

Modern EDR Sistemlerinde Makine Öğrenmesi ve Davranışsal Analiz ile Bilinmeyen Tehditlerin Tespiti

Mustafa Onur Erkan

Ağustos 2026

Özet

Geleneksel antivirüs çözümleri, daha önce tanımlanmış zararlı yazılımları imza ve benzeri göstergeler üzerinden tespit etmede etkili olsa da, bilinmeyen tehditler ve zero-day saldırıları karşısında tek başına yeterli olmayabilir. Bu nedenle modern endpoint güvenlik çözümlerinde yalnızca bilinen zararlıların tespitine değil, sistem üzerinde gerçekleşen davranışların analizine de önem verilmektedir.

Endpoint Detection and Response (EDR) sistemleri; process aktiviteleri, komut satırı bilgileri, dosya ve Registry değişiklikleri ile ağ bağlantıları gibi farklı telemetri verilerini toplayarak olaylar arasındaki ilişkileri inceleyebilir. Bu veriler davranışsal analiz, sezgisel yöntemler, tehdit istihbaratı ve makine öğrenmesi gibi farklı tekniklerle değerlendirilerek daha önce tanımlanmamış şüpheli aktivitelerin tespit edilmesine yardımcı olur. Makine öğrenmesi bu süreçte sınıflandırma, anomali tespiti ve risk skortlama gibi görevlerde kullanılabilir; ancak EDR sistemlerinin tamamı yalnızca makine öğrenmesine dayanmaz.

Bu çalışmada EDR sistemlerinin temel çalışma prensibi, davranışsal analiz ve makine öğrenmesinin tehdit tespitindeki rolü, bilinmeyen tehdit ile zero-day saldırısı arasındaki fark ve bu yöntemlerin sınırlamaları ele alınmaktadır. Ayrıca farklı tespit yöntemlerinin birlikte kullanıldığı hibrit yaklaşımın bilinmeyen tehditlerin tespitinde neden önemli olduğu örnek bir saldırı senaryosu üzerinden incelenmektedir.

İçindekiler

Özet.....	1
İçindekiler.....	2
1. Giriş.....	3
2. Geleneksel Antivirüs ve İmza Tabanlı Tespitin Sınırları.....	3
3. EDR Sistemlerinin Temel Çalışma Prensipleri.....	4
4. Davranışsal Analiz ile Tehdit Tespiti.....	5
5. EDR Sistemlerinde Makine Öğrenmesinin Rolü.....	6
6. Bilinmeyen ve Zero-Day Tehditlerin Tespiti.....	7
7. Hibrit Tehdit Tespit Yaklaşımı.....	7
8. EDR'nin Tespit Sonrası Müdahale Yetenekleri.....	8
9. Makine Öğrenmesi ve Davranışsal Analizin Sınırlamaları.....	9
10. Örnek Saldırı Senaryosu: Bilinmeyen Tehdidin EDR Tarafından Analizi.....	10
11. Sonuç.....	11
Kaynakça.....	12

1. Giriş

Siber güvenlikte temel amaçlardan biri, zararlı bir faaliyeti mümkün olduğunca erken fark etmek ve saldırganın sistem içinde ilerlemesini zorlaştırmaktır. Ancak saldırı yöntemleri sürekli değişmektedir. Günümüzde saldırganlar yalnızca doğrudan zararlı yazılım çalıştırmak yerine PowerShell, WMI veya rundll32 gibi işletim sisteminde zaten bulunan araçları da kötüye kullanabilmektedir. Bu nedenle sadece bilinen zararlı dosyaları tanımaya dayanan güvenlik yaklaşımı bazı saldırılarda yetersiz kalabilir.

Endpoint Detection and Response (EDR), uç noktada gerçekleşen olayları daha ayrıntılı biçimde izlemek ve şüpheli aktivitelerde müdahale edebilmek için kullanılan bir güvenlik yaklaşımıdır. Microsoft Defender for Endpoint belgelerinde EDR'nin saldırıları tespit etmeye, olayın kapsamını incelemeye ve tehditlere karşı yanıt vermeye yardımcı olduğu belirtilmektedir [1]. EDR'nin klasik antivirüsten önemli farklarından biri yalnızca dosyaya bakmamasıdır. İşlem (process) aktiviteleri, ağ bağlantıları, kullanıcı oturumları, Registry ve dosya sistemi gibi farklı kaynaklardan gelen veriler birlikte değerlendirilebilir [1].

Bu özellik özellikle daha önce tanımlanmamış tehditlerde önem kazanır. Bir dosyanın hash değeri tehdit istihbaratı veritabanında bulunmayabilir. Buna rağmen dosyanın başlattığı işlemler, değiştirdiği Registry anahtarları veya kurduğu ağ bağlantıları şüpheli olabilir. Bu nedenle modern endpoint güvenlik ürünleri imza tabanlı tespitini tamamen bırakmak yerine davranışsal analiz, sezgisel yöntemler, itibar kontrolleri ve ML gibi ek yöntemlerle destekleyebilmektedir [3][4].

Bu yazıda “bilinmeyen tehdit” ifadesi, güvenlik ürününün daha önce tanımlanmış bir imza veya göstergeyle doğrudan eşleştiremediği zararlı ya da şüpheli faaliyetler için kullanılmaktadır. “Zero-day saldırısı” ise daha özel bir kavramdır. NIST'e göre zero-day saldırısı, daha önce bilinmeyen bir donanım, firmware veya yazılım zafiyetinin istismar edilmesiyle gerçekleşir [8]. Bu yüzden her yeni zararlı yazılım zero-day değildir. Örneğin yeni hazırlanmış bir zararlı yazılım, kullanıcıyı kandıran bir e-posta ile sisteme girebilir ve hiçbir zero-day açığı kullanmayabilir.

Bu yazının amacı, EDR sistemlerinin bilinmeyen tehditleri hangi veriler üzerinden değerlendirdiğini açıklamak, makine öğrenmesinin bu yapıda nerede kullanıldığını göstermek ve zero-day ile bilinmeyen tehdit kavramlarını birbirinden ayırmaktır. Ayrıca neden tek bir tespit yöntemi yerine birden fazla yöntemin birlikte kullanıldığı üzerinde durulacaktır.

2. Geleneksel Antivirüs ve İmza Tabanlı Tespitin Sınırları

Geleneksel antivirüs çözümlerinin kullandığı temel yöntemlerden biri imza tabanlı tespittir. Daha önce zararlı olduğu belirlenmiş dosyalara ait hash değerleri, karakteristik kod parçaları veya başka dosya imzaları güvenlik ürünleri tarafından eşleştirme amacıyla kullanılabilir. IP adresi, alan adı ve URL gibi göstergeler ise IOC ve tehdit istihbaratı mekanizmaları tarafından ayrıca değerlendirilebilir. Yeni bir dosya veya gözlemlenen aktivite bu bilgilerden biriyle eşleştğinde tehdit daha hızlı tespit edilebilir. Bu yöntem özellikle bilinen zararlılara karşı oldukça kullanışlıdır ve modern güvenlik ürünlerinde de kullanılmaya devam etmektedir. Sorun, saldırganların bu göstergeleri değiştirebilmesidir. Zararlı bir dosyada yapılan küçük bir değişiklik bile farklı bir hash değeri oluşturabilir. Polimorfik zararlı yazılımlar kod yapılarını değiştirebilir. Dosyasız saldırılarda ise saldırgan PowerShell, WMI veya rundll32 gibi normalde meşru olan araçları kullanabilir. Böyle bir durumda sadece “bu dosya daha önce zararlı olarak görüldü mü?” sorusuna bakmak saldırının tamamını görmek için yeterli olmayabilir.

Zero-day saldırılarında bu sorun daha belirgin hale gelir. Güvenlik açığı veya kullanılan istismar yöntemi henüz bilinmiyorsa doğrudan o saldırıya ait hazır bir imza olmayabilir. Ancak saldırı başarılı olduktan sonra sistemde bazı izler oluşabilir. Örneğin beklenmeyen bir child process başlatılabilir, PowerShell çalıştırılabilir, Registry üzerinde kalıcılık için değişiklik yapılabilir veya dış bir sunucuya bağlantı kurulabilir. EDR, bu tür olayları izleyerek saldırının sonraki adımlarını tespit etmeye çalışır. Bu nedenle güncel endpoint güvenliğinde imza tabanlı tespit tek başına kullanılmaz; davranışsal ve bağlamsal yöntemlerle desteklenir. CISA'nın StopRansomware rehberi de uç noktalarda EDR kullanılmasını önermektedir [7]. Buradaki amaç imza tabanlı sistemi bırakmak değil, bilinen tehditlerde hızlı çalışan bu yöntemi bilinmeyen veya değişken tehditlere karşı ek analiz katmanlarıyla güçlendirmektir.

Tablo 1. İmza odaklı antivirüs ile modern EDR yaklaşımının genel karşılaştırması

Özellik	Geleneksel İmza Odaklı Antivirüs	Modern EDR Yaklaşımı
Temel soru	Bu dosya/gösterge daha önce zararlı olarak biliniyor mu?	Uç noktada ne oldu ve olaylar birbiriyle nasıl ilişkili?
Başlıca veri	Dosya, imza, hash	Davranış ve olay bağlamı üzerinden ek tespit imkanı
Bilinmeyen tehdit	Sınırlı görünürlük	Zaman çizelgesi, process ilişkileri ve olay korelasyonu
Olay inceleme	Genellikle sınırlı bağlam	Zaman çizelgesi, process ilişkileri, komut satırı, dosya/Registry/ağ olayları ve olay korelasyonu
Müdahale	Silme / karantina	Karantina, işlem durdurma, cihaz izolasyonu ve inceleme

3. EDR Sistemlerinin Temel Çalışma Prensibi

EDR sistemleri genellikle uç noktada çalışan bir ajan veya işletim sistemiyle bütünleşik sensörler üzerinden telemetri toplar. Bu veriler cihaz üzerinde analiz edilebildiği gibi merkezi bir sunucuya veya bulut ortamına da gönderilebilir. NIST SP 1800-24C içinde incelenen Symantec EDR örneğinde, davranışsal analiz kullanılarak potansiyel zararlı aktivitelerin belirlenebildiği ve olayların önceliklendirilebildiği görülmektedir [2]. Bu örnek bütün EDR ürünlerinin aynı mimariyi kullandığı anlamına gelmez, ancak davranış analizinin EDR içinde nasıl kullanılabileceğini göstermektedir.

Her EDR ürünü aynı miktarda veri toplamaz. Yine de sık kullanılan telemetri kaynakları arasında process başlatma ve sonlandırma olayları, parent-child process ilişkileri, komut satırı parametreleri, kullanıcı oturumları, dosya işlemleri, Registry değişiklikleri ve ağ bağlantıları bulunur. Microsoft'un EDR dokümantasyonunda çekirdek ve bellek yöneticisi gibi daha düşük seviyeli telemetrilerin de kullanılabildiği belirtilmektedir [1]. Bu nedenle EDR'yi tek bir algoritma olarak değil; veri toplama, analiz, korelasyon ve müdahale aşamalarından oluşan bir sistem olarak düşünmek daha uygundur.

EDR'nin önemli faydalarından biri, bu kayıtları tek tek göstermek yerine aralarındaki ilişkilerin incelenmesine yardımcı olmasıdır. Örneğin bir Office uygulamasının PowerShell başlatması, PowerShell'in kodlanmış bir komut çalıştırması ve sonrasında dış bir IP adresine bağlantı kurulması ayrı olaylar olarak kaydedilebilir. EDR bu olayları cihaz, kullanıcı, process kimliği ve zaman bilgisi üzerinden bir araya getirerek tek bir şüpheli olay zinciri olarak gösterebilir.

EDR sistemlerinde tespit ve müdahale aynı sürecin parçalarıdır. Bir olay yüksek riskli olarak değerlendirildiğinde sistem yalnızca alarm üretmekle kalmayabilir. Ürünün özelliklerine ve kurumun politikasına göre şüpheli aktiviteler engellenebilir, dosyalar karantinaya alınabilir veya cihaz ağdan izole edilebilir [1][4].

Şekil 1. Basitleştirilmiş EDR tespit ve müdahale akışı

Telemetri	Özellik / Sinyal	Hibrit Analiz	Korelasyon	Alarm / Yanıt
-----------	------------------	---------------	------------	---------------

4. Davranışsal Analiz ile Tehdit Tespiti

Davranışsal analizde temel olarak bir dosyanın veya process'in sadece kimliğine değil, sistem içinde ne yaptığına bakılır. Bir process'i hangi parent process'in başlattığı, hangi komutla çalıştığı, hangi dosyalara eriştiği, Registry üzerinde ne değiştirdiği ve hangi ağ adresleriyle iletişim kurduğu birlikte incelenebilir. Böylece saldırıda kullanılan dosyanın imzası bilinmese bile şüpheli bir davranış zinciri fark edilebilir.

PowerShell bunun için iyi bir örnektir. PowerShell sistem yöneticilerinin de kullandığı normal bir Windows aracıdır. Bu yüzden sadece powershell.exe çalıştı diye alarm üretmek çok fazla yanlış pozitif neden olabilir. MITRE ATT&CK, saldırganların PowerShell'i komut veya betik çalıştırmak için kötüye kullanabildiğini T1059.001 tekniği altında tanımlar [5]. EDR açısından önemli olan PowerShell'in çalışması değil, hangi process tarafından başlatıldığı, hangi parametrelerle kullanıldığı ve sonrasında ne yaptığıdır.

Process tree, bu bağlamı anlamının en kolay yollarından biridir. Örneğin "explorer.exe → chrome.exe" birçok kullanıcı için normal bir işlem zinciridir. Buna karşılık "winword.exe → powershell.exe → rundll32.exe" zinciri, özellikle bir belge açıldıktan hemen sonra ortaya çıkmış ve ardından dış bir sunucuya bağlantı kurulmuşsa daha şüpheli kabul edilebilir. Yine de tek bir process zincirine bakarak kesin karar vermek doğru değildir; başka telemetri verileri de değerlendirilmelidir.

Davranışsal tespit hem kurallarla hem de istatistiksel veya ML tabanlı yöntemlerle yapılabilir. Kural tabanlı yöntemde güvenlik uzmanının bilgisi belirli koşullara dönüştürülür. Örneğin bir Office uygulamasının PowerShell başlatması ve aynı zincirin kısa süre içinde dış bağlantı kurması bir alarm kuralının parçası olabilir. ML tarafında ise çok sayıda özellik birlikte değerlendirilerek elle yazılması zor olan örüntülere risk skoru verilebilir. Bu noktada Indicator of Compromise (IOC) ile bazı üreticilerin kullandığı Indicator of Attack (IOA) kavramlarını ayırmak faydalıdır. IOC; zararlı hash, URL, alan adı veya IP gibi gözlemlenebilir göstergelere odaklanır. IOA ise saldırı devam ederken görülen davranışları ifade etmek için kullanılan bir terimdir. IOA evrensel bir standart değildir; örneğin CrowdStrike bu kavramı saldırı sırasında gözlemlenen aktiviteler için kullanmaktadır [11]. Bilinmeyen bir tehditte hazır IOC olmayabilir, fakat davranış yine de şüpheli olabilir.

5. EDR Sistemlerinde Makine Öğrenmesinin Rolü

Makine öğrenmesi (ML), tek başına bir endpoint güvenlik ürünü değil, modern endpoint güvenlik sistemlerinde tehdit tespitini desteklemek amacıyla kullanılabilen bir analiz yöntemidir. Ayrıca her ML modelinin doğrudan EDR motorunun içinde çalıştığını söylemek doğru olmaz. Örneğin Microsoft'un mimarisinde istemci ve bulut tarafındaki ML motorları ile davranış izleme bileşenleri, EDR tarafında kullanılan sinyalleri destekleyebilmektedir [3]. Bu nedenle EDR'nin ML'den yararlanabildiğini söylemek, “EDR tamamen ML ile çalışır” demekten daha doğrudur.

Basit bir ML tespit süreci “telemetri → özellik çıkarımı → model → risk skoru → karar” şeklinde düşünülebilir. Örneğin parent-child process ilişkisi, komut satırı özellikleri, kısa sürede değiştirilen dosya sayısı, ağ hedefinin ne kadar nadir olduğu veya bir process'in sistemde daha önce ne sıklıkla görüldüğü modele özellik olarak verilebilir. Model bu verileri kullanarak bir dosyayı sınıflandırabilir veya bir davranışın ne kadar olağandışı olduğunu skorlayabilir.

Gözetimli öğrenmede model, daha önce zararlı veya normal olarak etiketlenmiş örneklerden öğrenir. Zararlı yazılım tespitinde statik dosya özelliklerini kullanan derin öğrenme çalışmaları bu yaklaşımın örneklerindendir [12]. Ancak bir modelin laboratuvar ortamında yüksek başarı göstermesi, gerçek bir kurum ağında aynı başarıyı göstereceği anlamına gelmez. Sonuçlar kullanılan veri setine ve ortama göre değişebilir.

Gözetimsiz öğrenme ve anomali tespitinde her saldırı türünü önceden etiketlemek zorunlu değildir. Bu tür yöntemler, etiketlenmemiş veriler içerisindeki olağandışı örnekleri belirlemeye çalışabilir. Örneğin Isolation Forest, veri noktalarını rastgele bölmeler kullanarak ayırır ve diğer örneklerden daha kolay izole edilen noktaları daha yüksek anomali skoruyla değerlendirebilir [13]. Bu yaklaşım bilinmeyen tehditlerde yararlı olabilir; fakat burada önemli bir problem vardır: anormal olan her şey zararlı değildir. Örneğin bir sistem yöneticisinin nadiren çalıştırdığı meşru bir script de anomali olarak görülebilir.

Microsoft Defender Antivirus gibi çok katmanlı endpoint güvenlik çözümlerinde tek bir analiz yöntemi veya ML modeli yerine farklı tespit motorları birlikte kullanılabilir. Davranış kuralları, itibar bilgileri, tehdit istihbaratı ve farklı modeller birlikte çalışabilir. Microsoft'un mimarisinde istemci tarafı ML, davranış izleme, komut satırı analizi, bellek taraması, sezgisel yöntemler ve bulut tarafındaki ML motorlarının birlikte kullanılabildiği açıklanmaktadır [3]. Bu durum ML'nin EDR'nin tamamı değil, çok katmanlı tespit sisteminin bir parçası olduğunu göstermektedir.

Tablo 2. Endpoint telemetrisinin davranışsal ve ML tabanlı analizde kullanımına örnekler

Telemetri / Özellik	Örnek Davranışsal veya ML Kullanımı
Parent-child işlem ilişkisi	Alışılmadık işlem zincirlerini belirleme veya skollama
Komut satırı	Kodlanmış, gizlenmiş veya olağandışı parametreleri değerlendirme
Dosya olayları	Kısa sürede toplu değiştirme veya şüpheli dosya oluşturma davranışını inceleme
Registry olayları	Kalıcılık amaçlı olağandışı değişiklikleri belirleme
Ağ bağlantıları	Nadir hedef, yeni alan adı veya beklenmeyen iletişim örüntüsünü değerlendirme
Zaman ve sıklık	Kullanıcının veya cihazın olağan davranışından sapmayı skollama
Dosya yapısal özellikleri	Statik zararlı yazılım sınıflandırması ve risk skollama

6. Bilinmeyen ve Zero-Day Tehditlerin Tespiti

Bilinmeyen tehditler konusunda sık yapılan bir hata, ML'nin doğrudan “bu saldırı zero-day” diyebilen özel bir teknoloji gibi düşünülmesidir. Aslında model çoğu zaman kullanılan açığın adını veya CVE numarasını bilmez. Bunun yerine saldırının sistemde oluşturduğu belirtilere bakar. Beklenmeyen process oluşturma, PowerShell çalıştırma, process injection benzeri davranışlar, kalıcılık girişimleri veya normal olmayan ağ bağlantıları bu belirtilere örnek verilebilir. Örneğin bir tarayıcıda daha önce bilinmeyen bir güvenlik açığının kullanıldığını düşünelim. NIST tanımına göre bu saldırı zero-day olabilir [8]. EDR veya başka bir endpoint koruma bileşeni açığı özel bir imza ile tanımayabilir. Fakat tarayıcının beklenmeyen bir child process oluşturmaya, bu process'in PowerShell başlatması ve sonrasında dış bir sunucuya bağlanması dikkat çekici davranışlar oluşturur. Davranış kuralları ve ML modelleri bu olayları birlikte değerlendirerek risk seviyesini yükseltebilir [3][4]. Bu yüzden EDR'nin bilinmeyen tehditlerdeki asıl avantajı, önceden bilinmeyen saldırının adını doğru tahmin etmek değildir. Daha önemli olan, saldırının sistem üzerinde bıraktığı davranışsal izleri toplamak ve bu olaylar arasındaki ilişkiyi ortaya çıkarmaktır. Saldırgan mümkün olduğunca az iz bırakmaya çalışsa bile komut çalıştırma, veri erişimi veya dış sistemlerle iletişim gibi işlemler genellikle bir miktar telemetri üretir.

Zero-day ile bilinmeyen zararlı yazılım aynı şey değildir. Zero-day, daha önce bilinmeyen bir zafiyetin saldırıda kullanılmasıyla ilgilidir [8]. Bilinmeyen zararlı yazılım ise güvenlik ürününün daha önce tanımadığı yeni veya değiştirilmiş bir zararlı olabilir. Bilinen bir malware zero-day açık üzerinden sisteme girebilir. Tersine, yeni bir malware hiçbir zero-day kullanmadan phishing, çalınmış parola veya yanlış yapılandırma üzerinden çalıştırılabilir.

Davranışsal ve ML tabanlı tespit bileşenleri sınıflandırma sonucu, anomali skoru, risk skoru veya güvenlik alarmı gibi farklı çıktılar üretebilir. Tek bir şüpheli olay her zaman saldırı anlamına gelmez. Ancak birden fazla şüpheli sinyal aynı zaman aralığında ve aynı process zincirinde görülürse olayın önceliği yükseltebilir. Bu, SOC analistlerinin binlerce ayrı log yerine daha anlamlı ve ilişkilendirilmiş olaylara odaklanmasına yardımcı olur.

7. Hibrit Tehdit Tespit Yaklaşımı

Modern endpoint güvenlik sistemleri genellikle tek bir tespit yöntemine bağlı değildir. İmza kontrolü bilinen zararlıları hızlı şekilde yakalayabilir. IOC eşleştirme bilinen kötü IP, alan adı veya hash değerlerini kontrol eder. Sezgisel kurallar belirli riskli davranışlara bakar. Davranışsal analiz olayın bağlamını inceler. ML ise çok sayıda özelliği birlikte değerlendirerek sınıflandırma veya anomali skoru üretebilir. Tehdit istihbaratı da dış kaynaklardan ek bilgi sağlar. Bu yöntemler birbirinin yerine geçmekten çok birbirini tamamlar. Örneğin yeni indirilen bir dosyanın hash değeri veritabanında olmayabilir. Statik ML modeli dosyayı orta riskli görebilir. Dosya çalıştıktan sonra Office uygulamasından türeyen bir PowerShell process'i oluşturmaya ve Registry üzerinde kalıcılık sağlamaya çalışması davranışsal riski artırabilir. Daha sonra bağlandığı alan adı hakkında kötü itibar bilgisi bulunursa olayın önceliği daha da yükselebilir. Korelasyon katmanı bu bilgileri tek bir alarm altında toplayabilir.

Hibrit yaklaşımın bir avantajı da tek bir yöntemin hata yapması durumunda diğer yöntemlerin ek kanıt sağlayabilmesidir. Saldırgan dosyanın hash değerini kolayca değiştirebilir; ancak saldırının bütün davranışını normal göstermek daha zor olabilir. Öte yandan ML yanlış pozitif üretirse davranışsal veriler, güvenilir imzalar veya analist incelemesi kararın düzeltilmesine yardımcı olabilir.

Bu nedenle ML, modern endpoint güvenlik platformlarında diğer tespit yöntemleriyle birlikte kullanılabilen önemli bileşenlerden biridir [3][4].

Hibrit yaklaşım MITRE ATT&CK ile de desteklenebilir. ATT&CK Enterprise Matrix, saldırganların kullandığı davranışları taktik ve teknikler altında sınıflandırır [6]. Bir EDR alarımının ATT&CK teknikleriyle eşleştirilmesi, analistin olayın saldırı sürecindeki yerini daha kolay anlamasına yardımcı olabilir.

Tablo 3. Hibrit endpoint tehdit tespit mimarisindeki temel katmanlar

Katman	Temel Görev	Güçlü Olduğu Alan
İmza / Hash	Bilinen zararlıyı eşleştirmek	Hızlı ve yüksek güvenilirlikli bilinen tehdit tespiti
IOC / Tehdit İstihbaratı	Bilinen kötü altyapı veya göstergeleri eşleştirmek	Güncel dış bağlam
Sezgisel Kurallar	Uzman bilgisini açık koşullara dönüştürmek	Belirli saldırı kalıpları
Davranışsal Analiz	Olay zincirini ve bağlamı incelemek	Living-off-the-land ve bilinmeyen davranışlar
Makine Öğrenmesi	Çoklu özelliklerden sınıflandırma veya anomali skoru üretmek	Büyük veri içinde karmaşık örüntüleri bulma
Analist Doğrulaması	Bağlamı ve iş etkisini değerlendirmek	Belirsiz olaylar ve olay müdahalesi

8. EDR'nin Tespit Sonrası Müdahale Yetenekleri

EDR'deki "Response" kısmı, sistemi sadece alarm üreten bir araç olmaktan çıkarır. Şüpheli bir olay tespit edildiğinde ürünün yeteneklerine ve kurumun politikasına göre otomatik veya analist onaylı müdahaleler yapılabilir. Microsoft Defender for Endpoint dokümantasyonunda cihaz izolasyonu, antivirüs taraması, dosyayı durdurma veya karantinaya alma ve gösterge tabanlı engelleme gibi yanıt işlemleri örnek olarak verilmektedir [1].

EDR ürününe bağlı olarak şüpheli veya zararlı process'lerin sonlandırılması gibi ek müdahale yöntemleri de bulunabilir. Dosya karantinası, şüpheli dosyanın normal şekilde kullanılmasını engeller. Cihaz izolasyonu ise ele geçirilmiş olabilecek bilgisayarın kurumsal ağ ile iletişimini büyük ölçüde sınırlar. Böylece saldırganın başka sistemlere geçmesi veya veri çıkarması zorlaştırılabilir. Bazı ürünler uzaktan inceleme, dosya toplama veya otomatik düzeltme gibi ek özellikler de sunabilir.

Otomatik müdahale her zaman en iyi seçenek değildir. Örneğin kritik bir sunucuda yanlış pozitif nedeniyle önemli bir process'in sonlandırılması hizmet kesintisine neden olabilir. Bu yüzden kurumlar alarımın güven seviyesine ve cihazın önemine göre farklı politikalar kullanabilir. Çok yüksek güvenle bilinen bir zararlı otomatik engellenirken, daha belirsiz bir anomali alarmı önce analistin incelemesine bırakılabilir.

EDR'nin olay zaman çizelgesi de inceleme sürecinde önemlidir. Analist sadece son alarmı değil, alarmdan önce ve sonra oluşan process zincirlerini, komutları, dosya ve Registry değişikliklerini ve ağ bağlantılarını inceleyebilir. Bu sayede saldırının ne kadar ilerlediği ve başka cihazların etkilenip etkilenmediği daha kolay araştırılabilir.

9. Makine Öğrenmesi ve Davranışsal Analizin Sınırlamaları

Makine öğrenmesi bilinmeyen tehditlerin tespitine yardımcı olsa da kusursuz değildir. En önemli problemlerden biri yanlış pozitif ve yanlış negatif dengesidir. Model çok hassas ayarlanırsa normal fakat nadir aktiviteleri saldırı olarak işaretleyebilir. Daha gevşek ayarlanırsa bazı saldırıları kaçırabilir. NIST AI RMF kaynaklarında da model doğruluğu değerlendirilirken false positive ve false negative oranlarının dikkate alınması gerektiği belirtilmektedir [9]. Bu denge SOC'un iş yükünü doğrudan etkiler.

Bir diğer problem eğitim verisidir. Gözetimli ML modelleri etiketli örneklerle ihtiyaç duyar. Eğer eğitim verisi belirli saldırı türlerine veya belirli sistemlere fazla ağırlık veriyorsa model başka ortamlarda aynı başarıyı göstermeyebilir. Ayrıca kurum içindeki kullanım alışkanlıkları zamanla değişir. Yeni programlar kurulabilir, kullanıcıların çalışma biçimi değişebilir. Bu durumda modelin öğrendiği normal davranış ile gerçek ortam arasındaki fark artabilir. Bu problem kavram kayması (concept drift) olarak adlandırılır.

Açıklanabilirlik de önemlidir. Modelin sadece “yüksek risk” demesi bir SOC analisti için yeterli olmayabilir. Analist hangi process ilişkisinin, komut satırının veya ağ bağlantısının bu skoru yükselttiğini görmek ister. Bu nedenle ML sonucunun, insanın inceleyebileceği davranışsal kanıtlarla birlikte gösterilmesi daha kullanışlıdır [9].

ML sistemleri saldırganlar tarafından da hedef alınabilir. Adversarial machine learning alanı, saldırganın model girdilerini değiştirmesi, eğitim verisini zehirlmesi veya modelin kararından kaçmaya çalışması gibi konuları inceler. NIST AI 100-2e2025 bu saldırıları ve olası savunma yöntemlerini sınıflandırmaktadır [10]. Bu yüzden yalnızca modelin kendisi değil, modele veri sağlayan sistemin de güvenli olması gerekir.

Davranışsal analizde de benzer şekilde bağlam önemlidir. Sistem yöneticileri PowerShell, PsExec veya WMI gibi saldırganların da kullandığı araçları normal işleri için kullanabilir. Çok katı kurallar bu normal işlemleri sürekli alarm olarak gösterebilir ve alarm yorgunluğuna neden olabilir. Bu yüzden tek bir olaya bakmak yerine kullanıcı, cihaz, zaman, process zinciri ve tehdit istihbaratı gibi farklı bilgiler birlikte değerlendirilmelidir.

Sonuç olarak ML ve davranışsal analiz, zero-day veya bilinmeyen tehdit sorununu tamamen çözmez. Bu yöntemlerin asıl katkısı, daha önce adı konmamış bir saldırının oluşturduğu sıra dışı davranışları fark etme ihtimalini artırmaktır. Yine de analist incelemesi, güncel tehdit istihbaratı, doğru yapılandırma ve diğer güvenlik katmanları gerekli olmaya devam eder.

10. Örnek Saldırı Senaryosu: Bilinmeyen Tehdidin EDR Tarafından Analizi

Aşağıdaki senaryo belirli bir EDR ürününün gerçek kuralını göstermemektedir. Amaç, farklı telemetri ve analiz katmanlarının bilinmeyen bir saldırıyı nasıl ortaya çıkarabileceğini basit bir örnek üzerinden açıklamaktır.

Aşama 1 – İlk erişim: Kullanıcı e-posta eki olarak gelen bir Office belgesini açar. Dosyanın hash değeri tehdit istihbaratı veritabanında bulunmadığı için yalnızca hash kontrolüne göre zararlı olarak işaretlenmez.

Aşama 2 – Şüpheli process zinciri: Belge açıldıktan kısa süre sonra WINWORD.EXE, PowerShell'i başlatır. PowerShell kodlanmış bir komut çalıştırır. Bu ilişki tek başına kesin saldırı kanıtı değildir, ancak davranışsal açıdan şüpheli kabul edilebilir.

Aşama 3 – Ağ bağlantısı: PowerShell veya onun başlattığı başka bir process daha önce cihazda görülmeyen bir dış alan adına bağlanır. Alan adı bilinen kötü bir adres olmasa bile hedefin nadir olması ve process bağlamı risk skorunu artırabilir.

Aşama 4 – Kalıcılık: Aynı process zinciri, kullanıcının oturum açmasından sonra otomatik çalışmak için Registry üzerinde bir değer oluşturmaya çalışır. Böylece olayın tek seferlik bir script çalıştırmadan daha ciddi olabileceğine dair yeni bir sinyal oluşur.

Aşama 5 – Korelasyon ve risk skoru: Sistem; Office → PowerShell ilişkisini, kodlanmış komutu, nadir ağ hedefini ve Registry değişikliğini aynı olay altında birleştirir. Davranış kuralları ve ML bileşenleri bu sinyallere göre risk seviyesini yükseltebilir. Zararlının adı bilinmese bile olay yüksek öncelikli olarak değerlendirilebilir.

Aşama 6 – Müdahale: Kurumun politikasına göre PowerShell process'i sonlandırılabilir, ilgili dosya karantinaya alınabilir veya cihaz izole edilebilir. Daha sonra SOC analisti olay zaman çizelgesini inceleyerek benzer aktivitelerin başka cihazlarda olup olmadığını araştırabilir.

Tablo 4. Örnek saldırı senaryosunda oluşabilecek endpoint sinyalleri

Gözlenen Olay	Neden Önemli?	Olası Analiz Katmanı
WINWORD → PowerShell	Office uygulamasından betik yürütme bağlamı	Davranış kuralı / işlem ağacı
Kodlanmış komut	Gizleme veya otomasyon göstergesi olabilir	Komut satırı analizi / ML
Nadir dış alan adı	Cihazın olağan ağ profilinden sapma	Anomali / tehdit istihbaratı
Registry kalıcılığı	Persistence davranışı	Davranış kuralı / ATT&CK eşleştirme
Birleşik olay zinciri	Tekil sinyallerden daha güçlü bağlam	Korelasyon / risk skoru

Bu örnek, bilinmeyen tehdit tespiti için tek bir "AI kararı" olmadığını göstermektedir. İlk hash kontrolü sonuç vermese bile process zinciri, komut satırı, ağ bağlantısı ve Registry değişikliği birlikte değerlendirilerek şüpheli bir saldırı görülebilir hale gelebilir. ML bu verilerin sınıflandırılması veya skorlanmasında yardımcı olurken, davranış kuralları, tehdit istihbaratı ve analist incelemesi de kararın önemli parçalarıdır.

11. Sonuç

Günümüzde sadece bilinen zararlı yazılım imzalarına dayanan bir endpoint güvenlik yaklaşımı yeterli değildir. Saldırganlar zararlı kodu değiştirebilir, sistemde bulunan meşru araçları kötüye kullanabilir ve bazı durumlarda daha önce bilinmeyen güvenlik açıklarından yararlanabilir. Bu yüzden savunma tarafının dosya imzasının yanında sistem davranışını da izlemesi gerekir. EDR sistemleri bu ihtiyaca, uç noktada gerçekleşen olayları toplayıp birbiriyle ilişkilendirerek cevap verir. Process ilişkileri, komut satırı bilgileri, dosya ve Registry değişiklikleri ile ağ bağlantıları birlikte incelendiğinde tek başına normal görünen olaylar daha anlamlı hale gelebilir. Özellikle living-off-the-land gibi meşru araçların kötüye kullanıldığı saldırılarda davranışsal analiz önemli bir avantaj sağlar.

Makine öğrenmesi modern endpoint güvenliğinde yararlı bir yöntemdir; ancak EDR'nin kendisi makine öğrenmesi değildir ve ML de doğrudan “zero-day tespiti” yapan sihirli bir çözüm değildir. ML; dosya sınıflandırma, davranış skorlama ve anomali tespiti gibi görevlerde kullanılabilir. Bazı platformlarda bu modeller başka koruma katmanlarında çalışıp EDR'ye sinyal sağlar. Önemli olan, ML sonucunun diğer güvenlik verileriyle birlikte değerlendirilmesidir.

Zero-day ile bilinmeyen tehdit arasındaki fark da bu yüzden önemlidir. Zero-day, daha önce bilinmeyen bir zafiyetin saldırıda kullanılmasıdır [8]. Bilinmeyen tehdit ise daha geniş bir kavramdır ve yeni bir zararlı yazılımı, daha önce görülmemiş bir davranış zincirini veya henüz tanımlanmamış başka saldırı aktivitelerini kapsayabilir. EDR çoğu zaman saldırının adını önceden bilmek yerine sistemde bıraktığı davranışsal izleri yakalamaya çalışır.

En mantıklı yaklaşım, farklı tespit yöntemlerini birlikte kullanmaktır. İmza ve IOC kontrolleri bilinen tehditlerde hızlı sonuç verir. Davranışsal analiz olayın bağlamını gösterir. ML çok sayıda veriden örüntü veya anomali bulmaya yardımcı olur. Tehdit istihbaratı güncel dış bilgi sağlar ve analist de bütün bu verileri gerçek sistem bağlamında değerlendirir. Bu şekilde hem bilinen hem de daha önce görülmemiş tehditlere karşı daha dengeli bir savunma oluşturulabilir.

Önümüzdeki yıllarda endpoint güvenliğinde daha gelişmiş davranış modellerinin, farklı cihazlardan gelen verilerin daha fazla ilişkilendirilmesinin ve açıklanabilir ML yöntemlerinin daha yaygın kullanılması beklenebilir. Ancak kullanılan teknoloji ne kadar gelişirse gelişsin, doğru veri toplama, güvenli yapılandırma ve yetkin güvenlik analistleri önemini koruyacaktır.

Kaynakça

- [1] Microsoft, "Overview of endpoint detection and response capabilities - Microsoft Defender for Endpoint," Microsoft Learn, Jun. 2, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/defender-endpoint/overview-endpoint-detection-response>. [Accessed: Aug. 19, 2026].
- [2] National Institute of Standards and Technology, National Cybersecurity Center of Excellence, "NIST SP 1800-24C: Securing Picture Archiving and Communication System (PACS)," Sec. 2.7.6, Symantec Endpoint Detection and Response. [Online]. Available: <https://www.nccoe.nist.gov/publication/1800-24/VoIC/index.html>. [Accessed: Aug. 19, 2026].
- [3] Microsoft, "Advanced technologies at the core of Microsoft Defender Antivirus," Microsoft Learn, May 22, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/defender-endpoint/adv-tech-of-mdav>. [Accessed: Aug. 19, 2026].
- [4] Microsoft, "Behavioral blocking and containment," Microsoft Learn, May 22, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/defender-endpoint/behavioral-blocking-containment>. [Accessed: Aug. 19, 2026].
- [5] MITRE, "Command and Scripting Interpreter: PowerShell, T1059.001," MITRE ATT&CK. [Online]. Available: <https://attack.mitre.org/techniques/T1059/001/>. [Accessed: Aug. 19, 2026].
- [6] MITRE, "Enterprise Matrix," MITRE ATT&CK. [Online]. Available: <https://attack.mitre.org/matrices/enterprise/>. [Accessed: Aug. 19, 2026].
- [7] Cybersecurity and Infrastructure Security Agency, "#StopRansomware Guide." [Online]. Available: <https://www.cisa.gov/stopransomware/ransomware-guide>. [Accessed: Aug. 19, 2026].
- [8] National Institute of Standards and Technology, "Zero Day Attack," Computer Security Resource Center Glossary. [Online]. Available: https://csrc.nist.gov/glossary/term/zero_day_attack. [Accessed: Aug. 19, 2026].
- [9] National Institute of Standards and Technology, "AI Risks and Trustworthiness," NIST AI Resource Center, AI RMF resources. [Online]. Available: <https://airc.nist.gov/airmf-resources/airmf/3-sec-characteristics/>. [Accessed: Aug. 19, 2026].
- [10] A. Vassilev, A. Oprea, A. Fordyce, H. Anderson, X. Davies, and M. Hamin, "Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations," NIST AI 100-2e2025, 2025, doi: 10.6028/NIST.AI.100-2e2025.
- [11] CrowdStrike, "What Are Indicators of Attack (IOAs)?," Jan. 16, 2025. [Online]. Available: <https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/indicators-of-attack-ioa/>. [Accessed: Aug. 19, 2026].
- [12] J. Saxe and K. Berlin, "Deep Neural Network Based Malware Detection Using Two Dimensional Binary Program Features," in Proc. 10th Int. Conf. Malicious and Unwanted Software (MALWARE), 2015, pp. 11–20, doi: 10.1109/MALWARE.2015.7413680.
- [13] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.