Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 28, S. 1008-1012, Kasım 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 28, pp. 1008-1012, November 2021

Research Article

Yazılım Güvenlik Açığı Veri Tabanları

Hakan Kekül[1*], Burhan Ergen[2], Halil Arslan[3]

[1*] Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Elazığ, Türkiye, (ORCID: 0000-0001-6269-8713), hakankekul@gmail.com
[2] Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, Türkiye (ORCID: 0000-0003-3244-2615), bergen@firat.edu.tr
[3] Sivas Cumhuriyet Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Sivas, Türkiye (ORCID: 0000-0003-3286-5159),harslan@cumhuriyet.edu.tr

(1st International Conference on Applied Engineering and Natural Sciences ICAENS 2021, November 1-3, 2021)

(**DOI:** 10.31590/ejosat.1012410)

ATIF/REFERENCE: Kekül, H., Ergen, B. & Arslan, H. (2021). Yazılım Güvenlik Açığı Veri Tabanları. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1008-1012.

Öz

Bir yazılım bileşeninin güvenlik açığı eğiliminin öngörülmesi, yazılım mühendisliğinin zorlayıcı araştırma alanlarından biridir. Bir bileşenin güvenlik açığı eğilimi hakkında önceden bilgi sahibi olmak, test çabasını ve süreyi önemli ölçüde azaltabilir. Yazılım güvenlik açıklarının belirlenmesi ve sınıflandırılması geliştiricilere yazılımın geliştirilmesinde doğru karar verme noktasında yardımcı olacaktır. Bu sebeple yazılımlarda tespit edilen açıklar çok uzun zamandır veri tabanlarına kaydedilmektedir. Farklı araştırma grupları tarafından pek çok veri tabanı oluşmuştur. Bu çeşitlilik her veri tabanına kendi içinde avantajlar ve dezavantajlar sağlamıştır. Bu çalışmada araştırmacıların çalışmalarında hangi veri tabanını kullanacaklarına karar vermelerine yardımcı olmak ve literatürde kullanılan en güncel ve erişime açık olanların sistematik bir listesi oluşturulmuştur. Yazılım güvenlik açığı tespiti ve sınıflandırmasında kullanan birçok farklı veri tabanının incelenmesi ve karşılaştırması yer almaktadır. Çalışmanın sonunda sonuçlar sunulmuş ve gelecekteki çalışmalar için yönlendirici tavsiyeler verilmiştir.

Anahtar Kelimeler: Yazılım Güvenliği, Yazılım Güvenlik Açığı, Zafiyet Veri Tabanları, Siber Güvenlik

Software Vulnerability Databases

Abstract

Predicting the vulnerability propensity of a software component is one of the challenging research areas of software engineering. Having prior knowledge of a component's vulnerability propensity can significantly reduce testing effort and time. Identifying and classifying software vulnerabilities will assist developers in making the right decision in software development. For this reason, vulnerabilities detected in software have been recorded in databases for a very long time. Many databases have been created by different research groups. This diversity has provided advantages and disadvantages to each database. In this study, a systematic list of the most up-to-date and accessible ones used in the literature was created to help researchers decide which database to use in their studies. There is a review and comparison of many different databases used in software vulnerability detection and classification. At the end of the study, the results are presented and guiding recommendations for future work are given.

Keywords: Software Security, Software Vulnerability, Weakness Databases, Vulnerability Databases.

[*] Sorumlu Yazar: hakankekul@gmail.com

1. Giriş

Siber güvenlik günümüz yazılım dünyasının en önemli kavramlarından biridir. Yazılımlar modern günlük yaşantımızın içinde çok önemli bir yere sahiptirler. Birçok durumda, yazılım sisteminde oluşabilecek sorunlar kötü sonuçlara sebep olabilmektedir. Bu nedenle, yazılımın güvenliğini belirleyebilmek çok önemlidir. Daha güvenli sistemlerin inşa edilmesi, son yıllarda tüm yazılım mühendisliği çabalarını yönlendiren amaç olarak karşımıza çıkmaktadır. Bu anlamda hata tahmini, yeniden kullanılabilirlik, yaşlanma tahmini, bilgi güvenliği ve yazılım ürün hattı gibi kavramların doğmasına neden olmuş ve bu alanlarda bilimsel çalışmalar yapılmıştır ve yapılmaya devam etmektedir. Yazılım güvenlik açıkları uzman kişiler tarafından manuel tespit edilerek sınıflandırılmakta, kategorize edilmekte ve skorlanmaktadır.

Yazılım kalitesi, güvenlik açığı özelliklerinin analizi, keşfi ve skorlarını tahmin ederek sınıflandıracak çalışmalar son yıllarda giderek artmaktadır. İlk olarak, yazılım kalite değerlerinin belirlenmesi için kullanılan yazılım metrikleri ayrıntılı olarak literatürde kullanılmıştır. Daha sonra yazılım güvenlik açığı analizi akademik bir ilgi alanı olarak tanımlanmıştır. İlk çalışmalarda geleneksel yaklaşımlar uygulanmış olsa da başarı istenen seviyede olamamıştır. Makine öğrenmesi ve veri madenciliği tekniklerinin yazılım bileşenlerinin güvenlik açıkları probleminde kullanımının arkasındaki motivasyon kaynağı bu algoritmaların farklı problemlerde ciddi başarılar elde etmesinde yatmaktadır (Kekül er al. 2021). Literatürde bu amaçla, yazılım bileşenlerinin güvenlik açığı analizi ve tespiti problemi için makine öğrenme ve veri madenciliği tekniklerini kullanan birçok farklı çalışma kategorik olarak yapılmıştır (Ghaffarian and Shahriari 2017).

Özellikle son dönemde tüm araştırmacılar makine öğrenmesi algoritmalarında kullanılacak özellikte veri setleri üzerine özellik mühendisliği çalışmalarının yapılmasını tavsiye etmektedir (Ghaffarian and Shahriari 2017; Spanos and Angelis 2018).

Ghaffarian vd. (Ghaffarian and Shahriari 2017), yazılım kırılganlık analizi ve keşif alanındaki makine öğrenmesi ve veri madenciliği tekniklerini kullanan birçok farklı çalışmanın kapsamlı bir incelemesini sunmaktadır. Bu alandaki farklı çalışma kategorilerini inceleyerek, hem avantajları hem de eksikliklerini tartışmakta ve alandaki zorlukları ve bazı keşfedilmemiş bölgeleri işaret etmektedir. Yazarlar çeşitli yazılım açıkları için yüksek düzeyde ayırt edici ve etkileyici güce sahip mühendislik açısından zengin özellikler içeren makine öğrenim sistemlerinin performansını artıracak özellik mühendisliği çalışmalarının yapılmasını önermektedir.

Wu vd. (Wu et al. 2020), makine öğrenimi tabanlı güvenlik hata raporları (SBR) tahmini için büyük ölçekli veri kümeleri oluşturma yaklaşımı önermektedir. Yaklaşık 80 bin hata raporu içeren OpenStack için veri kümesinin başlangıç sürümünü oluşturmuşlarıdır. Sonuç olarak, oluşturulmuş veri setlerinin kalitesini daha da artırmak için diğer yöntemleri (özellik seçimi, derin öğrenme gibi) dahil ederek veri seti oluşturma yaklaşımını geliştirmeyi önermektedir.

Williams vd. (Theisen and Williams 2020), çalışmalarında yıllardır biriken güvenlik açığı verilerinin büyük bir yapılandırılmamış veri grubu haline geldiğini belirtmektedir. Bu durumun verilerin kapsamlı analizini yapmak için gerekli araçların ve algoritmaların denenmesindeki eksiklikler yüzünden çoğunlukla keşfedilemeyen noktalar olduğunu vurgulamaktadır. Sonuçları, güvenlik açığı eğilimleri, evrimi ve etkileşimleri ile güvenlik açıklarına karşı genel ürün duyarlılığı konusunda önemli bir boşluk bulunduğunu ortaya koymaktadır. Güvenlik açığı verilerinin önemli özelliklerini anlamanın araştırmacıların ve endüstri uzmanlarının gelecek çalışmalarında daha güvenli sistemler geliştirmesinde, güvenlik açıklarından kaynaklanan sorunların azalmasında ve yeni akademik çalışma alanlarının doğmasına yol açacağı belirtilmektedir.

Fang vd. (Fang et al. 2020), güvenlik açıklarının sadece küçük bir bölümünün saldırganlar tarafından istismar edildiğini belirtmiştir. Bu nedenle istismar edilemez güvenlik açıkları ile diğerlerinin ayırt edilmesinin sınırlı kaynakların verimli kullanımını sağlayacağını vurgulamaktadır. Belirlenen güvenlik açıklarının sistemde yayınlanmasının zaman alması ve NVD veri tabanının kurumsal yapısından kaynaklanan etkilerden dolayı yetersiz kaldığı farklı toplulukların oluşturduğu veri tabanı açıklamalarının daha verimli özellikler içerdiği belirtilmektedir.

Yang vd. (Yang et al. 2020), istismar kodlarının yaklaşık yarısının güvenlik açığının ilanından iki hafta içerisinde kullanıldığı açıklamıştır. Bunun yanında ilan edilen açıların sadece %20'sinin istismara maruz kaldığı belirtilmektedir. Bu nedenle güvenlik açıklarının skorlarının doğru bir şekilde tahmin edilmesi ve açıkların önceliklendirilmesinin önemi vurgulanmaktadır.

Raducu vd. (Raducu et al. 2020), güvenlik açıklarını tespit etmek için farklı makine öğrenimi tekniklerinin ortaya çıktığını ve geliştirildiğini vurgulanmaktadır. Ancak, bu algoritmaların performanslarının veri kümeleri olarak bilinen çok miktarda verinin işlenmesine dayanan veri güdümlü motorlara ihtiyaç duyduğunu belirtmektedir.

İncelenen literatürün de gösterdiği üzere yazılım güvenlik açıklarının raporlanmaya başlanması ile birlikte akademik camianın bu alana ilgisi artmıştır. Alanın önemi dolayısı ile çalışmalar özellikle gelişmiş ülkelerde devlet desteği ile yapılmaktadır. Makine öğrenmesi algoritmalarının pek çok problemde başarı ile uygulanması sonrası özellikle 2012 yılından itibaren bu problem özelinde kullanıldığını görülmektedir (Ghaffarian and Shahriari 2017). Son yıllarda yapılan çalışmalar makine öğrenmesi temelli yaklaşımların kullanımı tavsiye etmektedir. Ancak literatür incelemelerinden de anlaşılacağı üzere yüksek başarım elde edilebilmesi için çalışmalarda yapılandırılmış ve özellikleri çıkarılmış veri setlerinin kullanılmasına ihtiyaç olduğu açıktır (Ghaffarian and Shahriari 2017; Miyamoto, Yamamoto, and Nakayama 2017; Raducu et al. 2020; Theisen and Williams 2020; Wu et al. 2020). Alana yön veren NVD veri seti yapısı itibari ile bunu sağlayamamaktadır. Bu durum araştırmacı topluğu tarafından farklı özellikler barındıran veri setleri oluşturmak yolu ile çözülmeye çalışılmıştır. Oluşturulan farklı veri setleri aynı güvenlik açıklarına yeni ve daha geniş bilgiler eklemiştir. Tüm bu veri setlerinin temel sorunu doğal dille ve uzmanların anlayacağı yapılar olarak oluşturulmuş olmalarıdır. Makine öğrenimi algoritmalarında doğrudan kullanımları uygun değildir.

2. Materyal ve Metot

2.1. Yazılım Güvenlik Açığı Kavramı

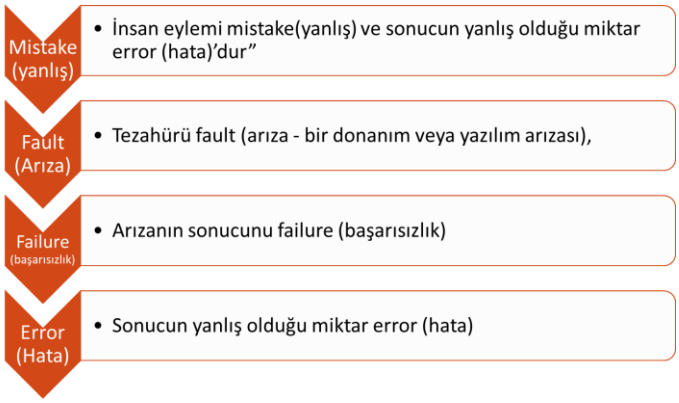
Yazılım güvenlik açığı kavramını tanımlarken IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğüne baktığımızda iki

temel terimi merkezine alan Krsul (Krsul 1998) ve Ozment (Ozment 2007) tanımlarını kabul ettiğini görmekteyiz. Bunlar;

Krsul (Krsul 1998); "yazılımın tanımlanmasında, geliştirilmesinde veya yapılandırılmasında, yazılımın çalıştırılmasında güvenlik politikasını ihlal edebilecek bir hata örneği".

Ozment; "Bir yazılım güvenlik açığı, uygulamanın örtük veya açık güvenlik politikasını ihlal edebilmesi için yazılımın teknik özelliklerinde, geliştirilmesinde veya yapılandırılmasında yapılan bir yanlışın bir örneğidir" (Ozment 2007).

Bu iki tanımdaki temel fark hata kelimesinin yanlış olarak değiştirmiş olmasıdır. IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğünde bu tanımlar benimsenmiştir (Committee and others 1990).



Şekil 1: Yazılım güvenlik açığını tanımlamada kullanılan terimler

IEEE Yazılım Mühendisliği Terminolojisi Sözlüğü 'ne baktığımızda şekil-1'deki dört anahtar terimin tanımlarının önemli olduğu anlaşılmaktadır. Bu terimlerin ilişkisinin bir özeti ve açıklanması "insan eylemi mistake(yanlış), tezahürü fault (bir donanım veya yazılım arızası), arızanın sonucunu failure (başarısızlık) ve sonucun yanlış olduğu miktar error (hata)'dur" (Committee and others 1990).

Bu tanımlardan, bir yazılım güvenlik açığı tanımında kullanılacak uygun anahtar terimin fault(arıza) (ayrıca kusur veya bug) olduğunu literatürde belirtilmiştir (Ghaffarian and Shahriari 2017).

Genel kabul gören tanıma göre yazılım güvenlik açığı şu şekilde tanımlanmaktadır; "Bir yazılım güvenlik açığı, bazı açık veya örtülü güvenlik politikasını ihlal etmek için kullanılabilecek şekilde yazılımın tasarımında, geliştirilmesinde veya yapılandırılmasındaki bir hatanın neden olduğu bir kusur örneğidir." (Ghaffarian and Shahriari 2017)

2.2. Güvenlik Açığı Veri Tabanları

Bir güvenlik açığı tespit edildiği zaman resmi yollar ile ilan edilmesi için uluslararası bir standart olan ve ABD İç Güvenlik Bakanlığı tarafından finanse edilen bir prosedür uygulanmaktadır. Bu işlem için yetkilendirilmiş kurum kar amacı gütmeyen MITRE şirketidir (Mitre Corporation 2020). Bu organizasyona üye olan pek çok ülkeye bağlı Bilgisayar acil müdahale ekipleri (Computer emergency response team –CERT) tarafından Ortak Güvenlik Açıkları ve Etkilenmeler (Common Vulnerabilities and Exposures – CVE) veri tabanına tespit edilen güvenlik açıkları kaydedilerek resmi süreç başlatılmış olmaktadır.

Çalışmamızda literatürde kullanılmış on iki farklı veri tabanı tespit edilmiş ve incelenmiştir. Veri tabanlarının tespitinde özellikle akademik çalışmalarda kullanılmış olması tercih edilmiştir. Tespit edilen on iki veri tabanı ayrıntılı bir şekilde incelenmiştir. Çalışmamıza hangi veri tabanlarının dahil edilip hangilerinin dışarda bırakılacağına karar vermek üzere iki seçim kriteri belirlenmiştir. Bunlar güncellik ve araştırmacıların kullanımına açık erişim imkânları sağlamalarıdır. Bu iki temel seçim kriterine uyan veri tabanları çalışmamız içerisinde incelenmiştir. Bu seçim sonucu yedi veri tabanı seçilmiştir.

Tablo 1. Veri tabanlarının genel bilgileri

Veri Tabanı	Güvenlik Skoru	Çözüm	İstismar Kodu	Referanslar	Test	Raporlama	İş Modeli	Veri Boyutu
CVE	×	×	×	✓	×	Herkes	Kamu	139.407
NVD	✓	×	(✓)	✓	×	Üyeler	Kamu	147.510
Exploit-DB	×	×	✓	✓	✓	Herkes	Kamu	42.962
SecutrityFocus	×	✓	(✓)	✓	✓	Herkes	Kamu	102.330
Rapid7	✓	×	×	✓	×	Çalışanlar	Ticari	171.816
Snyk	✓	×	×	✓	×	Çalışanlar	Ticari	6.012
SARD	×	×	×	✓	✓	Herkes	Kamu	177.184

Çalışmaya dahil edilen veri tabanları karşılaştırılırken güvenlik açığı skorlaması, veri boyutu, raporlama, iş modeli, güvenlik açığı için çözüm yöntemi, istismar kodlarının bulunup bulunmaması, referanslar ve güvenlik açıkları için test yapıp yapmama kriterlerine göre değerlendirilmiştir. Tablo 1'de tüm değerlendirme kriterlerine göre veri tabanlarının genel bilgileri verilmiştir.

2.2.1. CVE - Ortak Güvenlik Açıkları ve Etkilenmeler Sözlüğü

Tespit edilen güvenlik açıklarına uluslararası bir standart getirmek üzere MITRE tarafından bir araya getirilen büyük güvenlik organizasyonları ile birlikte 1999 yılında kurulmuştur. Genel olarak bilinen siber güvenlik açıkları için bir tanımlayıcı listedir. CVE girişlerinin kullanımı yazılımların güvenliği

hakkında uluslararası güvenirliği olan benzersiz bir güven sağlamaktadır. CVE, bir güvenlik açığı için tanımlayıcı, standartlaştırılmış bir açıklama, bir temel, endüstri tarafından onaylanmış herkese açık ve ücretsiz veri sağlamayı amaç edinmiştir. Temel misyonu farklı veri tabanları ile araçlar için aynı standartları oluşturmak, birlikte çalışabilirlik ve bilişim ekosisteminin güvenlik kapsamını iyileştirmektedir. Bir veri tabanından çok bir sözlük olarak kendisini tanımlamaktadır. Bilinen tüm büyük veri tabanları temelde CVE'de yayınlanan listeleri temel alarak oluşturulmuştur (CVE 2020). Standart ve güvenilir bilgi sağlamasının yanında listelerinin ham bilgiler barındırması ve ek bilgiler içermemektedir.

2.2.2. NVD - Ulusal Güvenlik Açığı Veri Tabanı

Ulusal Standartlar ve Teknoloji Enstitüsüne (NIST) bağlı olarak 2000 yılında oluşturulmuştur. Güvenlik açığı yönetimi verilerini, yönetimini, ölçümünü ve uyumluluğunu içeren bir veri tabanıdır. NVD, güvenlik kontrol listesi referansları, güvenlikle ilgili yazılım kusurları, yanlış yapılandırmalar, ürün adları ve etki metrikleri veri tabanlarını içermektedir. ABD Ulusal Güvenlik Bakanlığı'nın Ulusal Siber Güvenlik Bölümü tarafından desteklenmektedir (NVD 2020).

NVD çalışanlarının temel görevi CVE sözlüğünde yayınlanan güvenlik açığı listeleri üzerinde analiz yapmaktır. Bu aşamada CVE'lerde bulunan açıklamaları, referansları toplayabildikleri tüm ek verileri kullanırlar. NVD veri tabanında yayınlanan veriler için temel olarak; ilişkili etki metrikleri (Ortak Güvenlik Açığı Puanlama Sistemi - CVSS), güvenlik açığı türleri (Ortak Zayıflık Numaralandırması - CWE) ve uygulanabilirlik ifadeleri (Ortak Platform Numaralandırması - CPE) ve diğer ilgili meta veriler eklenir. Ancak NVD atadığı öznitelikler için güvenlik açığı testi yapmamaktadır. Yeni bilgilere göre verilerin CVSS puanları ve uygulanabilirlik ifadeleri değişebilir (NVD 2020).

2.2.3. Exploit-DB

Offensive Security tarafından 2004 yılında kamu hizmeti olarak kar amacı gütmeyen bir proje olarak doğmuştur. CVE sözlüğünde yayınlanan listeler ile uyumludur. Temel amaç, en kapsamlı istismar arşivine hizmet etmek ve bunları serbestçe erişilebilen ve gezinmesi kolay bir veri tabanında sunmaktır. Exploit veri tabanı, tanımlardan ziyade CVE listelerinde yayınlanan açıkların istismarlar edilebilirliğini gösteren PoC kodları (Proof of Concept Code) ve kavram kanıtları sağlamaktadır. PoC, bir saldırganın güvenlik açığını nasıl istismar edebileceğini açıklayan basit bir kod parçasıdır. Bu özelliği veri tabanını hemen eyleme geçirilebilir verilere ihtiyaç duyanlar için değerli bir kaynak haline getirmektedir. Ancak PoC kodu bulunmayan veriler ihmal edilmektedir (ExploitDB 2020).

2.2.4. SecurityFocus

Bağımsız güvenlik uzmanları tarafından oluşturulan bir topluluk tarafından 1999'da kurulmuştur. CVE listelerini temel alan SecurityFocus Güvenlik Açığı Veri Tabanı, güvenlik profesyonellerine tüm platformlar ve hizmetler için güvenlik açıkları hakkında en güncel bilgileri sağlamayı amaçlamaktadır. Bunun için haber bültenleri, teknik makaleler ve yazılar yayınlamaktadır. Posta listeleri sayesinde dünyanın her yerindeki üyeleri ile güvenlik sorunlarını tartışmaya olanak sağlamaktadır (SecurityFocus 2020). SecurityFocus en önemli ve en saygın güvenlik açığı veri tabanlarından biridir. NVD veri tabanındaki tanımlamalara göre SecurityFocus listelerindeki tanımlamalar güvenlik açığının etkisini ve sömürülebilirliğini daha spesifik olarak açıklamaktadır(Fang et al. 2020).

2.2.5. Rapid7

Birleşik güvenlik yönetimi çözümleri sağlayan bir güvenlik şirketi olan Rapid7 2000 yılında kurulmuştur. Güvenlik uzmanları ve araştırmacıların incelemesi için güvenlik açığı ve istismar için teknik ayrıntılar içeren bir veri tabanıdır. CVE listeleri ile uyumludur. Veri tabanında yayınlanan tüm istismar kodları Metasploit çerçevesine dahil edilmiştir. Kamu politikası olarak, tüketicilere fayda sağlayan ve sorumlu siber güvenlik uygulayıcılarını savunan politikaları, standartları ve mevzuatı şekillendirmek için hükümetler, şirketler, kar amacı gütmeyen kuruluşlar ve uzmanlarla birlikte çalışmayı benimsemiştir (Rapid7 2020).

2.2.6. Snyk

Snyk veritabanı, açık kaynak projeleri için ücretsiz olan kod değerlendirme araçları sağlayan kar amaçlı bir şirket tarafından oluşturulmuştur. Açık kaynak projelerinin geliştirilmesini desteklemek ve güvende kalmalarını sağlamaya yardımcı olmayı misyon edinmiştir(Snyk 2020).

2.2.7. SARD – Software Assurance Reference Dataset Project

Koleksiyon 2005 yılında Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından oluşturulmaya başlanmıştır. İlk duyurulduğunda Standart Referans Veri Kümesi - SRD kısaltılmış olarak adlandırılmıştır. Bu ad 2014 yılında Yazılım Güvencesi Referans Veri Kümesi – SARD kısalması olarak değiştirilmiştir. Bir dizi bilinen güvenlik açıklarını sağlayarak kullanıcıların, araştırmacıların ve yazılım geliştiricilerin güvenlik araçları geliştirmelerine yardımcı olmayı amaçlamaktadır. Ayrıca test senaryoları tasarımları, kaynak kodları, ikili dosyalar gibi verileri de sağlayarak yazılım yaşam döngüsünün tüm aşamalarını barındıran bir arşiv sunmaktadır. Bu, son kullanıcıların geliştirdikleri araçlarını ve araç geliştirme yöntemlerini test etmelerini ve değerlendirmelerini sağlar. Veri kümesi, "gerçek" (üretim), "suni" (test etmek için yazılmış) ve "akademik" (öğrencilerden) test senaryolarını içerir. Bu veri tabanı aynı zamanda bilinen hatalara ve güvenlik açıklarına sahip gerçek bir yazılım uygulaması içermektedir. Veri kümesi, çok çeşitli olası güvenlik açıklarını, dilleri, platformları ve derleyicileri kapsamaktadır. Veri kümesinin, birçok katılımcıdan test senaryoları toplayarak büyümektedir(SARD 2020).

3. Araştırma Sonuçları ve Tartışma

3.1. Tartışma

Yazılım güvenlik açıkları, insanların günlük işlemlerinin büyük çoğunluğunu bilgi sistemlerinin desteğiyle gerçekleştirildiği çağdaş toplumda büyük bir tehdit oluşturmaktadır. NVD veri tabanı istatistiklerine göre yazılım güvenlik açıklarında 2016'dan günümüze kadar %100 artış görülmektedir. Yazılım güvenlik açıklarındaki bu büyük artış, bu konunun siber güvenlik camiası için önemli bir araştırma alanı haline gelmesine neden olmuştur. Sürekli artan güvenlik açıkları sorunu, araştırmacıların onları tahmin etmeye çalışmasına yol açmaktadır (Kekül et al. 2021).

NVD'de yayınlanan Common Vulnerabilities and Exposures (CVE) teknik raporları 1988 yılından beri tespit edilebilmiş tüm güvenlik açıklarının doğal dille yazılmış bir kümesidir. Bu yönü

ile makinelerin değil insanların anlaması ve yorumlaması öngörülmüştür. Ancak NVD veri setindeki bilgilerin yetersizliğinden dolayı farklı topluluklar tarafından CVE verileri için oluşturulmuş farklı veri setleri bulunmaktadır. Ancak genellikle güvenlik açıklarına yeni özellikliler eklese de bu veri tabanları da doğal dille yazılmıştır. Ancak bu veri setleri temelde NVD gibi CVE teknik raporlarını temel almakta ve aynı güvenlik açıklarına yeni özellikler ve açıklamalar getirmektedir. Örneğin NVD'de bulunan açıklamaların yetersiz olduğunu düşünen uzmanlar tarafından SecurityFocus veri tabanı oluşturulmuştur. Aynı şekilde güvenlik açıklarının PoC kodlarının yer aldığı ExploitDB veri tabanı istismar kodu bulunan bir veri setidir. Farklı özelliklere sahip açık kaynak farklı veri tabanları mevcuttur. Ancak vurgulandığı gibi tamamı insanların anlayabileceği doğal dille hazırlanmış veri tabanlarıdır. Bunların doğrudan makine öğrenmesi algoritmalarında kullanılması mümkün değildir.

4. Sonuç

Sonuç olarak CVE sözlüğü ortak güvenilir bir standart ve alt yapı sağlamaktadır. Diğer veri tabanları CVE listelerini kullanarak kendi veri setlerini güncellemektedirler. NVD bu listelerdeki ham verilere uzman görüşleri ile yeni özellikler ve açıklamalar eklemektedir. Ancak eklenen bu özellikler ve açıklamaların yeterliliği tartışma konusudur. Bu sorundan dolayı alanın uzmanlarından oluşan topluluklar bu listelere daha anlaşılır ve kullanışlı özellikler ekleyerek yeni veri setleri oluşturmuşlardır. SecurityFocus ve ExploitDB bunların en başında gelen veri tabanlarıdır. Bu veri tabanlarının NVD'den temel farkları daha anlaşılır bilgiler içeren açıklamaları ve istimrar kodlarını içeren yapılarıdır. Rapid7 ve Snyk gibi ticari veri tabanları güvenlik açıklarının ticari değerleri yasal yollarla değerlendirilmektedirler. Ayrıca bu durum ticari güvenlik çerçevelerinin geliştirilmesini sağlayarak daha güvenli yazılım ürünlerinin çıkması için sektörü desteklemektedir. SARD veri tabanın sağladığı güvenlik test senaryoları en temel özelliğidir. Bu durum yazılım test mühendisliğinin gelişmesinde önemli katkılar sunmaktadır. Alandaki her veri tabanı farklı bir özelliği ile öne çıkmaktadır.

Gelecekteki çalışmalarımızda ilk hedef geniş bir veri tabanı imkanı bulunmasına rağmen makine öğrenmesi ve derin öğrenme algoritmalarında doğrudan kullanılamayan bu büyük yapıdaki verilerden tek ve kapsamlı işlenmiş ve yapılandırılmış bir veri tabanı oluşturmak olacaktır. Oluşturulacak veri tabanı açık kaynak prensibi ile araştırmacıların kullanımına sunulacaktır.

5. Teşekkür

Bu çalışma Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından 121E298 proje numarası ile desteklenmektedir.

Kaynakça

Committee, IEEE Standards Coordinating, and others. 1990. "IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos." *CA: IEEE Computer Society* 169.

CVE. 2020. "CVE." *Common Vulnerabilities and Exposures*. https://cve.mitre.org (July 25, 2020).

ExploitDB. 2020. "Exploit Database." https://www.exploit-db.com (July 25, 2020).

Fang, Yong, Yongcheng Liu, Cheng Huang, and Liang Liu. 2020. "Fastembed: Predicting Vulnerability Exploitation Possibility Based on Ensemble Machine Learning Algorithm." *PLoS ONE* 15(2): 1–28. http://dx.doi.org/10.1371/journal.pone.0228439.

Ghaffarian, Seyed Mohammad, and Hamid Reza Shahriari. 2017. "Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey." *ACM Computing Surveys* 50(4).

Kekül, H., Ergen, B., & Arslan, H. (2021). A multiclass hybrid approach to estimating software vulnerability vectors and severity score. Journal of Information Security and Applications, 63, 103028.

Kekül, H., Ergen, B., & Arslan, H. (2021). A New Vulnerability Reporting Framework for Software Vulnerability Databases.

Krsul, Ivan Victor. 1998. "Software Vulnerability Analysis." Purdue University.

"Mitre Corporation." 2020. https://www.mitre.org (July 25, 2020).

Miyamoto, Daisuke, Yasuhiro Yamamoto, and Masaya Nakayama. 2017. "Text-Mining Approach for Estimating Vulnerability Score." *Proceedings - 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2015*: 67–73.

NVD. 2020. "NVD." *National Vulnerability Database*. https://nvd.nist.gov (July 25, 2020).

Ozment, Andy. 2007. "Improving Vulnerability Discovery Models: Problems with Definitions and Assumptions." *Proceedings of the ACM Conference on Computer and Communications Security*: 6–11.

Raducu, Razvan, Gonzalo Esteban, Francisco J.Rodríguez Lera, and Camino Fernández. 2020. "Collecting Vulnerable Source Code from Open-Source Repositories for Dataset Generation." *Applied Sciences (Switzerland)* 10(4).

Rapid7. 2020. "Rapid7." https://www.rapid7.com/db/ (July 25, 2020).

SARD. 2020. "SARD-Software Assurance Reference Dataset Project." https://samate.nist.gov (July 25, 2020).

SecurityFocus. 2020. "SecurityFocus." https://www.securityfocus.com (July 25, 2020).

Snyk. 2020. "Snyk." https://snyk.io (July 25, 2020).

Spanos, Georgios, and Lefteris Angelis. 2018. "A Multi-Target Approach to Estimate Software Vulnerability Characteristics and Severity Scores." *Journal of Systems and Software* 146: 152–66. https://doi.org/10.1016/j.jss.2018.09.039.

Theisen, Christopher, and Laurie Williams. 2020. "Better Together: Comparing Vulnerability Prediction Models." *Information and Software Technology* 119(August 2019).

Wu, Xiaoxue et al. 2020. "CVE-Assisted Large-Scale Security Bug Report Dataset Construction Method." *Journal of Systems and Software* 160: 110456. https://doi.org/10.1016/j.jss.2019.110456.

Yang, Heedong, Seungsoo Park, Kangbin Yim, and Manhee Lee. 2020. "Better Not to Use Vulnerability's Reference for Exploitability Prediction." *Applied Sciences (Switzerland)* 10(7).