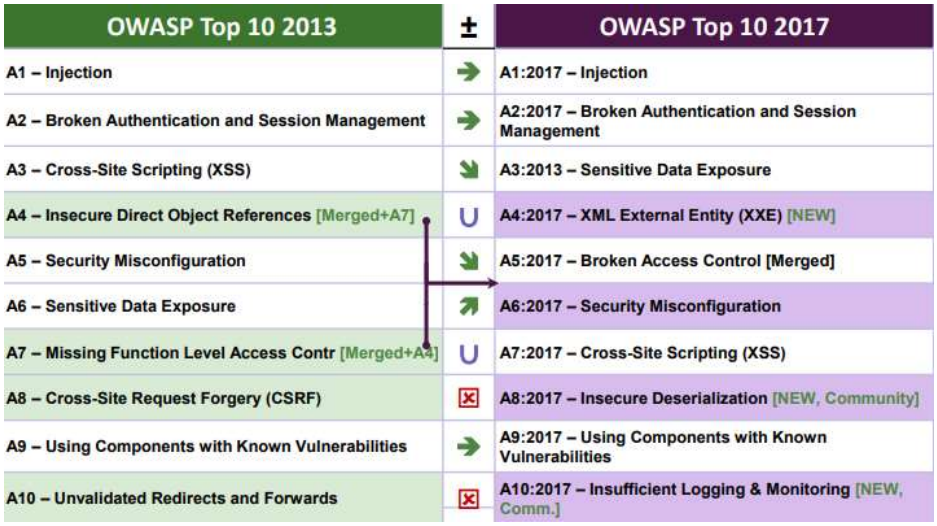
3 KARŞILAŞILAN TEHDİT TÜRLERİ

OWASP (Open Web Application Security Project), 2017 yılında en kritik olarak 10 adet web uygulamalarına ait güvenlik zafiyetlerini belirlemiştir.

1. **Injection**
   Kullanıcıdan alınan verileri çeşitli şekillerde değiştirmeye dayalı tehditlerdir. Sql Injection, Xml Injection, Code Injection, bunlardan sadece birkaçıdır.
2. **Broken Authentication and Session Management**
   Oturumun kontrol dışı olarak yönetilmesinden kaynaklı tehditlerdir. Oturum sabitleme (Session Fixation) ve oturumu tahmin etme (Session Prediction), olarak örnek verilebilir.
3. **Sensitive Data Exposure**
   Hassas ve önemli olan verilere erişimin çok kolay olması sonucunda oluşan tehditlerdir. Veri tabanı yönetim sistemine erişim yetkisinin birçok kullanıcıda olması örnek olarak verilebilir.
4. **Xml External Entity (XEE)**
   XML dış varlık enjeksiyonu (XXE), bir saldırganın bir uygulamanın XML verilerini işlemesine müdahale etmesine izin veren bir web güvenlik açığıdır. Genellikle bir saldırganın uygulama sunucusu dosya sistemindeki dosyaları görüntülemesine ve uygulamanın kendisinin erişebileceği herhangi bir arka uç veya harici sistemle etkileşime girmesine izin verir.
5. **Broken Access Control**
   Bazen yetkilendirme açığı olarak adlandırılan erişim kontrolü, bir web uygulamasının içeriğe erişim sağlamasına izin vermesidir. Geliştiricilerin güvenlik erişimlerine yeteri kadar önem vermemesi sonucunda ortaya çıkmaktadır. Güvenlik erişimlerinin daha sıkı kontrollerden geçmesi sonucunda bu açığın etkisi azalacaktır.
6. **Security Misconfiguration**
   Çalışan sistemlerin genel ayarlarının ilk başlangıç halinde bırakılması veya hatalı olarak düzenlenmesi sonucunda ortaya çıkan tehditlerdir. Apache veya IIS web sunucu yazılımlarının genel ayarlarının organizasyona veya işletmeye göre değiştirilmemesi veya hatalı olarak düzenlenmesi örneği verilebilir.

7. **Cross-Site Scripting (XSS)**
   Kullanıcının web tarayıcısında JavaScript kodlarını çalıştırmaya imkân sağlayan tehditlerdir. Reflected, Stored ve DOM türleri bulunmaktadır.
8. **Insecure Deserialization**
   Güvensizlik Serileştirme, bir uygulamanın mantığını kötüye kullanmak, hizmet reddi (DoS) saldırısı uygulamak veya seri hale getirildikten sonra isteğe bağlı bir kod yürütmek için güvenilmeyen veriler kullanıldığında ortaya çıkan bir güvenlik açığıdır.
9. **Using Components with Knows Vulnerabilities**
   Kullanılan yazılıma eklenmiş farklı kişiler tarafından yazılmış olan eklentilerden ortaya çıkan bir tehdit türüdür. Hazır içerik sistemleri (WordPress, Joomla vb.) içerisine eklenen farklı amaçlara yönelik eklentiler (Plug in) örnek olarak verilebilir.
10. **Insufficient Logging & Monitoring**
    Yetersiz Kayıt ve İzleme yukarıdaki risklerden biraz farklıdır. Doğrudan bir saldırıya yol açmasa da, bu risk, saldırıya ait bilgileri zamanında tespit edemeyecektir. Atağa ait bilgiler ne kadar geç elde edilirse o kadar uzun süre sistem devre dışı kalacaktır. Sonuç olarak maddi ve manevi olarak büyük zararlar ortaya çıkacaktır.

Güvenlik zafiyetleri, dönemine bağlı olarak değişmektedir. İşletim sistemleri, uygulamaların çalışması için gerekli alt yapıyı sağlayan yazılım topluluklarıdır. Yazılım, sürekli büyüyen bir yapıya sahiptir. İhtiyaca bağlı olarak bir güncelleme süreci içerisindedir. Bu yüzden her dönem farklı güvenlik zafiyetleri ortaya çıkabilir. İnternetin yaygın olmadığı yıllarda dosyalara imzalarını ekleyen ve bellekte sürekli çalışarak çalışabilen uygulamalara kendini ekleyen dosya virüsleri bir zamanlar oldukça etkindi. Diskin ilk yükleme alanına yerleşen Boot virüsleri de bir dönem bilgisayar dünyasını meşgul etmiştir. 1998 yılında ortaya çıkan CIH (Çernobil Virüsü), ana kartların Eprom (Erasable Programmable Read Only Memory) hafızasına veri yazarak donanıma zarar veren ilk virüs olarak tarihe geçmesi unutulmaz bir durumdur. İnternetin yaygın olarak kullanıldığı günümüzde ise zararlıların listesi daha da farklı olacaktır. Aşağıda OWASP'ın 2013 ve 2017 yılındaki zararlılar karşılaştırması görünmektedir.



Şekil 3. OWASP 2013 ve 2017 yılı karşılaştırması (Kaynak: https://www.owasp.org/images/b/b0/OWASP_Top_10_2017_RC2_Final.pdf)

3.1 Güvenlik Seviyesini Arttırma Sürecinde Dikkat Edilmesi Gerekli Noktalar

İhtiyaçlar, ortam vb. durumlardan dolayı yazılım güvenliği seviyesi ne yazık ki en üst derecelere çıkarılamamaktadır. Fakat kullanıcı ve geliştiriciler olarak bu tehditler hakkında temel bilgilere sahip olunursa daha güvenli bir erişim ortaya çıkacaktır. Hem kullanıcıların hem de geliştiricilerin temel güvenlik bilgilerini bilmeleri gerekmektedir.

Aşağıda yazılım geliştiricilerine yönelik temel güvenlik tedbirleri önerileri bulunmaktadır.

1. Veri tabanları, tüm verilerin yönetildiği yapılardır. Bu yapılar, SQL (Structered Query Language) dili adı verilen bir dil ile kontrol edilirler. Sql diline ait zafiyetlerden dolayı önemli kullanıcı bilgileri sızdırılabilir. Yetkili kullanıcıların isimleri, admin vb. bilinen isimler olmamalı ve şifreler veri tabanında MD5 gibi güçlü şifreleme teknikleri ile yazılmalı. Ayrıca kullanıcıların verdikleri şifre kuralları katı olmalı. Örneğin büyük harf, sayı, noktalama işaretleri içermeli, en az 8 karakter olmalı.

2. Oturum (Session) yönetimi esnasında ortaya çıkabilecek açıklar belirlenmeli. Gerekli tedbirler alınmalı. Geliştirilen programlama dili veya ortamına ait oturum yönetimi hakkında ayrıntılı bilgiler elde edilmeli. Sayfalar arasındaki verilerin şifreli bir şekilde iletilebilmesi sağlanmalı. Bu sayede kötü niyetli kullanıcıların oturuma sızmaları bir nebze de olsa engellenebilir.
3. Web tarayıcısı istemci yani kullanıcı tarafındaki yazılımdır. Web tarayıcılarını yönetmek için geliştirilen JavaScript ile ilgili açıklar tespit edilmeli. Gerekli tedbirler, sayfa içerisinde alınmalı. JavaScript kodları şifrelenerek gizlenmeli. Tarayıcının dili olan HTML çıktıları yerine düz metin çıktıları kullanmalı. Bu sayede yorumlanması engellenmiş olacaktır.
4. Üyelerin denetiminde olan sayfaların erişimlerinin başka kullanıcılar tarafından kontrolünü en aza indirmek için gerekli tedbirler alınmalı. Bunların başında, oturum yönetimi ile gerçek üyenin giriş yapıp yapmadığı kontrolü gelir. İlave olarak E-Posta üzerinden gerçek kişinin kontrolü sağlanabilir. Daha güvenli bir erişim gerekli ise SMS üzerinden kontrol da sağlanabilir.
5. Sistemlerin ilk varsayılan ayarlarına her zaman güvenilmemeli. İşletmeye veya organizasyona göre sistem yapılandırmaları yapılmalı. Apache, en çok kullanılan bir web sunucu yazılımıdır. httpd.conf dosyası içerisindeki ayarlamalar tekrar gözden geçirilmeli. Ayrıca Php ayarlarını koordine eden php.ini dosyası kontrol edilmelidir. Bu işlemleri yaparken yanlış ayarlamalardan uzak durmalı.
6. Önemli verilerin çok kolay ulaşılıp değiştirilmesi engellenmeli. Örneğin bir uygulamanın genel ayarlamaları bir kullanıcıya bağlanmamalı. Birden fazla onay işleminden geçerek değişikliğe izin verilmeli. Çoklu yetki ile bu durum sağlanabilir. Örneğin en az 3 kullanıcıdan E-Posta veya SMS onayını alarak değişiklik uygulanabilir.
7. Uygulamalara başlangıçta veya sonradan dahil edilen üçüncü parti yazılımlara dikkat edilmeli. Saldırıların büyük çoğunluğu buralardan gelmektedir. İçerik yönetim sistemlerinden (Content Management System) biri olan WordPress, kolay kurulumu ve kullanımı sayesinde çok tercih edilmektedir. Fakat bilgisiz yöneticiler, ihtiyaca göre güvensiz Wordpress eklentilerini kurarak sistemi güvensiz hale getirebilirler. Farkında olmadan web siteniz bu güvensiz işlemlerin sonucunda dünya çapında çalışan kredi kartı şifresini yayan bir grubun köle yazılımı olabilir.
8. Site içerisinde yapılan farklı web adreslerine yönlendirmelerin kontrol dışına çıkması engellenmeli. Yönlendiren sayfa ve mümkünse yönlendirilmiş sayfada kontroller yapılmalıdır. Eğer bu kontroller yeterli değilse web siteniz, illegal bir örgütün web sayfasına yönlendirme yapabilir. Özellikle kişilere açık ziyaret defteri, yorum vb. sayfalarda kod yorumlama engeli bulunması çok önemlidir. Php ile sayfayı geliştiriyorsanız htmlspecialchars fonksiyonu ile yorumlanabilen kodları düz metin haline getirmeniz gereklidir.
   Örnek:
   <?php
   $metin = "Merhaba Dünya";
   echo htmlspecialchars($metin);
   ?>
   Yukarıdaki kodların sonucunda metin şöyle yazılacaktır;
   Merhaba Dünya
   Fakat ifade şöyle kullanılırsa
   echo $metin;
   Aşağıdaki sonuç çıkacaktır.
   Merhaba **Dünya**
9. Sorgu string'leri (Query String) ile yapılan istek sayfalarında istek sahteciliğine yönelik tedbirler alınmalı. Değişkenler, istenilenin dışında bir değer alıp almayacağı kontrol edilmelidir.
   http://www.siteadresi.com/?istekno=5
   istekno değişkeni için farklı değerlerin girilmemesi için yazılım içerisinde gerekli kontroller yapılmalıdır.
   Ayrıca veri tabanına veri ekleme vb. işlemler, Form POST istek türü ile şifrelenerek yapılması daha güvenli bir ortam oluşturacaktır.
10. Sistemleri kullanan yöneticilerin şifre vb. bilgileri Google Drive, Microsoft OneDrive vb. bulut sistemlerinde bulunmamalı. Sadece kendilerine açık daha güvenli ortamlarda korunmalıdır.

4. SONUÇLAR

Yazılım geliştirmek uzun soluklu ve geniş bir alt yapıya dayalı bir süreçtir. Bu süreçte sistemli çalışmak oldukça önemlidir. Genelde tek veya birkaç kişi ile geliştirilen yazılımlarda güvenlik aşamalarına çok fazla başvurulmamaktadır. Bunun sonucunda güvenlik zafiyet düşük olan yazılımlar ortaya çıkacaktır. Daha kurumsal olan yazılım firmalarında sistemli çalışma konusu önemli bir yer almaktadır. Güvenli yazılımlar geleceğimizi etkilerler. Her yazılım bir büyüyen bir tohum gibidir. Büyüdükçe etrafına faydalar verir. Duruma göre olgunlaşır.

Yazılım geliştiricilere mutlaka tehditlere yönelik eğitimlerin verilmesi gereklidir. Temel güvenlik bilgilerine sahip olmaları sağlanmalıdır.

Ülkemizde özellikle son yıllarda bilişim güvenliği konusunda eğitim almış ve uygulama becerisi olan uzman sayısına olan ihtiyaç ciddi derecede artmıştır. Bu açığın ivedilikle kapatılması gereklidir. Çözüm, eğitimden geçmektedir. Liselerde ve üniversitelerin Ön Lisans derecelerinde "Bilişim Güvenliği" adı ile bölümler açılmalı ve vatandaşları bu konuya karşı duyarlı olmaya çağırmak gereklidir. Televizyonlarda veya İnternet sitelerinde kamu spotu başlığı altında bir farkındalık oluşturulmalıdır.

Not

Bu makale 01-03 Kasım 2019 tarihleri arasında Kocaeli'de gerçekleştirilen Uluslararası Marmara Fen Bilimleri Kongresinde (IMASCON 2019) sözlü bildiri olarak sunulmuş ve yeniden yapılandırılmıştır.

Kaynakça

[1] ALTINKAYNAK M., (2017). Uygulamalı Siber Güvenlik ve Hacking, Abaküs Yayınları, İstanbul

[2] DEMİR B., (2013). Yazılım Güvenliği Saldırı ve Savunma, Dikeyeksen Yayınları, İstanbul

[3] AKTAŞ O., Siber güvenlik: Hacking atölyesi, Gazi Kitabevi, Ankara, 2017

[4] OF M., Bilişim Güvenliği Politikalarının Temel İpuçları, Enscon 18 Bahar Sempozyumu, http://enscon.org/pdf/ensontammetinbildirikitabi.pdf, 2018

[5] OpenSAMM, http://www.cs.unh.edu/~it666/reading_list/SDLC/opensamm_1.0.pdf, [Online]. (Erişim Tarihi: 05.06.2019)

[6] OpenSAMM, https://www.opensamm.org/, [Online]. (Erişim Tarihi: 05.06.2019)

[7] DOM Tabanlı Cross Site Scripting (XSS) Zafiyeti, [Online]. https://www.netsparker.com.tr/blog/web-guvenligi/dom-tabanli-cross-site-scripting-xss-zafiyeti/, [Online]. (Erişim Tarihi: 05.06.2019)

[8] International Data Corporation (IDC), https://www.idc.com/about, [Online]. (Erişim Tarihi: 05.06.2019)

[7] Julian Assange kimdir? https://www.bbc.com/turkce/haberler-dunya-39975457, [Online]. (Erişim Tarihi: 05.06.2019)

[9] Siber saldırı ve doğal afetler enerji sektörü için risk oluşturuyor, http://www.konhaber.com/haber-siber_saldiri_ve_dogal_afetler_enerji_sektoru_icin_risk_olusturuyor-782802.html, [Online]. (Erişim Tarihi: 08.06.2019)

[10] PHP htmlspecialchars() Fonksiyonu,https://www.w3schools.com/php/func_string_htmlspecialchars.asp, [Online]. (Erişim Tarihi: 08.06.2019)

[11] Dijital in 2019, https://wearesocial.com/global-digital-report-2019, [Online]. (Erişim Tarihi: 10.11.2019)

[12] BÜLBÜL, İ. Yeni Bir Eğitim Ortamı Olarak İnternet, https://dergipark.org.tr/tr/download/article-file/296658, [Online]. (Erişim Tarihi: 20.11.2019)

[13] OWASP Top 10: Insufficient Logging & Monitoring Security Vulnerability Practical Overview, https://www.immuniweb.com/blog/OWASP-insufficient-logging-and-monitoring.html, 2018, [Online]. (Erişim Tarihi: 21.11.2019)

[14] Top 10-2017 A8-Insecure Deserialization, https://www.owasp.org/index.php/Top_10-2017_A8-Insecure_Deserialization, [Online]. (Erişim Tarihi: 21.11.2019)

[15] What is Insecure Deserialization?, https://www.acunetix.com/blog/articles/what-is-insecure-deserialization/, 2017, [Online]. (Erişim Tarihi: 21.11.2019)

[16] OWASP 2013 ve 2017 yılı karşılaştırması, https://www.owasp.org/images/b/b0/OWASP_Top_10_2017_RC2_Final.pdf), 2017, [Online]. (Erişim Tarihi: 19.11.2019)