Yazılım Güvenlik Açıklıklarının Analizi İle Olası Zafiyet Öngörüsü

Oğuz BOZOKLU, Celal Zaim ÇİL, Şeref SAĞIROĞLU

***Özet*–Yazılımlar, güvenliklerinin tam manasıyla sağlanamaması ve güvenilirliklerinde hala şüphelerin bulunmasından dolayı, siber güvenlik ve savunma için önemi artan tehditlerin başında gelmektedir. Yazılım piyasası içerisinde ise; örün (web) tarayıcıları, işlevleri gereği, önemli bir yer işgal etmektedirler. Yaygın olarak kullanılagelen örün tarayıcılara ait güvenlik açıklık verilerinden yararlanılarak gelecekteki olası zafiyete ilişkin yapılabilecek öngörüler; hem yazılıma karşı yaklaşımımızın sınırlarını daha net çizmemize, hem de kaynaklarımızı daha etkin kullanmamıza ve sonuçta karşılaşılabilecek risklerinde minimize edilmesine olanak tanıyacaktır.**

***Anahtar Kelimeler*— açıklık, keşif, tarayıcı, yazılım**

***Abstract*—Software systems are the main threats for cyber defense and security because there is not really secure and reliable software and are a lot of suspects on them. Furthermore, in the software market, web browsers keep their strong positions because of their functions. Predicting and estimating vulnerabilities of widely used web browsers, by utilizing their vulnerability data sets, we can both take realistic decisions about software that we used and allocate and use our resources effectively.**

***Index Terms*— vulnerability, discovery, browser, software,**

I. GİRİŞ

BÜYÜK oranda yazılım güvenliğine dayanan siber güvenliğin artırılmasına ilişkin olarak; özellikle kritik altyapılarda ve savunma alanında kullanılan bilgi sistemlerinde güvenli (secure) ve güvenilir (reliable) yazılım geliştirilmesine ve kullanılmasına yönelik ciddi programlar mevcuttur [1]-[2].

Yazılımlarda tespit edilen güvenlik açıklıkları (software vulnerabilities), konuyla ilgili kamu kurumları ve/veya şirketler tarafından kamuoyuna duyurulmak suretiyle, güvenlik zafiyetinin asgariye indirilmesi hedeflenmektedir.

Bu açıklıkları, oluşturdukları veri tabanları yoluyla ilgililere ileten başlıca organizasyonlar;

ABD Ticaret Bakanlığı'nın bir kuruluşu olan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standarts and Technology - NIST)[3], ABD İç Güvenlik Bakanlığı'na bağlı olarak görev yapan ABD Bilgisayar Olaylarına Müdahale Ekibi (United States Computer Emergency Response Team - US_CERT)[4] ve Bağımsız olarak faaliyetlerini sürdüren Açık Kaynak Açıklık Veri Tabanı (Open Sourced Vulnerability Database - OSVDB) [5], Bilgi Güvenliği alanında faaliyet gösteren ticari şirketler ve organizasyonlar olarak sıralanmaktadır [6]-[8].

Türkiye'de ise, bu konuda görevlendirilmiş [9] TÜBİTAK Bilgisayar Olaylarına Müdahale Ekibi (TR_BOME) tarafından, yazılım güvenlik açıklıkları "Güvenlik Bildirileri" örün (web) sayfası üzerinden ilgililer ile paylaşılmaktadır [10].

Yazılımlarda mevcut güvenlik açıklıkları çoğunlukla, kamuoyu ile paylaşılmadan önce saldırganlar tarafından bilinen, belirli ortamlarda istismar edilmiş (exploit) ve ilgili yazılımın geliştiricisi firma tarafından büyük olasılıkla güvenlik yaması hazırlanarak yayımlanmış hususları içermektedir.

Bu kapsamda; kişi, kurum ve organizasyonları zarara uğratan esas konu, üreticinin, kullanıcının ve dolayısıyla kamuoyunun bilgisi dâhilinde olmayan ve çoğunlukla ancak saldırı sırasında haberdar olunan sıfır(ıncı) gün ('0' Zero Day) güvenlik açıklıklarıdır [11].

Dolayısıyla, olası bir zafiyetin önceden kestirilebilmesi ya da bir yazılım sisteminde belirli bir zaman aralığında hangi miktarda güvenlik açığı olabileceğinin tahmini önem taşımaktadır.

Günümüzde çok fazla sayıda ve çeşitlilikte yazılım hayatımıza girmiş olduğundan, bunların her biri için uygun bir öngörü modeli geliştirilmesi ve hangi yazılım grubuna öncelik verileceği hususu çözüm gerektiren karmaşık bir problemdir.

Konuya ilişkin olarak yapılan bu çalışmada, yukarıda sıralı hususlara ışık tutulması hedeflenmiştir. Bu kapsamda; bu bildirinin ikinci bölümünde temel kavramlar ve açıklık yaşam döngüsü açıklanmakta, üçüncü bölümde literatürde konuya ilişkin yapılan çalışmalar sıralanmakta, dördüncü bölümde öncelik verilmesi gerekli yazılım grubunun ne olması gerektiği irdelenmekte, beşinci bölümde konuya ilişkin Türkiye'deki duruma dair genel bir bakış sunulmakta, çalışma, sonuç ve öneriler ile sonlandırılmaktadır.

II. KAVRAMLAR VE AÇIKLIK YAŞAM DÖNGÜSÜ

Yazılım güvenliği disiplininde kullanılan kavramlar, sayıca çok, sıklıkla birbiri yerine kullanılan, bazen karıştırılan veya gerçek anlamından uzaklaştırılan niteliktedir. Dolayısıyla bu kavramlara açıklık getirilmesi, çalışmaların sağlıklı yürütülmesi açısından önemlidir.

Kavramların açıklanmasında, İngilizce karşılık ve/veya karşılıklarına da yer verilmek suretiyle anlatım kolaylığı hedeflenmiştir.

IEEE tarafından yayımlanan Yazılım Mühendisliği Standart Sözlüğünde [12] yapılan tanıma göre; insan tarafından yapılan bir *yanlışın (mistake)* bir tezahürü olan

Oğuz BOZOKLU, K.H.O. SAVBEN, Doktora Öğrencisi, obozoklu@gmail.com
Prof.Dr.Celal Zaim ÇİL, Çankaya Üniversitesi, czaimcil@cankaya.edu.tr
Prof.Dr.Şeref SAĞIROĞLU, Gazi Üniversitesi, ss@gazi.edu.tr

kusur (fault, defect, bug, flaw*)* sonucunda *başarısızlık (failure)* gelebilir, başarısızlık sonucu hedeften sapma miktarı ise *hatadır (error)*.

Bütün kusurlar *başarısızlığa* yol açmaz, ancak bütün *başarısızlıklar kusurların* sonucudur.

Saldırılara açık olma hali olarak da betimlenebilecek yazılım güvenlik *açıklıkları* (software security vulnerabilities) içinse, yaygın kabul görmüş ve doktrine olmuş bir tanım bulunmamaktadır.

Schneider; tasarım, uygulama ya da işletmede zayıflık veya hata [13] olarak tanımlarken, Veri ve Bilgisayar Güvenliği Sözlüğünde *bilgisayar açıklığı;* bilgi sistem güvenlik süreçlerinde ve idari kontrollerdeki bir zayıflığın, yetkisiz erişim ve saldırılara açık olma hali yaratması olarak açıklanmıştır [14].

Çalışmanın devamında yazılım güvenlik açıklığı yerine anlatım sadeliği maksadıyla çoğunlukla 'açıklık' tabiri kullanılmıştır.

Açıklığın keşfi, duyurulması, önlem alınması gibi işlem adımları, gerçek hayatta tam anlamıyla karşılığını bulamayabilmektedir. Aşağıda sıralı ve Şekil 1'de gösterilen bu adımların doğrusal bir süreç yerine döngüsel nitelikte olduğu, bazen eşzamanlı olarak faaliyetlerin yürütüldüğü, bazen ise basamaklardan bir ya da birkaçının hiç oluşmadığı hatırdan çıkarılmamalıdır.

Teorik olarak tanımlanmış ve Ozment tarafından tasnif edilmiş açıklık yaşam döngüsü kapsamındaki tanımlar aşağıda sıralanmıştır [15]:

--Doğum Tarihi-DT (Injection Date): Açıklığı barındıran kod parçacığının ilk olarak yazılıma eklendiği ve derlendiği tarihtir.

--Piyasaya Sürülme Tarihi-PST (Release Date): Açıklığı ihtiva eden yazılımın piyasaya sürüldüğü tarihtir.

--Keşfedilme Tarihi-KT (Discovery Date): Açıklığın ilk kez fark edildiği tarihtir.

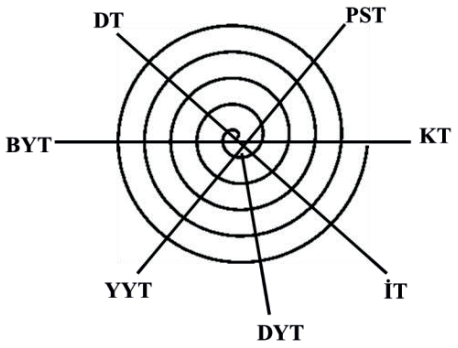
--İfşa Tarihi-İT (Disclosure Date): Açıklığı keşfedenin, yazılım geliştiriciyi ya da ilgili kurumu açıklığa ilişkin haberdar ettiği tarihtir.

--Duyuru Tarihi-DYT (Public Date): Açıklığın kamuoyuna duyurulduğu tarihtir. Genelde açıklık için geliştirilen yamanın yayımlanma tarihi ile aynıdır.

--Yama Yayımlama Tarihi-YYT (Patch Date): Açıklık için ilk düzeltmenin yayımlandığı tarihtir.

--Betik Yayımlama Tarihi-BYT (Scripting Date): Açıklığı istismar etmek üzere hazırlanmış ilk betiğin yayımlandığı tarihtir.

Yukarıda sıralı tarihleri tam ve gerçek bir şekilde tespit edebilmenin zorlukları ortadadır. Özellikle açıklık keşfetme ve ifşa tarihlerindeki belirsizlikler, yapılacak çalışmalarda mutlaka göz önünde bulundurulmalıdır.



Şekil.1. Açıklık Yaşam Döngüsü [15]

Yazılım açıklıklarının yaratılması, keşfi, ortadan kaldırılması vb. işlemlerde aktif olarak yer alan gerçek ve tüzel kişilere daha yakından bakmak gerekirse bunlar aşağıdaki şekilde sıralanabilir [15]:

--Bulucu *(Detector):* Yazılımdaki açıklığın farkına varan, açıklığı bulandır.

--Firma *(Vendor):* Ticari maksadı olsun ya da olmasın yazılımı geliştirendir, onun sahibidir.

--Firma Bulucusu *(Vendor Detector):* Görevi, yazılımdaki açıklıkları arayıp bulmak olan ücretli firma çalışanıdır.

--Bağımsız Bulucu *(External Vendor):* Sistemlerdeki açıklıkları arayarak bulan ancak firmanın maaşlı çalışanı olmayan kişidir.

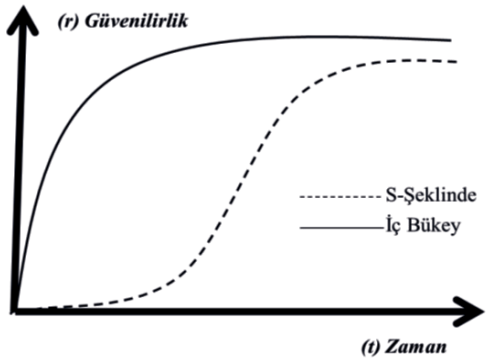
--İfşa Kurumu *(Disclosure Institution):*Bulucunun tespit etmiş olduğu açıklığı rapor ettiği yer, bu açıklığı firmaya ileten organizasyondur.

Açıklığın ifşasında sorumlu davranışta *(responsible disclosure)* bulunulması önemlidir. Yama yayımlanmadan önce açıklığın kamuoyuna duyurulması önemli sakıncalar doğurabilir.

III. LİTERATÜR

Yazılım Açıklıklarının Keşfine dönük modeller (Vulnerability Discovery Models -VDMs) başlangıçta büyük oranda Yazılım Güvenilirliği Geliştirme Modellerine (Software Reliability Growth Models - SRGMs) dayandırılmıştır.

Yaklaşık 30 yıldan fazla bir geçmişi olan Yazılım Güvenilirliğine ilişkin çalışmalar ve ortaya konulan modeller; Şekil 2'de görüleceği üzere iç bükey *(concave)* ve S-şeklinde *(S-shaped)* olmak üzere iki ana grupta toplanmıştır [16].



Şekil.2. Yazılım Güvenilirliği Ana Yaklaşımlar [16].

Yazılım Güvenilirliği Geliştirme yöntem ve modellerinden en önde gelenleri; Goel-Okumoto (G-O) [17], G-O S-Şeklinde [18], Hossain-Dahiya/G-O [19], Gompertz [20], Pareto [21], Weibull [22], Yamada Üssel [23] olarak sıralanmaktadır [16].

Yazılım Açıklık Keşfine dönük önde gelen çalışmalar ise, ABD Colorado Devlet Üniversitesi'nde Malaiya'nın önderliğinde yapılan araştırmalardır [24]. Bu araştırmalar kapsamında ortaya konulan önemli bir yaklaşım S-Şeklinde *(S-Shaped),* Zaman Tabanlı *(Time-Based)* "Alhazmi-Malaiya Lojistik Modeli" (AML)dir. Bu model ve diğer bazı temel yaklaşımlar Şekil 3'de gösterilmiştir.

Sunucu tarafında Windows NT 4.0, istemci tarafında ise Windows 98 işletim sistemleri açıklık verilerinden yararlanılmak suretiyle, takvim zaman aralıklarına dayandırılarak yapılan çalışmada; saldırganların da dâhil olduğu yazılım test grubunun, hedef yazılımı anladıkları ve gerekli bilgileri topladıkları "öğrenme safhası", yazılımın dikkat çekmeye başladığı ve yaygınlaştığı "doğrusal safha", yazılımla ilgili yamaların yayımlanma sıklığının azaldığı ve kullanıcıların yazılımın yeni sürümüne geçişe başladıkları "doygunluk safhası" olarak tanımlanmış ve açıklık/zaman düzleminde S şeklindeki bir eğri üzerine model oturtulmuştur [24].

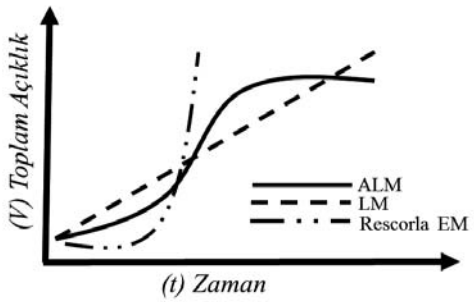
Çalışmada ayrıca, oluşturulacak modellerin, yazılımın yaygınlığı ile yakın ilişkili olduğu vurgulanmıştır. Bu kapsamda; yazılımın yaygın kurulum ve kullanımın, açıklık tespitine yönelik gayretleri de aynı oranda artıracağı varsayımı üzerine bir model inşa edilerek, Gayrete Dayanan (Effort Based) Model olarak literatüre sunulmuştur [24]. Yazılım Güvenilirliği Geliştirme Modelleri (Software Reliability Growth Models [SRGMs]) ile analoji kurularak geliştirilen söz konusu modelde, tanımlanan bir "gayret katsayısı" ile çalışmaya derinlik kazandırılmaktadır.

Çoklu sürüme sahip yazılımlarda açıklık tespitine dönük olarak açık kaynak kodlu Apache HTTP Web Sunucusu ve Mysql Veri Tabanı Yönetim Sistemi (DBMS) açıklık verileri kullanılarak yapılan bir diğer araştırmada da verilerin AML'ye büyük oranda uygunluğu görülmüştür [25].

AML Modelinin Internet Explorer ve Firefox örün (web) tarayıcılarının açıklık veri setlerine uyumluluğunun (goodness-of-fit) denetlendiği başka bir çalışmada, başarılı sonuçlar gözlemlenmiş, gelecekte ortaya çıkarılması beklenen açıklık miktarının tahmin edebilmesi için AML'den yararlanılabileceği belirtilmiştir. Çalışma kapsamında ayrıca, veri setleri, açıklık sebepleri ve açıklığın ciddilik seviyesi ele alınarak tasnif edildiğinde de, modelin öngörülerini destekleyici sonuçlara ulaşılmıştır [26].

Alhazmi ve Malaiya tarafından yapılan bir diğer çalışmada; Apache ve IIS örün (web) sunucularının AML ve Doğrusal (Linear) modele uygunluğu test edilmiş, bunun yanı sıra kısa ve uzun vade açıklık öngörü kabiliyetleri sınanmıştır. AML modelinde tatmin edici sonuçlar alınmakla birlikte, doğrusal modelin öngörüleri ve uyumluluğu dikkat çekici bulunmamıştır [27].

Win2000, WinXP, Red Hat 7.1 için AML ve doğrusal modeller için ile öngörü testi [28], değişik modellerin Win95, WinXP ve RedHat 6.2 için uyumluluk düzeylerinin kıyaslanması [29] araştırmaları, yapılmış olan dikkat çekici diğer bazı araştırmalardır.



Şekil.3. Temel Açıklık Bulma Yaklaşımları [29]

Rescorla [30] tarafından; Doğrusal Model ve Yazılım Güvenilirliği Modeline dayalı üssel yaklaşıma (Rescorla EM) uyumlulukları kapsamında yapılan çalışmada ise; Win NT4, Solaris 2.5.1, Free BSD 4.0 işletim sistemlerinde olumsuz sonuçlar alınmış, Red Hat 6.2 için belirgin neticeler gözlemlenmişse de, modelin öngörü yeteneklerine ilişkin bir analizde bulunulmamıştır.

MIT Lincoln Laboratuvarlarından Ozment [31], açıklık öngörü modellerinin dayandığı varsayımlar ve bu varsayımların karşılanması anlamında eleştiriler getirdiği çalışmasında dört noktaya dikkat çekmiştir. Bunları;

--Açıklıkların tespit edilmesi için geçen ya da harcanan zaman diliminde gösterilen gayretlerin ölçülebilirliğindeki zorluklar,

--Yazılımın kullanıldığı ortamla etkileşimindeki farklılıklar,

--Yazılım kodlarının statik olmadığı ve aslında her bir yamanın yeni açıklıklar yaratma potansiyeli bulunduğu,

--Açıklıkların bazı durumlarda birbirini tekrar eder nitelikte bağımlılıklar içerebildiği şeklinde sıralamıştır.

Ozment ayrıca; yapmış olduğu doktora tezi çalışmasında [15] OpenBSD yazılımının güvenlik açıklık verilerini irdeleyerek, ekonomik boyutunda ele alındığı derinlemesine bir analiz ortaya çıkarmıştır.

McQueen ve diğ. tarafından yapılan bir çalışmada ise; Sıfır Gün Açıklıkları (SGA) üzerine ortaya konulan yöntemdeki öngörüye göre, çalışmanın yapıldığı dönem itibariyle, 2006 yılı için, örnek olarak seçilecek herhangi bir günde ortalama 2500 ila 4500 civarında SGA'nın var olduğu belirtilmiştir [11].

Nguyen ve Sang tarafından, Mozilla Firefox JSE (Java Script Engine) hedef alınarak yapılan çalışmada ise,

karşılıklı bağımlılık grafikleri (Dependency Graphs) yöntemi ile açıklık öngörüsü sınanmış ve olumlu sonuçlar gözlemlenmiştir [32].

Başlangıçta Yazılım Güvenilirliği Geliştirme Modelleri olarak başlayan çalışmaların, son dönemde açıklık keşfetme ve öngörüye dayalı model arayışlarına yöneldiği anlaşılmaktadır. Bu noktada; öngörüde bulunulmak istenen yazılım grubunun seçimine ilişkin alınacak kararın önemi artmaktadır.

IV. AÇIKLIK ÖNGÖRÜSÜNDE BULUNULMASI HEDEFLENEN YAZILIM SEÇİMİ

Yazılım açıklık verilerinin analizi ile geleceğe dönük olarak olası bir kestirimin yapılması araştırmaları, eldeki kısıtlı veriye dayalı, dayandığı varsayımlar itibariyle karmaşık çözümlemeler gerektirebilecek problemli bir alandır. Ancak; güvenilir, öngörü yeteneği yüksek bir modelin geliştirilmesinin sağlayacağı faydalar büyüktür.

Öngörüde bulunulması hedeflenen yazılım grubunun seçilmesi, bu yazılımlara ait açıklık verilerinin tasnifi ve analize hazırlanması, verilerin bilimsel yöntemlerle geçerliliğinin sınanması, sağlıklı bir araştırma için temel adımlardır.

Böylesi bir gayretin merkezine; (a) son dönem bilgi sistem uygulamaların büyük oranda örün tabanlı geliştirilmesi, (b) kullanılacak olan açıklık veri tabanlarında örün tarayıcılarına ait verilerin, oransal olarak görece daha fazla olmasının araştırmanın niteliğine sağlayacağı olumlu katkı, (c) küresel ölçekte %90'lara varan pazar payları olması [33]–[35 nedenleriyle, örün tarayıcılardan Microsoft Internet Explorer, Google Chrome ve Mozilla Firefox yazılımlarının konulması gerektiği değerlendirilmektedir. Bu hususlar aşağıda başlıklar altında sunulmuştur.

A. Örün Tabanlı Uygulamalardaki Artış

Son 15 yıldır bilgi sistem iletişim teknolojilerindeki gelişmeler ve iletişim altyapı bant genişliğinin dikkate değer ölçüde artmış olması sebebiyle, tek başına çalışan sistem ve terminallerin yerini artık ağ tabanlı çalışan ve küresel hizmet veren yapılar almıştır.

Birçok bilgi sistem uygulamasının 'giriş kapısı' karakterine bürünen örün tarayıcılar ise adeta bu dönüşümün öncüleri olarak görev yapmışlardır.

Örün tarayıcılar marifetiyle bilgiyi paylaşan ağ tabanlı sistemlerin yaygınlaşması, saldırganların, hedeflerine tek bir yolla ulaşmaları zorunluluğunu azaltmış, örün tarayıcısına gömülü bir uygulamaya (built-in interpreter; HTML (Hyper Text Markup Language), JavaScript, CSS (Cascading Style Sheets, vb.) zararlı içeriği bir şekilde bulaştırmak suretiyle saldırıların boyutunu değiştirmiştir. Bu yolla uzaktan istismar edilebilir güvenlik açıklarının 2000-2007 yılları arasında %89,4 oranında arttığı rapor edilmiştir [36].

Yaşanan gelişmeler, alınacak güvenlik önlemlerinin en ön sıralarına örün tarayıcıları almamız gerektiğini işaret etmektedir.

B. Açıklık Veri Tabanlarında Bulunan Örün Tarayıcılara Ait Verilerin Oransal Fazlalığı

1997 yılından itibaren açıklık verilerinin tutulduğu ve 2005 yılından itibaren kamuyla paylaşılmaya başlandığı ABD Ulusal Açıklık Veritabanı (NVD) incelendiğinde [40], hâlihazırda **57046** adet açıklığın duyurulmuş olduğu, günde ise ortalama 16 adet duyuru yapıldığı görülmektedir [3].

Bu verilerden;

--Internet Explorer için, ilki 01/03/1997 tarihinde duyurulan "CVE-1999-1128" açıklığından, 10/07/2013 tarihindeki "CVE-2013-3166" açıklığına kadar toplam **1054** adet,

--Google Chrome için, ilki 30/09/2008 tarihinde duyurulan "CVE-2008-4340" açıklığından, 10/07/2013 tarihinde duyurulan "CVE-2013-2880" açıklığına kadar top**lam 877** adet,

--Mozilla Firefox için ise, ilki 27/07/2004 tarihinde duyurulan "CVE-2004-0718" açıklığından, 26/06/2013 tarihindeki "CVE-2013-1700" açıklığına kadar topla**m 1049** adet açıklığın kamuoyu ile paylaşıldığı görülmektedir.

NVD'de onbinlerce yazılıma dair açıklık duyuru verilerinin bulunduğu düşünüldüğünde, bu üç örün tarayıcıya ait **v**erilerin toplam duyuruların yaklaşık %5'ine tekabül etmesi ve görece oransal büyüklüğü dikkat çekicidir.

C. Pazar Payı Olarak Üç Tarayıcının Hâkimiyeti

Bu üç tarayıcının değişik kaynaklardan edinilen bilgilerle derlenen ve ayrıntısı Tablo 1'de sunulan pazar payı oranlarına bakıldığında dünya genelinde yaygın şekilde kullanıldıkları açıkça görülmektedir [33]-[35].

TABLO I
ÖRÜN TARAYICISI PAZAR PAYLARI [33]-[35.]

Tarayıcı	.statcounter [33]	.sitepoint [34]	.w3schools [35]
IE	42,68%	29,82%	12,6%
Chrome	25,44%	37,11%	27,7%
Firefox	20,01%	21,34%	52,9%
TOPLAM	88,12%	89,27%	93,2%

Ülke özelinde bakıldığında da; % 57,97 Chrome, %27,81 IE ve %8,52 Firefox olmak üzere toplamda %94,30'luk oranla Benzer bir sonuç göze çarpmaktadır. Türkiye pazarına yakın zamanda giriş yapan Yandex tarayıcının pazar payı %2,5-3 seviyelerindedir [33].

Şekil.4. Örün Tarayıcı Pazar Payı Değişim Eğilimleri [33].

Zaman ve kullanım yaygınlığı düzleminde oluşturulan Şekil 4'de ise;

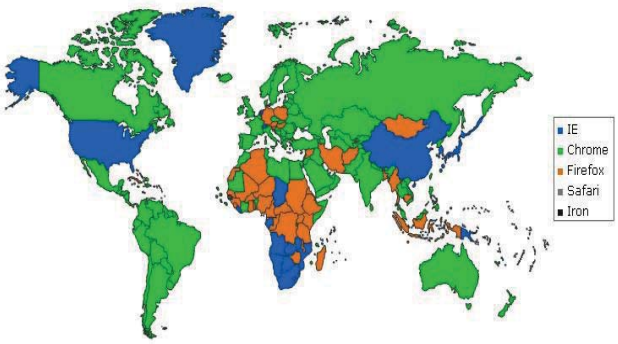
--IE kullanımın genelde ABD ve Çin'de daha çok tercih edildiği (Bknz. Şekil 5.) ve dünya genelinde pazar payının düşüş eğiliminde olduğu,

--Chrome örün tarayıcısının, kullanımının zaman içerisinde yaygınlaştığı,

--Yatay bir eğride seyreden Firefox'un daha tutucu ve sadık bir kullanıcı kitlesine hitap ettiği şeklinde yorumlanabilir.

V. TÜRKİYE'DEKİ DURUMA GENEL BİR BAKIŞ

Türkiye'de, siber güvenlik konusunda artan bir farkındalık ve bilinç söz konusu ise de gerekli örgütlenme ve mekanizmalar arzulanan düzeyde değildir [37]-[39].



Şekil.5. Kullanım Yaygınlığı Haritası [33].

Etkin ve sürdürülebilir bir yapıya kavuşturulması gerekli mekanizmalardan belki de en gerekli olanlardan birisi, yazılım açıklıklarından haberdar olunduğu, ilgililere duyurulduğu, doğru ve güvenilir açıklık öngörülerinin yapılabildiği bir yapının kurulmasıdır.

Mevcut durumda yazılım açıklıklarından görece önemli olduğu değerlendirilenler, TÜBİTAK [10] tarafından, örün sayfası yoluyla, "güvenlik bildirileri" adı altında, ilgililere duyurulmaktadır.

Söz konusu açıklık verileri; yazılım geliştirici firmalara veya ABD kaynaklı diğer ilgili kuruluşlara [3]-[8] dayandırılmaktadır.

Değerli bilgiler içeren ve birçok konuda her seviyeden kullanıcıya bilgi güvenliği konusunda yol gösterici olarak önemli bir işlevi yerine getiren www.bilgiguvenligi.gov.tr [10]'da, güvenlik bildirilerinin kapsamı; "*ülkemizde ve özellikle kamu kurumlarında sıkça kullanıldığı tespit edilen/öngörülen bilgi sistemi varlıklarında yeni bulunan açıklıklar hakkında detaylı bilgileri içerir. Bu açıklıklar TUBİTAK BİLGEM Siber Güvenlik Enstitüsü ağ güvenliği laboratuvarında tespit edilebileceği gibi, farklı açıklık veritabanlarından da alınabilir.*" şeklinde açıklanmaktadır.

TÜBİTAK tarafından yayımlanan güvenlik bildirilerine niceliksel olarak bakıldığında; **12 Mart 2008** ile **18 Haziran 2013** tarihleri arasında **623** adet güvenlik açıklığı ile ilgili duyuru yapıldığı görülmektedir. Aynı zaman aralığında NVD [3]'de **26849**, OSVDB [5]'de **45927,** Securityfocus [6]'da **35684** ve Rapid7 [7]'de ise **6154** adet güvenlik açıklığı ele alınarak duyurulmuştur.

Söz konusu duyuru sayfası [10]; site içi arama seçeneği sunmasına rağmen, açıklığa dair değişik parametrelere göre (açıklığın cinsi, üreticisi, tarihi, risk seviyesi vb.) açıklık veritabanını tarama olanağı sunmamaktadır.

Türkiye'de bilgi güvenliğine katkı kapsamında, yazılım açıklıkları konusu ile doğrudan ya da dolaylı olarak ilgili çeşitli platformlar [41]-[47] bulunmaktadır. Bu ve benzeri diğer platformların içeriğini ve yaygınlığını gün geçtikçe artırdığı görülmektedir.

VI. SONUÇ VE TARTIŞMA

Yazılımlardaki olası bir zafiyete dönük olarak geleceğe dair bir güvenlik açıklık öngörüsünde bulunabilmenin özellikle kritik altyapılar düşünüldüğünde hayati öneme haiz sonuçları olabileceği açıktır.

Bu yazılımların başında örün tarayıcıların her türlü sürümü gelmektedir.

Ülke içerisinde ise, büyük oranda dışa bağımlılık arz eden genel kullanıma dönük yazılımlara dair güvenlik açıklıklarının toplanması, derlenmesi, kendi içerisinde tutarlı bir şekilde sınıflandırılması, analizi ve duyurulması faaliyetlerinin bir takım zorluklar içerdiği anlaşılmaktadır.

Zaman içerisinde kamu başta olmak üzere [48] ülke sathında açık kaynak kodlu yazılımlar ve buna ek olarak özgün milli yazılımların yaygınlaşması öngörüldüğünden, yazılım açıklıklarının saldırganlardan önce tespit edilmesi, önlem alınması ve kullanıcılara duyurulması gün geçtikçe artan bir gereklilik olarak karşımıza çıkacaktır.

Bu çalışma kapsamında elde edilen bulgular ve öneriler aşağıda verilmiştir.

--Kısa sürede NVD, OSVDB, Securityfocus ve Rapid7 gibi güvenlik açıklığı duyurusu yapan yapılar kurulmalı veya bilgiguvenligi.gov.tr de bulunan yapı güncellenmelidir

--Yazılım güvenliği ve güvenilirliğini test eden merkezler açılmalıdır.

--Yazılımları Ortak Kriterlerde farklı seviyelerde test edebilen (EAL seviyeleri) yapılar oluşturulmalıdır.

--Artık günümüzde yazılımların aynı zamanda siber silahlar olduğu, özellikle amaca yönelik olarak geliştirilen yazılımlarla ülkelere büyük boyutta zararlar verilebildiği unutulmadan gerekli önlemler alınmalıdır.

--Ülkemizde yazılım güvenliği ve güvenilirliğine yönelik olarak üniversitelerde ar-ge çalışmalarına önem verilmelidir.

--Ülkemizde bu konuda belirli standartlar geliştirilmeli, kritik görevlerde kullanılan yazılımlar test edilmeden kesinlikle kullanılmamalıdır.

--Bu çalışma kapsamında yer alan örün tarayıcıları konusunda da mutlaka derinlemesine çalışmalar yapılmalı, gerekirse yeni bir örün tarayıcı da geliştirilmelidir.

Açıklıkların analizi ve olası açıklık öngörüsüne ilişkin küresel çerçeveden bir bakışı, yerel oluşum ve yapılanmayı genel olarak inceleyerek destekleyen bu çalışmanın, geleceğe dönük olarak kurulması ve geliştirilmesi hedeflenen özgün yapılar için ışık tutabileceği değerlendirilmektedir.

KAYNAKLAR

[1] Information Security Management System (ISMS), 2005,(ISO 27001)
[2] Common Criteria for Information Technology Security Evaluation (abbreviated as Common Criteria or CC) is an International Standard (ISO/IEC 15408)
[3] ABD Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), Açıklık Veri Tabanı (NVD) Örün (web) Sayfası. http://nvd.nist.gov.

[4] ABD Bilgisayar Olaylarına Müdahale Ekibi Örün (web) Sayfası. http://www.kb.cert.org/vuls

[5] Açık Kaynak Açıklık V eri Tabanı Örün Sayfası, http://osvdb.org.

[6] Security Focus, Symantec. Açıklık Veri Tabanı Örün Sayfası. http://www.securityfocus.com/vulnerabilities,

[7] Rapid7, http://www.rapid7.com/vulndb/index.jsp

[8] The MITRE Corporation, http://cve.mitre.org/,

[9] Bilgi Toplumu Stratejisi Eylem Planı (2006–2010), 88 Numaralı Eylem Maddesi, Devlet Planlama Teşkilatı (DPT).

[10] TÜBİTAK BİLGEM Bşk.lığı Bilgisayar Olaylarına Müdahale Ekibi (TR_BOME) Örün Sayfası, http://www.bilgiguvenligi.gov.tr/guvenlik-bildirileri/index.php

[11] Miles A. McQueen ve diğ., *Empirical Estimates and Observations of 0Day Vulnerabilities*, Hawaii International Conference on System Sciences

[12] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.121990, (Revision and reddgnation of (IEEEstd7921983),1990.

[13] Schneider, Trust in Cyberspace, 1998

[14] Dennis Longley ve Michael Shain, *Data & Computer Security: Dictionary of Standards Concepts and Terms*, 1987, Stockton Pr (December 1987)

[15] Ozment Andy, *Vulnerability Discovery & Software Security*, University of Cambridge Computer Laboratory Computer Security Group & Magdalene College, August 31, 2007, Degree of Doctor of Philosophy

[16] Wood Alan, *Predicting Software Reliability*, 1996, Tandem Computers

[17] A.L.Goel ve Okumoto, "*A Time Dependent Error Detection Model for Software Reliability and Other performance Measures*", IEEE Trans.Reliablity.Ağustos 1979, pp.206-211.

[18] S.Yamada, M.Ohba, ve Osaka, *"S-Shaped reliability Growth Modeling for Software Error Detection"*. IEEE Trans. Reliability. Aralık 1983,pp. 475-484

[19] S.Hossain ve R.Dahiya, "*Estimating the Parameters of Non-Homogeneous Poisson-Process Model of Software Reliability*", Aralık 1993, IEEE Trans. Reliability.pp. 604-612

[20] D.Keçecioğlu, *Reliablity Engineering Handbook*, Vol.2, Prentice-Hall, Englewood Cliffs,NJ,1991.

[21] B.Littlewood,"*Stochastic Reliability Growth:A Model for Fault Removal in Computer Programs and Hardware Design*" IEEETrans reliablity, Aralık 1981, pp.313-320.

[22] J.Musa,A.Iannino ve K.Okumoto, Software Reliability,McGraw-Hill,New York, 1987.

[23] S.Yamada, H.Ohtera, ve H.Narihisa "*Software Reliability Growth Models with Testing Effort*" IEEE Trans Reliablity. Nisan 2986,pp.19-23

[24] Omar H. Alhazmi, Yashwant K. Malaiya, *Quantitative Vulnerability Assessment of Systems Software*, Colorado State University,2005

[25] Jinyoo Kim, Yashwant K. Malaiya, Indrakshi Ray, *Vulnerability Discovery in Multi-Version Software Systems*, Computer Science Department, Colorado State University, Fort Collins,2007.

[26] Sung-Whan Woo, Omar H. Alhazmi and Yashwant K. Malaiya, *An Analysis Of The Vulnerability Discovery Process In Web Browsers*, Computer Science Department, Colorado State University, Fort Collins,

[27] Omar H. Alhazmi and Yashwant K. Malaiya, *Measuring and Enhancing Prediction Capabilities of Vulnerability Discovery Models for Apache and IIS HTTP Servers*, ,2006.

[28] Omar H. Alhazmi,, *Prediction Capabilities of Vulnerability Discovery Models*, Colorado State University Yashwant K. Malaiya, Ph. D., Colorado State University, 2006.

[29] Omar H. Alhazmi and Yashwant K. Malaiya, Senior Member, IEEE *Application of Vulnerability Discovery Models to Major Operating Systems*, 2008.

[30] Eric Rescorla, *Is Finding Security Holes a Good Idea?* RTFM, 2005.

[31] Ozment Andy, *Improving Vulnerability Discovery Models, Problems with Definitions and Assumptions*, MIT Lincoln Laboratory & University of Cambridge.

[32] Nguyen Viet Hung ve Le Minh Sang Tran, *Predicting Vulnerable Software Components with Dependency Graphs*, University of Trento, Italy,2010.

[33] Örün sayfası, http://gs.statcounter.com/

[34] Örün sayfası, http://www.sitepoint.com/browser-trends-march-2013/

[35] Örün sayfası, http://www.w3schools.com/browsers/browsers_stats.asp

[36] Stefan Frei ve diğ., *Understanding the Web browser threat: Examination of vulnerable online Web browser populations and the "insecurity iceberg"*,ETH Zurich Tech Report Nr. 288.

[37] E-Devlet Yuvarlak Masa Toplantısı, 10/03/2009, http://www.bthaber.com.tr/?p=10979&sayi=SAYI:810, , Sheraton Otel.

[38] Sağıroğlu Ş. Prof. Dr. http://ceng.gazi.edu.tr/secure/yazilardetay.aspx?yaziid=10, 06 Mayıs 2012 tarihinde erişim sağlanmıştır.

[39] Acarer Tayfun, BTK Başkanı, "*Siber Saldırılara Hazır Değiliz",* http://www.marmaraweb.com/web-tasarim-blogu/459-siber-saldirilara-hazir-degiliz.html,

[40] Mathias Karlsson, *The Edit History of the National Vulnerability Database and similar Vulnerability Databases*, Master's Thesis.Swiss Federal Institute of Technology, Zurich.

[41] Bilişim Platformu Örün Sayfası, http://www.bilisimplatformu.com/#

[42] E-Siber Örün Sayfasıhttp://www.e-siber.com/, 14 Temmuz 2013 tarihinde erişilmiştir.

[43] Forum Sayfası, http://www.frmtr.com/guvenlik-ve-guvenlik-aciklari/

[44] Örün Sayfası, http://www.olympos.net,

[45] Örün Sayfası, http://www.turk.internet.com/portal/index.php

[46] Örün sayfası, http://www.webguvenligi.org/,

[47] Örün sayfası, http://yazilimportal.com/

[48] M. Raşit ÖZDAŞ, "*Kamuda Açık Kaynak Kodlu Yazılım Kullanımı", Eylül 2012*, Çalışma Raporu – 4, Kalkınma Bakanlığı Bilgi Toplumu Dairesi, Ankara.