Mikroservis Tabanlı Ağ Uygulamalarında Zararlı Davranışların Saptanması
Detecting Malicious Behavior in Microservice Based Web Applications

Mustafa Özbek
Computer Engineering Department
Istanbul Technical University
TR34469, Istanbul, Turkey.
ozbekmus@gmail.com

Mehmet Tahir Sandıkkaya
Computer Engineering Department
Istanbul Technical University
TR34469, Istanbul, Turkey.
sandikkaya@itu.edu.tr

***Özetçe* —İnternet'teki zararlı davranışların sadece artan karmaşıklığı değil aynı zamanda sürekli artan yeni saldırı metotları da İnternet hizmetlerinin güvenilirliğini sarsarak modern topluma bir tehdit oluşturmaktadır. Bu çalışmada, mikroservis tabanlı bir ağ uygulamasındaki zararlı davranışlar, CRUD (ekleme, okuma, güncelleme, silme) davranış kalıpları gözlenerek saptanmaktadır. Bu yayının amacı, zararlı kullanıcıları (hatta güvenilir bir kullanıcının kötü niyetli ilk teşebbüsünü), mikroservislerin kullanım karakteristiklerine göre davranış gerçekleşir gerçekleşmez belirlemektir. Önerilen yöntem, OWASP Vakfı'nın en tehlikeli 10 ağ uygulama riskleri listesini göz önüne almaktadır. Bu bağlamda, sözü edilen saldırıları içeren bir veri kümesi, iyi huylu yaygın davranışlar da kapsanarak oluşturulup mikroservis tabanlı bir örnek ağ uygulamasında davranışların karakteristiği ölçülmüştür. İlgili veri kümesi daha sonra makina öğrenmesi algoritmalarından RandomForest, NaiveBayes, J48, AdaBoost, ZeroR, Bagging, Lojistik Regresyon, K-Star algoritmaları ile zararsız ve zararlı davranışlar olarak iki sınıfa ayrılmıştır. Deneyler sırasında karşılaşılan en iyi zararlı davranış algılama doğruluğu % 99,36 olarak RandomForest sınıflandırma algoritması ile gerçekleşmiştir. Zararlı davranışın belirlenmesinden sonra, ilgili kullanıcının mikroservislere erişimi önlenerek kaynakların boşa harcanması engellenmektedir.**

Anahtar Kelimeler*—*Zararlı Davranış, Ağ Saldırıları, Mikroservis, Makina Öğrenmesi, Ağ Uygulaması, Veri Sınıflandırma.

***Abstract*—Not only the increased complexity of the malicious acts on the Internet, but also the continuous increase of new attack methods compromise Internet-based services as a threat to the modern society. In this study, malicious behavior in a microservices-based web application is detected by measuring the patterns of CRUD (create, read, update, delete) access. The aim of this paper is to detect malicious users (or even the first malicious attempt of a trustworthy user) as soon as the action occurred according to the characteristics of the sequential use of microservices. The proposed approach renders OWASP Foundation's Top 10 critical web application security risks as possible attack vectors. Thus, a data set including such attacks together with mostly benign behavior is generated and measured on the microservices-based web application. The data set is then used to determine benign and malicious classes of behavior using RandomForest, NaiveBayes, J48, AdaBoost, ZeroR, Bagging, Logistic Regression and K-Star machine learning algorithms. The best malicious behavior detection accuracy encountered during experiments is an auspicious 99.36% using RandomForest classification algorithm. After the classification of malicious behavior, the respective user's further access to the microservices could be blocked to prevent the waste of resources.**

Keywords*—*Malicious Behavior, Web Attacks, Microservice, Machine learning, Web Application, Data Classification.

I. GİRİŞ

Zararlı davranışlar, önemli ekonomik kayıplar yaratmakta, mahrem bilgileri sızdırmakta, ağ uygulamalarını tehdit etmektedir. Bir bilgisayar sistemine zarar veren davranışın kaynağı güvenilen bir kullanıcı ya da şüpheli bir misafir kullanıcı olabilir. Ağda sunulan hizmeti kullanmak üzere kaydedilmiş güvenilen bir kullanıcı çoğu zaman virüs tarayıcı ya da sızma denetleme düzeneği gibi geleneksel güvenlik önlemlerinden etkilenmez. Oysa, bir kullanıcıya güvenilse dahi, kısa süreyle kullanıcının cihazlarını kullanan ya da erişim bilgilerini ele geçirmiş biri, tek seferde etkili bir saldırıda bulunabilir. OWASP (Açık Web Uygulamalarında Güvenlik Projesi) Kurumu, ağ uygulamalarına sıklıkla yapılan saldırıların listesini ve olası korunma yollarını yayınlamıştır [1], [2]. Bu yazıda, mikroservis yaklaşımıyla geliştirilmiş bir ağ uygulamasında, zararlı davranışları mikroservislerin olağan kullanım karakteristiklerinden ne kadar saptığına bakarak belirlemek üzere bir yöntem sunulmuştur. Çalışmada sunulan yöntemde, en büyük zorluk kullanıcı davranışlarının tamı tamına sınıflandırılmasıdır. Önerilen yöntemin zararsız davranışlar sırasında hizmeti kesmemesi ancak zararlı davranışları çeviklikle önlemesi beklenmektedir [3]. Bu nedenle, doğruluğu yüksek çözümler bulmak gereklidir.

Bu çalışmanın öncelikli amacı, bir ağ uygulamasının veritabanı ile haberleştiği hizmetleri mikroservisler biçiminde birbirinden yalıtarak tasarlayıp, elde edilecek güvenlik kazanımlarını belirlemektir. Yalıtılmış mikroservislere kullanıcı erişimleri ayrıntılı olarak düzenlenebildiğinden zararlı davranışta bulunan bir kullanıcı isteği belirlendikten sonra kullanıcının daha fazla eylemde bulunması kolaylıkla engellenebilmektedir [4]. Bu gerçeğe dayanarak, kullanıcıların davranışları mikroservisleri kullanım sıraları ve süreleri gözetilerek, mahremiyete zarar vermeden ölçülecektir. Ölçümün ardından yapılacak

sınıflandırma ile zararlı davranış belirlenebilir. Zararlı davranışta bulunan kullanıcının her biri yalıtılmış mikroservisler olarak sunulan kaynaklara erişimi kesilebilir. Ölçümlerde ele alınan ağ uygulamasında, alışıldık REST [5] yaklaşımıyla bir veritabanına erişmenin olası tüm yolları gösterilmiş, böylece kapsayıcı bir model kurulmuştur.

II. İLGİLİ ÇALIŞMALAR

Ağ hizmetlerinde zararlı davranışların saptanması üzerine yapay veri kümesi ile yapılan benzer çalışmada davranışların sınıflandırılmasında % 87,6 düzeyinde başarı RandomForest algoritması ile sağlanmıştır [6]. Ağ hizmetleri için yapılan bu çalışmada zararlı davranışın belirlenmesinin ardından kullanıcının erişimi kısıtlanabilmekte ancak tamı tamına engellenememektedir. Önerilen yöntemde, veritabanına erişim türüne göre ayrılmış ve yalıtılmış mikroservislerle oluşturulan bağımsız bir katman olarak kurgulanmıştır. Böylece, zararlı davranışın belirlenmesiyle kullanıcının mikroservis erişimi engellenerek, ağ uygulamasının güvenliği sağlanabilecektir.

Bundan başka, oturum bazında sıralama, işlemlerin çalışma frekansı ve sistem çağrılarını izleyerek önerilen benzer bir sınıflandırmayı zararlı yazılımlar için yapan bir çalışma vardır [3]. Zararlı yazılım belirlemek üzere yapılan bir başka çalışmada [7] J48 ve ANN (All-Nerest-Neighbor) algoritmaları ile % 96 oranında başarı oranı sağlanmıştır.

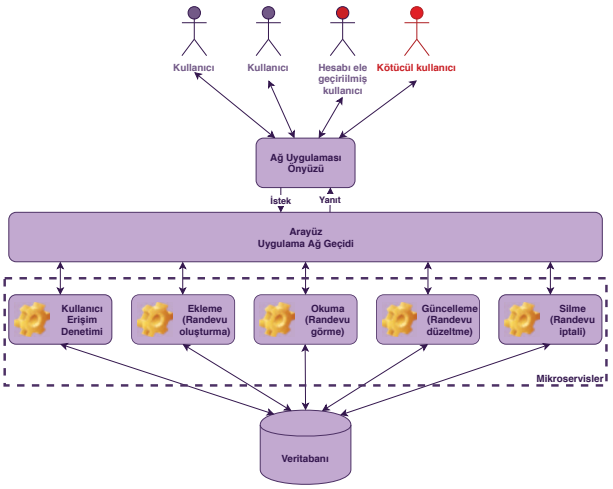
III. DENEY

Önerilen yöntemi göstermek üzere REST [5] yaklaşımıyla veri işlemeye olanak sağlayan, kullanıcıların çeşitli rezervasyonlarını yönetebildikleri bir ağ uygulaması seçilmiştir. Bu uygulama, veri erişimlerine göre (CRUD –ekleme, okuma, güncelleme ve silme) ayrılmış bağımsız mikroservisler kullanılarak gerçekleştirilmiştir. Mikroservis kullanımının yönetilmesi için bir ara katman kullanılmıştır. Şekil 1'de ele alınan uygulamanın yapısı görülebilir.

Şekil 1'deki tasarıma göre kullanıcıların erişebildikleri bir önyüz ve buraya gelen istekleri mikroservislere aktarmak üzere bir ağ geçidi bulunmaktadır. Veritabanı sorguları mikroservisler ile gerçekleştirildikten sonra uygulamanın oluşturduğu yanıt yine ağ geçidi üzerinden önyüze taşınır.

Gerçekleme sırasında Zuul API ağ geçidi [8] ve Eureka Server kulanılmış, mikroservisler Spring Boot Initializr [9] ile Java programlama dilinde yazılmıştır. Her mikroservisin veritabanı bağlantısı güvenlik ve yalıtım ilkeleri gözetilerek ayrı bağlayıcılar ile sağlanmış, böylece her bir mikroservisin veritabanına erişimi ayrıca yönetilmiştir [4].

Yukarıda belirtilen uygulama esas alınarak OWASP tarafından listelenen saldırılar ve bunun bir iki mertebe üstünde olağan istek Apache JMeter aracılığı ile oluşturulmuştur [10]. Uygulamaya gönderilen bu isteklerin mikroservisleri kullanım sırası ve mikroservisleri kullanım süreleri ölçülmüş ve saklanmıştır. Toplamda % 15 ve % 1 zararlı davranış oranıyla 1 500 000'er bağlantı içeren iki ayrı veri kümesi üzerinde çalışılmıştır. İki farklı oran kullanılmasının ardındaki neden, İnternet'teki saldırı trafiğinin oranının % 1'den % 30'a dek farklı yüzdelerle raporlanmış olmasıdır [11], [12].



Şekil 1. Zararlı davranışların belirlenmesi için geliştirilen örnek ağ uygulamasının yapısı.

Her mikroservisin çalışmaya başlaması ve çalışmasının sonlanması izlenerek mikroservislerin çalışma süreleri ölçülmüş, mikroservis kullanım sıraları saklanmıştır. JMeter ile oluşturulan eş zamanlı yapay trafik ile çok sayıda güvenilir ve kötücül kullanıcının aynı anda hizmet almasına öykünülerek gerçek hayata yakın bir veri kümesi elde edilmiştir. Örnek uygulamada Java programlama dili kullanılmış olsa da mikroservislerin diğer programlama dilleri ile yazılması ya da kullanılması olanaklıdır [13]. Gerçekte, mikroservis yaklaşımında hizmetlerin ayrılması ve yalıtımı gözetildiğinden her bir mikroservisin bir başka programlama dilinde yazılmış olması, tanım gereği, mümkündür. Bundan başka, örnek olarak sunulan uygulamadan farklı olarak, mikroservislerin başka altyapılar üzerine kurulabileceği ve farklı veritabanlarıyla haberleşebileceği de akla getirilmelidir.

IV. YÖNTEM

Sunulan ağ uygulamasında, kullanıcıların her isteği ağ geçidine gönderilmektedir. Kullanıcılarla ilişkilendirilmiş bu istekler, ağ geçidinden ilgili mikroservise o anda yürüttüğü görevler de gözetilerek yük dağılımı yapıldıktan sonra paylaştırılmaktadır. Uygulamada var olan tüm mikroservisler kendisine gelen istekleri aşağıda sayılan niteliklere göre ölçerek kaydeder.

- Kullanıcı adı
- Yapılan işlem
- Mikroservis yaftası
- İşlem süresi
- İşlem anındaki işlemci yükü
- Daha önce yürütülmüş işlemlerin sıralı listesi

Çalışmada elde edilen tüm sınıflandırma sonuçları bu niteleyiciler kullanılarak oluşturulmuştur.

Veri kümesi kullanılarak bir davranışın zararlı olup olmayışının sınıflandırılması için iyi bilinen makina öğrenmesi algoritmaları kullanılmıştır [14].

TABLO I. HATA MATRİSİ

		Öngörülen	
		Zararlı davranış	Olağan davranış
Gerçek	Zararlı davranış	TP (gerçek pozitif)	FN (yanlış negatif)
	Olağan davranış	FP (yanlış pozitif)	TN (gerçek negatif)

Sınıflandırma yöntemlerini kıyaslarken kullanılan temel kavramlar doğruluk, hata oranı, kesinlik, duyarlılık ve F-ölçütüdür [15]. Sunulan yöntemin başarısı, seçilen niteleyiciler, sınama kümesine ve veri kümesine atanan örnek sayıları ile yanlış sınıfa atılan örnek sayılarının niceliğiyle belirlenmektedir. Sınamada ulaşılan sonuçların başarımı Tablo I'de verilen dört durum ile gösterilir. Veri kümesinde bir yafta ile sınıfı belirlenmiş veriler *gerçek* başlığı altına gelecektir. *Öngörülen* başlığında da kullanılan sınıflandırma algoritmalarına göre belirlenen sonuçlar yer almaktadır.

- TP (Gerçek Pozitif) : Davranış, veri kümesinde zararlı olarak yaftalanmıştır ve sınıflandırma algoritmasına göre de zararlı bulunmuştur.
- FP (Yanlış Pozitif) : Davranış, veri kümesinde olağan olarak yaftalanmıştır ancak sınıflandırma algoritmasına göre zararlı bulunmuştur. Diğer deyişle, uygulamayı kullanma hakkı olan birinin erişimi haksız yere engellenmektedir.
- FN (Yanlış Negatif) : Davranış, veri kümesinde zararlı olarak yaftalanmıştır ancak sınıflandırma algoritmasına göre olağan bulunmuştur. Diğer deyişle, zararlı bir davranış yürütülmektedir ancak önerilen yöntemle bu davranış sezilememiştir.
- TN (Gerçek Negatif) : Davranış, veri kümesinde olağan olarak yaftalanmıştır ve sınıflandırma algoritmasına göre de olağan bulunmuştur.

Tabloda satırlar davranışların asıl sınıflandırmasını, sütunlar ise makina öğrenmesi ile kurulan modelin öngördüğü sınıflandırmayı gösterir [16].

Önerilen yöntemde, hem ağ hizmetlerinin devamlılığı hem de zararlı davranışın saptanmasıyla güvenliğin sağlanması gerekli görüldüğünden, oluşturulan modelin doğruluk oranının yüksek olması yeğlenir. Böylece olağan davranışın engellenmesi ya da zararlı davranışın hizmete zarar vermesi en aza indirilmiş olur. Bu oran, doğru sınıflandırılmış istek sayısının $(TP + TN)$, toplam istek sayısına $(TP + TN + FP + FN)$ oranıdır. Hata oranı, bu değerin 1'e tamlayanıdır. Diğer deyişle, yanlış sınıflandırma oranını görmek için pay değerine yanlış sınıflandırılmış istek sayısı olan $(FP + FN)$ yazılır [15].

$$Doğruluk = {}^{TP+TN}/_{FP+TP+FN+TN} \quad (1)$$

$$Hata\ Oranı = {}^{FP+FN}/_{FP+TP+FN+TN} \quad (2)$$

Bu ikisinin dışında kesinlik, duyarlılık ve F-ölçütüne bakılarak başarım belirlenebilir. Kesinlik, doğru öngörülen isteklerin tüm doğru öngörülere oranıdır. Duyarlılık ise doğru sınıflandırılmış pozitif isteklerin sayısının toplam pozitif istek sayısına oranıdır. Son olarak, F-ölçütü, kesinlik ve duyarlılığın harmonik ortalamasıdır.

$$Kesinlik = {}^{TP}/_{TP+FP} \quad (3)$$

$$Duyarlılık = {}^{TP}/_{TP+FN} \quad (4)$$

$$F\text{–}Ölçütü = 2 \times {}^{Kesinlik \times Duyarlılık}/_{Kesinlik + Duyarlılık} \quad (5)$$

Kullanıcı davranışları sınıflandırılırken 10 parçalı çapraz geçerleme (10-fold cross validation) kullanılmıştır. Bu yöntem ile veri seti 10 farklı parçaya bölünerek, ayrı ayrı 10 eğitim ve 10 test yapılır. Sonuçta, her eğitim ve test senaryosu için her seferinde modellemeyi sadece o aşamada sınıflandırmaya verilen eğitim kümesine göre hesaplayarak yeni bir doğruluk oranı elde edilir ve oranların ortalaması alınarak sınıflandırıcının ortalama başarı oranı bulunur [15]. Bu çalışmada, makina öğrenmesi modellerinin oluşturulması sırasında Weka yazılımı ve beraberinde sunulan makina öğrenmesi algoritmaları kullanılmıştır [17]. Ancak, yazılımla birlikte sunulan algoritmaların varsayılan parametreleri düzeltilerek başarım artırılmıştır.

Weka çatısında işlenen veri kümesine aşağıda kısaca açıklanmış sınıflandırma algoritmaları uygulanmıştır.

- **NaiveBayes** Temeli Bayes teoremine dayanır. Bir eleman için her durumun olasılığını hesaplar ve olasılık değeri en yüksek olana göre sınıflandırır [15].
- **RandomForest** Hiper parametre kestirimi yapılmadan da iyi sonuçlar vermesi hem regresyon hem de sınıflandırma problemlerine uygulanabilir olmasından dolayı önde gelen makina öğrenmesi modellerinden biridir [16].
- **J48** Doğrusal olmayan ve küçük boyutlu verilerin sınıflandırmasını yapan bir C4.5 karar ağacıdır [17].
- **AdaBoost** İçerdiği sınıflandırıcıların teker teker eğitim setinde çalışması sonucu doğruluk oranlarına göre ağırlıklarını dikkate alarak eğitmeyi sağlayan bir yöntemdir [16].
- **ZeroR** Veriyi, değerlerine göre gruplayarak gruplarda biriken veri sayılarını birbirleriyle kıyaslar ve frekansı yüksek olan grubu esas alarak bundan sonra gelecek tüm verilerin o gruba ait olduğuna karar verir [17].
- **Bagging** Var olan bir eğitim setini kullanarak yeni eğitim setleri oluşturup temel öğreniciyi yeniden eğitmeyi sağlayan bir yöntemdir [17].
- **Lojistik Regresyon** Bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan bir veri kümesini çözümlemek için kullanılan istatistiksel bir yöntemdir. Tüm verileri, ikili bir değişkenle ölçer, yani sadece iki olası sonuca göre sınıflandırır [16].
- **K-Star** Benzerlik fonksiyonlarıyla belirlendiği gibi, eğitim örnekleriyle aynı sınıftaki test örneğinin sınıfıdır. Diğer örnek tabanlı öğrenenlerden entropi tabanlı mesafe fonksiyonu kullanması yönüyle farklıdır [18].

Elde edilen sonuçların hata matrisine bakılarak parametreleri iyileştirilmiş her algoritma için doğruluk, kesinlik, duyarlılık ve F-ölçütü değerleri hesaplanmıştır. Gerçek veriye yakın bir veri kümesiyle çalışılması, gerçek hayatta da karşılaşılabilecek bazı çıkmazları modellere işlemiştir. Olağan bir kullanıcının, düzenli kullandığı işlemler dışında bir işlemi yaparken yaptığı hatalar ya da yeni görevleri yürütmesi de bazen zararlı davranış olarak sınıflandırılabilmektedir. Bu tür

TABLO II. KULLANILAN SINIFLANDIRMA ALGORİTMALARININ BAŞARIMI

Algoritma	Kümedeki zararlı davranışlar	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü
RandomForest	% 1	99,24	57,15	95,94	71,63
J48	% 1	99,17	57,49	96,01	71,91
AdaBoost	% 1	98,96	57,11	95,87	71,58
Naive Bayes	% 1	98,87	57,03	95,89	71,52
ZeroR	% 1	97,14	56,93	95,07	71,21
K-star	% 1	96,97	56,99	95,34	71,34
Bagging	% 1	96,74	56,21	94,31	70,44
Lojistik Regresyon	% 1	95,21	54,25	93,78	68,74
RandomForest	% 15	99,36	96,48	99,36	97,80
J48	% 15	99,21	98,31	98,18	98,24
AdaBoost	% 15	99,01	98,99	98,90	98,96
Naive Bayes	% 15	98,97	98,77	98,89	98,83
ZeroR	% 15	97,73	92,73	94,78	95,26
Bagging	% 15	97,73	97,65	97,75	97,70
Lojistik Regresyon	% 15	96,45	96,07	96,18	96,13
K-star	% 15	94,38	93,97	93,89	93,93

TABLO III. % 15 ZARARLI DAVRANIŞ İÇEREN VERİ KÜMESİNDE RANDOM FOREST İÇİN OLUŞAN HATA MATRİSİ

		Öngörülen		
		Zararlı davranış	Olağan davranış	Toplam
Gerçek	Zararlı davranış	223 567	1433	225 000
	Olağan davranış	8161	1 266 839	1 275 000
	Toplam	231 728	1 268 272	1 500 000

hatalı kullanımların da doğru sınıflandırılması, daha büyük ve gerçek veri kümelerinin elde edilmesiyle başarılabilir.

V. DENEY ÇIKTILARI VE VARGI

Oluşturulan makina öğrenmesi modellerinin başarımı toplu halde Tablo II'de sunulmuştur. En başarılı model, % 99,36 doğrulukla kNN ve N-Gram ile zenginleştirilen RandomForest algoritmasıyla elde edilmiştir ve hata matrisi Tablo III'de verilmiştir. Bir karar ağacı algoritması olan ve temeli ID3 ve C4.5 algoritmalarına dayanan J48 ile AdaBoost algoritmaları bunu izlemektedir [14]. İstatistiksel bir algoritma olan NaiveBayes, regresyona dayanan algoritmalardan Lojistik Regresyon ve örnek tabanlı sınıflandırma algoritmalarından K-star algoritmalarının da umut verici doğruluğa ulaştıkları görülmektedir. Dikkat çeken bir nokta, veritabanı erişim sıralamasının zararlı davranışları belirlemede etkin oluşudur. Bir başka deyişle, kullanıcıların sadece veriyi hangi sırayla işlediklerine bakılarak, ancak verinin kendisi ya da uygulamanın çalışma mantığı göz ardı edilerek zararlı davranışların sınıflandırılması olanaklıdır. Bu, aynı anda hem kullanıcı mahremiyetini gözeten hem de oldukça yüksek yaklaşıklıkla güvenliği sağlayan bir çözümdür. N-Gram ile zenginleştirilmiş sınıflandırmaların daha başarılı sınıflandırıcılar olmasıyla işlem sırasını gözeterek zararlı davranışların sınıflandırılabileceği varsayımı onaylanmıştır.

VI. SONUÇ VE ÖNERİLER

Sonuçlar incelendiğinde hem % 15 hem de % 1 zararlı davranış içeren veri kümelerinde RandomForest algoritmasının modelin sınanması sırasında % 99,24 – % 99,36 doğruluk oranlarıyla en iyi sonucu verdiği görülmektedir. Oluşturulan veri kümesinin, mikroservislerin kullanım sırasını N-gram ile niteleyicileriyle zenginleştirmesi, makina öğrenmesi modelinin başarımını arttırmış görünmektedir. Bunun dışında, hemen hemen tüm sınıflandırma yaklaşımlarında yüksek oranda doğruluk gözlenmesi ölçülen niteleyicilerin mikroservis mimarisinde güvenliği ve mahremiyeti artırmaya yönelik yaklaşımlarda kullanılabileceğinin göstergesidir. Mikroservis yaklaşımı, zararlı davranışın belirlenmesinden hemen sonra zarar veren kullanıcının veriye erişiminin engellenmesiyle ağ uygulamasının kaynaklarının yok yere kullanılmasının önüne geçer. Sunulan örnek uygulamada, asıllama aşamasını geçtikten sonra zararlı davranışta bulunan kullanıcı kaynaklara bir süre erişebiliyorsa da davranışın saptanmasıyla birlikte mikroservis erişimi engellendiğinden veri sızıntısı en aza indirilir. Önerilen yöntemin kullanılmaması durumunda, bir kere asıllanan kullanıcı, veritabanı üzerinde uygulamanın öngördüğü tüm işlemleri sınırsızca yapabilmektedir. Bunun ötesinde, mikroservislerin anlatıldığı biçimde kullanılması sadece zararlı davranışların önlenmesi sırasında değil, mikroservisler aracılığıyla hata hoşgörüsünün artırılması ve toparlanmanın hızlanması ile de yararlanırlık beklentilerine katkı sağlar.

KAYNAKLAR

[1] "OWASP Top 10 - 2013 The Ten Most Critical Web Application Security Risks," The Open Web Application Security Project, Tech. Rep., 2013, Available: www.owasp.org/index.php/Top_10_2013-Top_10.

[2] "OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks," The Open Web Application Security Project, Tech. Rep., 2017, Available: www.owasp.org/index.php/Top_10-2017_Top_10.

[3] R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen, and A. Czech, "Analysis of malware behavior: Type classification using machine learning," in *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. IEEE, 2015, pp. 1–7.

[4] C. Richardson. (2019, Jan) Pattern: Database per service. [Online]. Available: microservices.io/patterns/data/database-per-service.html

[5] R. T. Fielding and R. N. Taylor, *Architectural styles and the design of network-based software architectures*. University of California, Irvine Irvine, USA, 2000.

[6] C. D. Özdemir, M. T. Sandıkkaya, and Y. Yaslan, "Classifying Malicious Thread Behavior in PaaS Web Services," in *Proceedings of the 8th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER*, INSTICC. SciTePress, 2018, pp. 418–425.

[7] Y. Fan, Y. Ye, and L. Chen, "Malicious sequential pattern mining for automatic malware detection," *Expert Systems with Applications*, vol. 52, pp. 16–25, 2016.

[8] (2018, Dec) Zuul Proxy Server. [Online]. Available: github.com/Netflix/zuul

[9] (2019, Jan) Spring initializr. [Online]. Available: start.spring.io

[10] (2019, Jan) Apache JMeter. [Online]. Available: jmeter.apache.org

[11] "Imperva incapsula bot traffic report," Imperva, Tech. Rep., Jan 2017, Available: www.incapsula.com/blog/bot-traffic-report-2016.html.

[12] R. McMillan. (2008, Apr) Up to three percent of internet traffic is malicious, researcher says. [Online]. Available: www.csoonline.com/article/2122506/up-to-three-percent-of-internet-traffic-is-malicious--researcher-says.html

[13] M. Mazzara and B. Meyer, *Present and Ulterior Software Engineering*. Springer International Publishing, USA, 2017.

[14] R. D. King, C. Feng, and A. Sutherland, "Statlog: comparison of classification algorithms on large real-world problems," *Applied Artificial Intelligence an International Journal*, vol. 9, no. 3, pp. 289–333, 1995.

[15] E. Alpaydın, *Introduction to Machine Learning*. MIT press, 2014.

[16] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[17] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: A Machine Learning Workbench," in *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, 1994, pp. 357–361.

[18] J. G. Cleary and L. E. Trigg, "K*: An instance-based learner using an entropic distance measure," *Proceedings of Twelfth International Conference on Machine Learning*, pp. 108–114, 1995.