Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi 27(3),199-212, 2019

Journal of Engineering and Architecture Faculty of Eskisehir Osmangazi University 27(3),199-212, 2019

SON KULLANICILAR İÇİN ANOMALİ SALDIRI TESPİT SİSTEMLERİ

Kerim Can KALIPCIOĞLU[1*], Cengiz TOĞAY[2], Esra Nergis YOLAÇAN[3]

[1]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0003-4885-346X
[2]Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Bursa
ORCID No : https://orcid.org/0000-0001-5739-1784
[3] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0002-1655-0993

DOI : http://dx.doi.org/10.31796/ogummf.560747

Anahtar Kelimeler

Saldırı tespit sistemleri, Anomali tanıma, Makine öğrenmesi, Sistem çağrıları, Host-tabanlı

Öz

Günümüzde yaygın bir şekilde kullanılmakta olan imza tabanlı yaklaşımlar, özellikle sıfır gün saldırıları gibi henüz tespit edilmemiş saldırı vektörlerine karşı başarısız olmaktadırlar. Bu tip saldırılar genellikle en az bir sisteme zarar verdikten sonra tespit edilmektedir. Saldırıya ilişkin imza yapılan analizin ardından son kullanıcıların erişimine sunulur. Dolayısı ile bu süre zarfında kullanıcılar bu tip saldırılara karşı savunmasız kalırlar. Kritik noktalardaki bilgisayar sistemlerinin gerek güncelleme ve gerekse yeni uygulamaların kurulmasının ardından sıfır gün saldırıları ile karşılaşma riski bulunmaktadır. Bilindiği üzere, uygulamalar işletim sistemiyle sistem çağrı arayüzü üzerinden etkileşim kurarlar. Dolayısı ile uygulamalardan ya da sistemin tümünden toplanan sistem çağrı verisinde öğrenme sonrasında belirlenen anormal davranışlar bir saldırının varlığını işaret ediyor olabilir. Bu çalışmada, anomali tespit sistemleri için literatür taraması, kullanılabilecek veri kümeleri ve bunların karşılaştırmalı analizleri sunulmuştur.

ANOMALY INTRUSION DETECTION SYSTEMS FOR END-USERS

Keywords

Intrusion detection systems, Anomaly detection, Machine learning, System calls, Host-based

Abstract

Recent widely used signature-based approaches fail against unknown attack vectors, especially zero-day attacks. Such attack vectors are usually detected after giving damage to at least one system. Following the preparation of the attack signature, it is made available to end users. Therefore, during this period, users are vulnerable to such attacks. Computer systems at critical positions are at risk of encountering zero-day attacks after both updating and installing new applications. As is known, applications interact with the operating system using the system call interface. Hence, after the learning phase, abnormal behavior detected on system call information indicates a possible intrusion. This system call information may be collected both system-wide or only from user applications. In this study, literature review of the anomaly detection systems, data sets and comparative analyses of these datasets are presented.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 05.05.2019	Submission Date	: 05.05.2019
Kabul Tarihi	: 04.09.2019	Accepted Date	: 04.09.2019

1. Giriş

Bilgisayar sistemlerinin ekonomi ve ulusal güvenlik anlamında değer kazanmasıyla ticari kuruluşlar ve devletler arasındaki mücadele siber güvenlik ortamına taşınmıştır. Hatta terör ve suç örgütleri dahi fiziksel ortamda işlenen suçları desteklemek için interneti etkin bir şekilde kullanmaya başlamışlarıdır (Uma ve Ganapathi, 2013). Bu değişim saldırgan profiline de yansımıştır. Yeteneklerini göstermek için zararlı yazılım geliştiren saldırgan profili, siber saldırılardan maddi

* Sorumlu yazar; e-posta : kkalipcioglu@ogu.edu.tr

çıkar sağlamayı amaçlayan profesyonel saldırgan topluluğuna dönüşmüştür. Bu profesyonel saldırganlar fonlanarak önemli hedeflere siber saldırıların gerçekleşmesi sağlanmıştır (Farwell ve Rohozinski, 2011). Bu tip organize saldırılarla ilişkilendirilen gelişmiş kalıcı tehdit (APT - Advanced Persistent Threat) kavramı kurumların en büyük güvenlik çekincesi haline gelmiştir. Bu gelişmelerle beraber, bilgisayar kullanımının artması ve az yetenekli saldırganların gelişmiş yazılım araçlarını kullanarak siber saldırılara katılabilmesi bilgisayar suçlarından etkilenen kişilerin sayısını da önemli ölçüde artırmıştır.

Tüm bu gelişmelere karşı olarak; kullanıcı, kurum veya devletlerin güvenliğinin sağlanmasına yönelik çalışmalar hızlanmıştır. Araştırma kurumları tarafından verilen desteklerle beraber bilgisayar güvenliği popüler bir araştırma alanı olmuştur. Bu kapsamda, güncelleme politikaları, işletim sistemlerindeki açıklara yönelik çalışmalar, anti-virüs ve firewall gibi çok çeşitli birbirilerini destekleyici çalışmalar yapmışlardır. Son zamanlarda ise standart yöntemlere ek olarak saldırı tespit sistemleri (STS) konusunda çalışmalar gündeme gelmiştir.

STS, bilgisayar sistemlerini ve ağ trafiğini izleyen ve bu bilgiyi dışarıdan yapılan saldırıları, sistem kötüye kullanımlarını veya organizasyonun içinden yapılan saldırıları belirlemek için kullanan güvenlik sistemleri olarak tanımlanabilir (Sarmah, 2001). Bu tanım diğer güvenlik ürünleriyle STS arasındaki kesişimi göstermektedir. Bu yönüyle STS katmanlı bir güvenlik mimarisi oluşturmak için önemli bir araçtır.

Günümüzde STS kurumsal sistemlerde kullanılması gereken temel güvenlik ürünlerinden biri olarak görülmektedir. STS diğer güvenlik ürünleriyle beraber kullanıldığında katmanlı bir güvenlik mimarisi oluşturmak için kullanılabilir. Halihazırda, ev kullanıcılarının çoğu güvenlik duvarı ve anti-virüs yazılımlarının yanında STS kullanmaktadır. Bu şekilde STS diğer güvenlik ürünlerinin belirleyemediği saldırıları belirlemek için kullanılır. STS veri elde etme yöntemine göre saldırının bilgi toplamadan başlayarak erişim sonrasına (post exploitation) kadar olan her aşamasında farkında olma, bilgi toplama ve engelleme imkanına sahiptir. Ancak, STS'den başka şekillerde yararlanmak da mümkündür. STS bulunan tehditlerin raporlanması, güvenlik politikalarının tasarımı ve denetlenmesi, tespit edilen saldırılar hakkında bilgi toplanması ve saldırılara karşı önlemler geliştirilmesi gibi amaçlarla kullanılabilir (Bace ve Mell, 2001; Scarfone ve Mell, 2012).

Araştırmacılar saldırı tespitinde kullanılan sistemleri işledikleri verinin kaynağına, analiz stratejisine, saldırıya karşı verdiği tepkiye, yazılım mimarisine ve diğer birçok değişkene göre sınıflandırmışlardır (Axelsson, 2000; Debar, Dacier ve Wespi, 2000; Hindy ve diğ., 2018; Sabahi ve Movaghar, 2008; Scarfone ve Mell, 2012). STS analiz ettikleri verinin kaynağına göre ağ-tabanlı, kablosuz ağ tabanlı ve host-tabanlı şeklinde sınıflandırılabilir.

Ağ-tabanlı STS, ağ trafiğinin izlenmesi ile elde edinilen bilgileri kullanmaktadır. Kablosuz STS ise kablosuz ağ protokolünden elde edinilen bilgileri saldırının tespiti için kullanmaktadır. Host-tabanlı STS ise cihazdaki birçok farklı veri kaynağını analiz için kullanabilmektedir. Bunlardan; sistem ve uygulama günlük dosyaları (log), kullanıcı davranışları, sistem kaynaklarına doğrudan erişim ve işletim sistemi kütüphanelerinin (API) çağrılarına ilişkin veriler üzerinde çalışmalar yapılmıştır (Du, Li, Zheng ve Srikumar, 2017; Vokorokos ve Baláž, 2010). Günlük dosyalarının belirli bir formatta olması ve erişilebilir olması veri kümesi için avantaj sunarken aynı zamanda barındırdığı gürültü en büyük dezavantaj olarak karşımıza çıkmaktadır (Creech ve Hu, 2014). Sistem çağrıları uygulama yazılımının sistem kaynaklarına en temel erişimini modellemektedir. API çağrıları ise sistem çağrılarının eşdeğeri olarak düşünülmesine rağmen versiyonlarının sıkça değişmesi ve aynı işler için kullanılabilecek farklı sistem çağrılarının bulunması nedeniyle sistem tasarımını zorlaştırmaktadır (Hou, Saas, Chen ve Ye, 2016). Forrest, Hofmeyr, Somayaji ve Longstaff (1996) tarafından yapılan çalışmadan başlayarak yapılan çoğu çalışmada sistem çağrılarının aldığı argümanlar ihmal edilerek veri tek boyutlu dizilere dönüştürülmüştür. Literatürde başarımı artırmak ve taklit saldırılarını engellemek adına sistem çağrı argümanlarını dikkate alan modeller de sunulmuştur (Bhatkar, Chaturvedi ve Sekar, 2006; Kruegel, Mutz, Valeur ve Vigna, 2003; Liu, Jiang, Jin, Mao ve Chen, 2011; Maggi, Matteucci ve Zanero, 2010; Mutz, Valeur, Vigna ve Kruegel, 2006; Tandon ve Chan, 2003).

STS diğer yaygın sınıflandırmada ise analiz stratejilerine göre imza-tabanlı ve anomali-tabanlı olarak sınıflandırılabilir. İmza-tabanlı STS saldırı bilgisinden elde edilen imzaları kullanarak analiz işlemini yaparken, anomali-tabanlı STS veri üzerinde normal ve anormal davranışları belirlemeye çalışır. Birden fazla veri kaynağı veya analiz stratejisini benimseyen sistemler hibrit sistemler olarak adlandırılır (Debar ve diğ., 2000; Hindy ve diğ., 2018). Ticari uygulamalarda çoğunlukla imza-tabanlı STS kullanılmasına karşın, hem pazarlama hem de güncel ihtiyaçlara cevap vermek açısından güvenlik ürünü geliştiricileri bu teknolojilere ilgilerinin olduğunu her fırsatta belirtmektedirler. Anomali-tabanlı STS genel olarak imza-tabanlı sistemlere göre daha yüksek yanlış-pozitif sonuçlar üretse de sıfır gün saldırıları, kod karıştırma, şifreleme ve kendini değiştiren zararlı yazılımlar gibi bilgisayar güvenliğinin eski sorunlarına imza-tabanlı yöntemlerle tamamen çözüm bulunamayacağı açıktır (Canfora, Sorbo, Mercaldo ve Visaggio, 2015).

İmza tabanlı yöntemin bir diğer dezavantajı imza veri tabanının sürekli güncel tutulması gerekliliğidir. Saldırıların tespit edilmesi, önlemeye yönelik imzaların oluşturulması ve sonrasında imzaların son kullanıcıya aktarılması başlı başına zaman alan bir süreçtir. Bu nedenle saldırı fark edilse bile incelenmesi, imza üretilmesi ve bunun son kullanıcıdaki yazılıma ulaştırılması sürecinde saldırgan istenilen zararı verecek süreye sahip olabilmektedir (White ve diğ., 1999). Buna ek olarak, saldırganlar imza üretilme sürecini zorlaştırmak ve önlem alınmasını geciktirmek için de bazı yöntemler geliştirmişlerdir (CERT-UK). Anomali tanıma yöntemlerinin ise daha önce raporlanmamış saldırıları belirlemekte kullanılabileceğini değerlendirilmektedir (Duessel, Gehl, Flegel, Dietrich ve Meier, 2017; Maggi ve diğ., 2010; Patcha ve Park, 2007; Stavroulakis ve Stamp, 2010).

Sunulan çalışmada host ve anomali tabanlı STS yaklaşımları değerlendirilmiştir. Strace ve Sun Basic Security Module (BSM) gibi yardımcı yazılımlar ile işletim sistemi mekanizmaları vasıtasıyla çalışma sırasında elde edilen sistem ve API çağrı verilerini kullanan çalışmalar incelenmiştir ("Linux Programmer's Manual," 2017b).

Bölüm 2'de anomali tespitinde kullanılan yöntemler ve ilgili çalışmalardan bahsedilmiş, bölüm 3'de çalışmalarda kullanılan veya bu amaçla kullanılabilecek veri kümeleriyle ilgili ayrıntılı bilgi verilmiştir. Bölüm 4'de anomali tespiti ile ilgili yapılan çalışmalar ve veri kümeleri genel olarak ele alınmış bu bağlamda anomali tespit araştırmalarının genel sorunları tartışılmıştır. Bölüm 5'de ise sonuçlardan ve ileriki çalışmalarda yapılabileceklerden bahsedilmiştir.

2. Anomali Tespitinde Kullanılan Yöntemler

Anomali-tabanlı STS'de kullanılan yöntemler istatistiksel anomali tabanlı yöntemler, sınıflandırma tabanlı yöntemler, en yakın komşu tabanlı yöntemler, kümeleme tabanlı yöntemler ve enformasyon teorisi olarak alt başlıklarıyla birlikte gruplandırılmıştır. Kullanılan yöntemlerle ilgili çalışmalar Tablo 1'de özetlenmiştir. Buna ek olarak literatürde taklit (mimicry) saldırıları olarak geçen saldırılar da bu başlık altında derlenmiştir.

Tablo 1

Anomali-Tabanlı STS'de Kullanılan Yöntemler

Kullanılan Yöntem	Çalışmalar
Histogram kullanarak istatistiksel profil çıkarımı	(Cabrera, Lewis ve Mehra, 2001; Eskin, Lee ve Stolfo, 2001; Forrest ve diğ., 1996; Gupta ve Kumar, 2015; Hofmeyr, Forrest ve Somayaji, 1998; Kosoresow ve Hofmeyer, 1997; Marceau, 2000; Murtaza, Khreich, Hamou-Lhadj ve Couture, 2013; Tan ve Maxion, 2002; Tong ve Yan, 2017; Warrender, Forrest ve Pearlmutter, 1999)
Markov ve Saklı Markov Modeli	(Ali, Khan, Sajjad ve Khayam, 2009; Hoang, Hu ve Bertok, 2003; Hu, Yu, Qiu ve Chen, 2009; Warrender ve diğ., 1999; Ye, Li, Chen, Emran ve Xu, 2001; Yolaçan, Dy ve Kaeli, 2014)
Yapay Sinir Ağları	(Creech ve Hu, 2014; Ghosh, Schwartzbard ve Schatz, 1999; Haider, Creech, Xie ve Hu, 2016; Han ve Cho, 2005; Hou ve diğ., 2016; Kim, Yi, Lee, Paek ve Yoon, 2016)
Bayes Ağları	(Feng, Guan, Guo, Gao ve Liu, 2004; Haider ve diğ., 2016; Kang, Fuller ve Honavar, 2005; Mouttaqi, Rachidi ve Assem, 2017; Mutz ve diğ., 2006)
Destek Vektör Makineleri	(Chen, Hsu ve Shen, 2005; Eskin, Arnold, Prerau, Portnoy ve Stolfo, 2002; Haider ve diğ., 2016; Hu, Liao ve Vemuri, 2003; Xie, Hu ve Slay, 2014; Yao, Zhao ve Fan, 2006)
Kural Tabanlı Sistemler	(Canali ve diğ., 2012; Kang ve diğ., 2005; Lanzi, Balzarotti, Kruegel, Christodorescu ve Kirda, 2010; Lee, Stolfo ve Chan, 1997; Ye ve diğ., 2001)
Sonlu Durum Makineleri	(Sekar, Bendre, Dhurjati ve Bollineni, 2001)
Kaba Kümeler Teorisi	(Nauman, Azam ve Yao, 2016)
En Yakın Komşu Yöntemleri	(Borisaniya ve Patel, 2015; Deshpande, Sharma, Peddoju ve Junaid, 2018; Grimmer, Röhling, Kricke, Franczyk ve Rahm, 2018; Haider ve diğ., 2016; Liao ve Vemuri, 2002; Xie ve Hu, 2013; Xie, Hu, Yu ve Chang, 2015)
Kümeleme Yöntemleri	(Eskin ve diğ., 2002; Grimmer ve diğ., 2018; Xie ve diğ., 2015)
Enformasyon Teorisi	(Eskin ve diğ., 2001)

2.1. İstatistiksel Anomali Tabanlı Yöntemler

İstatistiksel modellerin ve testlerin veriler üzerindeki örüntüleri bulmakta kullanılmasıyla beraber öngörülen örüntüye şüpheli ölçüde uyumsuz olan uçdeğer (outlier) kavramı ortaya çıkmıştır (Hawkins, 1980). Uçdeğerler, gürültü veya ortaya çıkması daha düşük olasılıklı durumlardan kaynaklı olabilir. Bunlar dışındaki durumda ise uçdeğerler veriyi ürettiği kabul edilen stokastik süreçler tarafından üretilmemiştir (Anscombe ve Guttman, 1960; Kriegel, Kröger ve Zimek, 2010). İşte bu tip uçdeğerler anomali olarak kabul edilirler. Anomali-tabanlı STS, sisteme yapılan dış müdahalelerin sistem çağrı verilerinde istatistiksel anomaliler ortaya çıkaracağını düşünerek tasarlanmıştır. İstatistiksel anomali-tabanlı yöntemler verinin dağılımı ile ilgili ön kabulde bulunarak (parametrik yaklaşım) veya ön kabulde bulunmayarak (parametrik olmayan yaklaşım) oluşturduğu olasılıksal modeller ile anomalileri tespit etmeye çalışmaktadır (Chandola, Banerjee ve Kumar, 2009).

2.1.1. Histogram Kullanarak İstatistiksel Profil Çıkarımı

Histogram temelli yaklaşımlar, parametrik olmayan istatistiksel yöntemlerdendir (Chandola ve diğ., 2009). Bilgi birikimi, sistem çağrı dizi kümesinden iyi ve kötü davranışları modelleyebilecek sabit uzunluktaki alt-dizilerin (n-gram dizilerin) çıkarımı sonucunda oluşan veri tabanlarıdır. Çalışmalarda çoğu zaman iyi davranışı modelleyen veri tabanlarındaki alt-diziler sınıflandırma yapılacak dizi üzerinde aranarak anomali olup olmadığına karar verilmesi sağlanmaya çalışılmıştır.

İlk olarak New Mexico Üniversitesinden Forrest ve diğ. (1996) tarafından insan bağışıklık sistemi örnek alınarak yapılan çalışmada önerilen yöntemle her bir işlem için olması beklenen davranışı içeren bir veri tabanı oluşturulması öngörülmüştür. İnsan bağışıklık sistemleri örnek alınarak yapılan çalışmada bu veri tabanının işlemin "kendini" tanımladığı savunulmuştur. Bu veriler, sınıflandırılacak sistem çağrı dizisi üzerinde kaydırılan sabit uzunluktaki pencere ile karşılaştırılmıştır. Eğer pencerede ilk sırada bulunan sistem çağrısı için herhangi bir pozisyondaki sistem çağrısı önceki örneklerde bu pozisyonda görülmemiş ise bu aykırılık olarak geçmektedir. Örnek olarak, sendmail ve lpr yazılımları için farklı saldırılar denenmiş ve bu saldırıların sistem çağrı dizisine etkileri aykırılıkların olası aykırılık sayısına oranlanması ile değerlendirilmiştir. Kosoresow ve Hofmeyer (1997), önceki çalışmadan farklı olarak yerel ölçüme dayanan yöntemi önermişlerdir. Bu yönteme göre anomali ancak belirlenen sayıda öncül ölçümün eşik değerinden büyük olması durumunda oluşmaktadır. Hofmeyr ve diğ. (1998) yerel anomalilerin tespiti için veri tabanında bulunan alt-dizilere göre Hamming uzaklığı hesaplamışlar ve en anormal ölçümü sınıflandırma için kullanmışlardır. Warrender ve diğ. (1999) önceki çalışmalarda kullanılan stide yöntemine frekans bilgisini de ekleyerek t-stide adı verilen yöntemi uygulamışlardır. Buna göre veri tabanında "az" geçen örnekler değerlendirme dışı bırakılmış ve bu yöntemin başarımı düşürdüğü gözlemlenmiştir. Marceau (2000) pencere boyutu seçme problemine çözüm olarak büyük değerde seçilen bir pencere kullanılarak çıkarılan alt-dizilerden sonlu durum makinesi oluşturulabileceğini göstermiştir. Cabrera ve diğ. (2001) saldırılar için de benzer bir veri tabanının oluşturulabileceğini ve bu verilerin hem daha önceden belirlenmeyen saldırıları belirlemekte hem de belirlenen saldırıların tipini tespit etmekte kullanılabileceğini belirtmişlerdir. Eskin ve diğ. (2001) uygun pencere uzunluğunun içeriğe bağlı olduğu üzerinde durmuştur. Bunun yanında yine içeriğe bağlı olarak alt-dizi içerisindeki bazı dizilerin ihmal edilebileceğini ifade etmişlerdir. Bu nedenlerle olasılıksal suffix tree (sonek ağacı) benzeri SMT yöntemini sistem çağrı verisinde kullanmışlardır. Tan ve Maxion (2002) pencere uzunluğunun başarıma etkisini araştırmışlardır. Çalışmada koşullu entropinin stide başarımını göstermediği iddia edilmiş, bunun yerine saldırı dizilerinde en küçük yabancı alt-dizilerin bulunabilmesinin başarımı etkilediği öne sürülmüştür. Gupta ve Kumar (2015) dizi üzerinde tekil sistem çağrılarını takip eden çağrılar kümesini karşılaştırarak kötü amaçlı yazılımların belirlenebileceğini göstermişlerdir.

Murtaza ve diğ. (2013) sistem çağrılarını çekirdek modülleri ile ilişkilendirerek, ilgili durumların dizide görülme olasılıklarındaki sapmalara göre anomali tanıma yapmışlardır. Tong ve Yan (2017) Android uygulamalarından toplanan sistem çağrı verilerinden değişken uzunlukta pencerelerle çıkarılan alt-dizilerinin frekanslarını analiz ederek sınıflandırma yapmışlardır. Ayrıca, yeni tip zararlı yazılımların belirlenebilmesi için yeni örneklerin eğitim kümesine dahil edilmesinin gerekliliğini belirtmişlerdir.

2.1.2. Markov ve Saklı Markov Modelleri

Saklı Markov modeli (HMM - hidden Markov model) ardışıl veriler için kullanılan parametrik bir tekniktir (Chandola ve diğ., 2009). HMM'nin eğitimi sırasında veri kümesinden durumlar, çıktılar ve dönüşüm olasılıkları öğrenilir. Eğitim sonucunda elde edilen model istenilen davranışın meydana gelme olasılığını hesaplamak için kullanılır. Warrender ve diğ. (1999) HMM'yi sistem çağrılarına uygulamışlardır. Model başarımının stide ve RIPPER (repeated incremental pruning to produce error reduction) (Cohen, 1995) yöntemlerine göre daha iyi olduğunu belirtmişlerdir. Bunun yanında yöntemin eğitim zamanı açısından maliyetli olduğu görülmüştür.

Hoang ve diğ. (2003) t-stide ile HMM'yi bir modelde birleştirmişlerdir. Çalışmada t-stide tarafından uyumsuz olarak sınıflandırılan diziler HMM ile üretilme olasılığına göre bir kere daha sınıflandırmaya tabi tutulmuştur. Hu ve diğ. (2009) eğitim kümesinden benzer alt-dizileri çıkartmış ve artırmalı HMM yöntemini (Hoang ve Hu, 2004) uygulamışlardır. Bu sayede yanlış-pozitif oranı kabul edilebilir bir seviyede kalırken eğitim süresi düşürülmüştür. Yolaçan ve diğ. (2014) ön işlemden geçirilerek kümelenen veriler ile çoklu-HMM kullanmışlardır. HMM çıktılarını üstel ağırlıklı hareketli ortalama (EWMA - exponentially-weighted moving average) yöntemi (Roberts, 2000) ile filtreleyerek skorlamayı gerçekleştirmişlerdir.

Ali ve diğ. (2009) Markov zinciri temelli skor tahminini kullanarak adaptif eşikleme adını verdikleri algoritmayı geliştirmişlerdir. Bu algoritma ile model eşik değerlerinin seçimi problemine çözüm bulmayı amaçlamışlardır.

2.2. Sınıflandırma Tabanlı Yöntemler

Literatürde, farklı problemlerin çözümü için birçok sınıflandırma yöntemi ve algoritması geliştirilmiştir. Bu yöntemlerden host-tabanlı STS için kullanılan yöntemler bu bölümde incelenmiştir. Bunlar; yapay sinir ağları (YSA), Bayes ağları, destek vektör makineleri (SVM - support vector machine), kural tabanlı sistemler ve diğer sınıflandırma yöntemleri olarak sıralanabilir.

2.2.1. Yapay Sinir Ağları

Ghosh ve diğ. (1999) ileri-beslemeli YSA ve Elman ağını sistem çağrılarına uygulamışlardır. Model çıktılarını leaky bucket benzeri algoritmadan geçirerek skorlamışlar ve sınıflandırma yapmışlardır. Araştırmacılar Elman ağı benzeri tekrarlayan sinir ağlarının (RNN - recurrent neural network) sistem çağrılarında anomali tespiti problemine uygun olduğu sonucuna ulaşmışlardır. Han ve Cho (2005) ağ topolojisinin ve gizli katman nöron sayısının genetik algoritma tarafından optimize edilmesiyle oluşturulan evrimsel sinir ağını (ENN - evolutionary neural network) kullanmışlardır. Çalışmada evrimsel sinir ağlarının çok katmanlı algılayıcı (MLP - multilayer perceptron) tipindeki sinir ağlarına göre daha yüksek doğrulukla sınıflandırma yapmasının yanında eğitim süresini de azaltabileceği belirtilmiştir. Creech ve Hu (2014) değişken uzunlukta pencereler ile çıkarılan alt-dizilerden sözcük grupları oluşturmuşlardır. Ön işleme sonucunda ortaya çıkan veri tabanındaki frekans bilgisini kullanarak ELM (extreme learning machine) tipinde sinir ağıyla sınıflandırma yapmışlardır. Araştırmacılar uygulanan semantik yaklaşımın başarımı artırdığını göstermişlerdir. Kim ve diğ. (2016) dil modeli yaklaşımı ile semantik özellikleri yakalamayı amaçlamıştır. Bu amaçla doğal dil işlemede yaygın olarak kullanılan uzun kısa-süreli bellek (LSTM - long short-term memory) ağlarını sistem çağrı dizilerine uygulamışlardır. Bu çalışmayla RNN mimarilerinin sistem çağrılarında anomali tanıma için uygun olduğunu göstermişlerdir. Hou ve diğ. (2016) Android sistemlerde zararlı yazılımların belirlenmesinde yığılmış oto-kodlayıcı (SAE - stacked autoencoder) tipinde YSA kullanmışlardır. İlk olarak örneklerinin dinamik analizi sonucunda elde edilen sistem çağrı dizisi akışını temsil eden graf elde edilmiştir. Ardından bu graftan sistem çağrıları ile ilgili özellikler çıkartılmış ve bunlar derin sinir ağında girdi olarak kullanılmıştır. Deneylerde kullanılan graf yönteminin temel frekans tabanlı yöntemlere ve derin öğrenme yönteminin ise SVM, YSA, naive (saf) Bayes ve karar ağacı gibi bazı makine öğrenmesi yöntemlerine göre avantajlı olduğu görülmüştür.

2.2.2. Bayes Ağları

Feng ve diğ. (2004) dinamik Bayes ağlarını sistem çağrı dizilerine uygulamışlardır. Algoritmanın gerçek-zamanda zararlı davranışı tanıyabileceğini göstermişlerdir. Kang ve diğ. (2005) "bag of words" modeliyle çıkarttıkları frekans bilgisine naive Bayes yöntemini uygulamışlardır. Araştırmacılar deneyde iyi sonuçlar almasına rağmen, veri dağılımının yöntemin başarımını yüksek ölçüde etkilediğini belirlemişlerdir. Mutz ve diğ. (2006) sistem çağrı dizilerinde ve çağrı argümanlarında anomali tespiti için bir takım model geliştirmiştir. Birbirleriyle ilintili bu modelleri toplu şekilde değerlendirmek ve tek bir anomali skoru belirlemek için Bayes ağlarını kullanmışlardır. Mouttaqi ve diğ. (2017) Markov zincirleri ile beraber naive Bayes yöntemini kullanarak oluşturulan modeli dinamik kütüphane çağrılarına uygulamışlardır. Yapılan deneylerde tüm sistemi kapsayan yaklaşımın ve yüksek dereceli Markov zincirlerinin kullanıldığı modellerle daha yüksek başarı elde edildiği raporlanmıştır.

2.2.3. Destek Vektör Makineleri

Hu ve diğ. (2003) Robust SVM yönteminin eğitim kümesindeki gürültüden daha az etkilendiğini ve eğitim süresinin daha az olduğunu gözlemlemişlerdir (Song, Hu ve Xie, 2002). Chen ve diğ. (2005) frekans temelli özellik çıkarım yöntemleri kullanarak yaptıkları deneylerde SVM'nin YSA modellerine göre daha başarılı olduğunu görmüşlerdir. Bunun yanında terim frekansı-ters doküman frekansı (TF-IDF - term frequency-inverse document frequency) ile normalize edilen vektörlerin kullanımının başarımı artırdığı gözlenmiştir. Yao ve diğ. (2006) ilk olarak kaba kümeler teorisi temelli bir yaklaşımla özellik seçimi yapmışlar, ardından Gauss RBF (radial basis function) çekirdeğini kullanarak SVM'yi sistem çağrı dizilerine uygulamışlardır.

Bazı araştırmacılar ise gözetimsiz öğrenme için kullanılan one-class SVM yöntemini sistem çağrı dizisinden çıkarılan sabit uzunluktaki verilere uygulamışlardır (Eskin ve diğ., 2002; Xie ve diğ., 2014). Xie ve diğ. (2014) yöntemin düşük hesaplama maliyetli olduğunu belirtmişlerdir.

2.2.4. Kural Tabanlı Sistemler

Lee ve diğ. (1997) tarafından yapılan çalışmada, stide benzeri özellik çıkarımı sonrasında kural çıkarımı için RIPPER kullanılmıştır. Araştırmacılar saldırı dizilerinin eğitim için kullanımının önceden görülmeyen saldırıların belirlenmesinde başarımı düşürebileceğini belirtmişlerdir. Kang ve diğ. (2005) çağrı dizilerine "bag of words" modelini uygulayarak özellik çıkarımı yapmıştır. Bu frekans bilgisi üzerinde C4.5 karar ağacı (Quinlan, 1993) ve RIPPER kullanılarak sınıflandırma yapılmıştır. Bu yöntemlerden çıkarılan kuralların gerçek-zamanlı saldırı tespitinde kullanılabileceği belirtilmiştir. Bu şekilde makine öğrenmesi yöntemleri kullanılarak sistem çağrılarında anomali tanımaya yarayan kural kümelerinin çıkarılabileceği gösterilmiştir. Lanzi ve diğ. (2010) sistem çağrılarından normal davranışı gösteren erişim aktivite modellerinin (access activity models) çıkarılmasını ve bu kuralların işletim sistemi kaynaklarına erişimin denetlenmesi için kullanılmasını önermişlerdir. Bu amaçla hiyerarşik yapıdaki sistem kaynakları için sistem politikaları oluşturan bir algoritma geliştirmişlerdir. Canali ve diğ. (2012) zararlı yazılımların tanınması için karmaşık modeller kullanmaya gerek olmadığını ve imza-tabanlı sistemlere benzer basit kurallar kümesi kullanılarak da benzer başarımın elde edilebileceğini göstermiştir. Ancak yöntem kural kümesinin istenilen özelliklerde tutulması için yeni eşik değerlerinin kullanılması, performansın değerlendirilmesi ve eğitim süresi gibi zorluklar ortaya çıkarmıştır.

2.2.5. Diğer Sınıflandırma Yöntemleri

Sekar ve diğ. (2001) program durumumu ve sistem çağrı dizisini kullanarak sonlu durum makinesi oluşturmuşlardır. Program durumunu işletilen programın bellekteki yeri olarak kabul etmişlerdir. Bu şekilde program akışını öğrenmek için doğal bir yapı oluşturmaya çalışmışlardır.

Nauman ve diğ. (2016) üç sınıflı bir model ile seçimin ötelenebilmesi ve bu sayede modelin yeterli bilgi olduğunda seçim yapabilmesi üzerinde durmuşlardır. Bu modeli enformasyon ve oyun teorisinden yararlanarak oluşturulan kaba kümeler yaklaşımlarına uygulamışlardır. Çalışmada seçimin ötelenebilmesi kavramı literatüre kazandırılmıştır.

2.3. En Yakın Komşu Tabanlı Yöntemler

Araştırmacılar sistem çağrı verilerinden çıkarılan frekans temelli ağırlık vektörlerine k-en yakın komşu (k-NN - k-nearest neighbors) algoritmasını uygulayarak sistem çağrı dizilerindeki anomalileri belirlemeye çalışmışlardır. Liao ve Vemuri (2002) frekans yönteminin TF-IDF yöntemine göre dinamik ortamlarda kullanımının daha uygun olduğunu, Xie ve Hu (2013) ise ağırlık vektörü oluştururken kullanılan IDF ölçümüyle en iyi sonuçları elde ettiklerini belirtmişlerdir. Deshpande ve diğ. (2018) yöntemin ölçeklenebilir bulut sistemlerinde kullanımı için bir model önermişlerdir. Benzer bir çalışmada Borisaniya ve Patel (2015) çıkarılan alt-dizilerin frekans vektörü üzerinde k-NN algoritmasının diğer yöntemlere göre başarılı sonuçlar verdiğini belirtmişlerdir.

2.4. Kümeleme Tabanlı Yöntemler

Eskin ve diğ. (2002) spectrum kernel fonksiyonu (Leslie, Eskin ve Noble, 2001) ile özellikleri seyrek özellik uzayına haritalamışlar ve bu veriler üzerinde kümeleme algoritması kullanmışlardır. Xie ve diğ. (2015) sistem çağrı dizisi üzerinde frekans çıkarımı yaparak ortaya çıkan seyrek sistem çağrı vektöründe temel bileşen analizi (PCA - principal component analysis) ile boyut indirgemişlerdir. K-means kümeleme ile k-NN yöntemini karşılaştıran araştırmacılar, k-means kümeleme algoritmasının daha az hesaplama maliyeti ile daha isabetli tanımaya imkan verdiğini belirtmişlerdir. Grimmer ve diğ. (2018) sistem çağrı dizilerinden olasılıksal dönüşüm grafları oluşturarak bu graflardan özellik çıkarımı yapmışlardır. K-centers algoritmasını ve oluşturulan ensemble yöntemi kullanarak elde edilen sonuçlarla graf yönteminin özellik çıkarımında kullanılabileceğini göstermişlerdir (Ypma ve Duin, 1998).

2.5. Enformasyon Teorisi

Lee ve Xiang (2001) bilgi kazancından yola çıkarak ardışıl bağlılıkların kullanıldığı modellerde sınıflandırıcı başarımının koşullu entropi ile ilişkili olduğunu ifade etmişlerdir. Koşullu entropiyi model değişkenlerini belirlemede, göreli entropiyi de veri benzerliğini ölçmede kullanmışlardır. Bu çalışmayla ilk defa model çıktısı dışında da sistem çağrılarında anomali tanıma deneylerinde kullanılabilecek ölçütler geliştirilmiştir.

2.6. Taklit (Mimicry) Saldırıları

Denklem 1'de görülen sistem çağrı alfabesi üzerinde tanımlanan $L \subseteq \Sigma^*$ dili sistem çağrı dizilerinin kümesini, s ve s' ise programın iç durumunu temsil etmektedir.

$$\Sigma = \{tanımlı\ sistem\ çağrıları\} \quad (1)$$

Bu durumda taklit saldırıları, $\forall a \in L$ için olağan bir s durumundan güvensiz bir s' durumuna gitmekte kullanılabilecek ve model tarafından normal olarak tanımlanacak en az bir $s \xrightarrow{a} s'$ dönüşümü olması durumunda ortaya çıkar (Wagner ve Dean, 2001). Dolayısı ile, anomali olarak tanımlanması gereken $a = (a_1..a_n)$ sistem çağrı dizisi model tarafından saldırı olarak değerlendirilmez. Taklit saldırıları, veri kümesinin veya modelin yetersizliğinden ortaya çıkabilirler. Veri kümesinin yetersizliği durumunda ilgili sistem çağrı bilgisi eğitim kümesinde bulunmaz. Modelin yetersizliği durumunda ise eğitim kümesinde ilgili çağrı bilgisi bulunmasına rağmen model tarafından saldırı tespit edilemez.

Taklit saldırıları üzerine çalışma yapılırken saldırganların kullanılan STS ile ilgili tüm bilgiye sahip olduğu kabul edilmelidir. Bu bilgi sistemin herhangi bir girdiye vereceği cevabı bilmenin yanında model parametrelerini de içermektedir. Wagner ve Soto (2002) çalışmasında belirtildiği gibi sadece zafiyetin sömürülmesi sistem çağrı verisinde anomali oluşturmayabilir. Ancak ilk aşama sonrasında zararlı kodun çalışması öngörülen anomaliyi oluşturacaktır. Bunların yanında saldırganın program akışını tamamen yönetebildiği ve kendi kodunu çalıştırabildiği düşünülmelidir.

Bölüm 1'de bahsedildiği gibi sistem çağrı argümanları çoğu çalışmada ihmal edilmektedir. Bu model karmaşıklığını azaltıyor olsa da saldırganlara program akışı sürerken sistem çağrılarının argümanlarının istenilen işi görecek şekilde değiştirilmesi imkanı sağlamaktadır (Wagner ve Dean, 2001; Wagner ve Soto, 2002). Özellikle frekans özellikleri kullanan modeller için etkili olabilecek bir saldırı tipi ise no-op sistem çağrılarının kullanılması sayesinde gerçekleştirilir. Gerçek anlamda bir işlevselliği olmayan bir sistem çağrısı bulunmamakla beraber *getpid()*, *getuid(), geteuid()* ve *getegid()* grubu sistem çağrıları rahatlıkla saldırganlar tarafından kullanılabilir. Hatta diğer birçok sistem çağrısı kendisini işlevsiz hale getirecek argümanlarla beraber kullanılabilir. Sistem çağrı dizilerinin işlevsel eşdeğerlerinin birbiri yerine kullanılması da anomali tespit sistemlerini atlatmak için kullanılabilir. Bu yöntem için bağımsız sistem çağrılarının sıralamalarının değiştirilmesi ve eşdeğer sistem çağrılarının birbiri yerine kullanılması önerilmiştir (Wagner ve Soto, 2002). Bu yöntemlerin ikincisi modern işletim sistemlerinde pek etkin bir yol olarak gözükmemektedir. Ancak *clone() - fork()* ve *open() - openat()* örneklerinde görüldüğü gibi bazı durumlarda hala bu özellikten bahsedilebilir ("Linux Programmer's Manual," 2017a).

Tan, Killourhy ve Maxion (2002) stide gibi basit modeller için en küçük yabancı dizi özelliğini dikkate alarak el ile taklit saldırılarının gerçeklenebileceğini göstermişlerdir. Ancak karmaşık ve doğrusal olmayan modellerde bu tarz yaklaşımların uygulanması zordur. Kayacık, Zincir-Heywood ve Heywood (2007) denemelerini *stide* yöntemi üzerinde yapmış olmalarına rağmen, genetik programlama ile anomali skoruna bağlı uygunluk fonksiyonu kullanmışlar ve genellenebilir bir taklit saldırısı yöntemi geliştirmişlerdir. Sonraki çalışmalarında ise Markov modeli gibi başka yöntemler için de genetik programlamanın taklit saldırılarındaki başarımını göstermişlerdir (Kayacık, Zincir-Heywood, Heywood ve Burschka, 2009).

3. Veri Kümeleri

Anomali saldırı tespit sistemleri ile ilgili araştırmalar sistem ve kütüphane fonksiyon çağrıları üzerine yoğunlaşmış olsa da bu alanda kullanıma açık az sayıda veri kümesi vardır. Bazı çalışmalarda araştırmacılar taşınabilirliği denetlemek, karşılaştırma yapmak ve gerçek sistemlerde deneyler yapmak amacıyla kendi verilerini toplamak için yöntemler geliştirmişler ve kendi veri kümelerini oluşturmuşlardır (Lanzi ve diğ., 2010; Pendleton ve Xu, 2017). Pendleton ve Xu (2017) birçok veri kümesinde farklı iş parçacıklarının (thread) çalışma anındaki sistem çağrılarının karıştırılarak işlem (process) sistem çağrı dizisinin elde edildiğini belirtmiştir. İş parçacıklarının bazı işletim sistemlerinde temel iş birimleri olduğu ve çalışma zamanındaki deterministik olmayan ortam düşünülürse bu verilerde sistem çağrı dizileri arasındaki ardışıl bağlılığın öğrenilmesinin zor olduğu düşünülebilir. Araştırmacılar bu gerekleri göz önünde bulundurarak sistem çağrı veri kümesi üretmek için bir araç oluşturmuşlardır (Pendleton, 2017). Bu başlıkta anomali tespit çalışmalarında kullanılabilecek sistem ve kütüphane çağrı kütüphanelerinden bahsedilmiştir. Veri kümeleriyle ilgili genel bilgiler Tablo 2'de karşılaştırılmalı olarak gösterilmiştir. Tablo 3'de ise veri kümelerine göre ilgili çalışmalar gösterilmiştir.

Bu alanda yapılan çalışmalarda en çok kullanılan veri kümelerinden biri olan UNM veri kümesi New Mexico Üniversitesi (UNM) ve Massachusetts Teknoloji Enstitüsü (MIT) tarafından yapılan çalışmalar sonucunda toplanmıştır. Farklı tipte saldırılar için farklı ortamlardan toplanan veriler içerdiğinden ve diğer yöntemlerle karşılaştırma olanağı olduğundan eski bir veri kümesi olmasına rağmen yaygın olarak kullanılmıştır. Bazı araştırmacılar tarafından güncel saldırıları yansıtmayabileceği belirtilmiştir (Mouttaqi ve diğ., 2017).

1998 ve 1999 DARPA (Defense Advanced Research Projects Agency) Intrusion Detection Evaluation Dataset (IDEVAL) host STS araştırmalarında en yaygın

kullanılan veri kümeleridir (*DARPA Intrusion Detection Evaluation Dataset*, 1998; *DARPA Intrusion Detection Evaluation Dataset*, 1999). İlki 1998 yılında yapılan çalışmalar için oluşturulan veri kümesi STS performansının kıyaslanması için kullanılmıştır. Sonraki sene Windows NT sistemlerine yönelik veri kümesi ile genişletilen testlerde, amaç önemli bilgilerin saklandığı sistemlerde kullanılabilecek yüksek başarımlı STS geliştirilmesidir. Normal ve saldırı verilerinden oluşan DARPA IDEVAL, ağ verilerinin yanı sıra host-tabanlı sistemler için Solaris BSM ve Windows olay loglarını da içermektedir.

Australian Defence Force Academy Windows Data Set (ADFA-WD) ve Australian Defence Force Academy Windows Data Set with a Stealth Attacks Addendum (ADFA-WD:SAA) Windows İşletim Sistemleri için bulunabilecek nadir veri kümeleridir. Bu veri kümeleri aynı eğitim ve doğrulama verilerine sahip olmakla beraber, saldırı verileri farklı yeteneklerdeki saldırganları temsil etmektedir. Veri kümeleri ntoskrnl.exe ve sekiz ayrı dinamik kütüphaneye (DLL) yapılan çağrıları içermektedir.

Malrec, PANDA (Platform for Architecture-Neutral Dynamic Analysis) emulatör ortamı kullanılarak oluşturulan zararlı yazılım sandbox sistemidir. Araştırmacılar zararlı yazılımların çalışma zamanı davranışları ile alakalı bilgilerin toplanması için geliştirilen bu sistem üzerindeki çalışmaları neticesinde oluşturulan büyük boyutlu veri kümesini proje sayfasında yayınlamışlardır. Ayrıca, sistem çağrı verileri 465 GB boyutundaki bir arşiv dosyası olarak yayınlanmıştır. Bu veri kümesi diğerlerinden farklı olarak normal kabul edilebilecek sistem çağrı bilgisi içermemektedir.

Tablo 2

Veri Kümeleri ve Özellikleri

Veri Kümesi	Veri Tipi	Platform	Örnek Sayısı	Normal Örnekler	Saldırı Çeşidi	Etiket	İlgili Yayın
UNM	S	SunOS / Linux	100554	Var	14*	Var	(Forrest ve diğ., 1996; Hofmeyr ve diğ., 1998; Warrender ve diğ., 1999)
DARPA IDEVAL	S	Solaris	Bilinmiyor†	Var	58	Var	(Lippmann, Haines, Fried, Korba ve Das, 2000; R. P. Lippmann ve diğ., 2000)
DALHOUSIE SYSCALL	S	Redhat 6.2	32	Var	5	Var	(Kayacık, 2009)
ADFA-LD	S	Ubuntu 11.04	5951	Var	6	Var	(Creech, 2014)
ADFA-WD	K	Windows XP SP2	7724	Var	12	Var	(Creech, 2014)
ADFA-WD:SAA	K	Windows XP SP2	3047	Var	4‡	Var	(Creech, 2014)
MALREC SYSCALL	S	Windows 32-bit / 64-bit	66301	Yok	1270	Var	(Severi, Leek ve Dolan-Gavitt, 2018)

S: Sistem çağrı dizisi
K: Kütüphane fonksiyon çağrı dizisi
* Bir tanesi yazılım hatası olmak üzere on dört saldırı bulunmaktadır.
† Verilerin tümüne ulaşılamamıştır.
‡ Saldırı sayısı kullanılan gizleme yöntemlerinin sayısıdır.

Tablo 3

Veri Kümelerine Göre Yayınlar

Veri Kümesi	Çalışmalar
UNM	(Ali ve diğ., 2009; Cabrera ve diğ., 2001; Creech ve Hu, 2014; Eskin ve diğ., 2001; Feng ve diğ., 2004; Forrest ve diğ., 1996; Gupta ve Kumar, 2015; Hoang ve Hu, 2004; Hoang ve diğ., 2003; Hofmeyr ve diğ., 1998; Hu ve diğ., 2009; Kang ve diğ., 2005; Kim ve diğ., 2016; Kosoresow ve Hofmeyer, 1997; Lee ve diğ., 1997; Lee ve Xiang, 2001; Marceau, 2000; Mouttaqi ve diğ., 2017; Murtaza ve diğ., 2013; Nauman ve diğ., 2016; Tan ve Maxion, 2002; Warrender ve diğ., 1999; Yao ve diğ., 2006; Yolaçan ve diğ., 2014)
DARPA IDEVAL	(Ali ve diğ., 2009; Chen ve diğ., 2005; Creech ve Hu, 2014; Eskin ve diğ., 2002; Eskin ve diğ., 2001; Ghosh ve diğ., 1999; Han ve Cho, 2005; Hu ve diğ., 2003; Kang ve diğ., 2005; Kim ve diğ., 2016; Kruegel ve diğ., 2003; Lee ve Xiang, 2001; Liao ve Vemuri, 2002; Maggi ve diğ., 2010; Mutz ve diğ., 2006; Tandon ve Chan, 2003; Ye ve diğ., 2001)
DALHOUSIE SYSCALL	(Kayacık ve diğ., 2007; Kayacık ve diğ., 2009)
ADFA-LD	(Borisaniya ve Patel, 2015; Creech ve Hu, 2014; Grimmer ve diğ., 2018; Kim ve diğ., 2016; Xie ve Hu, 2013; Xie ve diğ., 2014; Xie ve diğ., 2015)
ADFA-WD	(Borisaniya ve Patel, 2015; Haider ve diğ., 2016; Mouttaqi ve diğ., 2017)
ADFA-WD:SAA	(Haider ve diğ., 2016)
ANDROID MGP	(Tong ve Yan, 2017)
Comodo CSS	(Hou ve diğ., 2016)

4. Tartışma

Sistem ve kütüphane fonksiyon çağrı dizileri üzerinde yapılan anomali tanıma çalışmaları gelişen yapay zeka modelleriyle beraber birçok modelde denenmiştir. Ancak verinin model tarafından nasıl işleneceği ve verinin neyi ifade ettiği farklı araştırmacılar tarafından farklı şekilde değerlendirilmiştir. Bazı araştırmacılar bu konulara açıklık getirmek için verinin doğasıyla ilgili çalışmalar yapmışlardır (Lee ve Xiang, 2001; Tan ve Maxion, 2002). Yapılan çalışmalara rağmen farklı yaklaşımlar sistem çağrı verilerinin farklı şekilde ele alınmasına yol açmıştır. Literatürde anomali tanıma yapılması gereken temel birimin ne olduğu ile ilgili farklı görüşler sunulmuştur. Bunun yanında çalışmalar temel birimin belirlenmesinde ölçeklenebilirlik ve performans gibi değişkenleri de göz önünde bulundurmuşlardır. Bazı çalışmalar bütün sisteme ait çağrı dizisini ele alırken bazıları ise işlemi birim kabul ederek çalışmalarını yapmışlardır. Hesaplama maliyeti, sistemin başarımı ve örüntülerin tanınması açısından hangi yöntemin daha yararlı olacağı önemli bir sorudur.

İşlemleri ve iş parçacıklarını birim alan anomali tespit sistemlerinin son kullanıcı ürünlerinde kullanılmasının bir zorluğu ilgili sistemde eğitim verilerinin toplanmasıdır. Sistemin belirli işlemler için önceden oluşturularak kullanıcıya sunulması buna bir çözüm olarak düşünülebilir. Ancak güncelleme gibi durumlarda bu sistemlerin güncellenmesinin gerekebileceği düşünülmelidir. Bu sorun sistemlerin ölçeklenebilirliğini önemli ölçüde kısıtlamaktadır.

Bir diğer sorun ise sistem çağrı dizilerinin modern sistemlerde her zaman gerekli bilgiyi verip vermeyeceğidir. Bölüm 2.6'da görülebileceği gibi anomali tespit sistemlerini yanıltmak bazı yöntemlerle mümkün olabilmektedir. Ancak bu durum verinin gerekli bilgiyi içermediğinden çok modelin yetersizliği olarak düşünülmektedir. Geleneksel anlamda işlemler işletim sisteminin yönettiği kaynaklara erişebilmek için sistem çağrı arayüzünü doğrudan veya dolaylı olarak kullanmak durumundadır. Farklı donanım ve yazılım ortamlarında sistem çağrılarının her zaman gerekli bilgiyi verip vermeyeceği tam olarak cevaplanmış bir soru değildir.

Bununla ilgili diğer bir sorun ise sistem veya kütüphane fonksiyon çağrı bilgilerinin ne kadar doğru bir şekilde elde edilebildiğidir. Bu kullanılan yönteme göre değişebilmekte, asıl olarak ise zararlı davranışa sahip kodun çalıştırıldığı yetkilere göre değişmektedir. Örneğin teoride GNU/Linux sistemlerde root yetkisine sahip bir kullanıcı hem ptrace gibi arayüzleri hem de çekirdek modülü destekli yöntemleri engelleyebilir ve yanıltabilir. Bunun dışında zararlı davranışa sahip kod kullanılan yönteme göre veri toplandığını fark ederek zararlı davranışı sergilemekten vazgeçebilir (Liţă, Cosovan ve Gavriluţ, 2018).

5. Sonuçlar

Sistem çağrılarında anomali tespit çalışmaları yirmi yıldan fazla süredir devam etmektedir. Ancak, bu alanda evrensel veri kümeleri üretmek için çalışmalar yapılmış olsa da teorik olarak tüm normal davranışı modellemesi gereken veri kümelerindeki eksiklikler kendini hissettirmeye devam etmektedir. Host tabanlı yaklaşımların günlük hayatta yer almasına yönelik ürün ve çalışmaların artması ile veri kümesinin artacağı değerlendirilmektedir.

Anomali tabanlı yaklaşımlar sıfır gün saldırılarını belirli bir oranda tespit edebilirken, normal davranışları da saldırı olarak belirleyebilmektedirler. Son kullanıcı ya da sistem yöneticilerinin, STS tarafından saldırı olarak tespit edilmiş olan davranışı incelemesi beklenmektedir. Böylece, anomali tespit sistemleri aracılığı ile tespit edilen ve analizi sonrasında saldırı olduğu belirlenen uygulamaya ilişkin imzanın çıkarılması mümkündür. İmza tabanlı sistemler imzasına sahip oldukları saldırıları doğrudan net bir şekilde tespit edebilirken, sıfır gün saldırılarını tespit edememektedirler.

Bu konuda yapılacak araştırmalarda literatüre yeni veri kümeleri kazandırmak kadar bu veri kümelerinin STS'nin başarımını nasıl etkilediği gibi konular da tartışılmalıdır. Bunun yanında yeni makine öğrenmesi ve derin öğrenme yöntemleri bu veriler üzerine uygulanmalıdır. Önemli bir konu da birbirlerinin zayıf ve güçlü yönlerini içeren imza ve anomali yaklaşımlarının veya farklı kaynaklardan elde edilen verilerin beraber kullanıldığı başarımı yüksek hibrit STS oluşturulması, bu sistemlerin gerçek ortamlarda kullanılmasıdır. Gerçek ortamlarda kullanımın gözlenmesiyle bu alanda yapılabilecek ileri çalışmalar için olası sorunlar da araştırmacıların dikkatini çekecektir.

Bu derlemede, sistem ve kütüphane fonksiyon çağrı dizisi verileri üzerinde yapılan anomali-tabanlı STS araştırmalarındaki literatür, ilgili araştırmacıların kullanılabilecekleri veri kümelerinin özellikleri ve kıyaslamalarına yer verilmiştir.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Ali, M. Q., Khan, H., Sajjad, A., & Khayam, S. A. (2009). *On achieving good operating points on an ROC plane using stochastic anomaly score prediction*. 16th ACM Conference on Computer and Communications Security, Şikago, USA.

Anscombe, F. J., & Guttman, I. (1960). Rejection of outliers. *Technometrics,* 2, 123-147. doi: https://doi.org/10.2307/1266540

Axelsson, S. (2000). *Intrusion detection systems: a survey and taxonomy*. Teknik rapor.

Bace, R., & Mell, P. (2001). *Intrusion detection systems* Erişim adresi: https://apps.dtic.mil/dtic/tr/fulltext/u2/a393326.pdf

Bhatkar, S., Chaturvedi, A., & Sekar, R. (2006). *Dataflow anomaly detection.* 2006 IEEE Symposium on Security and Privacy (S&P'06) Sunulmuş Bildiri.

Borisaniya, B., & Patel, D. (2015). Evaluation of modified vector space representation using ADFA-LD and ADFA-WD datasets. *Journal of Information Security, 6*(03), 250. doi:https://doi.org/10.4236/jis.2015.63025

Cabrera, J. B. D., Lewis, L., & Mehra, R. K. (2001). Detection and classification of intrusions and faults using sequences of system calls. *ACM SIGMOID Record, 30*(4), 25-34. doi: https://doi.org/10.1145/604264.604269

Canali, D., Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M., & Kirda, E. (2012). *A quantitative study of accuracy in system call-based malware detection.* International Symposium on Software Testing and Analysis Sunulmuş Bildiri.

Canfora, G., Sorbo, A. D., Mercaldo, F., & Visaggio, C. A. (2015). *Obfuscation techniques against signature-based detection: a case study.* 2015 Mobile Systems Technologies Workshop (MST) Sunulmuş Bildiri.

CERT-UK. *Code obfuscation*. Erişim adresi: https://www.ncsc.gov.uk/content/files/protected_files/guidance_files/Code-obfuscation.pdf

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: a survey. *ACM Computing Surveys,* (3), 15.

Chen, W.-H., Hsu, S.-H., & Shen, H.-P. (2005). Application of SVM and ANN for intrusion detection. *Computers and Operations Research,* (10), 2617-2634. doi: https://doi.org/10.1016/j.cor.2004.03.019

Cohen, W. W. (1995). Fast effective rule induction. A. Prieditis & S. Russell (Eds.), *Machine Learning Proceedings 1995* (115-123). San Francisco, ABD.

Creech, G. (2014). *Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks.* (Doktora Tezi), University of New South Wales, Canberra, Avustralya.

Creech, G., & Hu, J. (2014). A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns.

IEEE Transactions on Computers, (4), 807-819. doi: https://doi.org/10.1109/tc.2013.13

DARPA Intrusion Detection Evaluation Dataset. (1998). Erişim adresi: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

DARPA Intrusion Detection Evaluation Dataset. (1999). Erişim adresi: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset

Debar, H., Dacier, M., & Wespi, A. (2000). A revised taxonomy for intrusion-detection systems. *Annales des Télécommunications,* (7-8), 361-378.

Deshpande, P., Sharma, S., Peddoju, S., & Junaid, S. (2018). HIDS: A host based intrusion detection system for cloud computing environment. *International Journal of System Assurance Engineering Management,* (3), 567-576. doi: https://doi.org/10.1007/s13198-014-0277-7

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). *Deeplog: Anomaly detection and diagnosis from system logs through deep learning.* 2017 ACM SIGSAC Conference on Computer and Communications Security Sunulmuş Bildiri.

Duessel, P., Gehl, C., Flegel, U., Dietrich, S., & Meier, M. (2017). Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *International Journal of Information Security,* (5), 475-490. doi: https://doi.org/10.1007/s10207-016-0344-y

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. D. Barbará & S. Jajodia (Eds.), *Applications of Data Mining in Computer Security* (77-101). Massachusetts, ABD: Springer US. doi: https://doi.org/10.1007/978-1-4615-0953-0_4

Eskin, E., Lee, W., & Stolfo, S. J. (2001). *Modeling system calls for intrusion detection with dynamic window sizes.* DARPA Information Survivability Conference and Exposition II. DISCEX'01 Sunulmuş Bildiri.

Feng, L., Guan, X., Guo, S., Gao, Y., & Liu, P. (2004). Predicting the intrusion intentions by observing system call sequences. *Computers and Security,* (3), 241-252. doi: https://doi.org/10.1016/j.cose.2004.01.016

Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1996). *A sense of self for Unix processes.* 1996 IEEE Symposium on Security and Privacy, Sunulmuş Bildiri.

Ghosh, A. K., Schwartzbard, A., & Schatz, M. (1999). *Learning program behavior profiles for intrusion detection*. Workshop on Intrusion Detection and Network Monitoring. (51462), 1-13.

Grimmer, M., Röhling, M. M., Kricke, M., Franczyk, B., & Rahm, E. (2018). *Intrusion detection on system call graphs.* 25. DFN-Konferenz "Sicherheit in vernetzten Systemen" Sunulmuş Bildiri, Hamburg, Almanya.

Gupta, S., & Kumar, P. (2015). An immediate system call sequence based approach for detecting malicious program executions in cloud environment. *Wireless Personal Communications,* (1), 405-425. doi: https://doi.org/10.1007/s11277-014-2136-x

Haider, W., Creech, G., Xie, Y., & Hu, J. (2016). Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks. *Future Internet,* (3), 29. doi: https://doi.org/10.3390/fi8030029

Han, S.-J., & Cho, S.-B. (2005). Evolutionary neural networks for anomaly detection based on the behavior of a program. *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* (3), 559-570. doi: https://doi.org/10.1109/tsmcb.2005.860136

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11): Springer.

Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R. C., & Bellekens, X. J. A. (2018). A taxonomy and survey of intrusion detection system design techniques, Network Threats and Datasets. *CoRR*.

Hoang, X. A., & Hu, J. (2004). *An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls.* 2004 12th IEEE International Conference on Networks (ICON 2004) (IEEE Cat. No.04EX955) Sunulmuş Bildiri.

Hoang, X. A., Hu, J., & Bertok, P. (2003). *A multi-layer model for anomaly intrusion detection using program sequences of system calls.* 11th IEEE International Conference on Networks, 2003. ICON2003. Sunulmuş Bildiri.

Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion detection using sequences of system calls. *Journal of Computer Security,* (3), 151-180. doi: https://doi.org/10.3233/jcs-980109

Hou, S., Saas, A., Chen, L., & Ye, Y. (2016). *Deep4MalDroid: A deep learning framework for Android malware detection based on Linux kernel system call graphs.* 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW) Sunulmuş Bildiri.

Hu, J., Yu, X., Qiu, D., & Chen, H. (2009). A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. *IEEE Network,*

(1), 42-47. doi: https://doi.org/10.1109/mnet.2009.4804323

Hu, W., Liao, Y., & Vemuri, V. R. (2003). *Robust Support Vector Machines for anomaly detection in computer security.* International Conference on Machine Learning and Applications (ICMLA'03) Sunulmuş Bildiri, ABD.

Kang, D.-K., Fuller, D., & Honavar, V. (2005). *Learning classifiers for misuse and anomaly detection using a bag of system calls representation.* Sixth Annual IEEE SMC Information Assurance Workshop Sunulmuş Bildiri.

Kayacık, H. G. (2009). *Can the best defense be a good offense?: Evolving (mimicry) attacks for detector vulnerability testing under a 'black-box' assumption.* Dalhousie University.

Kayacık, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2007). *Automatically evading IDS using GP authored attacks.* IEEE Symposium on Computational Intelligence in Security and Defense Applications Sunulmuş Bildiri.

Kayacık, H. G., Zincir-Heywood, A. N., Heywood, M. I., & Burschka, S. (2009). *Generating mimicry attacks using genetic programming: a benchmarking study.* IEEE Symposium on Computational Intelligence in Cyber Security Sunulmuş Bildiri.

Kim, G., Yi, H., Lee, J., Paek, Y., & Yoon, S. (2016). LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems. *arXiv e-prints*.

Kosoresow, A. P., & Hofmeyer, S. A. (1997). Intrusion detection via system call traces. *IEEE Software,* (5), 35-42. doi: https://doi.org/10.1109/52.605929

Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). *Outlier detection techniques.* Erişim adresi: https://archive.siam.org/meetings/sdm10/tutorial3.pdf

Kruegel, C., Mutz, D., Valeur, F., & Vigna, G. (2003). *On the detection of anomalous system call arguments.* Computer Security – ESORICS 2003 Sunulmuş Bildiri, Berlin, Almanya.

Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M., & Kirda, E. (2010). *Accessminer: using system-centric models for malware protection.* 17th ACM Conference on Computer and Communications Security Sunulmuş Bildiri.

Lee, W., Stolfo, S., & Chan, P. (1997). *Learning patterns from Unix process execution traces for intrusion detection*. AAAI Workshop on AI Approaches to Fraud Detection and Risk Management.

Lee, W., & Xiang, D. (2001). *Information-theoretic measures for anomaly detection.* IEEE Symposium on Security and Privacy (S&P 2001) Sunulmuş Bildiri, ABD.

Leslie, C., Eskin, E., & Noble, W. S. (2001). *The spectrum kernel: A string kernel for SVM protein classification*. Pacific Symposium on Biocomputing.

Liao, Y., & Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security,* (5), 439-448. doi: https://doi.org/10.1016/s0167-4048(02)00514-x

Linux Programmer's Manual. (2017a). *Linux man-pages project.* Erişim adresi: http://man7.org/linux/man-pages/man2/fork.2.html

Linux Programmer's Manual. (2017b). *Linux man-pages project.* Erişim adresi: http://man7.org/linux/man-pages/man1/strace.1.html

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). *Analysis and results of the 1999 DARPA off-line intrusion detection evaluation.* International Workshop on Recent Advances in Intrusion Detection Sunulmuş Bildiri.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R. K., Zissman, M.A. (2000). *Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation.* DARPA Information Survivability Conference and Exposition (DISCEX'00) Sunulmuş Bildiri.

Liţă, C. V., Cosovan, D., & Gavriluţ, D. (2018). Anti-emulation trends in modern packers: a survey on the evolution of anti-emulation techniques in UPA packers. *Journal of Computer Virology Hacking Techniques,* (2), 107-126. doi: https://doi.org/10.1007/s11416-017-0291-9

Liu, A., Jiang, X., Jin, J., Mao, F., & Chen, J. (2011). *Enhancing system-called-based intrusion detection with protocol context.* IARIA Securware Sunulmuş Bildiri, Fransa.

Maggi, F., Matteucci, M., & Zanero, S. (2010). Detecting intrusions through system call sequence and argument analysis. *IEEE Transactions on Dependable and Secure Computing,* (4), 381-395. doi: https://doi.org/10.1109/tdsc.2008.69

Marceau, C. (2000). *Characterizing the behavior of a program using multiple-length N-grams*. 2000 Workshop on New Security Paradigms, County Cork, İrlanda.

Mouttaqi, T., Rachidi, T., & Assem, N. (2017). *Re-evaluation of combined Markov-Bayes models for host intrusion detection on the ADFA dataset.* 2017 Intelligent Systems Conference (IntelliSys) Sunulmuş Bildiri.

Murtaza, S. S., Khreich, W., Hamou-Lhadj, A., & Couture, M. (2013). *A host-based anomaly detection approach by representing system calls as states of kernel modules.* 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE) Sunulmuş Bildiri.

Mutz, D., Valeur, F., Vigna, G., & Kruegel, C. (2006). Anomalous system call detection. *ACM Transactions on Information and System Security (TISSEC),* (1), 61-93. doi: https://doi.org/10.1145/1127345.1127348

Nauman, M., Azam, N., & Yao, J. (2016). A three-way decision making approach to malware analysis using probabilistic rough sets. *Information Sciences,* , 193-209. doi: https://doi.org/10.1016/j.ins.2016.09.037

P. Farwell, J., & Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Survival*. doi: https://doi.org/10.1080/00396338.2011.555586

Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks,* (12), 3448-3470. doi: https://doi.org/10.1016/j.comnet.2007.02.001

Pendleton, M. (2017). syscall-dataset-generator: GitHub. Erişim adresi: https://github.com/marcusp46/syscall-dataset-generator

Pendleton, M., & Xu, S. *A dataset generator for next generation system call host intrusion detection systems.* MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM) Sunulmuş Bildiri.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*: Morgan Kaufmann Publishers Inc.

Roberts, S. W. (2000). Control chart tests based on geometric moving averages. *Technometrics,* (1), 97-101. doi: https://doi.org/10.2307/1271439

Sabahi, F., & Movaghar, A. (2008). *Intrusion detection: A survey.* 2008 Third International Conference on Systems and Networks Communications Sunulmuş Bildiri.

Sarmah, A. (2001). *Intrusion detection systems: definition, need and challenges*. SANS Institute Reading Room erişim adresi: https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343

Scarfone, K., & Mell, P. (2012). *Guide to intrusion detection and prevention systems (idps)*.

Sekar, R., Bendre, M., Dhurjati, D., & Bollineni, P. (2001). *A fast automaton-based method for detecting anomalous program behaviors.* IEEE Symposium on Security and Privacy (S&P 2001) Sunulmuş Bildiri.

Severi, G., Leek, T., & Dolan-Gavitt, B. (2018). *Malrec: compact full-trace malware recording for retrospective deep analysis.* International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment Sunulmuş Bildiri.

Song, Q., Hu, W., & Xie, W. (2002). Robust Support Vector Machine with bullet hole image classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C,* (4), 440-448. doi: https://doi.org/10.1109/tsmcc.2002.807277

Stavroulakis, P., & Stamp, M. (2010). *Handbook of information and communication security*: Springer Publishing Company, Incorporated.

Tan, K. M. C., Killourhy, K. S., & Maxion, R. A. (2002). *Undermining an anomaly-based intrusion detection system using common exploits*, Berlin, Heidelberg.

Tan, K. M. C., & Maxion, R. A. (2002). *"Why 6?" Defining the operational limits of stide, an anomaly-based intrusion detector*. 2002 IEEE Symposium on Security and Privacy.

Tandon, G., & Chan, P. K. (2003). *Learning rules from system call arguments and sequences for anomaly detection*.

Tong, F., & Yan, Z. (2017). A hybrid approach of mobile malware detection in Android. *Journal of Parallel and Distributed Computing,* 22-31. doi: https://doi.org/10.1016/j.jpdc.2016.10.012

Uma, M., & Ganapathi, P. (2013). *A survey on various cyber attacks and their classification*.

Vokorokos, L., & Baláž, A. (2010). *Host-based intrusion detection system.* 2010 IEEE 14th International Conference on Intelligent Engineering Systems Sunulmuş Bildiri.

Wagner, D., & Dean, R. (2001). *Intrusion detection via static analysis.* Proceedings 2001 IEEE Symposium on Security and Privacy (S&P 2001) Sunulmuş Bildiri.

Wagner, D., & Soto, P. (2002). *Mimicry attacks on host-based intrusion detection systems.* 9th ACM Conference on Computer and Communications Security Sunulmuş Bildiri.

Warrender, C., Forrest, S., & Pearlmutter, B. (1999). *Detecting intrusions using system calls: alternative data models.* 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344) Sunulmuş Bildiri.

White, S. R., Swimmer, M., Pring, E., Arnold, W. C., Chess, D. M., & Morar, J. (1999). *Anatomy of a commercial-grade immune system*.

Xie, M., & Hu, J. (2013). *Evaluating host-based anomaly detection systems: A preliminary analysis of ADFA-LD.*

6th International Congress on Image and Signal Processing (CISP) Sunulmuş Bildiri.

Xie, M., Hu, J., & Slay, J. (2014). *Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD.* 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) Sunulmuş Bildiri.

Xie, M., Hu, J., Yu, X., & Chang, E. (2015). *Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to ADFA-LD.* International Conference on Network and System Security Sunulmuş Bildiri.

Yao, J., Zhao, S., & Fan, L. (2006). *An enhanced Support Vector Machine Model for intrusion detection.* international conference on rough sets and knowledge technology Sunulmuş Bildiri, Berlin, Almanya.

Ye, N., Li, X., Chen, Q., Emran, S. M., & Xu, M. (2001). Probabilistic techniques for intrusion detection based on computer audit data. *IEEE Transactions on Systems, Man, Cybernetics-Part A: Systems Humans,* (4), 266-274. doi: https://doi.org/10.1109/3468.935043

Yolaçan, E. N., Dy, J. G., & Kaeli, D. R. (2014). *System call anomaly detection using multi-HMMs.* 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion Sunulmuş Bildiri.

Ypma, A., & Duin, R. P. (1998). *Support objects for domain approximation.* International Conference on Artificial Neural Networks Sunulmuş Bildiri.