T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

APT KAYNAKLI ATAKLARA KARŞI DAYANIKLI ETMEN TABANLI VE ONTOLOJİK VERİ SIZINTISI ÖNLEME SİSTEMİ

DOKTORA TEZİ

Emrah KAYA

Bilgisayar Mühendisliği Anabilim Dalı

OCAK 2024

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

APT KAYNAKLI ATAKLARA KARŞI DAYANIKLI ETMEN TABANLI VE ONTOLOJİK VERİ SIZINTISI ÖNLEME SİSTEMİ

DOKTORA TEZİ

Emrah KAYA

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. İbrahim ÖZÇELİK

OCAK 2024

Emrah KAYA tarafından hazırlanan “APT Kaynaklı Ataklara Karşı Dayanıklı Etmen Tabanlı ve Ontolojik Veri Sızıntısı Önleme Sistemi” adlı tez çalışması 30.01.2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı : .....

Jüri Üyesi : .....

Jüri Üyesi : .....

Jüri Üyesi : .....

Jüri Üyesi : .....

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum "APT KAYNAKLI ATAKLARA KARŞI DAYANIKLI ETMEN TABANLI VE ONTOLOJİK VERİ SIZINTISI ÖNLEME SİSTEMİ" başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

13/12/2023

Emrah KAYA

Aileme

TEŞEKKÜR

Doktora eğitimim boyunca her konuda değerli bilgi ve deneyimlerinden yararlandığım, araştırmamın her aşamasında yardımlarını yanımda bulduğum, her daim teşvik ve yönlendirmeleriyle ufkumu açan değerli danışman hocam Prof. Dr. İbrahim ÖZÇELİK'e en derin teşekkürlerimi sunarım.

Özellikle ontoloji araştırmaları ve yayınlamaları çalışmalarımda yönlendirmeleri ve yardımları ile yolumu açan değerli hocam Doç. Dr. Özgü CAN'a teşekkür ederim.

Gerek yapıcı öneri ve yönlendirmeleri gerekse de çalışmamın tamamlanması konusundaki teşvik ve destekleri için, saygıdeğer jüri üyeleri Prof. Dr. Ahmet SAYAR ve Prof. Dr. Numan ÇELEBİ hocalarıma teşekkür ederim

Ayrıca bu çalışmanın maddi açıdan desteklenmesine olanak sağlayan Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna (TÜBİTAK) (Proje No: 117E100) teşekkür ederim.

Emrah KAYA

İÇİNDEKİLER

Sayfa

KISALTMALAR

API : Application Programming Interface
APT : Advanced Persistent Threat
ART : Atomic Red Team
COM : Component Object Model
CNN : Convolutional Neural Network
DAR : Data at Rest
DIM : Data in Motion
DIU : Data in Use
DLL : Dynamic-link library
DLP : Data Leakage Prevention
IAT : Import Address Table
IDF : Inverse Document Frequency
IDS : Intrusion Detection System
IPC : Inter-process Communication
LSA : Latent Semantic Analysis
MKY : Merkezi Kontrol Yazılımı
NLP : Natural Language Processing
OKV : Ontoloji Karar Verici
OWL : Web Ontology Language
PTA : Purple Team Automation
RDF : Resource Description Framework
SIEM : Security Information and Event Management
SPARQL : SPARQL Protocol and RDF Query Language
SGD : Stochastic Gradient Descent
SVD : Support Vector Machine
SWRL : Semantic Web Rule Language
TF : Term Frequency
TLS : Transport Layer Security
TTP : Tactic Technique Procedure

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

APT KAYNAKLI ATAKLARA KARŞI DAYANIKLI ETMEN TABANLI VE ONTOLOJİK VERİ SIZINTISI ÖNLEME SİSTEMİ

ÖZET

Gelişmiş Israrcı Tehditler (Advanced Persistent Threat-APT) tarafından gerçekleştirilen veri sızıntıları, kurumlar ve devletler için her geçen gün artan bir sorun olmaktadır. Yapılan çalışmalar APT kaynaklı risklere karşı belli aşama kaydetse de veri sızıntısı önleme (Data Leakage Prevention-DLP) konusunda yeterli başarımı gösterememektedirler. APT'ler sofistike yöntemlerinin yanında içerik-tabanlı veri sızdırma yöntemleri de kullanmakta ve literatürde bulunan çözümleri aşarak veri sızıntılarına neden olmaktadırlar. Bununla beraber, sistem çağrılarının zararlı tespitinde kullanımının etkinliği ve anlamsal analizin bir saldırıya ait ilişkileri açığa çıkarma başarımı ortaya konulmuştur. Bu tez kapsamında, etmen tabanlı bir mimari içerisinde ontoloji temelli bir karar verme mekanizması ile APT kaynaklı veri sızıntılarını önleme sistemi, APTONSYS sunulmaktadır. Sunulan yaklaşım, bir saldırının sistem çağrısı, proses, içerik gibi alt-seviye detayları, MITRE ATT&CK çerçevesi içerisinde yer alan teknik ve taktiklerle anlamsal olarak ilişkilendirebilme imkânı sunmaktadır. İçerik tabanlı saldırılara karşı dayanıklı yeni bir içerik sınıflandırma algoritması ve içerik tabanlı saldırıları tespit etme yöntemi de çözüm içerisinde yer almaktadır. Ayrıca sistem üzerinde tespit edilen teknik ve taktiklere dayalı bir APT risk seviyesi tespiti yaklaşımı sunulmaktadır. Sunulan çözümün etkinliği gerçek senaryolar ve açık kaynaklı APT simülasyon araçları ile yapılan testler ile ortaya konulmuştur.

DLP sistemleri hassas içerikli verilerin kurum dışındaki ya da içindeki uygun olmayan konumlara aktarılmasını önlemeye yarayan sistemlerdir. Çeşitli içerik takip, eşleştirme ve sınıflandırma yöntemleri kullanarak durağan, hareket halinde ya da kullanımda olan verilerin hassasiyet seviyesini belirlemekte ve sonuca göre yapılmak istenen işleme izin vermekte ya da engellemektedirler. Basit ve genellikle yanlışlıkla yapılan sızıntıları engellemede başarılı olurlarken, son yıllarda artan şekildeki sızıntı olaylarının da gösterdiği üzere, maksatlı ve hedef odaklı sızıntılarda başarımlarını koruyamamaktadırlar.

APT'ler, hedef odaklı, çoğu durumda arkasında bir komuta kontrol mekanizması olan, hedef sistemlerde mevcut olan araçları da kullanabilen sofistike yöntemlere sahip saldırganlardır. APT'ler, hedef sistemde uzun süreli kalarak, hak yükseltme yaparak, sistem içerisinde yayılarak ve sistemde var olan prosesleri kendi amaçları için kullanarak DLP tespit mekanizmalarını atlatabilmektedirler. DLP çözümleri genellikle durumsal farkındalık taşımayan ve statik politika kurallarına dayanan denetim mekanizmaları içermektedirler. Bu kurallar, ilgili prosesin "çalıştırılabilir dosya", "kaynak" ve "hedef" gibi bileşenlerini dikkate almaktadırlar. APT kaynaklı saldırılarda ise sistem programlarını değiştirme, başka proseslerin hafıza alanına erişme, içeriği değiştirerek farklı konuma kaydetme, farklı ağ protokolleri kullanma, Powershell komut dosyaları kullanma, hak yükseltme gibi yöntemlerle bu politika kuralları aşılabilmektedir.

DLP çözümlerinin APT'lere karşı yapısal zayıflıklarının yanında, kullandıkları içerik eşleştirme ve sınıflandırma yöntemlerinin maksatlı veri sızdırma saldırılarına karşı da zayıflıkları bulunmaktadır. İçeriğin değerlendirilmesinde kullandıkları istatistiksel analiz, parmak izi çıkarma ve düzenli ifadeler gibi yöntemler, içerik üzerinde yapılabilen basit ve küçük değişiklikler ile atlatılabilmektedir. İçerik üzerinde yapılabilecek bölüm, kelime ya da harf değiştirme saldırıları bu yöntemleri atlatabilmektedir. Ayrıca eş veya çok anlamlı kelime kullanma, kitap şifreleme, özet çıkarma gibi birçok farklı yöntemle içerik, orijinal halinden çıkarılarak DLP sistemleri etkisiz hale getirilebilmektedir. Veri sızdırma en temel amaçlarından birisi olan APT'ler de benzer içerik tabanlı saldırıları kullanarak DLP sistemlerini atlatmaktadırlar.

Bu tez çalışması ile DLP sistemlerinde APT kaynaklı veri sızıntılarının önlenmesi amacıyla bütüncül bir sistem önerisi geliştirilmiştir. Bu kapsamda, ontoloji bilgisine dayalı etmen tabanlı bir sistem sunulmuştur. Bu sistem ile, ölçeklenebilir ve çevrimiçi olarak APT riski tespit edilebilmekte ve veri kaçırma atakları ile ilişkisi kurulabilmektedir. Ayrıca sistem içerisinde geliştirilen özgün içerik sınıflandırma algoritması ile maksatlı veri kaçırma saldırılarına dayanıklı bir içerik sınıflandırma gerçekleştirilebilmektedir.

APT ve zararlı tespiti yönünde yapılan çalışmalar, APT'lerin hedefe ulaşana kadar izledikleri adımlar APT'ler arasında farklılık göstermekle beraber davranışsal yönün aynı kaldığı ve başarılı bir tespit için bir davranışsal analiz modeline ihtiyaç duyulduğunu göstermektedir. Bu kapsamda MITRE ATT&CK'ye dayalı olarak bir APT ile ilişkilendirilen Teknik ve Taktikleri tespit etmenin, savunma sisteminin geliştirilmesi, bakımı ve sağlamlığı açısından önemli bir yöntem olduğu görülmektedir. Bu sebeple, sunulan çözümde bu yönde bir çıkarım mekanizması oluşturulmuştur.

APT'lerin özellikleri sebebiyle tek başına çalıştırabilir dosya analizi yapılması APT kaynaklı saldırıların bütün aşamalarında tespit sağlamak için yeterli değildir. Bu statik analiz adımına ek olarak sistemdeki proseslerin sürekli takip edilerek dinamik bir analiz gerçekleştirilmesi gerekir. Dinamik analiz için olay kayıtlarından (event logs) faydalanmak gibi yöntemler var olmakla beraber, bunlar sadece önceden tanımlanmış, belirli olayların tespit edilmesini sağlayabilirler. Bunun yanında, bu olay kayıtları Tehdit Algılama Sistemleri tarafından sürekli takip edildiği ve onları tetikleyebilecek durumlar sabit olduğundan APT'ler bu tür olay günlüklerini oluşturacak işlemleri yapmamaya çalışırlar. Bu noktada, işletim sistemi kütüphanelerinin uygulama programlama arayüzü (Application Programming Interface-API) çağrılarını kullanmak önemli bir fayda sağlayacaktır. Bu çağrılar, literatürde "sistem çağrıları" olarak adlandırılır. Kullanılan araç ne olursa olsun, sistem üzerinde gerçekleştirilen her işlem, bir sistem çağrısı ile gerçekleşir. Bu nedenle, APT'ler sistem çağrılarını kullanmaktan kaçınamazlar. Bir sistem çağrısının türü, kaynağı, kullanıcısı ve işlem gibi unsurlar APT'lerin TTP'leri ile ilişkilendirilebilirse saldırının ilerleyişi belirlenebilir. Bu nedenle tez kapsamında sunulan sistemde sistem çağrılarına ait bileşenler ve bunların ilişkileri değerlendirilerek APT davranışları tespit edilmektedir.

Bir sistemi oluşturan bileşenler arasındaki ilişkilerin tespit edilmesinde ontoloji kullanımı ve anlamsal analiz literatürde önemli bir yer tutmaktadır. Çeşitli çalışmalarda saldırının kaynak, hedef, operasyon ve içerik gibi bileşenlerinin arasındaki anlamsal ilişkinin saldırı tespiti açısından önemi ortaya konulmuştur. Bu sebeple, tez kapsamında sistem çağrısına ait kategori, kaynak, hedef, kullanıcı hesabı

gibi bileşenlerin, proses güvenlik değerlendirmesinin, içerik ve içerik atak tespiti durumlarının bütüncül olarak tanımlandığı ve ilişkilendirildiği özgün bir ontoloji, APTON sunulmuştur. Ontolojideki sınıf ve ilişkiler kullanılarak özgün APT teknik, taktik ve risk tespit kuralları sunulmuştur. Bu yönüyle APT tespiti ve DLP yaklaşımlarının birleştirildiği yeni bir çözüm sunmaktadır.

APT'lerin birden fazla yürütülebilir dosya kullanabilmesi, sistemde var olan yazılım ve araçlardan faydalanmaları ve farklı bilgisayarlarda operasyonlarına devam edebilmeleri nedeniyle tekil proses ya da bilgisayar bazlı bir değerlendirme yerine, bütün ağdaki bilgisayarları içeren sistem genelinde bir analiz yapılması gerekmektedir. Bu sebeple, sunulan sistemde uç sistemlerde çalışan etmenler aracılığıyla sistem çağrılarının toplanması ve değerlendirilmeleri sağlanmaktadır. Etmenler arasında veri paylaşımı yapılması ile sonuçları birbirleriyle paylaşarak sistem genelinde bir tespit gerçekleştirmesi sağlanmıştır.

DLP çözümlerinin APT'lerin içerik tabanlı saldırılarına karşı zayıflıklarına karşı literatürde bütüncül bir çözüm bulunmamakla beraber gerek saldırılara karşı direnci arttıracak gerekse de sınıflandırma başarımı sağlayacak yöntemler bulunmaktadır. Bu yöntemler ışığında, tez kapsamında içerik tabanlı saldırılara karşı dayanıklı, çok aşamalı özgün bir içerik sınıflandırma yöntemi sunulmuştur. Ayrıca proseslerin eriştiği içeriklerin sürekli takibi ile bilinen içerik tabanlı atakların tespiti sağlanarak APT davranış tespitine katkı sunulmaktadır.

Son olarak, önerilen modeller ve sistem iki aşamalı olarak test edilerek başarımı gösterilmiştir. Öncelikle önerilen ontoloji modeli sistemden toplanan verilerle bağımsız olarak test edilmiştir. İçerik sınıflandırma algoritması da veri kümeleri ile bağımsız olarak test edilmiştir. Nihai olarak, kurulan ağ üzerinde APT ataklarının icra edilmesi ve bunun neticesinde sistem tarafından tespitlerin yapılması ile tez kapsamında sunulan etmen tabanlı APT veri sızıntısı tespit sisteminin başarımı bütüncül olarak sergilenmiştir.

AGENT BASED AND ONTOLOGICAL DATA LEAKAGE PREVENTION SYSTEM AGAINST ADVANCED PERSISTENT THREATS

SUMMARY

Data leakage caused by Advanced Persistent Threats (APTs) is a growing concern for organizations and governments. Although recent studies have made progress addressing APT risks, they still lack sufficient capabilities for data leakage prevention (DLP). APTs employ content-based methods for data exfiltration alongside their sophisticated methods, increasing the risk of data exfiltration and reducing the effectiveness of the current solutions in the literature. On the other hand, using system calls has proven to be effective in malware detection and semantic analysis has been used for inferring relations regarding an attack successfully. This thesis proposes APTONSYS, an agent-based system that utilizes ontology-driven reasoning mechanism to prevent the data leakage caused by APTs. The proposed approach establishes semantic connections between low-level details of an attack, such as system call, process, and content information with the APT Technique and Tactics defined within the MITRE's ATT&CK framework. A novel content classification method and mechanism to detect content-based attacks executed by APTs are also integrated into the solution. Further, an APT Risk definition is introduced by using Techniques and Tactics that are applied in the system. The effectiveness of the solution is presented using experimental tests using data from real-life scenarios and open-source APT simulation tools.

DLP systems are applications that prevent the transfer of sensitive content to inappropriate locations outside or inside an organization. By employing various content tracking, matching, and classification methods, they determine the sensitivity level of data at rest, in-motion, or in-use. Regarding the outcome, they allow or block actions. While they are successful in preventing simple and often accidental leaks, they have been unable to maintain their effectiveness in deliberate and targeted leaks, as evidenced by the increasing number of leakage incidents in recent years.

APTs are attackers that utilize sophisticated methods, often with a command-and-control mechanism. They employ targeted methods and can utilize tools existing within target systems. They bypass DLP detection mechanisms by remaining in the target system for extended periods, elevating privileges, spreading within the system, and repurposing existing processes for their objectives. DLP solutions typically utilize static policy-based control mechanisms that lack situational awareness. These rules take into account components such as the executable file, source, and destination of the process in question. However, in APT-based attacks, these policy rules can be circumvented by methods like altering system programs, accessing memory areas of other processes, saving content in different locations, using different network protocols, employing PowerShell scripts, and elevating privileges.

Besides the structural vulnerabilities of DLP solutions against APTs, their employed content matching and classification methods also possess weaknesses against purposeful data exfiltration attacks. Methods such as statistical analysis,

fingerprinting, and regular expressions employed in content classification can be bypassed through simple and minor alterations to the content. Attacks involving changes in sections, words, or letters on the content can evade these methods. Additionally, using synonymous or polysemous words, employing book ciphers, extracting summaries, and many other different methods can render the content ineffective against DLP systems by altering it from its original form. APTs, whose primary goal is data leakage bypass DLP systems by utilizing such content-based attacks.

In this thesis, a comprehensive system proposal has been developed to prevent APT-sourced data leaks in DLP systems. Within this scope, an agent-based system relying on ontology knowledge has been presented. This system enables scalable and online detection of APT risks and establishes a connection with data leakage attacks. Additionally, the system incorporates a novel content classification algorithm that facilitates resilient content classification against purposeful data leakage attacks.

In the context of research conducted towards APT and malware detection shows that while the steps taken by APTs may vary until they reach their target, their behavioral aspect remains consistent. This signifies the need for a behavioral analysis model for successful detection. Accordingly, it is observed that associating Techniques and Tactics with an APT based on MITRE ATT&CK is a crucial method for the development, maintenance, and robustness of the defense system. Therefore, the proposed solution includes such an inference mechanism.

Due to the characteristics of APTs, conducting executable file analysis alone is insufficient to detect APT-originated attacks at all stages. A dynamic analysis is also needed on top of this static analysis by continuous monitoring of system processes. While methods like leveraging event logs exist for dynamic analysis, they can only identify predefined, specific events. Moreover, as these event logs are constantly monitored by Intrusion Detection Systems and triggered by certain conditions, APTs avoid performing actions that generate such event logs. In this context, utilizing Application Programming Interface (API) calls of operating system libraries becomes crucial. These calls are commonly known as “system calls” in literature. Regardless of the tool used, every operation executed on the system is conducted through a system call. Hence, APTs cannot evade using system calls. If the type, source, user, and operation of a system call can be associated with APTs' Tactics, Techniques, and Procedures (TTPs), the progression of an attack can be determined. Therefore, within the framework of this thesis, APT behaviors are identified by evaluating components related to system calls and their relationships.

The literature shows that the use of ontology and semantic analysis holds significant importance in identifying relationships among components constituting a system. Numerous studies have highlighted the significance of semantic relationships between components such as the source, target, operation, and content concerning attack detection. Therefore, within the scope of this thesis, a novel ontology, APTON has been presented. It allows to define and correlate system call properties, such as category, source, target and user account with process security assessment, content classification, and content attack detection results. Novel APT technique, tactic, and risk detection rules are presented using the classes and relations in the ontology. This presents a novel solution that merges APT detection with DLP approaches.

APTs can utilize multiple executable files, leverage existing software and tools within the system, and persist operations across different computers. Therefore, it is

imperative to conduct an analysis across the entire system, encompassing all computers on the network. Hence, in the presented system, the collection and evaluation of system calls are facilitated through agents operating on endpoint systems. These agents can share results with each other, leading to a comprehensive detection across the system.

While there is no comprehensive solution in the literature addressing the weaknesses of DLP solutions against content-based attacks used by APTs, there exist methods that can enhance resistance against such attacks and improve classification accuracy. By using these methods, a more resilient, multi-stage content classification method against content-based attacks is presented in this thesis. Additionally, by continuously monitoring the content accessed by processes, the detection of known content-based attacks is performed, aiding in APT behavior detection.

Finally, the proposed models and system were tested in two stages to demonstrate their performance. Firstly, the proposed ontology model was tested independently with data collected from the system. The content classification algorithm was also independently tested with data sets. The comprehensive performance of the agent-based APT data leakage detection system was demonstrated by executing APT attacks on the established network and subsequently detecting them by the system.

1. GİRİŞ

1.1. Tezin Kapsamı

Veri Sızıntısı Önleme (Data Leakage Prevention-DLP), gerek ticari gerekse de devlet kurumlarına ait maddi, stratejik, fikri ve manevi öneme sahip her türlü hassas bilgi ve verinin ilgili kurumun dışına ya da kurum içinde yanlış kişilere maksatlı veya maksatsız olarak kaçırılması, bu kaçırma işleminin tespit edilmesi ve engellenmesi ile ilgilenen bir çalışma alanıdır. Son dönemlerdeki DLP alanındaki çalışmalar, veri sızıntısı olaylarının sıklık ve büyüklüğünde artış olduğunu göstermektedir (InfoWatch Analytics Center, 2018). Bu olayların sayısındaki sürekli artış, bu alandaki ilgiyi artırmış ve araştırma çabalarının miktarını ve ölçeğini de dikkate değer bir şekilde artırmıştır (Alneyadi ve ark., 2016; Shabtai ve ark., 2012). Bununla birlikte, artan çalışma sayısına rağmen, veri sızıntısı olaylarının artması, mevcut çözüm yöntemlerinin etkinliği konusunda ciddi soru işaretlerine yol açmıştır.

Veri güvenliği bağlamında, verinin üç farklı durumdan birinde olduğu kabul edilir: (Sealpath, 2020; Shabtai ve ark., 2012):

- Durağan Veri (Data At Rest - DAR): Kullanıcın bilgisayarında, bir taşınabilir diskte ya da bulut gibi bir çözümün içerisinde kalıcı olarak saklanmış dosyaları ifade eder.
- Hareket Halindeki Veri (Data In Motion- DIM): Ağ üzerinde akan ya da akmak üzere olan verileri ifade eder. Genel olarak http, ftp, smtp gibi trafikleri belirtmekle beraber ağ üzerindeki her türlü haberleşme içerisindeki veriler bu sınıfa girer.
- Kullanımdaki Veri (Data In Use - DIU): Kullanıcının ya da işlemlerin erişiminde olan, kullanıcı bilgisayarında herhangi bir işlemin hafıza alanında yer alan veriler olarak tanımlanabilir. Bu yönüyle, DAR ve DIM'e göre daha geniş ve muğlak bir veri kümesini ifade eder. Bu ifadenin muğlaklığı, kullanıcı ekranında açık bir metin içerisinde yer alan verilerin, internet tarayıcısında yazmakta olduğu ya da okuduğu metnin, panoya kopyalanmış verilerin, bir yazıcı ya da taşınabilir hafızaya yazılmak üzere olan verilerin hepsinin bu sınıfa girmesinden kaynaklanır.

Esasında birçok veri türü öncelikle DIU durumunda olup daha sonra DIM ya da DAR durumuna geçmektedir.

DAR ve DIM türündeki verileri korumak için oluşturulmuş genel yöntemler bulunmaktadır ve piyasada bulunan çoğu ticari DLP ürünü, DAR ve DIM koruma çözümleri sunmaktadır. DAR genellikle şifreleme ve erişim kontrol mekanizmaları aracılığıyla korunurken; DIM, uç noktalar arasındaki güvenli bağlantılar, örneğin Taşıma Katmanı Güvenliği (Transport Layer Security-TLS) ile koruma altına alınabilmektedir. Ancak, DIU türündeki verilerin güvenliği söz konusu olduğunda genel bir çözüm eksikliği bulunmaktadır. Bir proses verilere eriştiğinde, verinin proses belleğindeki güvenliği, işletim sistemi ve uygulama düzeyindeki önlemlere dayanmaktadır. Bunu dikkate alan bazı DLP çözümleri, yerel dosyalara erişimi kontrol etmek, kaydetme/gönderme işleminin hedefini, kullanıcısını veya zamanını politikalara dayalı olarak sınırlamak için uygulama eklentileri gibi uç nokta çözümlerini içermektedir (Forcepoint, 2022; Symantec, 2022). Bu çözümler ayrıca olay raporlama işlevini artırmak ve sistem üzerindeki kontrolü iyileştirmek için uç noktalarda etmenlere sahip olabilmektedirler. Her durumda, verilerin sızdırılmadan önce ilgili prosesin belleğine yüklenmesi gerekmektedir. Tüm olası kullanım senaryolarında veri, DAR veya DIM durumuna geçmeden önce DIU durumunda olacaktır. Bu nedenle, DIU güvenliği için uç noktalarda prosesleri takip edebilecek önlemler gerekmektedir. Ayrıca DLP sisteminin etkin olabilmesi için, bu uç nokta çözümlerinin kullanıcı, içerik, işlem, zaman ve hedef gibi DLP politikasının çeşitli unsurlarını toplayabilen ve kontrol edebilen yetenekte olması gerekmektedir.

Bu sebeplerle, DLP sistemleri uç nokta çözümlerine yönelmiş olsalar da çoğunlukla insan faktörlerinden kaynaklanan istenmeyen hatalara ilişkin veri sızıntılarını dikkate almaktadırlar. Halbuki organizasyonların hem içinden hem de dışından gerçekleştirilen saldırılardan kaynaklanan veri sızıntıları sürekli bir artış içerisindedir (Verizon, 2023). Özellikle Gelişmiş Israrcı Tehditlerle (Advanced Persistent Threat-APT) beraber, hedef sistemlerde halihazırda mevcut olan araçlar kullanılarak DIU durumundaki verilerin manipüle edildiği sofistike yöntemlerin uygulanması ve böylece mevcut DLP tespit mekanizmalarının atlatılması problemi doğmuştur. Mevcut DLP çözümlerinin APT'lere karşı başlıca sorunu, genellikle statik politika kurallarına dayanmaları ve DIU'yu kapsamlı bir şekilde izlememeleridir. Bu amaçla kullanılan ve ilgili prosesin “çalıştırılabilir dosya”, “kaynak” ve “hedef” bileşenlerini kontrol eden

bir DLP politika kuralı örnek alınırsa, bir APT açısından bu kuralın aşılmasının birden çok yolu vardır: APT aktörü, böyle bir durumda proses enjeksiyonu (MITRE, 2019b) kullanarak "çalıştırılabilir dosya" bileşenini, içerik tabanlı bir saldırı gerçekleştirip (MITRE, 2019c) sonrasında dosyanın içeriğini farklı bir konuma kaydederek (MITRE, 2019d) "kaynak" bileşenini ve http(s) dışında bir protokol kullanıp (MITRE, 2019e) içeriği uzak bir konuma taşımak için bir PowerShell komut dosyasını kullanmak yoluyla (MITRE, 2019f) "hedef" bileşenini devre dışı bırakabilir. Bu senaryoda, politika kuralının tüm kaynak dosya, hedef ve yürütülebilir unsurları atlatıldığı görülmektedir. (Digital Guardian, 2019) raporunda da veri sızıntılarının %61 oranında bu tür kötücül yazılımlar ile gerçekleştiği dikkate alındığında DLP sistemlerinin kötücül yazılımlara karşı zayıflığı ortaya konulmaktadır.

Mevcut DLP çözümlerinin APT'lere karşı bir diğer zayıflığı ise APT'lerin hedef odaklı olması, sistemler üzerinde bir aksiyon gerçekleştirmeden yayılarak sistem üzerinde uzun süre fark edilmeden kalabilmesidir. APT'ler hedef sistemde kalıcılığı sağladıktan ve yetki artırma (privilege escalation) işleminde başarılı olduktan sonra sistemde fark edilmeden yayılma yeteneğine sahip olurlar. Sistemde uzun bir süre varlığını sürdürerek önemli verileri komut ve kontrol kanalı üzerinden sızdırırlar. Mandiant kurumunun hazırladığı rapora göre, APT1 hedef sistemlerde ortalama 365 gün olmak üzere en uzun 4 yıl 10 ay süreye kadar faaliyet göstermiştir (Mandiant Intelligence Center, 2013). Mevcut DLP çözümleri, sızdırma denetimi yaparken hem APT'lerin bu davranış örüntüsünü dikkate almamaktadırlar, hem de "durumsal farkındalık" (situational awareness) özelliği taşımamaktadırlar. Geçmiş işlemler, DLP'nin davranışını ya da tespit ettiği sızdırma riski seviyesini değiştirmemektedir. DLP sistemi, sadece veri sızdırma işlemlerini dikkate alarak APT'nin o aşamaya gelene kadarki faaliyetlerini bir risk durumu olarak tespit edememektedir. DLP üzerinde tanımlanan proses, kaynak, kullanıcı, hedef gibi politika kuralları ile durumsuz (stateless) olarak tespit sağlamaya çalışmaktadırlar. Bu yaklaşım ise tespit için yeterli olmayıp sistemde yayılmış ve yerleşmiş olan APT'nin tanımlanan kurallar dışında bir yol ile kaçırma işlemini gerçekleştirme ihtimalini arttırmaktadır.

DLP sistemlerinin bahsedilen yapısal zayıflıklarının yanında, içerik sınıflandırma algoritmaları açısından da belli problemleri bulunmaktadır. DLP sistemlerinde içerik veya bağlam (context) dikkate alınabilmektedir. Bağlamın değerlendirilmesi açısından, dosyanın sahip olduğu oluşturan kişi, dosya türü, dosya konumu, yaratılma

tarihi, özel etiketler gibi üst veri (metadata) bilgiler dikkate alırlar. İçeriğin değerlendirilmesinde ise istatistiksel analiz, parmak izi çıkarma (fingerprinting) ve düzenli ifadeler (regular expressions) gibi yöntemler içerik eşleştirme ve sınıflandırma algoritmalarında kullanılarak içeriğin korunması sağlanmaktadır (Alneyadi ve ark., 2016).

Bağlama dayalı yöntemlerde, dosyada ya da üst verilerinde kasıtlı ya da kasıtsız olarak yapılan basit değişiklikler ve tanımlamalar, dosyanın gizli olarak sınıflandırmasını engelleyebilir. Örneğin dosyayı yaratan kişinin ve dosya konumunun dikkate alındığı bir DLP kuralında, dosya içeriği farklı bir kullanıcı hesabı ile farklı bir konuma taşındığında DLP kuralı atlatılabilmektedir. Bağlam bilgisinin tek başına yeterli olmaması sebebiyle DLP sistemleri sınıflandırma yaparken içeriği de dikkate almaktadırlar. Ancak içerik sınıflandırma yöntemlerinde özellikle APT kaynaklı içerik saldırılarına karşı zayıflıklar bulunmaktadır. İçerik üzerinde yapılabilen basit ve küçük değişiklikler DLP sistemlerini atlamada etkili olabilmektedir. DLP sistemlerinin içerik sınıflandırmada kullandığı parmak izi çıkarma yöntemi, doküman içerisindeki özel kelime ya da bölüm desenlerini ayırt ederek doküman için bir parmak izi çıkartmaktadır. İçerik üzerinde yapılabilecek bölüm, kelime ya da harf değiştirme saldırıları bu yöntemi atlatabilmektedir. Benzer şekilde, düzenli ifadelere dayalı yapılan sınıflandırma yöntemleri de yer değiştirme ve yerine koyma (substitution) saldırılarında tespit sağlayamamaktadırlar. İçerik üzerinde bahsedilen bu saldırılardan başka, eş veya çok anlamlı kelime kullanma, kitap şifreleme (book cipher), özet çıkarma gibi bir çok farklı yöntemle içerik, orijinal halinden çıkarılarak DLP sistemleri atlatılabilmektedir (Canbay ve ark., 2017; Mustafa, 2013).

Veri sızdırma en temel amaçlarından birisi olan APT'ler de içerik tabanlı saldırıları kullanarak DLP sistemlerini atlatmaktadırlar. Özellikle hedef sistemde tespit edilmeme çabalarının sonucu olarak şifreleme gibi işletim sistemi kütüphanelerine bağlı yöntemler yerine, uygulama hafıza alanı içerisinde gerçekleştirilebilecek yer değiştirme, yeri koyma gibi yapısal dönüşüm atakları ve eş/çok kelime, kitap şifreleme gibi karartma saldırıları tercih etmektedirler. Nitekim, yapılan analiz sonucu APT10'un Vigenère saldırısı ile verileri kaçırdığı tespit edilmiştir (Suguru Ishimaru, 2022).

1.2. Tezin Amacı ve Önerilen Çözüm Yöntemi

Bu tez çalışması ile DLP sistemlerinde APT kaynaklı veri sızıntılarının önlenmesi amacıyla bütüncül bir sistem önerisi geliştirilmiştir. Bu kapsamda, ontoloji bilgisine dayalı etmen tabanlı bir sistem sunulmuştur. Bu sistem ile, ölçeklenebilir ve çevrimiçi olarak APT riski tespit edilebilmekte ve içerik kaçırma atakları ile ilişkisi kurulabilmektedir. Ayrıca sistem içerisinde geliştirilen özgün içerik sınıflandırma algoritması ile maksatlı veri kaçırma saldırılarına dayanıklı bir içerik sınıflandırma gerçekleştirilebilmektedir. Bu bölümde sunulan çözüm genel hatları ile anlatılmıştır.

DLP sistemlerinin bahsedilen problemlerine yönelik literatürdeki APT ve zararlı tespiti yönünde yapılan çalışmalardan faydalanmak mümkündür. APT'lerin hedefe ulaşana kadar izledikleri adımlar APT'ler arasında farklılık gösterse de birçok çalışma davranışsal yönün aynı kaldığını göstermekte olup başarılı bir tespit için davranışsal bir analiz modeline ihtiyaç duyulmaktadır. Bu bağlamda, saldırı adımlarını farklı aşamalara ayırmak yaygın bir yaklaşımdır. (Li ve ark., 2016) ve (Atapour ve ark., 2018) çalışmalarında APT saldırı yaşam döngüsü farklı sayıda aşamalarda modellenmiştir. Daha detaylı ve kapsamlı bir liste, MITRE ATT&CK çerçeve yapısı (MITRE, 2020a) tarafından sunulmuştur. Bu yapıda, APT'lerin davranışları bir matris yapısı içerisinde, taktik, teknik ve prosedürlerin (TTPs) ilişkisini belirtecek şekilde sınıflandırılmıştır. Taktikler saldırının aşamalarını belirtmekte ve her bir taktik, saldırının ilgili adımını gerçekleştirmek için izlenen prosedürleri açıklayan çeşitli Teknikleri içerir. Bu hiyerarşik yaklaşım, bir saldırının unsurları ile TTP'ler arasında ve nihayetinde APT ile ilişkilendirme konusunda etkili bir yol sunar. Bir sistemde gerçekleştirilen saldırı prosedürleri tanımlanabildiğinde, bunlara karşılık gelen teknik ve taktiklerle ilişkilendirilebilir. Bu sayede, saldırı aktörü, niyeti ve saldırının mevcut durumu belirlenir. Durum doğru bir şekilde belirlendikten sonra, bu tehdit aktörüne karşı en etkili savunma yöntemleri önerilebilir ve daha fazla savunma sağlanabilir. Bu nedenle, MITRE ATT&CK'ye dayalı olarak bir APT ile ilişkilendirilen TTP'leri tespit etmek, savunma sisteminin geliştirilmesi, bakımı ve sağlamlığı açısından önemli bir yöntemdir. Ayrıca, (Bianco, 2014) çalışmasında yer alan "Pyramid of Pain" yapısında önerildiği gibi, TTP'lerin başarılı bir şekilde tespit edilmesi, saldırganları kullandıkları teknikleri tamamen değiştirmeye ve yeni bir araç geliştirmeye zorlar. Bu yaklaşım, savunma mekanizmasının etkinliğini artırır. Bu nedenle, tez kapsamında, MITRE ATT&CK ile ilişkilendirilen TTP'leri tespit eden bir yöntem önerilmektedir.

APT'lerin özellikle sisteme ilk bulaşma aşamasında zararlı yazılımlardan faydalanabilmektedirler. Zararlı yazılımlar henüz çalışmadan, çalıştırılabilir dosyaların statik olarak analiz edilmesi, bu aşamada önemli bir savunma başarımı sunmaktadır (Bai ve ark., 2014; Kozachok & Kozachok, 2018; Schultz ve ark., 2001). Ancak APT'lerin bahsi geçen özellikleri sebebiyle tek başına çalıştırabilir dosya analizi yapılması, APT kaynaklı saldırıların bütün aşamalarında tespit sağlamak için yeterli değildir. Buna ek olarak sistemdeki proseslerin sürekli takip edilerek dinamik bir analiz gerçekleştirilmesi gerekir (Singh ve ark., 2019). Dinamik analiz için çeşitli bilgi kaynakları bulunmaktadır. Modern işletim sistemlerinde, sistem genelindeki işlemler genellikle olay kayıtlarına (event logs) kaydedilir ve bu olay kayıtları zararlı tespitinde kullanılabilir. Ancak bu kayıtlar, APT'lerin gerçekleştirebileceği tüm faaliyetlerin tamamını yakalama konusunda yeterli değildir; çünkü sadece önceden tanımlanmış, belirli olayların tespit edilmesini sağlayabilirler. Bunun yanında, bu olay kayıtları Tehdit Algılama Sistemleri (Intrusion Detection Systems-IDS) tarafından sürekli takip edildiği ve onları tetikleyebilecek durumlar sabit olduğundan APT'ler bu tür olay günlüklerini oluşturacak işlemleri yapmamaya çalışırlar. Bu noktada, işletim sistemi kütüphanelerinin uygulama programlama arayüzü (Application Programming Interface-API) çağrılarını kullanmak önemli bir fayda sağlayacaktır. Bu çağrılar, literatürde "sistem çağrıları" olarak adlandırılır. Kullanılan araç ne olursa olsun, sistem üzerinde gerçekleştirilen her işlem, bir sistem çağrısı ile gerçekleşir. Bu nedenle, APT'ler sistem çağrılarını kullanmaktan kaçınamazlar. Bir sistem çağrısının türü, kaynağı, kullanıcısı ve işlem gibi unsurlar APT'lerin TTP'leri ile ilişkilendirilebilirse saldırının ilerleyişi belirlenebilir. Bu nedenle tez kapsamında sunulan sistemde sistem çağrılarına ait bileşenler ve bunların ilişkileri değerlendirilerek APT davranışları tespit edilmektedir.

Bir sistemi oluşturan bileşenler arasındaki ilişkilerin tespit edilmesinde ontoloji kullanımı ve anlamsal analiz (semantic analysis) literatürde önemli bir yer tutmaktadır. (Mustafa, 2013) çalışmasında bir saldırının kaynak, hedef, operasyon ve içerik bileşenlerinin arasındaki anlamsal ilişkinin saldırı tespiti açısından önemi ortaya konulmuştur. (Choi ve ark., 2015; Han ve ark., 2021; Kim ve ark., 2019; Lajevardi & Amini, 2019; Luh ve ark., 2016) çalışmalarında APT'lerin kullandıkları zararlıların davranışları ve tespitleri ontoloji kullanılarak gerçekleştirilmiştir. Bu sebeple, tez kapsamında sistem çağrısına ait kategori, kaynak, hedef, kullanıcı hesabı gibi

bileşenlerin, proses güvenlik değerlendirmesinin, içerik ve içerik atak tespiti durumlarının bütüncül olarak tanımlandığı ve ilişkilendirildiği özgün bir ontoloji sunulmuştur. Ontoloji üzerine eklenen özgün APT teknik, taktik ve risk tespit kuralları ile tespit sağlanmaktadır. Bu yönüyle APT tespiti ve DLP yaklaşımlarının birleştirildiği yeni bir çözüm sunmaktadır.

APT'lerin birden fazla yürütülebilir dosya kullanabilmesi, sistemde var olan yazılım ve araçlardan faydalanmaları ve farklı bilgisayarlarda operasyonlarına devam edebilmeleri nedeniyle tekil proses ya da bilgisayar bazlı bir değerlendirme yerine, bütün ağdaki bilgisayarları içeren sistem genelinde bir analiz yapılması gerekmektedir. Bununla beraber, sistem çağrılarının önemi ve sistem çağrılarının sadece hedef bilgisayarlarda toplanabilmesi sebebiyle her bilgisayarda bu işlemin gerçekleştirilmesi gerekir. Sistem çağrılarının doğası gereği, kısa sürede çok sayıda sistem çağrısı gerçekleşebilmektedir. Ayrıca, her sistem çağrısının bileşenlerinin, içerik gizlilik tespitinin ve içerik üzerindeki saldırı durumunun değerlendirilmesinin de yapılması gerekmektedir. Bu sebeplerle, değerlendirme işlemlerinin uç sistemlerde yapılması gerek içerik güvenliği gerek performans artışı gerekse de ağ iletişimini en aza indirmek için en faydalı seçenektir. Bu sebeple, tez kapsamında sunulan sistemde uç sistemlerde çalışan etmenler kullanılmaktadır. APT'lerin ağ üzerine yayılma eğiliminde olması ve saldırı aşamalarının farklı bilgisayarlarda gerçekleşebilmesi nedeniyle, etmenlerin elde ettikleri sonuçları birbirleriyle paylaşarak sistem genelinde bir tespit gerçekleştirmesi önerilmiştir.

Sunulan çözümün DLP özelliklerinin sağlanması, APT'lerin içerik tabanlı saldırılarına karşı dayanaklı bir içerik sınıflandırma yöntemi ile mümkün olabilecektir. Bu probleme karşı literatürde bütüncül bir çözüm bulunmamakla beraber gerek saldırılara karşı direnci arttıracak gerekse de sınıflandırma başarımı sağlayacak yöntemler bulunmaktadır. Kelime ekleme, kelime çıkarma gibi saldırılara karşı kelime temelli n-gram ve k-skip-n-gram yöntemlerinin başarılı olduğunu çeşitli çalışmalarla gösterilmiştir (Alneyadi ve ark., 2016). Kelimeler arasındaki boşluklara ve kelime içerisindeki harflere ilişkin yapılan saldırılarda ise sözlüksel karşılaştırma yapılarak ve harf temelli n-gramlar kullanılarak kelimelerin düzeltilmesi mümkündür (Martins & Silva, 2004; Priya ve ark., 2017). Eş anlamlı ya da çok anlamlı kelimeler kullanma, karartma (obfuscation) ve özet çıkarma türündeki saldırılarda ise Gizli Anlam Analizi (Latent Semantic Analysis - LSA) yöntemi kullanılması ile dokümana ait örtük

bilginin açığa çıkarılması sağlanarak sınıflandırma başarımı arttırılmaktadır. Bu yöntemler ışığında, tez kapsamında içerik tabanlı saldırılara karşı dayanıklı, çok aşamalı özgün bir içerik sınıflandırma yöntemi sunulmuştur. Ayrıca proseslerin eriştiği içeriklerin sürekli takibi ile bilinen içerik tabanlı atakların tespiti sağlanarak APT davranış tespitine katkı sunulmaktadır.

1.3. Tezin Katkısı

Bahsedilen problemlere ilişkin tez kapsamında sunulan çözüm, prosesin ilgili tüm unsurlarını sürekli olarak izleyen ve bileşenlerin ilişkilerini dinamik olarak değerlendiren çok katmanlı bir tespit mekanizmasıdır. İlgili işleme ait sistem çağrısı, proses, kaynak, hedef, kullanıcı hesabı, içerik, içerik üzerindeki değişiklikler ve önceki tespit sonuçlarının bir arada değerlendirilebilmesine imkân sunan bir sistem ve sistemin merkezinde yer alan ontoloji tabanlı karar verme mekanizması sunulmuştur. Bu sistemin temelini, proseslerin işletim sisteminde gerçekleştirmek istedikleri herhangi bir işlemde kullanmak zorunda oldukları sistem çağrılarının sürekli takip ve analizi oluşturmaktadır. Sistem çağrılarının bileşenleri ve proses bilgileri MITRE ATT&CK içerisinde tanımlanan APT teknik ve taktikleri temel alınarak değerlendirilmektedir. Bileşenler arasındaki ilişki, sunulan APTON ontolojisi ve ontoloji içerisindeki tespit kuralları kullanılarak çıkarım yapılmaktadır. Sistem çağrısının bileşenlerinden olan içerik ve içerik üzerindeki değişiklikler, sunulan yeni yöntem ile sınıflandırılmakta ve APT'lerin kullandığı içerik temelli saldırılara karşı daha dayanıklı bir değerlendirme yapılabilmektedir. Sunulan etmen tabanlı sistem ile sadece tek bir noktada değil, sistem genelinde bütüncül bir değerlendirme yapılabilmektedir. APT'lerin teknik ve taktikleri dikkate alınarak geliştirilen APT Risk kavramı ile sistem üzerindeki APT kaynaklı veri kaçırma risk seviyesi belirlenebilmektedir.

Bu yönleriyle, tezin sunduğu katkılar şu şekilde sıralanabilir:

- Etmen tabanlı bir sistem önerisi ile sistem çağrılarından APT teknik, taktik ve risk değerlendirmesinin bütün ağ genelinde çevrimiçi yapılabilmesine imkân sunan bir sistem sunulmaktadır.
- Veri sızıntısını önlemek için APT davranışlarını tespit etmek amacıyla bir anlamsal analiz modeli sunulmuştur. Bu amaçla, MITRE ATT&CK çerçevesi içerisinde tanımlanmış APT teknik ve taktiklerini sistem çağrısı bileşenleri, proses,

içerik sınıflandırması ve içerik atak durumu ile ilişkilendiren özgün APTON ontolojisi ve tespit kuralları oluşturulmuştur.

- APT'lerin veri kaçırma amaçlı içerik tabanlı saldırılarına karşı dayanıklı özgün bir içerik sınıflandırma yöntemi sunulmuştur.
- Bilinen içerik tabanlı saldırıların tespiti sağlanarak bu bilginin APT davranış ve risk tespiti için kullanımı sağlanmıştır.
- MITRE ATT&CK içerisinde tanımlanan teknik ve taktik bilgilerine dayalı yeni bir APT risk durumu tespiti yaklaşımı sunulmuştur.

1.4. Tez Sunum Planı

Sunulan tez 7 bölümden oluşmaktadır. Çözüm sunulan problemin tanımı, tez çalışmasının amacı ve tez organizasyonunun yer aldığı bu 1. bölümden sonraki plan şu şekildedir:

Bölüm 2 - Temel bilgiler ve literatür çalışmaları: Bu bölümde, tez kapsamında ele alınan kavramlar detaylandırılmakta, güncel literatürün probleme ilişkin sunduğu çözümler ele alınmaktadır.

Bölüm 3 - APTON: APT kaynaklı veri sızıntılarının tespitini sağlamak için yeni bir ontoloji: Bu bölümde sistemin esas karar verme mekanizmasını oluşturan APTON, APT tespit Ontolojisi sunulmaktadır. Ontolojinin geliştirilmesine temel oluşturan ihtiyaçlar açıklanarak ontoloji içerisindeki sınıflar ve ilişkiler detaylandırılmaktadır. APT teknik ve risk tespit kurallarının da açıklandığı bu bölüm ile sistem çağrılarına ait bilgilerden APT tespiti çıkarımının nasıl yapıldığı ortaya konulmaktadır.

Bölüm 4 - İçerik sınıflandırma algoritması geliştirilmesi: Bu bölümde mevcut DLP sistemlerinin içerik eşleştirme ve sınıflandırma başarımını düşüren içerik tabanlı ataklara karşı dayanıklı yeni bir içerik sınıflandırma algoritması sunulmaktadır. Literatürde farklı atak türlerine karşı başarım sağlayan farklı metotların bir araya getirilmesi ve uygun sıra ile uygulanması ile elde edilen bu yaklaşım detaylı olarak anlatılmaktadır. Ayrıca metnin önceki ve sonraki hallerini karşılaştırarak içerik tabanlı atakların tespitini sağlayan İçerik Atak Tespiti yöntemi de bu bölümde sunulmaktadır.

Bölüm 5 – APTONSYS: Etmen tabanlı APT veri sızıntısı tespit sistemi: Sistemin bileşenlerinin ve etmen mimarisinin anlatıldığı bu bölümde, tez kapsamında sunulan sistem bütüncül olarak ele alınmaktadır. Teoriden pratiğe geçişte gerekli olan modüller

ve çalışma şekilleri anlatılmakta, sistem üzerinde çevrimiçi olarak sistem çağrılarına ait bilgilerin nasıl toplanarak ontoloji karar verme aşamasına aktarıldığı detaylandırılmaktadır. Ayrıca farklı bilgisayarlarda çalışan etmenlerin bütüncül karar verme işlemini nasıl gerçekledikleri de bu bölümde yer almaktadır.

Bölüm 6 – Deneysel çalışmalar ve bulgular: Sunulan çözümlerin öncelikle bağımsız olarak sonrasında ise bütünleşik olarak testlerinin nasıl yapıldığı ve elde edilen sonuçların sunularak değerlendirildiği bu bölüm ile APTON ontolojisinin, içerik sınıflandırma algoritmasının ve nihayetinde APTONSYS sisteminin başarımı ortaya konulmaktadır.

Bölüm 7 – Sonuç ve tartışma: Bu bölümde tez çalışması ile elde edilen sonuçlar değerlendirilerek çalışmaların bilime ve endüstriye olası katkıları sunulmuştur. Tez çalışması sonrasında yapılabilecek çalışmalar değerlendirilerek çalışmanın ulaşması muhtemel ufuklar ele alınmaktadır.

2. TEMEL BİLGİLER VE LİTERATÜR ÇALIŞMALARI

2.1. Veri Sızıntısı Önleme (Data Leakage Prevention-DLP) Sistemleri

2.1.1. Veri sızıntısı önleme

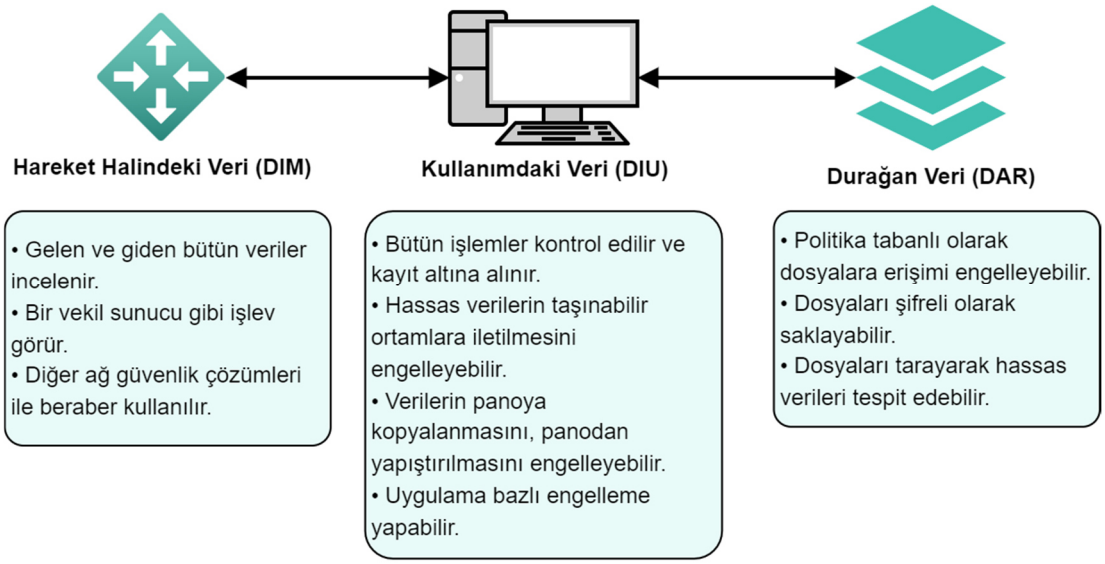
Veri sızıntısı (data leakage-data breach), özel ya da hassas verilerin yeterli izni olmayan bir varlığın erişimine sunulmasıdır. Bu tanımdaki varlık, bir kişi, kurum ya da yazılım olabilir. Sızıntı, kasıtsız ya da kasıtlı olarak gerçekleşebilir. Veri tipleri finansal bilgiler, fikri mülkiyet içeren bilgiler, çalışan, müşteri ya da hasta bilgileri, kredi kartı verileri gibi çok çeşitli olabilmektedir. Veri sızıntısının kurumlar ya da devletler üzerinde doğrudan ve dolaylı olarak ciddi etkisi olmaktadır. Müşteri ya da kişilerin gizliliğinin ihlal edilmesi sonucunda oluşan hukuki ve parasal cezalar ve tazminatlar; bu bilgilerden faydalanacak üçüncü kişilerin ve rakiplerin açacağı zararlar; şirketin ticari itibarının ve satışlarının olumsuz etkilenmesi; firmanın hisse senedi kayıpları gibi birçok boyutta etkisi olmaktadır. Devletler açısından da benzer zararlar oluşmakta, ülke itibarı ve güvenliği riske girmektedir.

(Ponemon Institute LLC & IBM, 2023) raporunda yayınlanan bilgilere göre, Dünya genelinde gerçekleşen veri sızıntısı vakalarında bir sızıntı olayının ortalama maliyeti 4,45 milyon ABD doları seviyesine çıkarken, sızdırılan her bir verinin (kaydın) açtığı zarar 165 ABD doları olmuştur. Gerek bu rapor, gerekse de önceki yıllarda yayınlanmış olan raporlar (InfoWatch Analytics Center, 2018; Verizon, 2023), veri sızıntısı olaylarının boyut, sayı ve zarar açısından sürekli bir artış içerisinde olduğunu göstermektedir.

Veri Sızıntısı Önleme (Data Leakage Prevention-DLP), bahsedilen bu veri sızıntılarının tespit edilmesi ve engellenmesi ile ilgilenen bir çalışma alanıdır. DLP sistemi üzerinde tanımlanan politikalar aracılığıyla hassas verilerin uygun olmayan yerlerde saklanmamasını, bu yerlere gönderilmemesini ve bu hassas verilere yetkisiz erişilmemesini otomatik olarak sağlar. Bunu yaparken kullanıcıların ihtiyaç duydukları araçları ve hizmetleri kullanmalarına izin verir. Bu amaçla farklı hassasiyet seviyeleri ve kullanıcı erişim kontrolünü yönetmek üzere ayarlanabilir.

Bir DLP sistemini tanımlayan en önemli özellik, içerik konusunda bilgi sahibi olmasıdır (content-awareness). Giriş bölümünde de bahsedildiği gibi, veri güvenliği bağlamında, verinin Durağan Veri (Data At Rest - DAR), Hareket Halindeki Veri (Data In Motion- DIM), Kullanımdaki Veri (Data In Use - DIU) olmak üzere üç farklı durumdan birinde olduğu kabul edilir (Sealpath, 2020; Shabtai ve ark., 2012). Bir DLP sistemi bunların hepsine ya da birkaçına karşı önlemler içerebilir. Şekil 2.1'de bir DLP sisteminde verinin olması muhtemel durumları sergilenmiş ve bu durumlarda yapılabilecek DLP koruma işlemleri örneklendirilmiştir.

Bu alanda gerek akademik gerekse de ticari çalışmalar ve çözümler artarak sunulmaya devam etmektedir. Veri sızıntısı olayların sayısındaki sürekli artış, bu alandaki ilgiyi, araştırma çabalarının miktarını ve ölçeğini de dikkate değer bir şekilde artırmıştır (Alneyadi ve ark., 2016; Shabtai ve ark., 2012). Ancak, artan çalışma sayısına rağmen, veri sızıntısı olaylarının artması, mevcut çözüm yöntemlerinin etkinliği konusunda ciddi soru işaretlerine yol açmaktadır.



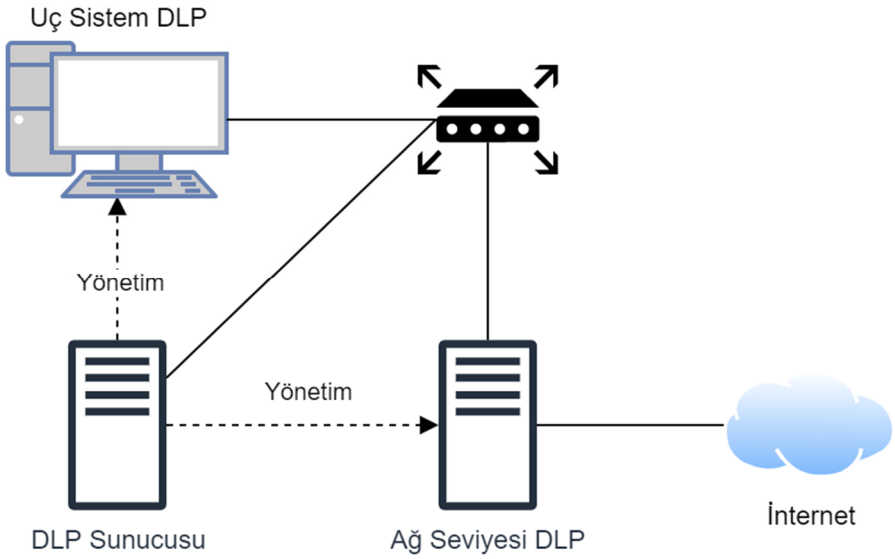
Şekil 2.1. DLP sisteminde verinin durumları.

2.1.2. DLP teknolojisi

Bu bölümde DLP sistemlerinin genel özellikleri ve yetenekleri detaylandırılmaktadır. DLP sistemleri bazı farklılıklar içermekle beraber genel olarak Şekil 2.2'deki ana bileşenlere sahip olmaktadırlar. Uç sistem DLP (Endpoint DLP) sistemdeki hedef bilgisayarlara doğrudan kurulan ve genelde bu bilgisayarlar üzerindeki DIU ve DAR türündeki verileri korumakla görevlidirler. Uç sistem DLP'ler ilgili bilgisayara gelen ve giden verileri inceleyip DIM türündeki verileri de kontrol edebilirler. Ağ seviyesi

DLP (Network DLP) ise yerel ya da genel ağ (LAN ve WAN) erişimlerinin arasına girerek ağ trafiği üzerinden DIM türündeki verileri inceleyerek DLP kontrolü yapmaktadır. Sistemde yer alan bir DLP sunucu ile bu DLP kontrolcülerinin davranışları, politikalar gibi ayarlar kontrol edilebilirken bu sistemlerden toplanan sonuçlar da DLP sunucusunda kaydedilebilir ve daha ileri değerlendirme ve aksiyon adımlarına aktarılabilir.

DLP sisteminin ağ ya da uç sistem üzerinde olmasından bağımsız olarak sahip olduğu özellikler ve teknolojiler vardı. İlerleyen bölümlerde bunlar detaylandırılmaktadır.



Şekil 2.2. Genel DLP sistemi bileşenleri.

2.1.2.1. DLP politikaları

DLP politikalarının tanımlanması, bir DLP sisteminin koruma adımlarının başında gelir. Hangi verinin nasıl korunacağını belirleyen bu kurallar sayesinde DLP sistemi otomatik olarak sonuç üretebilmektedir. Hedef kuruma göre hassas veri ve nasıl korunacağına ilişkin tanımlar değişebileceği için özelleştirilebilir olmaları önemlidir. Her durumda, bir DLP politikasının şu ana bileşenleri içermesi gerekir:

1. Teşhis

DLP'nin ilk aşaması hangi verilerin hassas olduğunun belirlenmesidir. Bunu sağlamak için içeriği analiz etmek, dosya üst-verilerinden faydalanmak, verinin hedef ya da kaynağını dikkate almak gerekebilir. Teşhis aşaması verinin hassasiyet seviyesini de belirler ve gizli, kuruma özel, genel gibi sınıflandırabilir. Bir DLP politikası, teşhis aşamasının nasıl yapılacağını, hangi verilerin politikanın geri kalanı için dikkate alınacağını içermelidir.

2. Yetkilendirme

DLP politikasının bir diğer özelliği ise hangi kullanıcıya hangi politikanın uygulanacağının belirlenebilmesidir. Bunun klasik erişim kontrol yöntemlerinden farkı, salt dosya konumu ya da içeriğine göre bir kısıtlama yapmak yerine bu bileşenlerle beraber yapılmak istenen işlem, hedef konum, zaman gibi politikanın bütün bileşenlerini dikkate alarak bir yetkilendirme yapılmasıdır. Bu sayede örneğin kısıtlı bir kullanıcının ilgili içeriğe erişmesine izin verilirken bu veriyi eposta ile göndermesi engellenebilmektedir.

3. Koruma/Engelleme yöntemi

DLP politikasının ihlal edildiğinin tespit edilmesi durumunda yapılacak işlemlerin belirlendiği kısımdır. DLP politikasının bu aşaması uygulamaya göre oldukça değişebilmekle beraber, genel olarak öncelikle ihlalin kullanıcıya ve/veya DLP yöneticisine raporlanmasını mutlaka içerir. Sonrasında işlemin engellenmesi ile verinin şifrelenmesi gibi ek aksiyon adımları içerebilir.

2.1.2.2. İçerik tespiti ve sınıflandırılması

DLP sisteminin öncelikle korunması gereken içerikleri tespiti gerekir. Bir kurumda hangi konumda hangi hassasiyet seviyesindeki verilerin bulunduğunun tespiti önemlidir. Bu aşamada genelde bilgisayarlar ve sunucular üzerinde yer alan dosyalar, veri tabanları gibi durağan durumda (DAR) olan verilerin taranması ve hassas verilerin belirlenmesi şeklinde gerçekleşmektedir. DLP sisteminin yeteneklerine bağlı olarak hareket halindeki (DIM) ve kullanımdaki veriler de (DIU) çevrimiçi taranarak hassas içerikler tespit edilebilmektedir.

İçerik tespiti ve sınıflandırılması konusunda farklı yöntemler yer almaktadır. Bunlar 2.5.1 bölümünde detaylandırılmıştır.

2.1.2.3. Verinin korunması

Uç sistem DLP sistemlerinde verileri barındıran dosyalara doğrudan erişim sağlanabildiği için daha kapsamlı çözümler sunulabilmektedir. Öncelikle farklı dosya türlerini desteklemek için çeşitli eklentiler yer almaktadır. Farklı türde ofis dosyaları (Word, Excel), PDF dosyaları, sıkıştırılmış dosyalar, html dosyaları, Java sınıf dosyaları gibi birçok farklı türde dosyanın içerisindeki esas metin gerek ticari (Micro Focus & OpenText, 2021) gerekse de açık kaynak kütüphaneler (Apache, 2023) ile

elde edilebilmektedir. Bu sayede içerik bazlı bir değerlendirme yapılmasını sağlayarak içeriğin farklı dosya türlerine aktarılması sonucu sistemin atlatılmasının önüne geçilmektedir.

Uç sistemlerde tespit öncesi ve sonrasında işletim sistemi seviyesinde çeşitli aksiyonlar sağlanabilmektedir. En açık olan aksiyon, yapılmak istenen işlemin engellenmesidir. Dosyanın uygun olmayan bir hedefe kopyalanması uç sistemdeki DLP yazılımı ile sağlanabilmektedir. Yine DLP'nin işletim sistemi seviyesinde elde edebildiği yetkiler ile kullanımda olan veriler (DIU) üzerinde de denetim yapılabilmektedir. Örneğin herhangi bir proses tarafından panoya kopyalanan veriler takip edilerek tespit edilen hassas verilerin panodan silinmesi sağlanabilmektedir (Symantec, 2022).

Uç sistem DLP'ler donanımsal olarak da engellemeler yapabilmektedir. USB flaş disklerin kullanımı tamamen ya da sadece izin verilen cihazlar seviyesinde kısıtlanabilmekte; hassas verilerin bu tür cihazlara kopyalanması engellenebilmektedir.

Eposta türündeki uygulamaların kontrolü de gerek uç sistem DLP'de gerekse ağ seviyesi DLP çözümünde gerçekleşmektedir. Uç sistemdeki eposta istemcisi üzerinde DLP kontrolleri ile eposta içeriği ya da eklentiler taranabilmektedir (Forcepoint, 2016). Ağ seviyesinde ise eposta sunucusu üzerinde benzer kontroller gerçekleştirilmektedir.

Uç sistem DLP'ler kullanıcıya aksiyon konusunda seçenekler sunabilmektedir. Bir ihlal tespit edildiğinde bunun otomatik engellenmesi, kayıt tutularak izin verilmesi, kullanıcıya ya da sistem yöneticisine bildirim gönderilmesi, kullanıcıdan bu işlem için gerekçe yazması istenerek izin verilmesi ya da ilgili dosyanın hedefe otomatik şifrelenerek gönderilmesi gibi seçenekler yer almaktadır. Bu işlemler verinin hassasiyet seviyesine ya da kullanıcının yetki seviyesine göre farklılaştırılabilmektedir (Forcepoint, 2022).

2.1.2.4. İşlem kaydının tutulması

DLP sistemleri sadece anlık ihlallerin tespiti için değil, geriye dönük vaka analizi için de işlem kayıtlarının tutulmasını sağlar. Bu kayıtlar ihlal durumuna ait yer, zaman, kişi ve ihlal şekli gibi bilgilerin yanında ihlale konu olan hassas verileri de içerebilir. Bu sebeple ayrıca şifrelenmiş bir veri tabanında tutulurlar. DLP sisteminin hatalı tespitleri

çok olursa bu kayıtların da işlevi azalacağı gibi, ilgili veri tabanında yük oluşturacaktır. Bu sebeple kayıt tutma işlemlerinin ve buna ait altyapının da ölçeklenebilir olması önemlidir.

2.1.3. DLP sistemlerindeki zorluklar

DLP sistemlerinin hem etkin bir teşhis ortaya koyması hem de sistemlerin kullanımına engel olmayacak şekilde çalışması gerekir. Ayrıca sistem yöneticisi müdahalesini en aza indirecek bir otomasyon sunmalıdır. Bu konuya ilişkin genel zorluklar bu bölümde değerlendirilmektedir.

2.1.3.1. Veri sızıntı kanalları

Veriye erişmek ve paylaşmak için bileşenler arasında çeşitli kanallara ihtiyaç vardır. Bileşenler ve kişiler arasında veri paylaşımı gerekiyorsa, bu kanalların açık kalması gerekir. Ancak, bu kanallar veri sızıntıları için de önemli bir yoldur. (InfoWatch Analytics Center, 2018) raporunda da sızıntıların çok büyük bir bölümünün ağ aracılığıyla gerçekleştiği görülmektedir. Ayrıca taşınabilir bellek, eposta, yazıcı gibi kanalların da önemli birer sızıntı yolu olduğu sunulmaktadır. Bir bilgisayardan taşınabilir bellek, CD, harici disk gibi ortamlara; internet ya da ağdaki bir bilgisayara, bir akıllı telefona, yazıcıya, bulut sistemlerine, eposta sunucularına sürekli ya da aralıklı olarak bağlantıların kurulması gerekmektedir. Bu kanalların bazıları kolayca yönetilebilirken, diğerleri tamamen güvenli hale getirilmek için önemli bir çaba gerektirebilir. Bu kanalların veri sızıntısını engelleyecek ama işlevini koruyacak şekilde kontrol edilmesi, hatalı tespitlerin en aza indirilerek sistemin kullanımında performans sorunu yaşatmayacak bir yöntem izlenmesi gerekmektedir.

2.1.3.2. İnsan kaynaklı sızıntılar

İnsan faktörü, DLP konusunda hem saldıran hem kurban tarafında etkili olan önemli bir bileşendir. Özellikle APT'ler ile beraber saldırı vektörleri, kendi başına önceden tanımlı bir algoritmayı yürüten yazılımlar yerine, insanların sürekli kontrol ve erişim sağladığı birer araç olmuştur. Bu da hem saldırı vektörünün öngörülebilirliğini azaltmış hem de atağın daha sofistike olmasına neden olmuştur.

Kurbanlar tarafında da insan faktörü etkili olmaktadır. (Verizon, 2023) raporunda özellikle APT kaynaklı veri sızıntılarında sisteme ilk bulaşma işlemi bir eposta ya da bir ofis dokümanı ile gerçekleşmiştir. Bu vektörler ise esas olarak insan faktörünü hedef alan, dikkatsiz kullanıcıların bu epostalardaki bağlantıları ya da dosyaları

açmaları ile etkili olmaktadırlar. Raporda fidye yazılımlarının %45'inin masaüstü paylaşımı, %35'inin ise eposta yolu ile sisteme bulaştığı tespit edilmiştir. Bu yöntemler ise doğrudan insan faktörünün hedef alındığı saldırıları göstermektedir.

Bu durumun sebebi, insan davranışını öngörmenin zor olması, birçok psikolojik ve sosyal faktörden etkilenebilir yapıda olmasıdır. Verinin gizlilik düzeyini tanımlamak, erişim haklarını belirlemek veya bir DLP'nin manuel ayarlandığı durumlarda tespit eşiğini ayarlamak gibi birçok işlemde insanların öznel değerlendirmeleri etkili olur. Ayrıca, çoğu kuruluşun sıkı düzenlemeler ve yönergeler içeren güvenlik politikalarına uyum genelde insanların bunları dikkate almasına bağlıdır.

Kişilerin sistemi kullanımını zorlaştıran DLP yaklaşımları da insanların veri sızıntılarına yol açmasına sebep vermektedir. Örneğin, USB bağlantı noktalarının belli yetkideki kullanıcılar için devre dışı bırakıldığı bir DLP sisteminde, eğer kişilerin bu USB bağlantılarını kullanmaları işlerinin daha kolay yapılmasını sağlayacaksa, erişim ayrıcalıkları kullanıcılar arasında kasıtlı olarak ya da sosyal mühendislik yoluyla paylaşılabilir (Orgill ve ark., 2004).

2.1.3.3. Uygun yetkilendirme

Gerek genel siber güvenlik önlemlerinde gerekse de mevcut DLP sistemlerinde kullanıcıların yapabildiği işlemler, farklı yetkilendirme seviyeleri ile yönetilmektedir. DLP'ler özelinde, örnek olarak ne tür verilerin hangi hedeflere iletilebileceği belli yetki seviyeleri belirlenebilmektedir. Böylece örneğin sadece yetkili kişiler hassas verileri eposta ile gönderebilmektedirler. Bu yetkilerin uygun şekilde ayarlanması açık bir gereklilik olmakla beraber, özellikle değişen çalışan ve iş durumuna göre bu yetkilendirmelerin aktif olarak güncellenmesi ihtiyacı esas dikkate alınması gereken kısımdır. Örneğin görevi değiştirilen veya işten çıkarılan çalışanların erişim hakları iptal edilmezse eski ayrıcalıklarını kullanarak verilere kötü niyetle erişebilirler. APT'lerin de bu pasif hesapları kullandıkları ve buna karşı bu tip hesapların periyodik kontrol edilmesi gerektiği (CISA, 2020) bilgilendirmesinde de belirtilmiştir.

2.1.3.4. Ölçeklenebilirlik

Özellikle kurumsal yapılarda DLP sisteminin ele alması gereken uç nokta ve veri miktarı büyük boyutlara ulaşabilmektedir. DIU ve DIM durumundaki verilerin dikkate alındığı çözümlerde sistemin kullanıcı sayısının yanında işlenen ve iletilen veri miktarı toplam hassas veri miktarının çok ötesinde olabilir. Bu sebeple iş akışını etkilemeden

verilerin işlenebilmesi gerekir. Uç sistem tabanlı DLP çözümlerinde veri analizinin en azından belli bir kısmının uç noktada yapılması gerekir. Aksi durumda bütün uç birimlerin veriyi analiz için tek bir noktaya iletmesi hem hata olasılığını arttıracak hem de performans açısından ölçeklenebilir olmayacaktır.

Ağ tabanlı DLP'lerde bazı işlevler güvenlik duvarları ya da Saldırı Tespit Sistemleri (Intrusion Detection System-IDS) tarafına aktarılabilir ve böylece veri üzerindeki mükerrer kontroller kaldırılabilir. (Mogull, 2010) çalışmasında, ağ tabanlı bir DLP sisteminin bir vekil sunucu ile entegre olarak denetim yapabileceğini anlatmıştır.

2.1.3.5. İçerik üzerindeki işlemler

Bir DLP sisteminin en temel özelliği işlediği içeriklerin hassasiyetini belirleyebilmektir. Bununla beraber, kullanıcıların içerikler üzerinde değişiklik yapmaları da en temel iş akışlarındandır. Ancak bu ihtiyaç, DLP sisteminin bu temel özelliğini gerçekleştirmesini güçleştirmektedir.

DLP sistemlerinde yer alan içerik eşleştirme ve sınıflandırma yöntemleri, bazı varsayımlara göre tasarlanmıştır. Örneğin kelime ya da örüntü eşleştirme yapan yöntemler hassas verilerin hep belirtilen kalıplarda yer alacağını varsayarlar. Halbuki kasıtlı ya da kasıtlı içerik değişiklikleri kelimelerde küçük farklar oluşturarak bu eşleştirme yöntemlerini atlatabilirler. Kelimelerdeki bazı harfleri sayılarla değiştirmek, boşlukları silmek gibi basit ataklar bu yöntemleri etkisiz hale getirebilir.

İçeriğin bir özetleme fonksiyonu (hash function) ile özetinin çıkarılarak sınıflandırma yapılan yöntemlerin de benzer problemleri vardır. İçerikteki küçük değişiklikler özet sonucunu değiştirmektedir. Gerek uç sistem tabanlı DLP'lerde gerekse ağ tabanlı DLP'lerde bu problemi aşmak için özeti çıkarılacak içeriklerin daha küçük parçalarda ele alınması tercih edilmektedir. (Shu & Yao, 2013) çalışmasında sunulan ağ tabanlı DLP'de parçalı imzalama yapılması önerilmiştir. Özet fonksiyonlarında da bu tür değişikliklerin etkilerini dikkate alarak daha etkin özet çıkartmayı amaçlayan Rabin–Karp algoritması (Cormen, Thomas H. ve ark., 2022) gibi yöntemlerden faydalanılabilir.

İçerik üzerinde ayrıca şifreleme ve steganografi türünden değişiklikler de yapılabilmektedir. Gizli bir doküman şifrelenerek sonrasında eposta ile hedefe doğrudan eposta içeriği olarak ya da bir eklenti olarak gönderilebilir. DLP sistemi bu şifreli içeriği tanımlayamayacağı için sınıflandırmada başarısız olacaktır.

Steganografik ataklarda ise hassas veriler başka bir metnin ya da resmin içerisinde gömülerek DLP atlatılabilmektedir.

Tez önerisinde de yer aldığı gibi, uç sistem DLP'de içerik henüz şifreli ya da steganografi ile saklı değilken yapılabilecek kontroller ile bu problem hafifletilebilir. Ağ temelli DLP'de ise ağ seviyesindeki şifrelemeyi aşmak için SSL vekil sunucular ile entegre çözümler önerilmektedir.

2.2. Gelişmiş Sürekli Tehdit (Advanced Persistent Threat-APT)

APT'ler "sofistike ve hedef odaklı kötü amaçlı yazılımlar" (Lajevardi & Amini, 2019), "yüksek profilli şirketler ve hükümetlere özgü bilgilere yönelik olarak sofistike ve kaynakları bol olan saldırganlar tarafından gerçekleştirilen siber saldırılar" (Chen ve ark., 2014) ve "bir hedef organizasyondan bilgi elde etmek, operasyonlarını sabote etmek veya her ikisini birden yapmak amacıyla kötü amaçlı, organize ve son derece sofistike uzun vadeli veya tekrarlayan bir ağ sızma ve sömürme operasyonu yürüten bir varlık" (Ahmad ve ark., 2019) olarak tanımlanabilirler. Bu tanımlar, APT'lerin gelişmiş ve karmaşık yeteneklerini öne çıkarırken, aynı zamanda bilgi elde etme hedeflerini vurgulamaktadır. APT'ler ayrıca bulaşıcı bir yapıya sahiptir, çünkü saldırganlar tüm bir ağa yayılabilmektedirler ve özel araçlar kullanarak sistem üzerinde uzun bir süre boyunca algılanmadan kalabilmektedirler.

2.2.1. APT davranış modeli

APT'ler öncelikle hedef sistem üzerinde kalıcılığı sağlar ve ardından sistemde yayılır. Temel amaçları hassas bilgileri sızdırmak veya hedef sistemde sabotaj yapmaktır. Bununla beraber, ilk olarak istihbarat toplama ve keşif yöntemlerini kullanarak hedef sistemin özelliklerini ve işlevselliğini incelerler. Bu sayede görevin ilerlemesini engelleyecek olası sorunları azaltmayı hedeflerler. APT'lerin hedefe giden yolda takip ettiği araçlar ve adımlar arasında farklılık olsa da birçok çalışma davranışsal yönlerinin aynı kaldığını göstermektedir. Bu sebeple başarılı bir tespit için bir APT davranışsal analiz modeli kullanılmasına ihtiyaç duyulmaktadır.

Bu bağlamda, APT saldırı adımlarını farklı aşamalara ayırmak yaygın bir yaklaşımdır. (Li ve ark., 2016), APT saldırı yaşam döngüsü modeli için dört aşama tanımlamışlardır: Hazırlık, Erişim, Konak, Hasat (Preparation, Access, Resident, Harvest). Genellikle erişim aşamasında bilinen güvenlik açıkları kullanılır. Ardından,

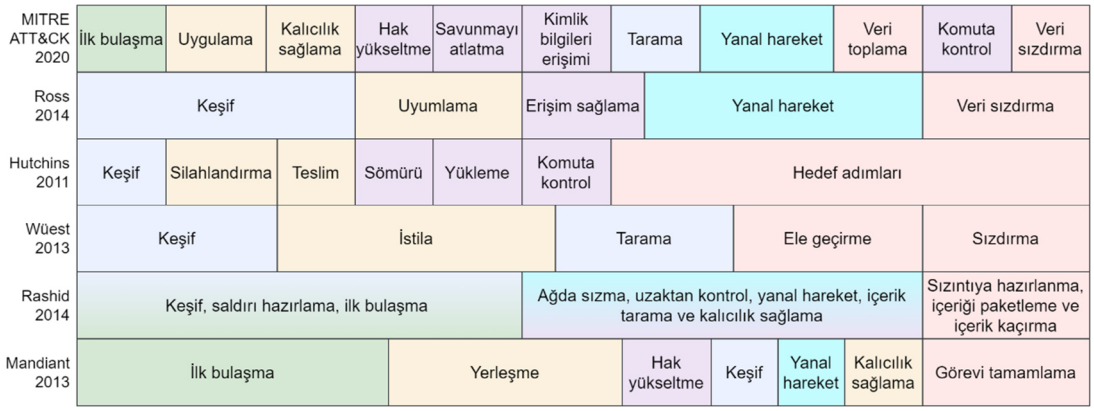
saldırgan, tüm bağlı düğümler dahil olmak üzere ortam hakkında ayrıntıları belirleyerek, yapılandırma bilgilerini toplar ve saldırı araçlarını gizlice sürdürür, böylece hedef sistem içinde kalır. Başarılı bir sızma sonrasında, saldırganın talimatları yürütmesine, kötü amaçlı yazılımı güncellemesine ve veri sızdırmasına olanak tanıyacak uzaktan kontrol aranır.

Benzer şekilde, (Atapour ve ark., 2018), APT özelliklerini tanımlamak ve diğer çok aşamalı saldırılardan ayırt edilebilir yönlerini ortaya koymak için teorik bir yaklaşım sunmuştur. Bu çalışmada APT davranışını tanımlayan soyut modeller oluşturulmuştur. Araştırma, Keşif, Silahlandırma, Teslimat (Delivery), Sömürü (Exploitation), Kurulum, Komuta ve Kontrol ve Hedef Adımları (Actions on Objectives) adımlarından oluşan bir Öldürme Zinciri Modeli (Kill Chain Model) bağlamında APT davranışlarını tanımlamıştır.

Hedef odaklı saldırılar farklı çalışmalarda ele alınmış ve farklı saldırı fazlarına ayrılmıştır. (Väisänen ve ark., 2016) çalışmasında ele alınan çeşitli çalışmalar ve bu tez çalışması kapsamında ele alınan MITRE ATT&CK'ın yaklaşımları Şekil 2.3'te birleştirilmiştir. Bu şekilde ilgili çalışmalardaki farklı aşamalandırmalar görülmektedir. Şekilde, aşamalar farklı sayılarda ele alındığı görülmekle beraber ilk bulaşmadan sonuca kadarki süreçte benzer akışların var olduğu görülmektedir. (Brewer, 2014), aşamaları keşif, uyumlama, erişim sağlama, yanal hareket ve veri sızdırma olarak belirlerken; (Hutchins ve ark., 2011) keşif, silahlandırma, teslim, sömürü, yükleme, komuta kontrol ve hedef adımları olarak belirlemiştir. (Wüest, 2013) keşif, istila, tarama, ele geçirme ve sızdırma olarak aşamalandırmış; (Rashid ve ark., 2014) daha az adım ile sunarken, (Mandiant Intelligence Center, 2013) çalışmasında ilk bulaşma, yerleşme, hak yükseltme, keşif, yanal hareket, kalıcılık sağlama ve görevi tamamlama olarak belirlemiştir.

Şekil 2.3'te ayrıca bu çalışmalardaki benzer adımlar ayrı renkler ile gösterilmiştir. Bu renklendirme ile genel olarak benzer akışların sunulduğu daha iyi görülmektedir. Hedef sisteme öncelikle bir ilk bulaşma aşaması ile giriş yapılmaktadır. Bazı çalışmalar bu ilk bulaşma öncesinde de sosyal mühendislik gibi çalışmalar ile bir keşif aşaması öngörmüştür. Sonrasında çeşitli yöntemlerle sistem üzerinde yerleşerek kalıcılık sağlanmaktadır. Sonrasında hak yükseltme, kimlik bilgilerine erişim işlemleri ile hedefe doğru ilerlemek için gerekli yetkiler elde edilmektedir. Bu aşamadan sonra sistem üzerinde tarama yapılarak asıl hedefe doğru yayılma yolları keşfedilmektedir.

Keşfedilen bu yollar kullanılarak yanal hareket ile sistem üzerinde yayılan saldırılar, veri toplama ve sızdırma ile son bulmaktadır. Bu aşamaların bazı noktalarında komuta kontrol ile uzaktan erişim ve gerekli yazılımların yüklenmesi sağlanabilmektedir.



Şekil 2.3. Farklı çalışmalardaki APT davranış modellerini karşılaştırması.

Şekil 2.3'te yer alan MITRE ATT&CK çerçevesi (MITRE, 2020a) tarafından bu davranışların daha detaylı ve kapsamlı bir listesi sunulmuştur. MITRE ATT&CK, kötü niyetli davranışların sınıflandırılmasını sağlayan bir bilgi tabanıdır. Bu bilgi tabanı ile siber güvenlik alanına katkıda bulunan ortak bir dil oluşturmuştur. Esas olarak taktikler, teknikler ve prosedürler (TTP'ler) arasındaki ilişkiyi temsil eden bir matristir (Şekil 2.4). APT'lerin gerçekleştirdikleri taktikleri, ilk bulaşma (initial access), uygulama/ yürütme (execution), kalıcılık sağlama (persistence), hak yükseltme (privilege escalation), savunmayı atlatma (defense evasion), kimlik bilgileri erişimi (credential access), tarama (discovery), yanal hareket (lateral movement), veri toplama (collection), komuta kontrol (command and control) ve veri sızdırma (exfiltration) gruplarına ayırır. Her grup, saldırının belirli bir adımını başarıyla gerçekleştirmek için izlenen prosedürleri açıklayan çeşitli Teknikleri içerir. Bu hiyerarşik yaklaşım, bir saldırının unsurları ile TTP'ler arasında etkili bir şekilde eşleştirme yapma imkânı sunmakta ve nihayetinde ilgili APT ile ilişkilendirme yapılmasına imkân vermektedir. Bir sistemde gerçekleştirilmiş olan işlemler tanımlanabilirse, bunlar ilgili teknikler ve taktikler ile ilişkilendirilebilir. Böylece, tehdit aktörü, niyeti ve saldırının mevcut durumu belirlenebilir. Böylece mevcut durum doğru şekilde tanımlandığında, bu tehdit aktörüne karşı en etkili şekilde nasıl savunma yapılabileceği ve sorunun ortadan kaldırılabileceği belirlenebilir. Bu nedenle, MITRE ATT&CK'ye dayalı olarak bir APT'lere ait TTP'lerin tespiti, etkili bir savunma sisteminin geliştirilmesi, bakımı ve etkinliği açısından önemli bir yöntemdir. Ayrıca, Ağrı Piramit'inde (Bianco, 2014)

önerildiği gibi, TTP'lerin başarılı bir şekilde tespit edilmesi saldırganları kullandıkları teknikleri tamamen değiştirmeye ve yeni bir araç geliştirmeye zorlar. Bu yaklaşım, savunma mekanizmasının etkinliğini artıracaktır. Bu nedenle, bu tez kapsamında, MITRE ATT&CK ile ilişkili olarak TTP'lerin tespitini sağlayan ontoloji tabanlı bir yöntem önerilmektedir.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels	
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy	
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels	
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication	
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption	

Şekil 2.4. MITRE ATT&CK teknik/taktik matrisi.

2.2.2. APT tespitinde sistem çağrılarının kullanımı

Anormallik tespiti ve kötü amaçlı yazılım analizi konusunda, prosesler tarafından yapılan sistem çağrılarının toplanması ve değerlendirilmesi yaygın olarak kullanılmaktadır. Bu sistem çağrıları, genellikle şüpheli prosese dll enjeksiyonu ile yapılan "araya girme" müdahalesi ile yakalanmaktadır.

(Nguyen ve ark., 2003) çalışmasında dosyalarla ve proseslerle ilgili sistem çağrıları veri kümesini toplamak için kancalama kullanılmıştır. Kullanıcılar, dosyalar ve proseslere ilişkin modeller ve ilişkiler analiz edilmektedir.

(Moon ve ark., 2017) çalışmasında APT saldırılarını önlemek için davranış analizine dayalı bir saldırı tespit sistemi (Intrusion Detection System-IDS) sistemi önerilmiştir. Önerilen IDS sistemi, sistem çağrılarını izlemekte, ardından bu çağrıları sistem içinde gerçekleştirdikleri işlemin türüne göre (dosya oluşturma, kayıt defteri erişimi, ağ bağlantısı vb.) sınıflandırmaktadır. Son olarak ise, bir karar ağacı kullanarak APT saldırılarını tespit etmektedir.

(Ki ve ark., 2015) çalışmasında, kötü amaçlı yazılımları tespit etmek için sistem çağrısı dizisi analizine dayanan bir yaklaşım geliştirilmiştir. Sistem çağrıları, Microsoft Detours kütüphanesi (Microsoft, 2002) kullanılarak kancalama yoluyla hedef işlevlerin özel yapılmış işlevlerle değiştirilmesiyle yakalanmaktadır. Zararlı ve masum yazılımlar çalıştırılarak toplanan sistem çağrısı dizileri, bir imza veri tabanında depolanmakta, sonrasında "DNA dizisi hizalama algoritmaları" ile en uzun ortak dizilerini belirlemeye çalışılmaktadır. Yüksek başarım oranları sergilemesine rağmen, bu yaklaşımın da bazı problemleri vardır. En önemlisi, nispeten uzun sürelerde çalışmakta ve yüksek işlemci ve hafıza gereksinimine ihtiyaç duymasıdır. Bu sebeple, gerçek zamanlı tespit için uygun değildir. Ayrıca, imza tabanlı olması davranışsal analiz açısından atlatılmaya açık bir taraf oluşturmaktadır.

APT sınıflandırması konusundaki bir çalışma ise (Mirza ve ark., 2014)'tür. APT'lerin erken aşamalarında meydana gelen izinsiz girişlere karşı farklı yöntemlerin bir arada kullanıldığı bir güvenlik mekanizması sunulmuştur. Yaklaşım, ana sistem içinde meydana gelen API çağrılarının yanı sıra "syslog", "snort" ve "nagios"tan gelen raporlar ve bildirimlerle beslenen bir "Güvenlik Bilgisi ve Onay İzleme" (Security Information And Event Management - SIEM) sistemi etrafında kurulmuştur. Sistem çağrıları, açık kaynaklı bir ana bilgisayar tabanlı saldırı önleme sistemi olan Ambush (Weeks, 2013) tarafından toplanır. Bu çalışmanın kapsamı, diğer saldırı vektörlerinden meydana gelebilecek veri kaybını önlemede boşluk bırakan sıfırıncı gün saldırılarını tanımlamak ve önlemekle sınırlıdır.

(Ravi & Manoharan, 2012), Windows API işlev çağrısı sırasına dayalı olarak kötü amaçlı yazılım tespitine yönelik bir yaklaşım önerir. Önerilen yöntem, hedef prosesin içe aktarma adres tablosunu (Import Address Table-IAT) değiştirerek API çağrılarını kancalamak; ardından ilgili sistem çağrısının her tekrarı için bir sıra numarası üretmek şeklinde çalışır. Öğrenme ve analiz, öncelikle sistem çağrısı sırasına dayanmaktadır. Çevrimdışı öğrenme sırasında, sistem çağrıları 4-gramlık dizilerinden sınıflandırma kuralları çıkarılmaktadır. Destek ve güven eşiği değerlerini karşılayan kurallar, kural veri tabanına eklenmektedir. Çevrimiçi analiz aşamasında ise sistem çağrısı dizileri ile birlikte kural veri tabanı kullanılarak, sistem çağrısı dizilerini 4-grama bölerek tüm 4-gramların kötü amaçlı ve iyi huylu ortalama güven değerlerini ayrı ayrı hesaplanarak proses sınıflandırması yapılmaktadır.

Bu çalışmaların incelenmesi neticesinde sistem çağrılarının kullanımının APT'lerin davranış analizine dayalı sınıflandırmasında etkili olabileceği görülmektedir. APT'lerin belirli bir parmak izi olmamasına rağmen, kullandıkları komutların işletim sistemi seviyesinde karşılık geldiği fonksiyonlar bulunmaktadır. Bir tekniğin uygulanmasında farklı yollar olmakla beraber, sonuçta işletim sisteminde aynı sistem çağrısını çağırmaktadır.

2.3. Ontoloji ve Siber Güvenlik Alanında Ontoloji Kullanımı

APT davranış modelinin incelenmesi ve literatürdeki diğer yöntemlerin bu davranış modelini ifade etmede yetersiz olduğunun değerlendirilmesi neticesinde davranışsal analiz (behavioral analysis) yöntemleri ile bu problemin çözülebileceği görülmüştür. Davranışsal analizin gerek genel bilişim sistemlerinde gerekse de siber güvenlik alanındaki en önemli kuramlarından birisi olan ontoloji ve ontolojinin kullanımı bu bölümde detaylandırılmıştır.

2.3.1. Taksonomi

Bir ontoloji, içerisinde bir taksonomi barındırmaktadır. Bu sebeple, öncelikle taksonomi kavramı detaylandırılmıştır.

Sözlük tanımına göre bir taksonomi, sınıflandırma şemasının belirlenmiş kriterlere göre gruplara veya kategorilere sistematik bir şekilde düzenlendiği bir sınıflandırma sistemidir. Genel tanımıyla, sınıflandırma bilimi ve pratiğidir. Yunancada düzen anlamındaki taxis ve kanun anlamındaki nomos kelimeleri ile türetilmiştir. Bilginin hiyerarşik ve sistematik olarak sınıflandırılması ve bu sınıfsal bağlarla organize edilmesini sağlar.

Simpson'a göre (Simpson, 1961), sınıflandırmaların oluşturulmasında kullanılan özelliklerin en az iki karşıt duruma bölünebilmesi gerektiğini belirtir. Ayrıca, taksonomik özelliklerin, söz konusu nesneden gözlemlenebilir olması gerektiğini belirtir

(Amoroso, 1994; Landwehr ve ark., 1994; Lindqvist & Jonsson, 1997) gibi çalışmalar ile bilgisayar güvenliği alanında yeterli ve kabul edilebilir bir taksonominin sahip olması gereken özelliklerini belirlemiştir. Özet olarak, aşağıdaki özellikleri bir taksonomi için gerekli görmüşlerdir:

Birbirini dışlama (Mutually exclusive): Bir kategorideki sınıflandırma, diğer tüm sınıflamaları dışlar; çünkü kategoriler örtüşemez.

Kapsamlılık: Bütün kategoriler bir arada değerlendirildiğinde tüm olası sınıflandırmaları içerirler.

Belirsiz olmama: Sınıflandırma kim tarafından yapılırsa yapılsın, kategori açık ve net olmalıdır. Böylece sınıflandırma belirsizlik içermemelidir.

Tekrarlanabilirlik: Sınıflandırma kim tarafından yapılırsa yapılsın, her seferinde aynı sınıflama oluşmalıdır.

Kabul edilebilirlik: Taksonomi mantıklı ve sezgisel olmalıdır, böylece genel kitle tarafından kabul görebilir olmalıdır.

Yararlılık: Taksonomi, araştırma alanına daha detaylı bir kavrayış ve içyüzünü kavrama sağlamak için kullanılabilir olmalıdır.

Anlaşılırlık: Taksonomi, uzmanlık düzeyinden daha az bilgiye sahip olanlar için kullanılabilir olmalıdır.

Uyumluluk: Taksonominin terimleri, belirlenmiş terminolojiye (siber güvenlik terminolojisine) uygun olmalıdır.

Objektiflik: Sınıflandırma için gerekli özellikler, gözlem altındaki nesnede ayırt edici nitelikte ve açıkça gözlemlenebilir şekilde var olmalıdır.

Belirleyici: Sınıflandırma için gerekli özelliği çıkarmak için takip edilebilecek net bir prosedür olmalıdır.

Belirlilik (Deterministic): Sınıflandırma için gerekli özelliğin değeri benzersiz ve net olmalıdır.

Bu özellikler içerisindeki "kabul edilebilirlik" ve "anlaşılırlık" biraz daha öznel değerlendirmeler içerebileceğinden her taksonomide aranması uygun olmayabilir.

Yapılan değerlendirmelerde MITRE ATT&CK çerçevesinin bu özellikleri taşıdığı görülmektedir. Sadece belli tekniklerin birden çok taktik altında sınıflandırılması "birbirini dışlama" özelliğine aykırı gibi görülebilir. Örneğin, "T1015-Accessibility Features" tekniği (MITRE, 2019a) hem kalıcılık hem de hak yükseltme taktiklerinin altında yer almaktadır. Ancak bu durumda tekniğin farklı aşamalarının farklı amaçlar için kullanıldığı ve bu aşamalarda yapılan işlemlerin farklı olması sebebiyle esas

olarak hangi taktik altında gerçekleştirildiği belirlenebilmektedir. Bu yönüyle de birbirini dışlar bir durum sergilediği söylenebilir.

2.3.2. Ontoloji

Ontoloji ve bilgi temsili (knowledge representation) oldukça yakın ilişkili kavramlardır. Bilgi temsili, en temel ifadesi ile bir nesnenin yerine geçebilen veya onun yerine kullanılabilen kavramlar olarak ifade edilebilir. Böylece, bir sistemin ilişkili nesne hakkında akıl yürütmesine olanak sağlar. Bilgi temsili ayrıca nesnenin özünü tanımlayan terimleri belirleyen ontolojik taahhütlerin bir kümesidir. Başka bir deyişle, nesnenin ilişkilerini açıklayan üst verilerdir.

Bir ontoloji, ele alınan etki alanındaki (domain) kavramların biçimsel ve açık bir tanımı olan sınıflar (konseptler); her bir kavramı tanımlayan çeşitli ayırt edici belirteçleri ve nitelikleri gösteren özellikler (yuvalar (slot)) ve bu özellikler üzerinde yer alan kısıtlamalardan meydana gelir. Böylece bir ontoloji, sınıfların bireysel örnekleri (instance) ile birlikte bir bilgi tabanı oluşturur. Gerçekte, ontolojinin bittiği ve bilgi tabanının başladığı ince bir çizgi bulunmaktadır.

Sınıflar, çoğu ontolojinin odak noktasıdır. Sınıflar, etki alanı içindeki kavramları tanımlar. Örneğin, kitap sınıfı tüm kitapları temsil eder. Belirli olan kitaplar bu sınıfın örnekleridir. Örneğin fiziksel olarak var olan bir tane Sefiller kitabı, Sefiller kitapları sınıfının bir örneğidir. Bir sınıf, üst sınıftan daha spesifik kavramları temsil eden alt sınıflara sahip olabilir. Örneğin, tüm kitapların sınıfını klasik, macera, kişisel gelişim gibi kitaplara bölebiliriz. Alternatif olarak, tüm kitapların sınıfını basılı ve dijital kitaplara bölebiliriz.

Özellikler (yuvalar), sınıfların ve örneklerin özelliklerini tanımlar: “Suç ve Ceza” kitabı felsefi bir roman türündedir; bu kitap Dostoyevski tarafından yazılmıştır. Bu örnekte kitap hakkında iki özellik bulunmaktadır: “felsefi roman” değerine sahip olan “tür” özelliği ve “Dostoyevski” değerine sahip olan “yazar” özelliği. Sınıf düzeyinde, kitap sınıfının örneklerinin tür, yazar, yayınevi, basım yılı gibi özellikleri olabileceğini söyleyebiliriz.

Bu örnekteki bazı özelliklerin değerleri de esasında başka sınıfların örnekleri olabilir. Örneğin “Yayınevi” özelliğinin değeri “Sakarya Basımevi” olduğunu varsayarsak, bu özellik “Yayınevi” sınıfının bir alt sınıfı olacaktır. Benzer şekilde “Yazar” özelliğindeki “Dostoyevski” değeri, “Yazar” sınıfının bir alt sınıfı olan “Roman

Yazarları" sınıfının bir örneği olabilir. Bu sınıfların da kendilerine ait özellikleri vardır. Yazar sınıfının "yazdı", Yayınevi sınıfının "basım yaptı" özellikleri olabilir.

Buna göre, bir ontolojiyi geliştirmek, genel olarak şunları içerir:

- Ontolojideki sınıfları belirleme.
- Sınıfları taksonomik (alt sınıf-üst sınıf) hiyerarşisi ile yerleştirme.
- Özellikleri tanımlama ve bu özellikler için izin verilen değerleri belirleme.
- Örnekler için özellik değerlerini doldurma.

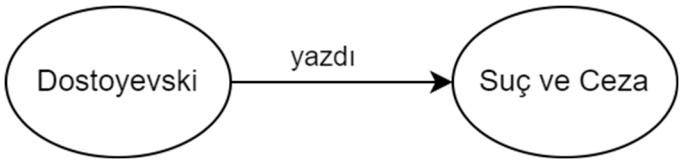
2.3.3. Ontolojinin yazılımsal olarak temsil edilmesi

İnternet, aslında insan tüketimi için geliştirilmiştir ve üzerindeki her şey makine tarafından okunabilir olmasına rağmen bu veri makine tarafından anlaşılabilir değildir. İnternetteki herhangi bir şeyi otomatikleştirmek çok zordur ve içerdiği bilgi miktarı nedeniyle bunun manuel olarak yönetilmesi mümkün değildir. Bu sebeple W3C konsorsiyumu tarafından bu verileri tanımlamak için bir üst-veri kullanma imkanı sunan Kaynak Açıklama Çerçevesi (Resource Description Framework - RDF) geliştirilmiştir (Ora Lassila & Ralph R. Swick, 1999). Böylece internetteki kaynakları tanımlayan veri bilgisayarın anlayabileceği şekilde ifade edilebilmiştir. Sonrasında RDF-S (W3C, 2014) ve OWL (W3C, 2012) standartları ile RDF üzerine çeşitli eklentiler yapılarak geliştirmeler devam etmiştir. Günümüzde bu standartlar, ontolojileri temsil etmek için kullanılmaktadır.

RDF ile nesneler ve ilişkiler üçlüler (triple) kullanılarak tanımlanır. Özne (subject), yüklem (predicate) ve nesne (object) olarak genellenebilecek bu üçlüler ile tanımlamalar yapılmaktadır. OWL ile bu gösterim daha basitleştirilerek XML içerisinde bir hiyerarşik yapıda gösterim sağlanmıştır.

Şekil 2.5'te önceki örnekte yer alan basit bir ilişki gösterilmiştir. Şekil 2.6'da ise bu ilişkinin RDF ve OWL ile nasıl temsil edildiği sergilenmiştir. Burada öncelikle ontolojiye ait bazı üst bilgiler verildikten sonra sınıf tanımlamaları (owl:Class), sınıflara ait özelliklerin tanımlamaları (owl:ObjectProperty) ve sınıfların örnekleri (owl:NamedIndividual) yer almaktadır. Şekil 2.5'te gösterilen "Dostoyevski yazdı Suç ve Ceza" ilişkisi Dostoyevski örneğinin altında yazdı özelliğinin eklenmesi ile ifade edilmiştir. Belirtildiği gibi OWL ile bu ilişki hiyerarşik olarak ifade edilmektedir. Bu durum basit RDF ile ifade edilseydi "<http://sakarya.edu.tr/ornek#Dostoyevski> <http://sakarya.edu.tr/ornek#yazdı> <http://sakarya.edu.tr/ornek#Suç_ve_Ceza>"

şeklinde bir üçleme ile ifade edilecekti. Her iki gösterim de doğrudur ve bu ikisi arasında dönüşüm yapılabilmektedir.



Şekil 2.5. Örnek bir ontoloji ilişkisi.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://sakarya.edu.tr/ornek-ontoloji#"
     xml:base="http://sakarya.edu.tr/ornek-ontoloji"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:ornek="http://sakarya.edu.tr/ornek-ontoloji"
     xmlns:ornek1="http://sakarya.edu.tr/ornek#">
    <owl:Ontology rdf:about="http://sakarya.edu.tr/ornek-ontoloji"/>

    <owl:Class rdf:about="http://sakarya.edu.tr/ornek#Kitap"/>
    <owl:Class rdf:about="http://sakarya.edu.tr/ornek#Yazar"/>

    <owl:ObjectProperty rdf:about="http://sakarya.edu.tr/ornek#yazdı">
        <rdfs:subPropertyOf
          rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <rdfs:domain rdf:resource="http://sakarya.edu.tr/ornek#Yazar"/>
        <rdfs:range rdf:resource="http://sakarya.edu.tr/ornek#Kitap"/>
    </owl:ObjectProperty>

    <owl:NamedIndividual rdf:about="http://sakarya.edu.tr/ornek#Dostoyevski">
        <rdf:type rdf:resource="http://sakarya.edu.tr/ornek#Yazar"/>
        <ornek1:yazdı rdf:resource="ornek#Suç_ve_Ceza"/>
    </owl:NamedIndividual>
    <owl:NamedIndividual rdf:about="ornek#Suç_ve_Ceza">
        <rdf:type rdf:resource="http://sakarya.edu.tr/ornek#Kitap"/>
    </owl:NamedIndividual>
</rdf:RDF>
```

Şekil 2.6. Örnek ontolojinin RDF ve OWL ile temsil edilmesi.

2.3.4. Zararlı tespitinde ontolojinin kullanımı

Siber güvenlik alanındaki en belirgin ihtiyaç, zararlı hareketlerin tespiti ile önlenmesi ve bu bilginin diğer bilgisayarlara/sistemlere/hedeflere ulaşmadan önce bu sistemlerde çalışan saldırı önleme sistemlerine iletilerek o sistemlerin de korunmasıdır. Yani, makine tarafından anlaşılabilir/işletilebilir kuralların çıkarılması ve bu kuralların insan müdahalesine ihtiyaç kalmadan paylaşılması ve uygulanabilmesidir.

Bu noktada eskiden beri kullanılan temel çözüm, zararlı imzalarının (hash) çıkarılması ve bu imzalar üzerinden zararlı tespiti yapılmaya çalışılmasıdır. Ancak zararlı üzerindeki basit değişikliklerle zararlının farklı varyasyonlarının çıkarılabilmesi ile bu yöntem atlatılabilmektedir (Katz ve ark., 2014; Rashid ve ark., 2014). Daha kapsamlı saldırılarda zararlı içerisinde sanal makine kullanımı gibi yeteneklerin kullanılması ile zararlının tespit sistemini atlatması olasılığının çok daha fazla olabileceği görülmüştür (Kafka, 2018).

Bir sonraki seviye, zararlıya ait çalıştırabilir dosyanın belirli özelliklerinin (feature) çıkarılması ve bu özellikler üzerinden bir sınıflandırma yapılarak benzer zararlıların tespitinin sağlanması olmuştur. Tez kapsamında statik analiz kısmında çözüme dahil ettiğimiz bu yöntemle, imza karşılaştırmaya göre çok daha başarılı sonuçlar alınsa da zararlıların kod karmaşıklaştırma, biçim değiştirme, sanal makine kullanma gibi özellikleri içerebilmesi statik analizin tek başına yeterli olmamasına neden olmuştur.

Daha ileri bir önlem olarak, sistemde çalışan sistem çağrılarının dinamik olarak takip edilmesi ile bu tip gizli hareketler tespit edilmeye çalışılmıştır (Canfora ve ark., 2015; Ravi & Manoharan, 2012). Ancak özellikle APT'ler tarafından kullanılan/suistimal edilen PowerShell, Vscript gibi esasında zararlı olmamakla beraber çeşitli zayıflıklara (vulnerabilities) kapı aralayan normal/masum programlar üzerinden ataklar yapılması ile hangi programların/sistem çağrılarının izleneceği ve nasıl değerlendirileceği önem kazanmıştır. Ayrıca birden çok programın atağın belli adımlarını gerçekleştirmesi sebebiyle sadece proses bazlı değil, sistem genelinde ve süreğen bir izleme ve tespitin yapılması gerekmiştir.

Bu sonuçlar ışığında gerek başarılı bir tespitin yapılabilmesi gerekse de zararlıların çalıştırılabilir dosya bağımsız analizini sağlayan bir yöntem geliştirilebilmesi için sistemde yer alan öğelerin ve ilişkilerinin doğru ve genelleştirilebilir bir şekilde tanımlanması gerektiği değerlendirilmiştir. Bu sebeple, davranışsal analiz ve bu kapsamda ontoloji kavramı ve sistemi bir ontoloji olarak ifade etmek doğal bir sonuç olarak ortaya çıkmaktadır.

Literatürde de özellikle atak vektörlerinin ve APT kaynaklı tehditlerin tanımlanmasında ontoloji kavramının öne çıktığı görülmüştür. (Woo ve ark., 2013) çalışmasında siber saldırıların tespitinde, alt fonksiyonların/aksiyonların tanımlanarak, bu aksiyonların ve etki ettikleri öğelerin bir arada değerlendirmesi ile saldırının

davranış (behavior) modellenmesinin nasıl olması gerektiği tartışılmıştır. Bir aksiyon ve bu aksiyonla ilişkili kullanıcı, bilgisayar, veri gibi bileşenlerin bir arada değerlendirilmesi ile aksiyon > davranış > atak çıkarımlarının ve atak tespitinin sağlanabileceği anlatılmıştır.

(Grégio ve ark., 2016) çalışmasında ise bir zararlının davranışının tespitinde proses, sistem çağrısı, kaynak, hedef ilişkisinin modellenmesi ve ontoloji içerisinde değerlendirilmesi ile özellikle bilinmeyen/yeni programların risk/zararlılık seviyesinin tespitine nasıl gidilebildiği ve bir sistem üzerindeki uygulaması anlatılmıştır.

(Jacob ve ark., 2008) çalışması da daha çok zararlılar üzerine yoğunlaşmakla ve statik analizi öne çıkarmakla beraber, benzer şekilde birleştirme kodu (assembly code) komutlarının her bir alt kümesinin birer davranışa karşılık gelebileceği ve bu yolla tespitin sağlanabileceğini anlatmaktadır. Bu çalışmadaki bir diğer dikkat çeken yaklaşım, her bir davranışın bir durum makinası (state machine) ile ifade edilmesi gerektiğinin belirtilmesidir.

2.4. APT Tespitinde Davranışsal Analizin ve Ontolojinin Kullanımı

Siber güvenlik alanındaki davranışsal analiz yaklaşımı APT özelindeki çalışmalarda da görülmektedir.

(Choi ve ark., 2015) tarafından APT saldırı davranışlarını analiz etmek için bir ontoloji önerilmiştir. Önerilen çözümün temel amacı, belirli API çağrılarını saldırı davranışlarıyla ilişkilendirerek APT saldırılarını belirlemektir. Örneğin, InternetOpen API çağrısı, "Yetkisiz web sitesi erişimi" APT saldırı davranışına eşlenmiştir. Makaledeki önerilen ontoloji, bu davranışları ve saldırı yoğunluğu, saldırı alanı ve saldırı konusu gibi ek özellikleri merkezi bir APT Saldırı nesnesi ile ilişkilendirmeyi amaçlamaktadır. Önerilen yaklaşım ve ontoloji ilgi çekici olsa da kötü amaçlı yazılımın saldırıyla nasıl ilişkilendirildiği gibi bazı özellikler belirsizdir.

(Luh ve ark., 2016) çalışması, (Hutchins ve ark., 2011) tarafından sunulan “Kill-Chain” modeline dayalı bir APT saldırı ontolojisi önermiştir. Önerilen metot, “Kill-Chain” modeline dayalı olarak, APT saldırısını keşif, silahlandırma, teslim etme gibi aşamalara bölmek ve aşamaları alt sınıflara bölmektir. Böylece tespit edilen zararlı faaliyetler bu aşamalarla ilişkilendirilmektedir. Örneğin, Keşif aşaması altında elde etme, analiz ve tanımlama alt sınıfları bulunmaktadır. Bu çalışma ile APT saldırılarını

tespit etmek için etkili olabilecek bir yaklaşım önerilmektedir. Bununla birlikte, önerilen yaklaşım, "yanlış pozitif" tespitlere karşı oldukça etkili olan proses skorunu dikkate almamaktadır. Ayrıca, saldırı aşamalarının Kill-Chain modeli yerine MITRE ATT&CK gibi daha kapsamlı bir bilgi tabanı ile ilişkilendirilmesi, mevcut ve gelecekteki tehditler için tespitin ifadesini ve etkinliğini artıracaktır.

(Lajevardi & Amini, 2019), önceden tanımlanmış bir APT tespit politikasının ihlal edilip edilmediğini belirlemek için düşük seviye ayrıntılar olan API çağrıları ve ağ olaylarından yararlanan bir anlamsal yaklaşım önermektedir. Önerilen yöntem, işletim sistemi olayları ve ağ olayları arasındaki ilişkilerden kötü amaçlı bir işlem olup olmadığını veya güvenlik politikalarının ihlal edip edilmediğini tespit etmeyi amaçlamaktadır. Sistem, işletim sistemi ve ağ olaylarını tanımlamak için OWL-DL biçimini kullanır. Bir "olay", bir "özne" ile; "özne", eylemi uyguladığı "nesne" ve eylemin "zaman damgası" ile ilişkilendirilmiştir. Konu, nesne, eylem ve zaman damgası değişkenlerinin bir araya getirilmesi sonucunda güvenlik politikalarına dayalı bir sinyal mekanizması oluşturur. Makale, sistemdeki olayları APT davranışıyla ilişkilendirerek tez kapsamında sunulan öneriye benzer bir yaklaşım sunmaktadır. Ancak, kötü amaçlı yazılımları tekil olarak ele alarak sistem genelinde bir değerlendirme sunmamaktadır. Bu sebeple APT'lerin gerçekleştirebildiği görünüşte zararsız olayları gerçek saldırıyla ilişkilendirmekte yetersiz kalmaktadır.

(Kim ve ark., 2019) çalışmasında bir APT tehdit ontolojisi önerilmiş ve genel güvenlik bilgisi ve etki alanına özel ontolojiler ile entegre edilmiştir. Yazarlar, MITRE'nin matrisine benzeyen bir APT saldırı şablonu sunmuşlardır. Bu amaçla raporlar, bloglar ve web siteleri gibi çeşitli kaynaklardan bilgi toplamışlardır. Daha sonra, bu APT tehdit bilgisini diğer ontolojilerle entegre ederek sistem bileşenlerinin (sunucular, bilgisayarlar gibi) bir APT saldırısına karşı herhangi bir risk oluşturup oluşturmadığını değerlendirmişlerdir. Çalışmadaki APT ontolojisi geniş olmamakla beraber, sistem ayrıntılarını APT saldırılarıyla bağlantı kurma konusunda örnek bir yaklaşım sunmaktadır.

(Bryant & Saiedian, 2020), MITRE ATT&CK matrisini kill-chain çerçevesi ile entegre ederek "Güvenlik Bilgisi ve Onay İzleme" (Security Information And Event Management-SIEM) tespit yeteneklerini geliştirmişlerdir. Sistemdeki farklı kayıtları toplayarak ve birleştirerek gelişmiş bir saldırı tespit yöntemi sunulmaktadır. Sunulan ontolojinin detayları verilmese de sistem kayıtlarındaki belirli parametreleri farklı

saldırı yöntemleriyle ilişkilendirerek alarm oluşturulduğu görülmektedir. Bununla birlikte, tüm saldırıların kolayca tanımlanabilir olay kimlikleri oluşturmadığını unutmamak gerekir. Bu sorunu aşmak için sistem çağrıları hakkında bilgi toplayabilen bir uç nokta tabanlı sistem gereklidir. Böylece, saldırıyla ilişkilendirilen kritik eylemler, nesneler ve aktörlerin çıkarımı mümkün olacaktır.

(Han ve ark., 2021), sistem çağrılarını APT kötü amaçlı yazılımlarla ilişkilendiren bir ontoloji kullanılmaktadır. Yöntem, TF-IDF tabanlı bir algoritma kullanarak APT'lerin her davranışı için bir API çağrısı dizisi oluşturmakla başlamaktadır. Böylece, her APT davranışı bir API çağrı dizisiyle temsil edilir. Bu sayede, API çağrı dizilerini inceleyerek APT ve niyetini tanımlama imkânı sağlar. Bu çalışma, API çağrıları gibi düşük seviye ayrıntıları APT davranışları ile ilişkilendirme konusunda iyi bir örnek sunmaktadır. Bununla birlikte, kötü amaçlı yazılımın uzun bir süre boyunca adımlarını gerçekleştirdiği ve arada ek veya farklı adımlar kullandığı gerçek yaşam senaryolarında API çağrı dizilerine bağımlılık sorun oluşturacaktır. Ayrıca, işlemin kaynağı ve hedefi gibi diğer ayrıntıları kullanmaması ve saldırıya dahil olan kullanıcı hesaplarını eksik bırakması, çözümün risk değerlendirme yeteneğini azaltmaktadır.

(Kaya ve ark., 2019) çalışmasında tez kapsamında sunulan ontoloji yaklaşımı genel çerçevesi ile sunulmuştur. Sunulan ontoloji, tezin 3. bölümünde detaylandırılmıştır.

2.5. İçerik Sınıflandırma

2.5.1. İçerik sınıflandırma yöntemleri

DLP alanında ilgili verinin ya da dosyanın hassasiyet seviyesini belirlemek için üst-verilerden başka, doğrudan işleme konu olan içeriğin çeşitli yöntemlerle analiz edilerek tespit sağlanmaktadır. Bu yöntemler bu bölümde detaylandırılmaktadır.

2.5.1.1. Anahtar kelime eşleştirme

Anahtar kelime eşleştirmesi, en basit analizi yöntemlerindendir. Önceden belirlenmiş bir anahtar kelime listesine dayanılarak, metin içerisinde bu kelimeler aranmaktadır. Bu yöntem hızlıdır ve hassas belgelerin belirli kelimeler veya metin dizeleriyle tanımlanabildiği bir durumda etkili olabilir. Hedef kelimelerin sayıca az olduğu ve kelimelerin orijinal halleri ile metinlerde geçme ihtimalinin yüksek olduğu durumlarda daha etkilidir. Ancak, çoğu gerçek durumda başarılı tespit sağlanamamaktadır. Sosyal güvenlik numaraları, kredi kartı numaraları gibi dinamik değerler bu tür bir yöntem ile

keşfedilememektedir. Daha çok yapılandırılmış verilerde (structured data) kullanılması uygun olmaktadır.

2.5.1.2. Kurallı ifadeler

Kurallı ifadeler (regular expressions) ile yapısal olarak aynı kalan, fakat değer olarak değişen kimlik numarası, banka hesap numarası gibi veriler tespit edilebilmektedir. Örneğin ^[1-9]{1}[0-9]{9}[02468]{1}$ ifadesi ile T.C. kimlik numarası tespiti yapılabilmektedir. İçerik üzerinde maksatlı bir kaçırma saldırısı olmadığı varsayımında yapısı ya da deseni belli olan veriler için etkili ve hızlı bir tespit sağlar.

2.5.1.3. İmzalama

Özellikle DAR durumunda olan veriler için imzalama (fingerprinting) yöntemi de DLP çözümlerinde yer almaktadır. Bunun için hassas olarak belirlenmiş bir dokümanın tamamının ya da parçalarının bir özetleme fonksiyonu (hash function) ile özetinin çıkarılması; daha sonrasında bu özet ile diğer dokümanlardan aynı şekilde elde edilmiş özetlerin karşılaştırılması yöntem olarak uygulanır. Bu yönüyle hızlı bir şekilde büyük dokümanların karşılaştırılması sağlanabilmektedir. Ayrıca dosyanın sahip olduğu dosya yaratılma tarihi, yazar ya da başka üst-veriler de bu imzaya eklenebilmektedir.

Özellikle dosyanın parçalı olarak imzasını çıkartmak içerik üzerindeki değişikliklere karşı daha dayanıklı bir çözüm sunar. Dosyanın bazı kısımları değişse bile diğer kısımların aynı kalabilmesi ve önceki özetlerle eşleşmenin kısmen de olsa sağlanmasına imkân sunar. Ancak bununla beraber parçalama seviyesinin ne kadar olacağı net değildir. Çok küçük parçalara ayrılması, benzer ifadelerin hassas olmayan dokümanlarda da geçme olasılığını arttırmaktadır. Bu durumda imzaların tespit yeteneği azalacaktır ve hatta yanlış tespitler yapabilecektir. Bu sorunu azaltmak için (Gessiou ve ark., 2011) çalışmasında bilinen hassas olmayan kelimelerin imzalama öncesi dokümandan çıkarılması önerilmiştir.

2.5.1.4. Makine öğrenmesi

Daha çok sistem yöneticisinin müdahalesini gerektiren, manuel işlemler içeren önceki yöntemler, çok verinin oluştuğu ve işlendiği ortamlarda pratik ve uygulanabilir olmamaktadır. Makine öğrenmesi yöntemleri eğitim sonrası hassas içeriklerin tespitinin otomatik yapılmasını sağlamaya çalışır. Bu yöntemde öncekilerden farklı olarak SVM, Naive Bayes gibi daha karmaşık yöntemler kullanılabilmektedir. En

önemli avantajları, kelime ya da ifade bazında bir kural tanımlamak yerine, eğitim kriterlerinin belirlenerek DLP sisteminin otomatik olarak kendisini eğitmesinin sağlanması ve çok daha detaylı ve yetenekli sınıflandırma algoritmaları kullanılabilmesidir. Sistem yöneticisinin eğitim aşamasında veri kümesini doğru seçmesi sistemin başarımı için en önemli aşamadır. Eğitim kümesindeki verilerin hassaslık seviyesinin etiketlenmesi de DLP yöneticisinin yapması gereken önemli bir adımdır. Eğitim kümesinin gerçek verileri doğru ve yeterli yansıtmaması sınıflandırma başarımını doğrudan etkilemektedir.

2.5.1.5. Doğal dil işleme

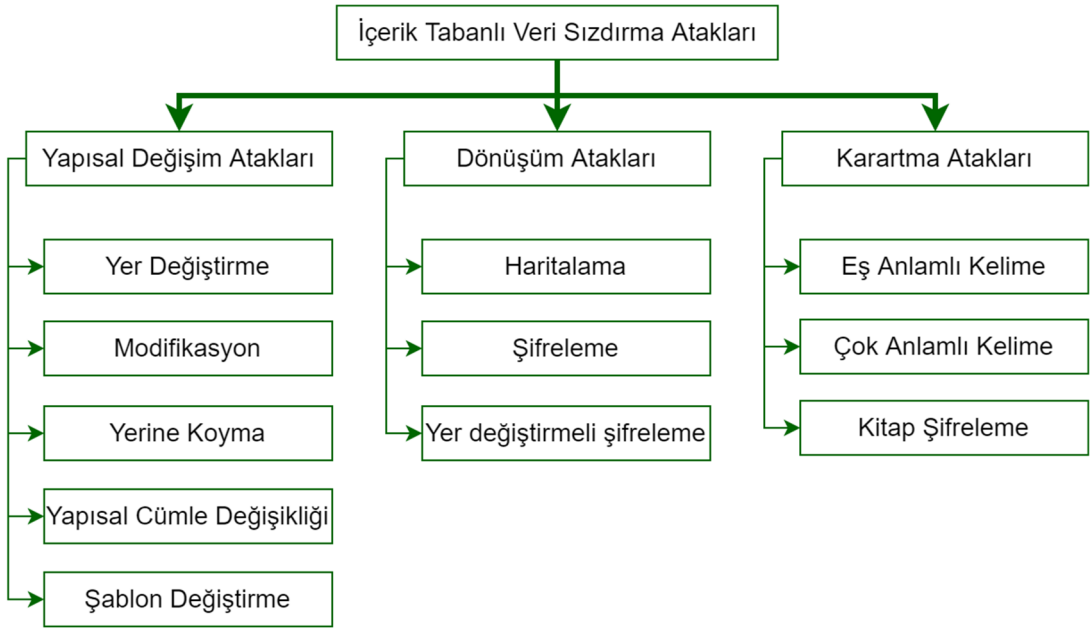
Doğal Dil İşleme (Natural Language Processing-NLP), insan dilini anlamak, yorumlamak ve işlemek için bilgisayar sistemlerinin kullandığı bir alandır. Metinlerin anlamlarını anlama, öznitelikleri çıkarma, model oluşturma ve sınıflandırma gibi işlemlerde NLP teknikleri yoğun olarak kullanılır. Metin madenciliği, kelime vektörleştirme, kelime öznitelikleri, dil modelleme ve derin öğrenme gibi NLP yöntemleri, içerik sınıflandırmada oldukça etkilidir.

NLP'nin sahip olduğu dilin karmaşıklığını ve doğal dil yapılarını anlama, metinden anlamsal çıkarım yapabilme gibi dil modelleme yöntemleri ile özellik çıkarımı yapıldıktan sonra diğer makine öğrenmesi yöntemleri kullanılarak sınıflandırma yapılır. Bu yönleriyle, kelimelerin eş anlamlıları ile değiştirilmesi, cümle ya da paragrafların değiştirilmesi, sayısal ifadelerin kelimelerle değiştirilmesi gibi işlemler neticesinde sınıflandırma başarımını korumada başarılı bir yöntemdir.

2.5.2. İçerik tabanlı veri sızdırma atakları

İçerik tabanlı veri sızdırma atakları, ilgili içeriğin DLP sistemini atlatacak şekilde değiştirerek DLP sisteminin içeriğin hassas olduğunu anlamasını engellemeye yönelik yöntemlerdir. (Mustafa, 2013) çalışmasında özellikle APT'ler tarafından bu yöntemlerin nasıl kullanıldığını değerlendirmiş ve bir sınıflandırma yapmıştır. (Canbay ve ark., 2017) çalışmasında ise bazı karakter bazlı ataklar gösterilmiştir. (Shapira ve ark., 2013) çalışmasında kelimeleri eş anlamlılarla değiştirme, cümlelerde kelimelerin yerini değiştirme, bazı kelimeleri kaldırma gibi cümle tabanlı ataklar sergilenmiştir.(Karlberger ve ark., 2007) çalışmasında ise kelimeleri yakın anlamlı değiştirme ve kelimeler üzerinde harf bazında değişiklikler yaparak karartma saldırıları gerçekleştirilmektedir.

Bu çalışmaların ortak değerlendirilmesi sonucunda içerik tabanlı ataklar Şekil 2.7'de gösterildiği şekilde sınıflandırılmıştır. İlerleyen bölümlerde bu atak türleri açıklanmaktadır.



Şekil 2.7. İçerik tabanlı veri sızdırma atakları sınıflandırması.

2.5.2.1. Yapısal değişim atakları

Yapısal değişim atakları, doküman içerisindeki kelimelerin veya cümlelerin yerlerinin değiştirilmesi, bazı kelimelerin kaldırılması türünden atakları kapsar.

Yer değiştirme (Transposition):

Bu atak ile doküman içerisindeki cümlelerinin, paragrafların ya da bölümlerin yerleri değiştirilir. Bu tür ataklar parmak izi çıkarma (fingerprinting) türünden tespit yöntemlerine karşı etkilidir.

Yapısal cümle değişimi:

Bu atakta doküman içinde seçilen cümleler anlamları aynı kalacak şekilde yapısal olarak değiştirilerek farklılaştırılır. Örnek olarak, "Bugün Ali Ankara'ya gidecek" cümlesi değiştirilerek "Ankara'ya bugün Ali gidecek" şeklinde anlamı bozulmayacak şekilde yeniden oluşturulmaktadır.

Şablon değiştirme:

Kurumsal sistemlerde dokümanların belli şablonlara bağlı kalınarak oluşturulması genel bir uygulamadır. Örneğin, bir Word dosyasının belli başlıkları, belli üst verileri içermesi istenebilir. Bu şablonlardaki bilgilere göre DLP sistemleri tanımlama

yapabilmektedir. Şablon değiştirme saldırılarında bu şablonlar değiştirilerek dosyanın farklı bir şablona sahip olduğu algısı oluşturulur.

Şablon kullanımına ilişkin DLP sistemlerinde yer alan bir diğer yaklaşım da belli kurallı ifadelerin (regular expressions) dikkate alınmasıdır. Bu yaklaşımda belli kurallı ifadelerin dokümanlarda var olmasına göre sınıflandırma yapılır. Bu tespit sistemlerine yönelik ataklarda içeriklerin bu şablonlara uygunluğu bozularak, mesela kimlik numarasına boşluklar, kesme işareti gibi karakterler eklenerek biçimi değiştirilir.

Yerine koyma:

Yerine koyma (substitution) atağı ile dokümandaki kelimeler ya da ifadeler farklı anlamlara gelecek şekilde değiştirilir. Böylece doküman anlamsal olarak da farklı bir sınıflandırmaya dahil edilmiş olur ve sızdırma gerçekleşebilir. Saldırgan dokümandaki olası kelimeleri öngörerek, kendisinin bildiği bir eşleştirme ile değişimi gerçekleştirir. Mesela "Bu sene şirket yüz milyar zarar etti" cümlesini, "Bu sene takım yüz gol yedi" şeklinde değiştirebilir.

2.5.2.2. Dönüşüm atakları (transformation)

Dönüşüm saldırılarında dokümanlar asıl biçimlerinden tamamen uzaklaşarak başka bir biçime dönüşmektedir. Çıktı, bir metin olabileceği gibi, sayısal ya da resim de olabilir. Şifreleme gibi yöntemler de bu kapsama girdiğinden en çok kullanılan veri kaçırma yöntemlerindendir.

Haritalama:

Haritalama (mapping) yöntemi ile verilen değeri başka bir değere "haritalayan" bir *f* fonksiyonu ile içerik ya da içeriğin parçaları farklı bir değere dönüştürülür. Bu sonuç değeri metinsel olabileceği gibi sayısal da olabilir. Burada önemli olan *f* fonksiyonunun tersi olan bir *f'* fonksiyonunun var olmasıdır. Böylece saldırgan veriyi elde ettikten sonra geri çevrim yapabilmektedir.

Şifreleme:

Genelde açık anahtarlı şifreleme yöntemi tercih edilerek yapılan bu yöntemde saldırganın açık anahtarı ile şifrelenerek DLP sisteminden kaçırılan içerikler, saldırganın kendisinde bulunan gizli anahtar ile şifresi çözülerek elde edilir. Genelde fidye yazılımları tarafından sıkça kullanılır. Bu tür ataklarda dosyaların çevrimiçi

kontrol edilmesi ve atağın gerçekleşmeden önlenmesi önemlidir; çünkü şifrelenmiş verinin normal şartlarda çözülmesi mümkün değildir. Bu sebeple tez kapsamında sunulan çözümde çevrimiçi çalışan bir içerik atak kontrol modülü önerilmiştir.

Yer değiştirmeli şifreleme:

Bu şifreleme yönteminde metindeki harfler alfabe içerisinde belirtilen sayıda ileri ya da gerideki harf ile değiştirilir. Sezar şifreleme yöntemine de benzeyen bu yöntemde örnek olarak İngilizce alfabesini esas alarak 10 karakter ilerletilmesi istenirse, "a" harfi "k" harfi ile, z harfi ise "j" harfi ile değiştirilir. Çok basit bir yöntem olsa da DLP sistemlerini atlatmada etkilidir.

2.5.2.3. Karartma atakları (obfuscation)

Bu ataklar ile kelimelerin ya da cümlelerin anlamları gizlenmeye çalışılır. Yöntem olarak DLP sisteminin kelimelere dayalı olan hassas içerik tespit algoritmaları hedef alınarak hassas içerikler hassas değilmiş gibi gösterilmeye çalışılır.

Eş anlamlı kelime (synonym):

Bu yöntemde kelimeler eş anlamlıları ile değiştirilir. İmza çıkartma, anahtar kelime ya da kurallı ifade eşleştirme türünden DLP yöntemlerini atlatmak için kullanılabilir.

Çok anlamlı kelime (polysemy):

Benzer şekilde çok anlamlı kelimelerin de sözlükteki karşılıklarından herhangi birisi seçilerek doküman içerisinde değiştirilir. Eş anlamlı kelime değiştirme yöntemine ek olarak kelime bazlı anlamsal analiz yapan DLP algoritmalarını atlatmak için kullanılabilir.

Kitap şifreleme (book cipher):

Bu yöntemde seçilmiş bir başka doküman anahtar olarak kullanılarak hedef dokümandaki kelimeler, bu anahtar metin kullanılarak sayılar ile değiştirilir. Anahtar metin içerisindeki kelimelerin metinde kaçıncı sırada geldiği dikkate alınarak bir şifreleme yapılabilir. Örneğin anahtar metin "İstiklal Marşı" olsun. Hedef metin ise "Dağları aştım tek başıma" olsun. İstiklal Marşı içerisinde "dağları" kelimesi 73., "aşarım" kelimesi 71., "tek" kelimesi 99., "başım" kelimesi 225. sırada geçmektedir. Böylece hedef metin "73 71 99 225" olarak değiştirilerek şifreleme gerçekleştirilmiş olur. Örnekte de görüldüğü üzere, Türkçe gibi sondan eklemeli dillerde değiştirme algoritmasının anahtar metindeki karşılıkları bulmak için gövdeleme gibi yetenekleri

olması gerekir ve anlam tam olarak korunamayabilir. Ancak İngilizce gibi dillerde daha başarılı şifreleme gerçekleştirilebilir.

3. APTON: APT KAYNAKLI VERİ SIZINTILARININ TESPİTİNİ SAĞLAMAK İÇİN YENİ BİR ONTOLOJİ

3.1. Motivasyon ve Ontolojinin Gereksinimleri

APT'ler tarafından gerçekleştirilen veri sızıntısı saldırıları, standart kötü amaçlı yazılımlara göre daha karmaşık yöntemler içerir. Saldırı, sisteme sızdıktan hemen sonra gerçekleştirilmek yerine zaman içinde yayılır. Ayrıca saldırı vektörleri (kötücül yürütülebilir dosyalar veya kütüphaneler) saldırganın komuta ve kontrol sistemleri tarafından güncellenebilir. Sisteme yapılan sızma hareketi bir bilgisayardan başlayıp asıl veri sızdırma işlemi, ağdaki başka bir makinede gerçekleşebilir. Bunların yanında, saldırganlar hedef sistemlerde mevcut olan programları veya özellikleri kullanabilirler. Uygulamaların ve işletim sisteminin bilinen ya da bilinmeyen zayıf noktalarından yararlanarak onları hedefleri için kullanabilirler (Singh ve ark., 2019).

APT'lerin veri sızdırma saldırılarının bu özellikleri, zararlı dosyaların algılanması için parmak izi çıkarma (fingerprinting) gibi seçeneklerin APT'lerin bütün işlemlerinin tespiti için kullanılması seçeneğini ortadan kaldırır. Literatürde zararlı yazılımların özelliklerinin çıkarılarak bunların bilinen kötü amaçlı yazılımlarla karşılaştırılması gibi statik analiz yöntemleri, saldırıları tespit etme noktasında belli bir başarı ortaya koymuştur (Bai ve ark., 2014; Kozachok & Kozachok, 2018; Schultz ve ark., 2001). Ancak APT'lerin anlatılan özellikleri sebebiyle bu çözüm tek başına yeterli değildir ve saldırıların tespiti için sistemin sürekli takip edilerek dinamik bir analiz gerçekleştirilmesi gerekir (Singh ve ark., 2019). Bu dinamik analiz işlemi neticesinde, sistemde gerçekleşen olaylar ve işlemler APT davranışları ile ilişkilendirmelidir. Ayrıca APT'lerin birden fazla yürütülebilir dosya kullanabilmesi ve sistemde var olan yazılım ve araçları kullanabilmeleri nedeniyle sistem genelinde bir analiz yapılarak birden çok çalıştırılabilir dosyanın aynı hedef için işlem yaptıklarının çıkarımının yapılması gereklidir.

Modern işletim sistemlerinde, sistem genelindeki işlemler genellikle olay kayıtlarında (event logs) kaydedilir ve bu olay kayıtları zararlı tespitinde kullanılmaktadır. Ancak bu kayıtlar, APT'lerin gerçekleştirebileceği tüm faaliyetlerin tamamını yakalama

konusunda yeterli değildir; çünkü sadece önceden tanımlanmış, belirli olayların tespit edilmesini sağlayabilirler. Bunun yanında, bu olay kayıtları Tehdit Algılama Sistemleri (Intrusion Detection Systems-IDS) tarafından sürekli takip edildiği ve onları tetikleyebilecek durumlar sabit olduğundan APT'ler bu tür olay günlüklerini oluşturacak işlemleri yapmamaya çalışırlar.

Zararlı ve APT tespitinde alternatif bir yaklaşım, işletim sistemi kütüphanelerinin uygulama programlama arayüzü (Application Programming Interface-API) çağrılarını kullanmayı içerir, genellikle "sistem çağrıları" olarak adlandırılır. Kullanılan araç ne olursa olsun, sistem üzerinde gerçekleştirilen her işlem, bir sistem çağrısı ile gerçekleşir. Bu nedenle, APT'ler sistem çağrılarını kullanmaktan kaçınamazlar. Ancak, sistem çağrıları düşük seviye işlemlerdir ve sadece sistem çağrısının adı ile bir saldırı tespit edilemez. Bununla birlikte, bir sistem çağrısının türü, kaynağı, kullanıcısı ve işlem gibi unsurlar APT'lerin taktik-teknik-prosedürleri (Tactic Technique Procedure-TTP) ile ilişkilendirilebiliyorsa, saldırının ilerleyişi belirlenebilir.

Bu nedenle, APT kaynaklı veri sızıntılarını tespite yönelik bir ontoloji aşağıdaki temel özelliklere sahip olmalıdır:

- Sistem çağrıları gibi düşük seviyeli olayların TTP tespit mekanizmalarına entegrasyonu sağlamalıdır.
- Sistem çağrıları, kaynaklar, işlemler ve kullanıcılar arasındaki anlamsal ilişkiyi oluşturmalıdır. Böylece bu unsurlar arasındaki ilişkiler aracılığıyla TTP'ler çıkarılabilir.
- Veri toplama, veri gizleme ve veri sızdırma ile ilişkili APT davranışlarına duyarlı olmalıdır.

Sonraki bölümde, bu özellikleri karşılayan ontolojinin bileşenlerini ve her bir bileşenin gerekliliği anlatılmaktadır. Sonrasında ise tez kapsamında geliştirilen ontoloji ve bileşenleri arasındaki ilişkiler detaylandırılmıştır.

3.2. Ontoloji İçerisindeki Kavramlar

Bu bölümde, öncelikle yukarıda bahsedilen ihtiyaçları karşılayacak olan ontolojinin sahip olduğu bileşenler (sınıflar) açıklanmaktadır. Sonrasında bu sınıflar arasındaki ilişkilendirmeler (object property) açıklanacak ve genel ontoloji sunulacaktır.

3.2.1. Sistem çağrısı (SystemCall)

Sistem çağrısı, işletim sistemindeki herhangi bir işlemin en basit öğesidir. Temelde işletim sistemi kütüphanesi içerisinde gerçekleştirilen bir API çağrısıdır. Bu, çekirdek düzeyinde (kernel-level) veya kullanıcı düzeyinde (user-level) olabilir. Her iki durumda da kullanılan araç veya yönteme bağlı olmadan, bir kaynağa yönelik bir işlem gerekiyorsa, kaynağa erişmenin tek yolu ilgili sistem çağrısıdır. Örneğin, Windows işletim sistemindeki bazı sistem çağrıları aşağıdaki gibidir:

- ReadFile, WriteFile, OpenFile, GetFileAttributesEx, WriteRegistry, ReadRegistry sistem çağrıları, dosya ya da kayıt defterine (registry) erişmek için açık bir isteği ortaya koymaktadır.
- GetAccountsInfo, GetNetworkConfigInfo, NetUserGetLocalGroups sistem çağrıları, sistemden veri toplama amacını göstermektedir.
- CreateToolhelp32Snapshot sistem çağrısı, bir prosesin başka bir prosesin hafıza alanını okuması yoluyla prosesler arası bir veri transferini belirtmektedir.
- CreateProcess yeni çocuk prosesler oluşturulmasını ve dolayısıyla bu prosesler ile bütün verilerin paylaşılabilmesini sağlamaktadır.
- WinHttpSendRequest, TransmitFile ağ üzerinde veri gönderme hedefini ortaya koymaktadır.

Bu örnekler, sistem çağrılarının bir prosesin davranışını anlamak için oldukça önemli olduklarını ortaya koymaktadır. Bu sebeple, sistem çağrıları esasında zararlı yazılım tespitinde yaygın olarak kullanılmaktadır (Canzanese ve ark., 2015; Han ve ark., 2021; Nissim ve ark., 2018; Prachi. ve ark., 2023). Bu sebeplerle, sunulan ontoloji modelinde sistem çağrısı en etkili bileşen olarak yer almaktadır ve ontoloji sistem çağrısının ontolojideki diğer bileşenlerle ilişkisini temel alarak çıkarım yapacak şekilde tasarlanmıştır.

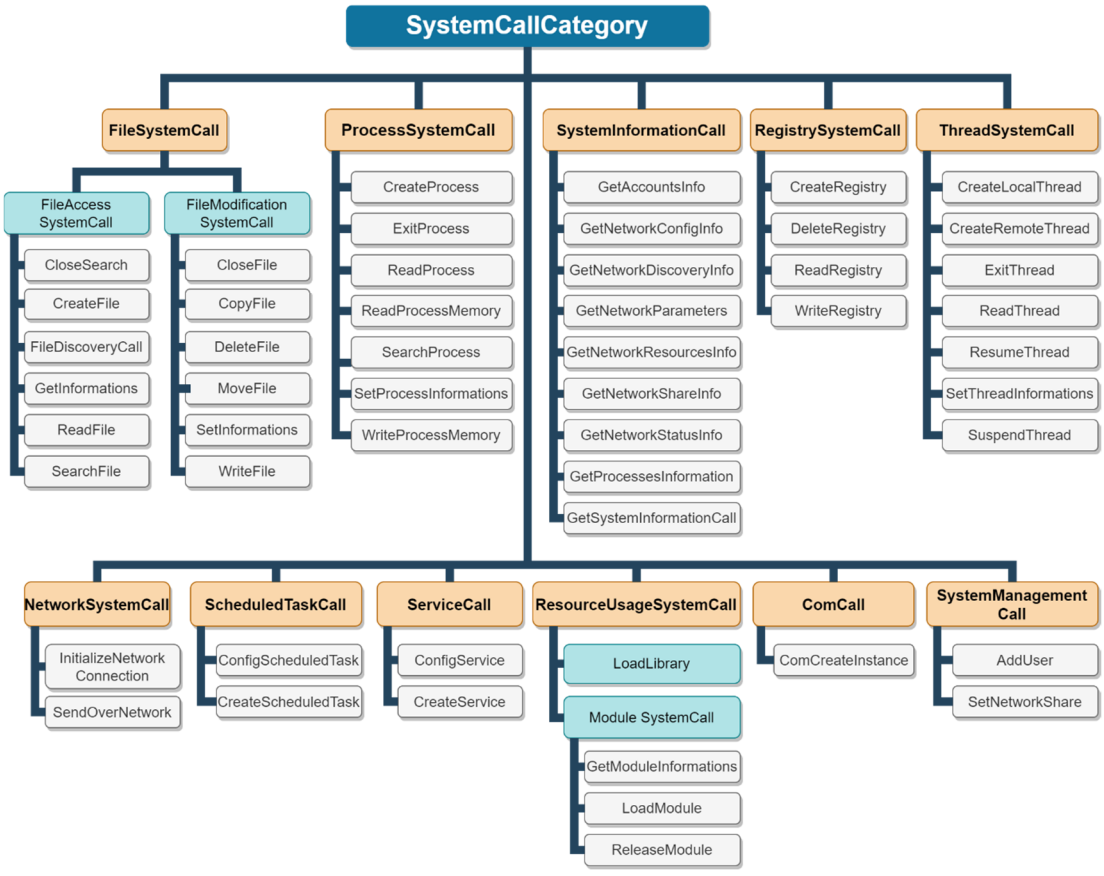
3.2.2. Sistem çağrısı kategorisi (SystemCallCategory)

Bir işletim sisteminde, anlamsal olarak benzer görevleri yerine getiren birçok API fonksiyonu bulunur. Örneğin, ReadFile ve OpenFile işlevlerinin her ikisi de bir dosya okuma işlemi gerçekleştirirken; ReplaceFile, WriteFile ve MoveFile fonksiyonları bir dosyaya yazma işlemi gerçekleştirir. Ayrıca, farklı özelliklere sahip olmakla beraber

aynı görevi yerine getiren işlevler de bulunmaktadır, örneğin CopyFile, CopyFileEx ve CopyFileTransacted fonksiyonları bu şekildedir.

İyi tasarlanmış bir ontolojide, bu tür sistem çağrıları aynı API kategorisinde temsil edilmelidir. Böylece, aynı özellikle olan ontoloji bireyleri aynı şekilde değerlendirilebilir. Aksi halde, her biri için ontolojide ayrı birer sınıf ve ilişki kurulması gerekecektir. Bu da ontolojinin hem çalışma performansını hem de yönetilebilirliğini azaltacaktır.

Bu nedenle, sunulan ontolojide anlamsal olarak benzer işlemler aynı olarak değerlendirilmiştir. Bu yaklaşım, ontolojideki SWRL kurallarının ve SPARQL sorgularının sayısını ve karmaşıklığını azaltmada da yardımcı olmaktadır. Esasında bu yaklaşım literatürdeki çalışmalarda da uygulanmıştır ve işlem davranışını tanıma konusundaki başarısı gösterilmiştir (Amer & Zelinka, 2020; Moon, Daesung ve ark., 2014). Bu bilgiler ışığında, SystemCallCategory sınıfı, Windows API ve yukarıda bahsedilen çalışmalardan elde edilen bilgilere dayanarak oluşturulmuştur Şekil 3.1'de gösterilen alt sınıflara ayrılmıştır.



Şekil 3.1. SystemCallCategory alt sınıfları.

3.2.3. Proses ve proses güvenlik seviyesi (Process, ProcessTrustLevel)

Yukarıda bahsedildiği üzere, bir sistem çağrısının özellikleri, bir prosesin davranışını belirlemede önemlidir. Ancak bu özellikler, onu çağıran proses bağlamında düşünülmelidir. Aksi takdirde yanlış tespitlerin oluşması (false-positive) kaçınılmazdır. Örneğin, ReadFile fonksiyonu bir dosyayı okumak için kullanılır. Eğer ReadFile, Windows'taki Notepad uygulaması tarafından kullanılıyorsa, bu olay kendisi başlı başına bir güvenlik sorunu olmayabilir. Ancak APT'ler kendi uygulamaları içerisinden Notepad gibi zararsız prosesleri oluşturabilir ve bunlara kötücül kodlarını enjekte edebilirler. Yani bir Notepad uygulaması güvenilir olmayan bir proses tarafından oluşturulursa, bu işlem ata prosesin Notepad prosesi üzerinde tam kontrol sahibi olmasını sağlar. Sonuç olarak, Notepad işleminin ReadFile sistem çağrısı, düzenli bir dosya okuma işlemi yerine bir APT saldırısının bir parçası olabilir.

Bununla beraber, gizli kalma çabaları nedeniyle, APT'ler genellikle saldırılarını işletim sisteminde zaten mevcut olan programlar, komut dosyaları veya araçlar kullanarak gerçekleştirme eğilimindedir. Ancak, saldırının ilk aşamalarında özellikle özel yazılım veya değiştirilmiş (compromised) bir dosya kullanmaları gerekebilir (Shabtai ve ark., 2012). Bu nedenle, sistemdeki her proses için bir "Proses Güvenilirlik Seviyesi" özelliği olması gerekmektedir. Bunun oluşturulması iki aşamalı olarak sağlanabilir:

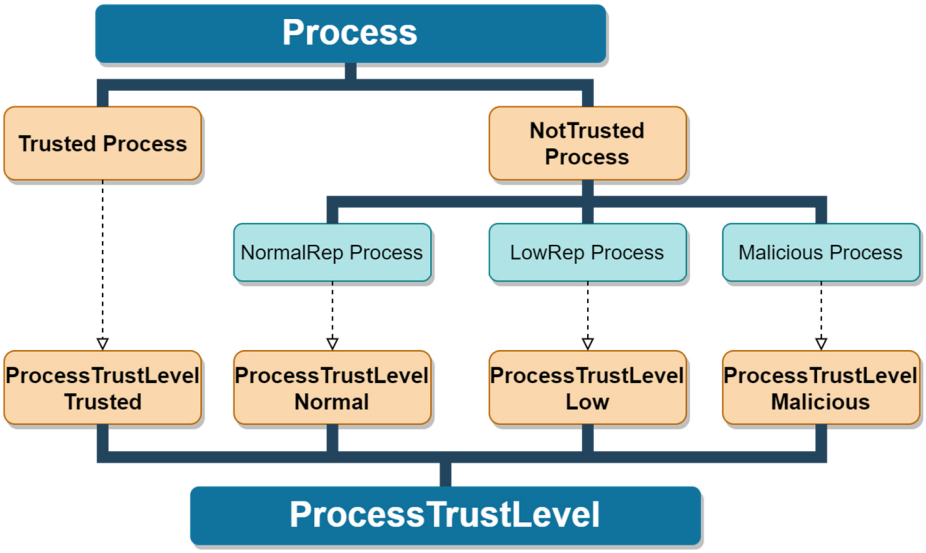
1. Literatürde yaygınca kullanılan bir yöntem olarak, Çalıştırılabilir dosyaların statik analizi kullanılarak ilk skor oluşturulabilir (Garg & Yadav, 2019; S. Gupta ve ark., 2016; Hassen ve ark., 2017)
2. Çocuk prosesler, ata proseslerinin skorlarını devralırlar. Yani ata prosesin skor çocuk prosesin skorundan düşükse, çocuk prosesin skoru da düşürülür. Bir diğer ifadeyle, bir çocuk proses asla ebeveyninden daha yüksek bir güvenilirlik seviyesine sahip olamaz, ancak çocuk prosesin çalıştırılabilir dosya güvenilirlik seviyesine bağlı olarak daha düşük bir skoru olabilir.

Proses skorunun oluşturulması, 5.3.2 bölümünde, Proses Sınıflayıcı modülü içerisinde detaylandırılmıştır. Bahsedilen yönteme uygun olarak ontoloji içerisinde oluşturulan Process ve ProcessTrustLevel sınıflarına ait alt sınıflar Şekil 3.2'de gösterilmiştir. Bu resimde, Process sınıfındaki bireylerin sahip oldukları ProcessTrustLevel seviyesine göre ayrışmaları belirtilmiştir. Örneğin, ProcessTrustLevelTrusted tipinde bir

ProcessTrustLevel'a sahip olan Prosess bireyi, TrustedProcess olarak belirlenmektedir.

3.2.4. İçerik, içerik gizlilik seviyesi ve veri konumu (Content, ContentPrivacyLevel, DataLocation)

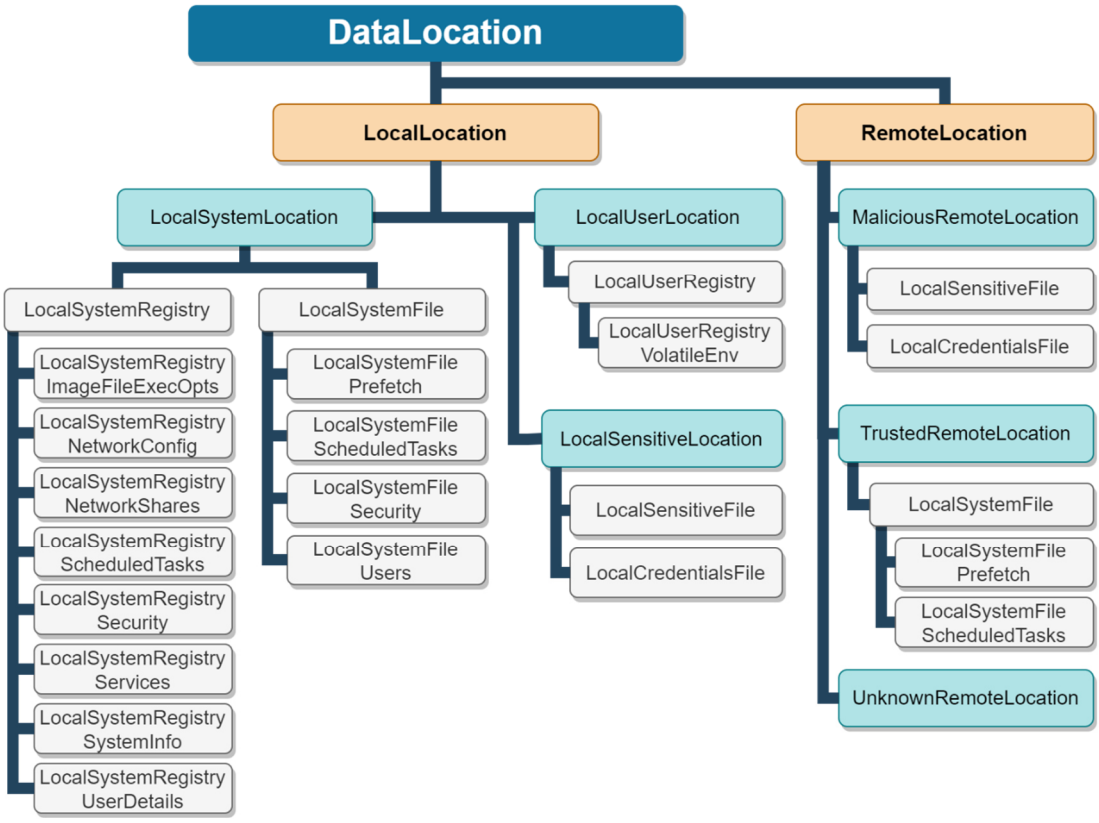
Proseslerin eriştiği içeriklerin incelenmesi bir DPL sisteminde çalışan analiz süreçlerinin vazgeçilmez bir parçasıdır. ReadFile, WriteRegistry, TransmitFile gibi sistem çağrıları, bir içerik kaynağı veya hedefine sahiptir. İçerik, diskte yer alan bir dosya, bir kayıt defteri anahtarı veya ağdan alınan bazı veriler olabilir. Bu durumda içeriğin değerlendirilebilmesi için içeriğin iki özelliğine dikkate alınmalıdır:



Şekil 3.2. Process ve ProcessTrustLevel alt sınıfları.

1. İçeriğin konumu (DataLocation), prosesin davranışını ve içerik akışının yönünü ortaya koyduğu için önemlidir. Bu bir dosya, yerel bir makinedeki bir kayıt defteri anahtarı veya uzak bir bilgisayardaki bir konum olabilir. 3.4.2 APT teknik tespit kuralları bölümünde detaylandırılacağı üzere, bazı APT teknikleri belli başlı konumlardaki verilerle ilgilenmektedir. Örneğin, Windows işletim sisteminde Kimlik Bilgilerinin Sızdırılması (T1003) APT tekniği, Kayıt Defteri'nin HKLM\SECURITY veya HKLM\SAM bölümüne erişim gerektirir. Bu durumda, kayıt defteri okuma işleminin yeri, tespitte önemli bir faktör haline gelir. Bu sebeple, sunulan ontolojide sistemdeki önemli konumlar, uygulanan APT tekniğini belirlemek için sınıflandırılmıştır. Buna göre oluşan, DataLocation sınıfının alt sınıfları Şekil 3.3'te gösterilmiştir.

2. İçeriğin gizlilik seviyesi (ContentPrivacyLevel), kaçırılmak istenen içeriği belirlemede son derece önemlidir. Zaten DLP sistemlerinin genel amacı, önemli olarak belirlenmiş verilerin korunmasıdır. Bu sebeple önerilen DLP sisteminde bu yönde bir değerlendirme olması ve bunun ontoloji içerisinde yer alması doğal bir sonuçtur. İçeriğin sınıflandırmasında, tez kapsamında zararlı yazılımların veri kaçırma saldırılarına karşı dayanıklı bir algoritma geliştirilmiştir. Buna ilişkin detaylar 4. Bölümde detaylandırılmıştır. Ontoloji kapsamında vurgulanması gereken, tez kapsamında içeriklerin Kurumsal Gizli (G), Hizmete Özel (Ö) veya Tasnif Dışı (TD) olarak 3 ayrı sınıfta değerlendirilerek ele alındığı ve ontolojide de bu şekilde alt sınıflara sahip olmasıdır. 3.4.2 APT teknik tespit kuralları bölümünde kurumsal sınıfa dahil olan veriler üzerinde yapılan işlemlerdin özel olarak değerlendirildiği detaylandırılacaktır.



Şekil 3.3. DataLocation alt sınıfları.

3.2.5. İçerik tabanlı saldırı (ContentAttack)

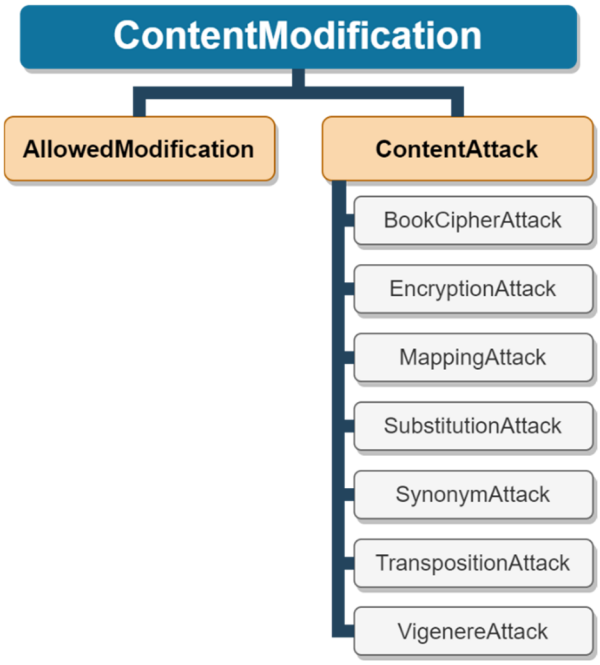
Başta da belirtildiği üzere, APT'lerin önemli verileri dışarı çıkarma amaçları oldukça belirgindir (Lemay ve ark., 2018). Bununla beraber, DLP sistemleri geliştikçe, kriptografik yöntemleri kullanmayan ve bellek içi değişiklikleri kullanan içerik tabanlı

kaçınma saldırıları popülerlik kazanmaktadır. Literatürdeki çalışmalar, bu tür saldırıların bilinen sınıflandırma algoritmalarını atlatmada başarılı olduğunu göstermektedir (Blasco ve ark., 2012; Matasano, 2007; Mustafa, 2013). Bu saldırılar, genellikle özel bir yazılım, kütüphane veya kritik sistem çağrıları (örneğin CryptEncrypt fonksiyonu) kullanmadan içeriği değiştiren saldırılar olmaktadır.

Bu saldırılar kriptografik sistem çağrılarını ve kütüphanelerini kullanmadıkları için, bunlara bağlı oluşabilecek olası uyarıları ve kayıtları tetiklemezler. Bunun yanında, bu ataklar sonucunda oluşan içerikler, şifrelenmiş veriler gibi değildir. Kelime, cümle, harf değişimleri gibi yöntemler kullandıkları için bir DLP sistemi için normal metinlerden ayrıştırılması güçtür. Ayrıca şifrelenmiş ya da sıkıştırılmış verilerin tespitinde kullanılan entropi seviyesi gibi parametreler açısından da şüpheli bir değere sahip değillerdir. Örneğin, Vigenère şifresi, herhangi bir kütüphane kullanmaya gerek kalmadan, uygulama içerisinde metin üzerinde yapılabilecek basit işlemlerle elde edilebilir. Bu basitliği ile beraber orijinal metni bir DLP sisteminin tanımasını olanaksız kılacak şekilde değiştirebilir.

Buna yönelik olarak tez kapsamında sunulan çözüm, prosesleri sürekli takip ederek okudukları ve yazdıkları verileri karşılaştırmaktır. Proseslerin verilere erişimlerin takip edilmesi halihazırda birçok ticari DLP çözümü tarafından uygulanmaktadır (McAfee, 2022; Symantec, 2021). İçerik Atak Tespiti yönteminin tez kapsamında nasıl gerçeklendiği 4.3 bölümünde detaylandırılmıştır.

Ontoloji içerisinde, bu özellik ContentModification ve alt sınıfları ile temsil edilmektedir. Her bir SystemCall nesnesi bir ContentModification nesnesi ile ilişkilendirilmektedir. Saldırı tespit edilmediği durumda AllowedModification, saldırı tespiti durumunda ise ne tür saldırı yapıldığını belirten ContentAttack'ın alt sınıflarından birisi yaratılmaktadır. Bu alt sınıflar Şekil 3.4'te gösterilmiştir.



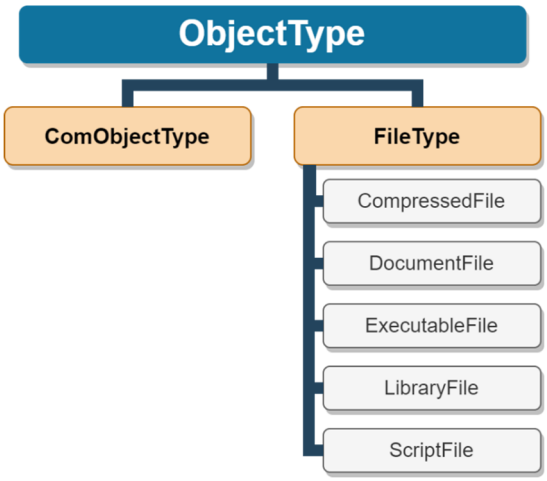
Şekil 3.4. ContentModification alt sınıfları.

3.2.6. Nesne tipi (ObjectType)

Sistem çağrıları farklı işlemleri gerçekleştirmek için kullanılmaktadır. Yeni prosesler oluşturabilir, belirli kütüphaneleri yükleyebilir veya bazı komut dosyalarına erişerek bunları çalıştırabilirler. Bu tür davranışların ayırt edilmesi, prosesin niyetini belirlemede önemlidir. SystemCallCategory içerisinde bu ayrım yapılmaktadır. Ancak, hangi kaynaklara eriştikleri de davranışın amacını belirlemek için gereklidir. Örneğin, T1016 tekniği (System Network Configuration Discovery), ipconfig, arp, route gibi sistemde var olan araçları kullanır. Bir sistem çağrısı bu komutları kullanarak yeni bir proses oluşturmak istediğinde bu durum değerlendirilmelidir. Çünkü bu davranış, ontolojideki prosesin güvenirlik seviyesi gibi diğer kavramlarla ilişkilendirildiğinde APT kaynaklı bir T1016 tekniğinin tespitini sağlayabilir. Başka bir örnek olarak, Uzak Masaüstü Protokolü (T1076) tekniği, tscon.exe veya mstsc.exe gibi sistemdeki belirli yerleşik uygulamalara erişim gerektirir. Güvenilmeyen bir hedef işlem, bu uygulamaları parametre olarak kullanarak CreateProcess çağrısı ile bir alt işlem oluşturmaya çalışıyorsa, bu bir APT Tekniği olarak kabul edilebilir

Benzer şekilde, APT'ler sistemde bazı iyi bilinen kitaplıkları (DLL) kullanabilirler; kötü amaçlı kod parçalarını çalıştırmak için PowerShell dosyaları gibi komut dosyalarını kullanabilirler; ya da önemli Component Object Model (COM) nesnelerini sistemden önemli bilgiler toplamak için kullanabilirler. Bu tür durumlarda, hedef nesne/dosya türü dikkate alınarak değerlendirme yapılması gerekmektedir. Bu sebeple

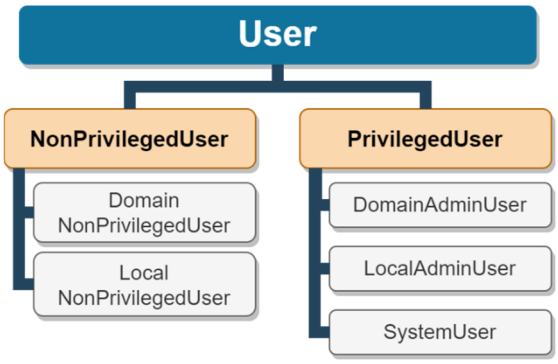
sunulan ontoloji içerisinde bu bilgi ObjectType sınıfı ile karar verme sürecine entegre edilmiştir. Dosya tiplerine göre değerlendirme yapılabilmesi için ObjectType sınıfı Şekil 3.5'te gösterilen alt sınıflara ayrılmıştır.



Şekil 3.5. ObjectType alt sınıfları.

3.2.7. Kullanıcı türü (User)

Yapılan işlemin hangi kullanıcı hesabı ile gerçekleştiği bilgisi, yalnızca eylemden sorumlu kullanıcı hesabını belirlemekle kalmaz, aynı zamanda mevcut işlemin bir APT tekniğinin parçası olup olmadığını tespit etmek için de gereklidir. Örneğin, T1015 tekniği (Accessibility Features) içinde, "System" isimli kullanıcı (Windows'ta) erişilebilirlikle ilgili komutları çalıştırmak için kullanılır (MITRE, 2019a). Bu tür senaryolarda, prosesi çalıştıran kullanıcı hesabının türü, yapılan işlemin kötü amaçlı olup olmadığını belirlemede önemli bir ipucu sağlayabilir. Bu sebeple, kullanıcı hesabı türünün ontolojiye entegre edilmesi önemlidir. Tez kapsamında kurumsal bir sistem ve APT'lerin saldırı tipleri dikkate alınarak, User sınıfı Şekil 3.6'da gösterilen alt sınıflara ayrılmıştır. Bu noktada kullanıcıların erişim yetkileri ana ayrım kriterini oluşturmuştur.



Şekil 3.6. User alt sınıfları.

3.2.8. APT teknik, taktik ve riski (AptTechnique, AptTactic, AptRisk)

Önerilen ontoloji modelinde, MITRE ATT&CK (MITRE, 2020a) içerisinde yer alan APT Teknikleri ve Taktikleri AptTechnique ve AptTactic sınıfları ile birebir eşleşecek şekilde tanımlanmıştır. Bu yaklaşımla sistem tarafından tespit edilen APT davranışları, geniş kabul ve kullanım alanı olan MITRE sınıflandırmasıyla doğrudan uyum sağlamaktadır.

Bununla beraber, çeşitli APT saldırı analiz raporları (FireEye, 2019, 2015; Mandiant Intelligence Center, 2013; PwC, 2017; Tok & CeliKtas, 2019) ve APT3 Saldırı Planı (Korban ve ark., 2017), belirli teknikler bir araya geldiğinde APT'lerin veri sızdırma riskinin arttığını ortaya koymaktadır. Bu çalışmalar, aynı zamanda APT'lerin veri sızdırmadan önce keşif ve veri toplama gibi belirli adımları izlemesi gerektiğini de göstermektedir. Bu nedenle, sistemdeki APT kaynaklı riskin belirlenerek veri kaçırma durumuna ne kadar yaklaşıldığını ifade etmek aksiyon oluşturulması noktasında oldukça faydalı olacaktır. Örneğin, Keşif (Discovery) ve Yatay Hareket (Lateral Movement) Taktiklerinin bir araya gelmesi sistemde yayılmakta olan bir APT riskini gösterirken, buna Veri Toplama (Data Collection) taktiğinin eklenmesi ile APT'nin yayılma aşamasını tamamlayarak veri toplama aşamasına geçtiğini belirtir. Böyle bir risk tanımı, özellikle sistem içinde koruyucu önlemler uygulanırken gerekli olmaktadır. Örneğin risk seviyesi arttıkça, internet iletişimini kısıtlamak, verilere erişimi kontrol etmek ve kullanıcı izinlerini yöneterek saldırının ilerlemesini durdurmak ve bu yolla veri sızdırma çabasını önlemek sağlanabilir.

Bu sebeplerle, ontoloji içerisinde MITRE ATT&CK Taktiklerini ve Tekniklerini kullanarak bunları risk tanımlaması olarak temsil eden AptRisk sınıfı eklenmiştir. Bu risk tanımlamaları, Tablo 3.1'de gösterildiği gibi seviyelendirilmiştir. Bu APT risk seviyeleri ve karşılığı olan taktik ve teknikler, bahsi geçen analiz raporları, APT3 saldırı planı ve bu yöntemlerin açık kaynaklı uygulamalarına dayalı olarak geliştirilmiştir. Bu çalışmalarda, APT'lerin genellikle daha az riskli bir bilgisayara başlayarak buradan yayılmaya eğilimli olduğunu göstermektedir. Bu ilk bilgisayar üzerinde belirli keşif adımlarını gerçekleştirdikten sonra işlemlerine devam etmek için kalıcılık elde etmeye çalışmaktadırlar. Sonraki adımda ise daha değerli hedeflere yayılma çabası gösterirler. Yayıldıkça hedefteki değerli verileri belirlemek için veri toplama teknikleri uygularlar. Hedefteki verilerin değerli olduğunu belirlediklerinde, veri sızdırma adımlarına devam ederler. Hedefte bir DLP sistemi varsa, onu atlatmak

için içerik saldırı yöntemleri kullanabilirler. Ontoloji içerisinde bu aşamalar AptRisk'in alt sınıflarında yansıtılmıştır.

3.2.9. Bilgisayar (Host)

Ontoloji içerisindeki bilgisayar (host) sınıfı, bir APT saldırısı tarafından etkilenen düğüm(leri) belirtir. Bu sınıf, saldırının akışını ve saldırıya uğrayan belgelerin konumunu temsil etmek için yardımcı olmaktadır. Her bir sistem çağrısı, proses ve APT tekniği, taktiği, riski ve içerik bir host öğesi ile ilişkilidir.

Tablo 3.1. AptRisk altsınıfları ve karşılık gelen APT teknik ve taktikleri.

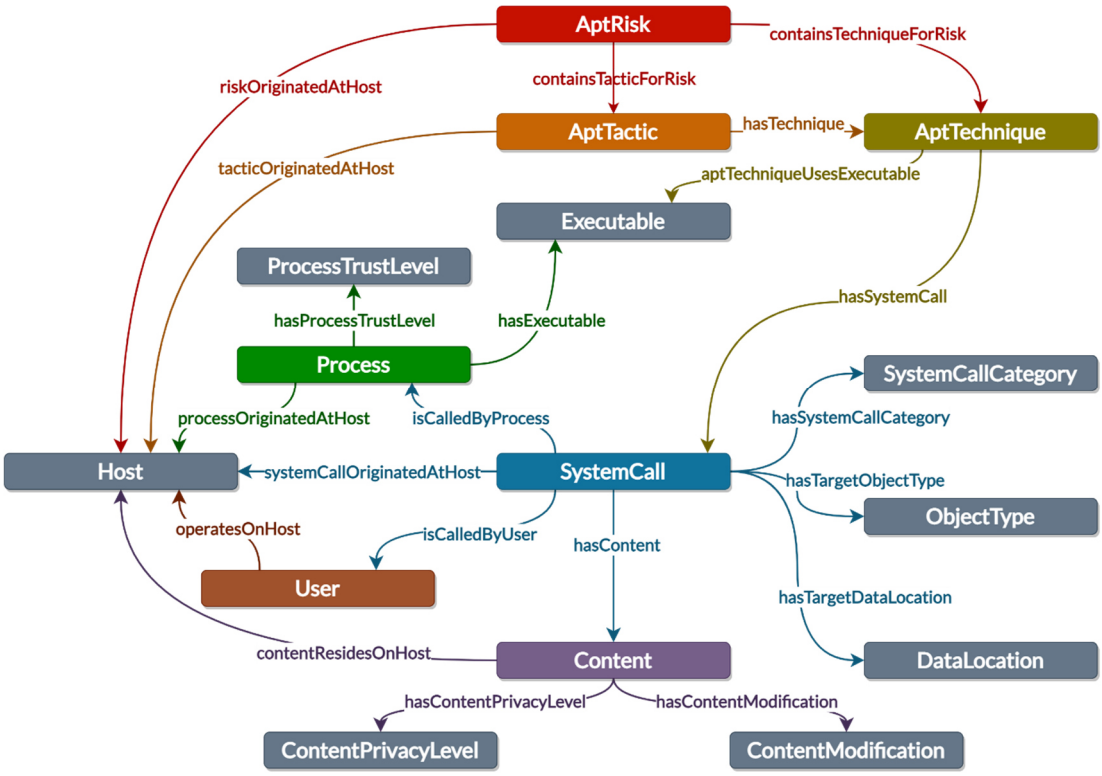
APT Risk Sınıfı	Kullanılan APT Taktik ve Teknikleri
İçerik-tabanlı atak içeren APT Veri Sızdırma Riski (APT Data Exfiltration Using Content Attack Risk)	Discovery, Lateral Movement, Collection, Exfiltration, Content Attack
APT Veri Sızdırma Riski (APT Data Exfiltration Risk)	Discovery, Lateral Movement, Collection, Exfiltration
APT Veri Toplama Riski (APT Data Collection Risk)	Discovery, Lateral Movement, Collection
Bulaşıcı APT Riski (Contagious APT Risk)	Discovery, Lateral Movement
Kalıcı APT Riski (Persistent APT Risk)	Discovery, Persistence
Muhtemel APT Riski (Possible APT Risk)	Discovery

3.3. Ontoloji İçerisindeki Kavramlar Arasındaki İlişkiler

Ontolojinin salt bir topoloji olmanın ötesinde, bir karar verme aracı olmasını sağlayan en önemli özelliği bileşenler arasındaki anlamsal ya da yapısal ilişkilerin tanımlanabilmesidir. Bu ilişki tanımları sayesinde ileriki bölümde detaylandırılan APT teknik tespit kuralları çalışabilmektedir. Bu bölümde, ontoloji terminolojisinde "nesne özellikleri (object property)" olarak ifade edilen bu ilişkiler anlatılacaktır.

Sunulan ontolojinin genel görünümü, Şekil 3.7'de sergilenmiştir. Bu resimdeki ilişkilere ait açıklamalar Tablo 3.2'de verilmiştir. Ontolojideki kavramlar tanımlanırken de belirtildiği üzere, APT Tekniklerinin çıkarılmasında kullanılan en temel kavram, SystemCall sınıfıdır. SystemCall sınıfının diğer sınıflarla ilişkileri, sistemdeki proseslerin davranışlarını ve dolayısıyla APT tekniklerinin varlığını belirlemek belirleyici olmaktadır. Bu sebeple teknik tespit kuralları bölümünde

detaylandırılacağı üzere, sistem çağrısının sahip olduğu ilişkiler belli kriterleri sağlıyorsa bu durumda bir APT tekniği olduğu tespit edilmektedir.



Şekil 3.7. Ontoloji kavramları arasındaki ilişkiler.

3.4. Ontoloji Tespit Kuralları

Ontoloji tespit kuralları, "Semantic Web Rule Language (SWRL)" dilinde yazılmış, ontolojideki bileşenlerin arasındaki ilişkilerin belli durumda olup olmadığını kontrol ederek daha önce ontolojide tanımlanmamış bilgilerin çıkarımını sağlayan kural tanımlarıdır. İlerleyen bölümlerde örnekler ile daha detaylı açıklanmaktadır.

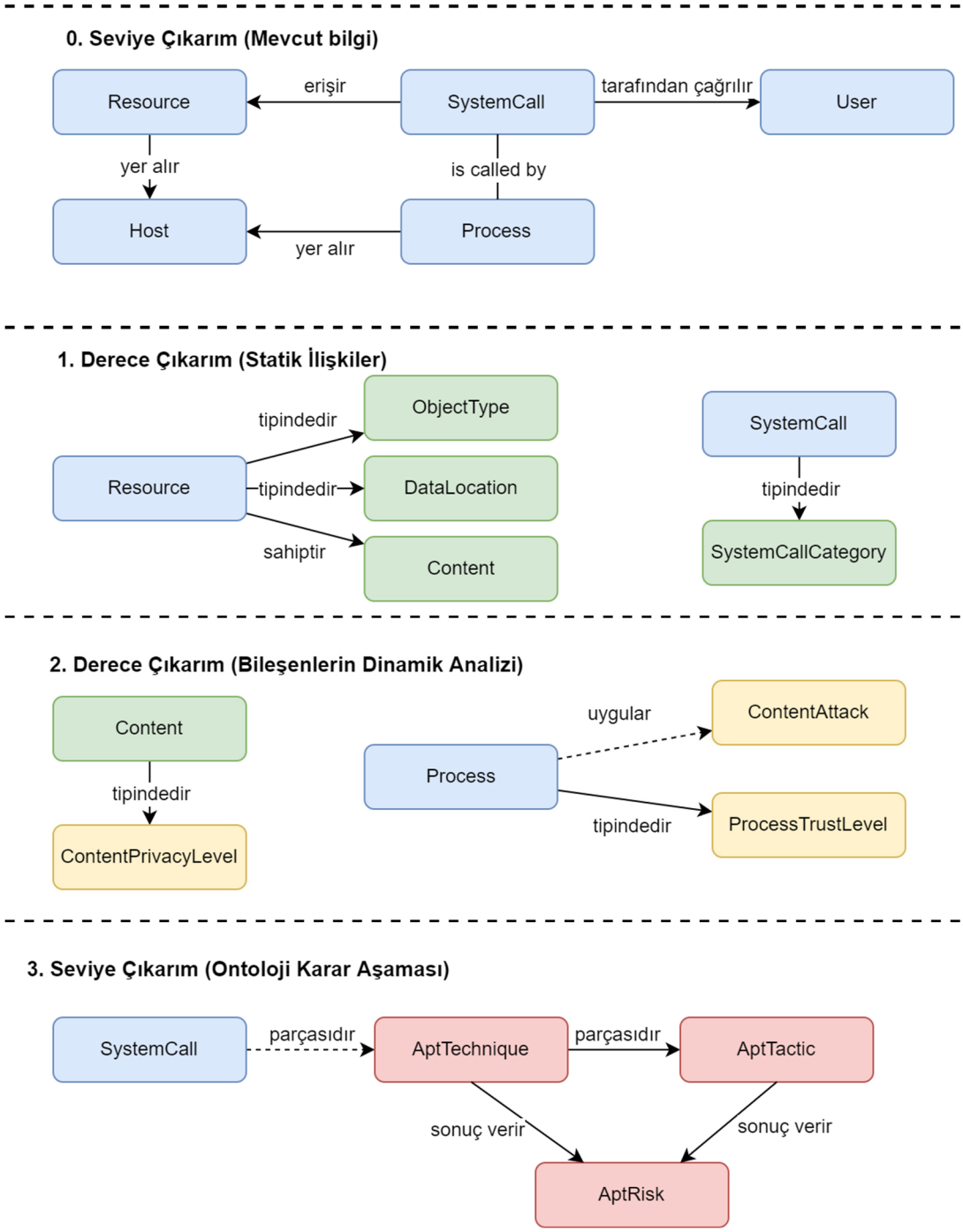
3.4.1. Sistem çağrılarından APT risk çıkarımı akışı

Teknik ve risk tespit kuralları detaylandırılmadan önce bu bölümde sistem çağrılarından APT risk çıkarımı akışı genel olarak anlatılmaktadır.

Ontolojinin çıkarım yetenekleri sayesinde sistem çağrısının temel öğeleri analiz edilerek ve MITRE matrisindeki APT teknik ve taktikleri ile ilişkilendirilmektedir. Bu çıkarım işlemi, Şekil 3.8'de gösterildiği gibi birkaç adım gerektirir.

Tablo 3.2. Ontoloji içerisindeki nesne özellikleri.

Nesne Özelliği	Alan (Domain)	Sınır (Range)	Açıklama
containsTacticForRisk	AptRisk	AptTactic	APT riskinin belirtilen APT taktiğini içerdiğini gösterir.
containsTechniqueForRisk	AptRisk	AptTechnique	APT riskinin belirtilen APT tekniğini içerdiğini gösterir.
hasTechnique	AptTactic	AptTechnique	APT taktiğinin belirtilen APT tekniğini içerdiğini gösterir.
hasSystemCall	AptTechnique	SystemCall	APT tekniğinin belirtilen sistem çağrısını kullandığını gösterir.
isUsedByTactic	AptTechnique	AptTactic	APT tekniğinin belirtilen APT taktiği tarafından kullanıldığını gösterir.
contentCreatedAtHost	Content	Host	İçerik belirtilen bilgisayarda oluşturulmuştur.
hasContentModification	Content	Content Modification	İçerik üzerinde ne tür bir içerik değiştirme yapıldığını belirtir.
hasContentPrivacyLevel	Content	Content PrivacyLevel	İçeriğin gizlilik seviyesini belirtir.
isUtilizedByAptTechnique	Executable	AptTechnique	Çalıştırılabilir dosyanın belirtilen APT tekniğin tarafından kullanıldığını gösterir.
systemCallOriginatedAtHost	SystemCall	Host	Sistem çağrısının belirtilen bilgisayarda çalıştırıldığını gösterir.
processOriginatedAtHost	Process	Host	Prosesin belirtilen bilgisayarda çalıştırıldığını gösterir.
techniqueOriginatedAtHost	AptTechnique	Host	APT tekniğinin belirtilen bilgisayarda ortaya çıktığını gösterir.
tacticOriginatedAtHost	AptTactic	Host	APT taktiğinin belirtilen bilgisayarda ortaya çıktığını gösterir.
riskOriginatedAtHost	AptRisk	Host	APT Riski'nin belirtilen bilgisayarlarda ortaya çıktığını gösterir.
hasExecutable	Process	Executable	Proses'in belirtilen çalıştırılabilir dosya kullanılarak oluşturulduğunu belirtir.
hasProcessTrustLevel	Process	ProcessTrustLevel	Proses'in belirtilen güvenilirlik seviyesine sahip olduğunu gösterir.
hasContent	SystemCall	Content	Sistem çağrısının belirtilen içeriğe erişim sağladığını belirtir.
hasSystemCallCategory	SystemCall	SystemCall Category	Sistem çağrısının belirtilen kategoride olduğunu gösterir.
hasTargetDataLocation	SystemCall	DataLocation	Sistem çağrısının belirtilen konum sınıfındaki kaynaklara erişim sağladığını gösterir.
hasTargetObjectType	SystemCall	ObjectType	Sistem çağrısının belirtilen tipteki kaynaklara erişim sağladığını gösterir.
isCalledByProcess	SystemCall	Process	Sistem çağrısının belirtilen proses tarafından çağrıldığını gösterir.
isCalledByUser	SystemCall	User	Sistem çağrısının belirtilen kullanıcı tarafından çağrıldığını gösterir.



Şekil 3.8. Sistem çağrısından APT risk tespitinde çıkarım adımları.

0. seviyedeki çıkarım, esasında bir çıkarımdan öte verilen bilgilerin ontoloji diline aktarılmasıdır. Sistem çağrısının eriştiği veya değiştirdiği kaynak, sistem çağrısını yürüten işlem, işlemi yürüten kullanıcı ve işlemin bulunduğu ana bilgisayara ait bilgiler, bir sistem çağrısı yakalandığı anda doğrudan bilinmektedir.

1. seviye çıkarım, ilk katmanda çıkarılan özelliklerin bilinen sınıflarla statik eşlenmesini ve sistem çağrısının eriştiği içerik verisinin çıkarılmasını içerir. Bu

adımda sistem çağrısı, bir sistem çağrısı kategorisine sınıflandırılır. Benzer şekilde, kaynak, kaynağın konumuna ve özelliklerine bağlı olarak bilinen nesne türüne eşlenir. Kaynağın içeriği de bu adımda çıkarılır.

2. seviyedeki çıkarım, sistem çağrısının öğelerinin dinamik bir analizini gerektirir. Bu, içeriğin gizlilik seviyesini belirleme, işlemin güven düzeyini belirleme ve içerik tabanlı bir saldırının gerçekleştirilip gerçekleştirilmediğini belirleme gibi unsurları içerir. Bu işlemler için makine öğrenmesi ya da kural tabanlı algoritmaların çalıştırılarak ilgili bileşenlerin incelenmesi gerekmektedir.

Bu analiz adımları tamamlandığında, 3. Seviyede tüm sonuçlar toplanır ve sistem çağrısının bir APT saldırısının bir parçası olup olmadığını belirlemek için ontoloji tabanlı bir çıkarım çalıştırılır. APT Teknikleri ve Taktikleri tanımlandığında, sistemdeki APT risk seviyesi belirlenmiş olur.

3.4.2. APT teknikleri tespit kuralları

Sunulan ontolojide, APT tekniklerinin varlığı, SWRL kuralları kullanılarak çıkarılır. Bu kurallar, SystemCall ve ilgili nesneler arasındaki ilişkileri içerir. Bu kriterler karşılandığında, belirtilen APT tekniğinin sistemin içinde var olduğu sonucuna varılır.

Bu kuralları çalıştırmak için sistemde geçici olarak en genel sınıf olan owl:Thing sınıfına sahip olan bir birey yaratılır. Bu birey hasSystemCall nesne özelliği ile ilgili sistem çağrısı ile ilişkilendirilir. Daha sonra SWRL kuralları ile sistem çağrısını ve diğer bireylerin ilişkilerinin belli bir durumda olduğu tespit edilince bu bireyin sahip olduğu esas sınıfın ne olduğu çıkarım yapılır.

Örnek olarak, sunulan ontolojide içerik tabanlı saldırılar, T1022, Veri Şifreleme tekniği (Data Encrypted), ile değerlendirilmektedir. T1022'yi tespit etmek için kullanılan SWRL kuralı aşağıdaki gibidir:

$$owl{:}Thing(?\,aptTech), NotTrustedProcess(?\,p),$$
$$Content(?\,content), ContentAttack(?\,contentMod),$$
$$WriteFile(?\,sysCallCategory),$$
$$hasSystemCall(?\,aptTech, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$$
$$hasSystemCallCategory(?\,sysCall, ?\,sysCallCategory),$$
$$hasContent(?\,sysCall, ?\,content),$$
$$hasContentModification(?\,content, ?\,contentMod)$$
$$\rightarrow DataEncrypted(?\,aptTech)$$

Bu kural, belirli bir AptTechnique bireyinin DataEncrypted türünde olup olmadığını belirlemek için çıkarım motoru tarafından kullanılır. Kural şu şekilde çözümlenebilir:

Tespiti yapılmış, bilinen bireyler ve tipleri:

- Thing türünde, esas sınıfı bilinmeyen bir birey (muhtemel AptTechnique türünde).
- NotTrustedProcess türünde bir birey.
- Content türünde bir birey.
- Bir ContentAttack bireyi (alt sınıfı önemli değil).
- WriteFile türünde bir sistem çağrısı kategorisi.

Bu bireyler arasındaki bilinen ilişkiler:

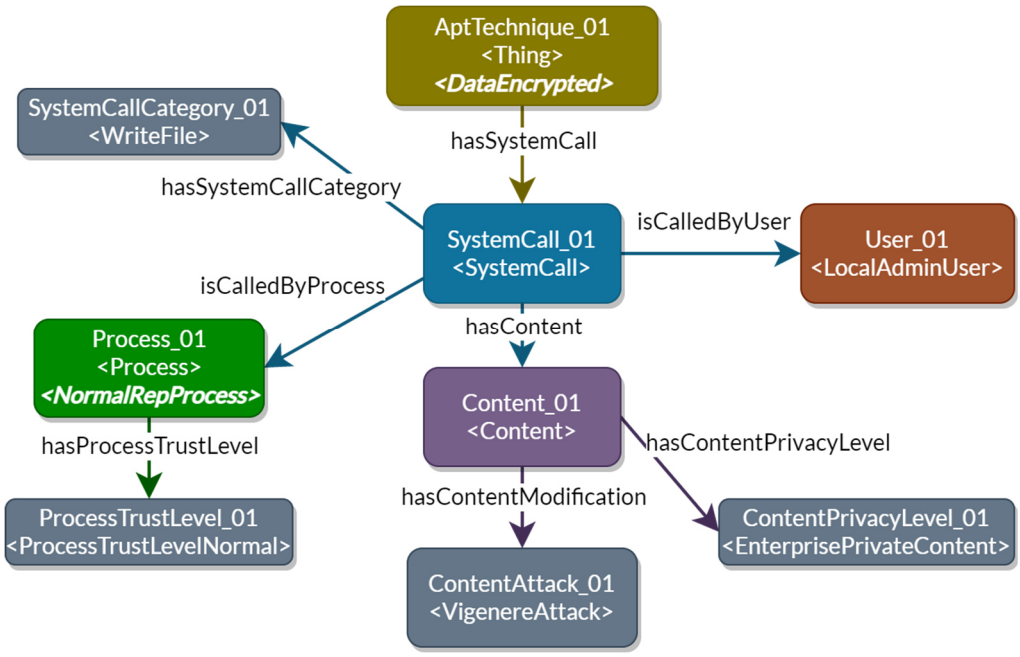
- Thing türünde yaratılmış geçici birey, belirtilen sistem çağrısını kullanmaktadır (hasSystemCall).
- Sistem çağrısı belirtilen NotTrustedProcess türündeki proses tarafından çağrılmaktadır (isCalledByProcess).
- Sistem çağrısı WriteFile türündendir (hasSystemCallCategory)
- Sistem çağrısı belirtilen içeriğe erişim sağlamaktadır (hasContent)
- Belirtilen içerik üzerinde bir içerik atak saldırısı tespit edilmiştir (hasContentModification)

Bu koşullar karşılanıyorsa, başta Thing türünde yaratılmış olan, esas türü bilinmeyen AptTechnique bireyinin DataEncrypted türüne sahip olduğu çıkarımı yapılır. Başka bir deyişle, bir DataEncrypted tekniği uygulandığı çıkarılabilir. Diğer APT teknikleri de SWRL kuralları ile benzer bir yaklaşım kullanılarak tespit edilmektedir.

Belirtmek gerekir ki, buradaki SWRL kuralı içerisinde gözükmeyen, ontoloji içerisindeki diğer daha basit kurallar ile alt sınıflık gibi ilişkiler de kullanılmaktadır. Bu örnekteki prosesin NotTrustedProcess olduğu, "LowRepProcess hasProcessTrustLevel some ProcessTrustLevelLow" ve "LowRepProcess isSubclassOf NotTrustedProcess" kuralları ile önceden belirlenmiştir.

Bu çıkarımın görsel bir anlatımı Şekil 3.9'da gösterilmiştir. Kutuların en üstündeki etiketler bireylere verilen adlardır. "<>" işaretleri içindeki etiketler, bu bireylerin tespiti yapılmış, bilinen sınıflarıdır. Kalın italik olarak yazılan sınıf adları ise ontoloji motoru tarafından çıkarım yapılarak belirlenen sınıflardır. Şekilde, öncelikle

Process_01 bireyinin sınıfının çıkarımının yapıldığı ve APTTeknik_01'in asıl sınıfının DataEncrypted olarak çıkarıldığını göstermektedir.



Şekil 3.9. T1022 tekniği tespit çıkarım akışı.

MITRE matrisinin başka bir yönü, bazı APT tekniklerinin birden fazla APT taktiği altında listelenmiş olmasıdır. T1015-Erişilebilirlik Özellikleri (Accessibility Features) tekniği buna bir örnektir. Bu tekniğin farklı zamanlarda çalıştırılabilen iki ayrı adımı vardır. İlk adım, bir erişilebilirlik uygulamasını zararlı bir yazılımla değiştirerek sistemde kalıcılık sağlanmasıdır. İkincisi ise bir erişilebilirlik uygulaması yerine bu kötü amaçlı yazılımın, Sistem kullanıcısının haklarıyla çalıştırılmasıdır. Böylece T1015 tekniği hem Süreklilik (Persistence) hem de Hak Yükseltme (Privilege Escalation) taktiklerinin bir parçası olabilmektedir. Bu nedenle, ilk adımı tespit etmek ve bu ilk adımın sonucunu kullanarak ikinci adımı tespit etmek, hangi APT taktiğinin gerçekleştirildiğini ayırt edebilmek için gereklidir. Sunulan ontolojide, ilk adımı tespit etmek için aşağıdaki SWRL kuralı kullanılmaktadır:

$$owl{:}Thing(?\,aptTechnique), owl{:}Thing(?\,aptTactic),$$
$$NotTrustedProcess(?\,p), WriteFile(?\,sysCallCategory),$$
$$LocalSystemFile(?\,dataLoc), AccessibilityExecutable(?\,fileType),$$
$$PrivilegedUser(?\,user), Executable(?\,exe),$$
$$SystemCall(?\,sysCall),$$
$$hasSystemCall(?\,apt, ?\,sysCall),$$
$$isCalledByProcess(?\,sysCall, ?\,p),$$

$$hasSystemCallCategory(?sysCall, ?sysCallCategory),$$
$$hasTargetDataLocation(?sysCall, ?dataLoc),$$
$$hasTargetObjectType(?sysCall, ?fileType),$$
$$isCalledByUser(?sysCall, ?user)$$
$$\rightarrow$$
$$isUtilizedByAptTechnique(?exe, ?aptTechnique),$$
$$AccessibilityFeatures(?aptTechnique),$$
$$Persistence(?aptTactic)$$

Bu kural şu şekilde özetlenebilir: Eğer erişilebilirlik uygulamalarından birisine güvenilmeyen bir proses kullanarak WriteFile sistem çağrısıyla bir yazma işlemi yapılıyorsa (dosya muhtemelen kötücül bir yazılımla değiştiriliyorsa), bu olay T1015 saldırısı olarak tespit edilir. Bunun yanında, bu durumun bir Süreklilik (Persistence) taktiği olduğu belirlenir ve bu işlemde kullanılan çalıştırılabilir dosya, T1015 saldırısının bir parçası olarak ilişkilendirilir.

İkinci adımda ise aşağıdaki SWRL kuralı kullanılmaktadır:

$$owl:Thing(?aptTactic), SystemCall(?sysCall),$$
$$CreateProcess(?sysCallCategory), Executable(?exe),$$
$$AccessibilityFeatures(?aptTechnique), PrivilegedUser(?user),$$
$$hasSystemCallCategory(?sysCall, ?sysCallCategory),$$
$$isUtilizedByAptTechnique(?exe, ?aptTechnique),$$
$$isCalledByUser(?sysCall, ?user)$$
$$\rightarrow PrivilegeEscalation(?aptTactic)$$

Bu kuralda, belirtilen çalıştırılabilir dosyanın CreateProcess sistem çağrısı kullanılarak çalıştırılması kontrol edilir. Kullanıcı ve Erişilebilirlik Özellikleri (Accessibility Features) APT tekniğinin varlığı gibi diğer özellikler de kontrol edilir. Bu koşullar karşılandığında, Yetki Yükseltme (Privilege Escalation) taktiği uygulandığı belirlenmiş olur.

Tekniklere ait kurallar belirlenirken teorik ve pratik bilgiler bir arada kullanılmıştır. Öncelikle MITRE'nin her bir teknik ile ilgili sayfasındaki bilgiler ve referanslardan hareketle APT'lerin ilgili tekniği nasıl uyguladıkları araştırılmıştır. İlgili tekniğin icrası için kullanılması mümkün ve gerekli olan sistem çağrıları Windows API (Microsoft, 2021) içerisinden incelenerek uygun olan sistem çağrıları belirlenmiştir. Benzer şekilde, ilgili sistem kaynaklarının (kayıt defteri, dosya, vb.) hangileri olduğu

konu ile ilgili çeşitli web sitelerinden araştırılarak belirlenmiştir. Sonrasında ise bu tekniklerin simülasyonlarını içeren Atomic Red Team (*Atomic Red Team*, 2020), Red Team Automation (*Red Team Automation*, 2021), Purple Team Automation (*Purple Team ATT&CK Automation*, 2021) ve MITRE Caldera (MITRE, 2021) yazılımlarında ilgili tekniğin nasıl gerçeklendiği incelenmiştir. Bu açık kaynaklı yazılımların gerçekleştirdikleri sistem çağrıları toplanmış ve önemli sistem çağrılarının ve kaynakların belirlenmesi sağlanmış ve yukarıda bahsedilen araştırma sonuçları ile çapraz kontrol edilmiştir.

MITRE tekniklerinin çokluğu ve APT'lerin çeşitliliği nedeniyle tez çalışmasında örnek olabilecek bir APT üzerine odaklanılmıştır. Yapılan araştırma neticesinde (FireEye, 2015; Micah Yates, 2017; Ned Moran ve ark., 2015; Symantec, 2016) gibi çeşitli analiz raporlarında veri sızdırma çabaları ve yöntemleri vurgulanan APT3'ün (MITRE, 2020b) modellenmesi uygun görülmüştür. APT3'ün seçilmesindeki bir diğer önemli faktör de MITRE'nin APT3'ün saldırı adımlarını detaylı olarak açıkladığı ve saldırıları modelleme yöntemleri sunduğu "APT3 Saldırı Planı" (Korban ve ark., 2017) çalışmasının var olmasıdır. Bu plan ilerleyen zamanlarda mevcut siber güvenlik çözümlerinin bu saldırılara karşı test edildiği bir çalışma içerisinde kullanılarak sonuçları yayınlanmıştır (MITRE, 2018). Bunların yanında bahsi geçen Atomic Red Team, Red Team Automation gibi açık kaynaklı saldırı simülasyonlarında APT3'e ait saldırıları modellemek için yeterli kaynak olduğu belirlenmiştir.

Bu kaynaklardan faydalanarak, yukarıdaki kurallara benzer şekilde, ontoloji içerisinde APT3'ün kullandığı 19 Tekniği tespit etmek için 55 kural oluşturulmuştur. EK 1'de bu kurallar detaylandırılarak nasıl oluşturuldukları anlatılmıştır.

3.4.3. APT risk tespit kuralları

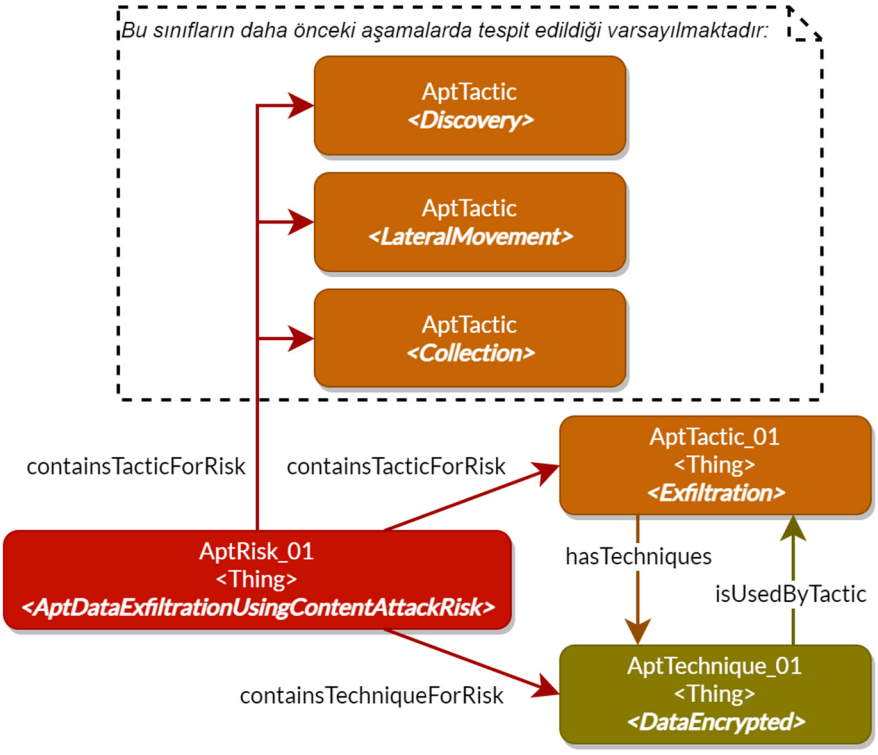
APT Teknikleri tespiti sonrasında, AptTactic ve AptRisk sınıf tanımlamaları kullanılarak sistemdeki APT riski değerlendirilmektedir. MITRE matrisine göre her bir Teknik, bir Taktik altında yer alır. Ontolojide bu ilişki, AptTactic hasTechniques some AptTechnique OWL tanımı aracılığıyla tanımlanır. MITRE'de bir tekniğin birden fazla taktik altında yer aldığı durumlarda, bir önceki bölümde belirtildiği gibi SWRL kuralları aracılığıyla ayrım sağlanarak taktik sınıfı belirlenir.

Benzer şekilde, AptRisk sınıfları, SWRL kuralları ile AptTactic ve AptTechnique sınıfları ile ilişkilendirilir. 3.2.8 bölümünde belirtilen risk seviyelendirmesine göre bu

sınıflar arasında ilişki kurulmuştur. Örneğin, en riskli durumu ifade eden İçerik-tabanlı atak içeren APT Veri Sızdırma Riski (AptDataExfiltrationUsingContentAttackRisk) sınıfına ait kural şu şekilde tanımlanmıştır:

$$APTDataExfiltrationUsingContentAttackRisk\ equivalentTo$$
$$(containsTacticForRisk\ some\ Discovery)$$
$$and\ (containsTacticForRisk\ some\ LateralMovement)$$
$$and\ (containsTacticForRisk\ some\ Collection)$$
$$and\ (containsTacticForRisk\ some\ Exfiltration)$$
$$and\ (containsTechniqueForRisk\ some\ DataEncrypted)$$

Şekil 3.10'da AptTactic_01 nesnesinin hasTechniques ilişkisi aracılığıyla asıl sınıfının Exfiltration olarak tespit edildiği, daha sonrasında ise diğer taktiklerin de tespit edilmiş olmalarıyla beraber AptRisk_01 nesnesinin sınıfının APTDataExfiltrationUsingContentAttackRisk olarak belirlendiği gösterilmektedir. Diğer risklere ait SWRL kuralları EK 1'de gösterilmiştir.



Şekil 3.10. İçerik-tabanlı atak içeren APT Veri Sızdırma Riski çıkarımı.

SWRL kuralları yardımıyla sınıfların belirlenmesinin ardından, tespit edilen APT Teknikleri, Taktikleri ve Riskleri, SPARQL sorguları kullanılarak elde edilebilir.

Şekil 3.11, Şekil 3.12 ve Şekil 3.13'te sırası ile bu sonuçları elde etmek için kullanılan SPARQL sorguları ve örnek veri kümesinden gelen sonuçlar gösterilmektedir. Bu sorgular aracılığıyla sistemdeki tespit işlemleri tamamlanmaktadır.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX apton: <http://www.semanticweb.org/user/ontologies/2021/1/apton#>

SELECT ?individual ?inferredAptTechnique
WHERE {
        ?individual rdf:type ?inferredAptTechnique .
        ?inferredAptTechnique rdfs:subClassOf apton:AptTechnique .
}
```

individual	inferredAptTechnique
AptTechniqueDataCompressed3	DataCompressed
AptTechniqueDataCompressed2	DataCompressed
AptTechniqueDataCompressed1	DataCompressed
AptTechniqueCreateAccount1	DataCompressed
AptTechniqueSystemOwnerUserDiscovery1	SystemOwnerUserDiscovery
AptTechniqueCreateAccount1	CreateAccount
AptTechniqueDataCompressed1	CreateAccount
AptTechniqueCredentialDumping2a	CredentialDumping
AptTechniqueDataEncrypted1	DataEncrypted

Şekil 3.11. Tespit edilen APT tekniklerini elde etmek için SPARQL sorgusu ve örnek sonuçlar.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX apton: <http://www.semanticweb.org/user/ontologies/2021/1/apton#>

SELECT ?individual ?inferredAptTactic
WHERE {
        ?individual rdf:type ?inferredAptTactic .
        ?inferredAptTactic rdfs:subClassOf apton:AptTactic .
}
```

individual	inferredAptTactic
AptTacticExfiltration1	Exfiltration
AptTacticLateralMovementX	LateralMovement
AptTacticDiscovery1	Discovery
AptTacticCollectionX	Collection

Şekil 3.12. Tespit edilen APT taktiklerini elde etmek için SPARQL sorgusu ve örnek sonuçlar.

SPARQL query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX apton: <http://www.semanticweb.org/user/ontologies/2021/1/apton#>

SELECT ?individual ?inferredAptRisk
WHERE {
   ?individual rdf:type ?inferredAptRisk .
   ?inferredAptRisk rdfs:subClassOf+ apton:AptRisk .
   FILTER NOT EXISTS {
      ?individual rdf:type ?childOfInferredAptRisk .
      ?childOfInferredAptRisk rdfs:subClassOf ?inferredAptRisk .
      FILTER (?inferredAptRisk != ?childOfInferredAptRisk )
   }
}
```

individual	inferredAptRisk
AptRiskPossibleAptActivityRisk	PossibleAptActivityRisk
AptRiskAptDataExfiltrationUsingContentAttackRisk1	AptDataExfiltrationUsingContentAttackRisk

Şekil 3.13. Tespit edilen APT risklerini elde etmek için SPARQL sorgusu ve örnek sonuçlar.

4. İÇERİK SINIFLANDIRMA ALGORİTMASI GELİŞTİRİLMESİ

4.1. Motivasyon ve Algoritmanın Gereksinimleri

DLP sistemlerinin önceki bölümlerde bahsedilen yapısal zayıflıklarının yanında, içerik sınıflandırma algoritmaları açısından da zayıflıkları bulunmaktadır. DLP sistemlerinde içerik ve bağlam dikkate alınabilmektedir. Dosyayı oluşturan kişi, dosya türü, dosya konumu, yaratılma tarihi, özel etiketler gibi üst veri (metadata) bilgiler bağlamın değerlendirilmesi açısından ele alınırlar. İçerik ise istatistiksel analiz, parmak izi çıkarma (fingerprinting) ve düzenli ifadeler (regular expressions) gibi yöntemler ile sınıflandırmaya çalışılır (Alneyadi ve ark., 2016).

Bağlama dayalı yöntemlerde, dosyada ya da üst verilerinde kasıtlı ya da kasıtsız olarak yapılan basit değişiklikler ve tanımlamalar, dosyanın gizli olarak sınıflandırmasını engelleyebilir. Örneğin dosyayı yaratan kişinin ve dosya konumunun dikkate alındığı bir DLP kuralında, dosya içeriği farklı bir kullanıcı hesabı ile farklı bir konuma taşındığında DLP kuralı atlatılabilmektedir. Bağlam bilgisinin tek başına yeterli olmaması sebebiyle DLP sistemleri sınıflandırma yaparken içeriği de dikkate almaktadırlar. Ancak içerik sınıflandırma yöntemlerinde özellikle zararlı yazılım kaynaklı içerik saldırılarına karşı zayıflıklar bulunmaktadır. (Blasco ve ark., 2012; Canbay ve ark., 2017; Matasano, 2007; Mustafa, 2013) çalışmalarından ve faydalanılarak oluşturulan Tablo 4.1'de gösterildiği üzere, DLP sistemlerinin içerik sınıflandırmada kullandığı yöntemlerin her birinin birden çok içerik tabanlı saldırıya karşı zafiyeti bulunmaktadır. Bununla beraber APT'ler, tespit edilmeme çabalarının sonucu olarak şifreleme gibi işletim sistemi kütüphanelerine bağlı yöntemler yerine, uygulama hafıza alanı içerisinde gerçekleştirilebilecek yer değiştirme, yeri koyma gibi yapısal dönüşüm atakları ve eş/çok kelime, kitap şifreleme gibi karartma saldırıları tercih etmektedirler. Ayrıca şifrelenmiş verilerdeki entropi değerleri belirgin olarak bir saldırı olduğunu gösterebilir (Davies ve ark., 2022). Yapısal dönüşüm ve karartma saldırıları sonucu oluşan dokümanlardaki entropi değerleri ise şifrelenmiş veriler gibi değerlere sahip olmamaktadır ve DLP sistemleri için normal bir dosyaya benzemektedirler. Bu sebeplerle APT'lerin bu yöntemleri tercih etmesi beklenebilir.

Nitekim, yapılan analiz sonucu APT10'un Vigenère saldırısı ile verileri kaçırdığı tespit edilmiştir (Suguru Ishimaru, 2022).

Tablo 4.1. DLP sistemlerinin kullandığı yöntemler ve saldırılara karşı zayıflıkları.

İçerik Eşleştirme ve Sınıflandırma Yöntemi	Etkili Olan İçerik Saldırısı Türü
İmza Çıkartma (Fingerprinting)	Yer Değiştirme Cümle Yapısı Değiştirme Yerine Koyma Eşleştirme/Anahtarlama Şifreleme Kitap Şifreleme Eş/Çok Anlamlı Değişikliği
Özel Kelime Eşleştirme	Yerine Koyma Eşleştirme/Anahtarlama Şifreleme Kitap Şifreleme Eş/Çok Anlamlı Değişikliği
Düzenli İfade Eşleştirme	Eşleştirme/Anahtarlama Şifreleme Kitap Şifreleme Eş/Çok Anlamlı Değişikliği
Doğal Dil İşleme (NLP)	Cümle Yapısı Değiştirme Yerine Koyma Eşleştirme/Anahtarlama Şifreleme
Gizli Anlam Analizi (LSA)	Şifreleme Kitap Şifreleme Eş/Çok Anlamlı Değişikliği

Tez kapsamında sunulan sistemde, sistem çağrısının eriştiği içeriğin doğru olarak sınıflandırılması, APT risk seviyesinin belirlenmesinde ve içeriğin korunmasında önemli bir yer tutmaktadır. Bu sebeple, APT'lerin içerik tabanlı saldırılarına karşı dayanaklı bir içerik sınıflandırma yöntemi geliştirilmesi gerekmiştir. Bu probleme karşı literatürde bütüncül bir çözüm bulunmamakla beraber ilerideki bölümlerde önerilen ön işleme aşamaları sonrasında literatürde bulunan yöntemlerin bir arada kullanılması ile sınıflandırma başarımının artırılması mümkündür. n-gram ve k-skip-n-gram yöntemlerinin kelime ekleme/çıkarma gibi yapısal dönüşüm saldırılarına karşı başarılı olduğunu gösterilmiştir (Alneyadi ve ark., 2016). Sözlüksel karşılaştırma yapılması ve harf temelli n-gramlar kullanılarak kelimelerin düzeltilmesi yöntemlerinin ise kelimeler arasındaki boşluklara ve kelime içerisindeki harflere ilişkin yapılan saldırılarda başarılı olduğu incelenmiştir (Martins & Silva, 2004; Priya ve ark., 2017). Gizli Anlam Analizi (Latent Semantic Analysis - LSA) yönteminin eş

anlamlı ya da çok anlamlı kelimeler kullanma, karartma (obfuscation) ve özet çıkarma türündeki saldırılarda sınıflandırma başarımı arttırdığı belirtilmiştir (Cosma & Joy, 2012; Du ve ark., 2015). Bu yöntemler ışığında, tez kapsamında içerik tabanlı saldırılara karşı dayanıklı, çok aşamalı özgün bir içerik sınıflandırma yöntemi sunulmuştur.

Probleme ilişkin farklı yöntemler TÜBİTAK 1001 117E100 numaralı projesi kapsamında tez yazarının da ortak olduğu (Kesenek ve ark., 2021) çalışmasında ele alınmış olup (Kesenek, 2019) yüksek lisans çalışmasında detaylandırılmıştır. Bu tez kapsamında, bu çalışmalarda elde edilen sonuçlar ışığında kaydedilen en başarılı yöntem ele alınmıştır.

İçerik sınıflandırma yöntemine ilişkin sunulan çözümün yanı sıra, içerik tabanlı saldırıların varlığının tespitinin de faydalı olacağı değerlendirilmiştir. Bazı içerik tabanlı saldırılarda içeriğin orijinal hali ile saldırı yapılmış hali arasında yapılan karşılaştırma sonucunda içerik üzerinde bir atak yapıldığı ve ne tür bir atak yapıldığı tespit edilebilmektedir. Bu durumun tespiti hem içeriğin korunmasına katkı sunarken hem de APT davranışının ve riskinin daha detaylı belirlenmesine imkân sunmaktadır. Bu sebeple, tez kapsamında sunulan sistemde proseslerin eriştiği içeriklerin sürekli takibi sağlanarak bilinen içerik tabanlı ataklar tespit edilmeye çalışılmaktadır. Bu tespitin nasıl yapıldığı ilerleyen bölümlerde detaylandırılmaktadır.

4.2. Algoritma Akışı

Makine öğrenmesine dayalı yöntemlerin genelinde olduğu gibi sunulan yöntemde de öncelikle öğrenme gerçekleştirilmektedir. Öğrenme aşamasında önişleme, özellik çıkarımı, özellik seçimi ve sınıflandırma olmak üzere dört aşamalı bir yöntem uygulanmaktadır. Şekil 4.1'de sergilenen önişleme ve özellik çıkarımı akışına göre öncelikle bir önişleme yapılmaktadır. Bu yolla sonraki özellik çıkarımı aşamasında çıkarılmak istenen özelliklerin hem içeriği daha iyi temsil etmesi sağlanmakta, hem de yazım düzeltimi ile olası harf bazlı saldırıların özellikleri yok etmesinin önüne geçilmiş olur. Sonrasında hem kelime hem de harf bazında simge (token) çıkartma işlemi yapılarak hem n-gram hem de LSA yöntemleri ile özellik çıkarımı yapılmaktadır.

Özellik çıkarımı aşamasını Şekil 4.2'de sergilenen özellik seçimi ve sınıflandırma aşamaları takip etmektedir. Özellik seçimi aşamasında Terim Frekansı-Ters Doküman

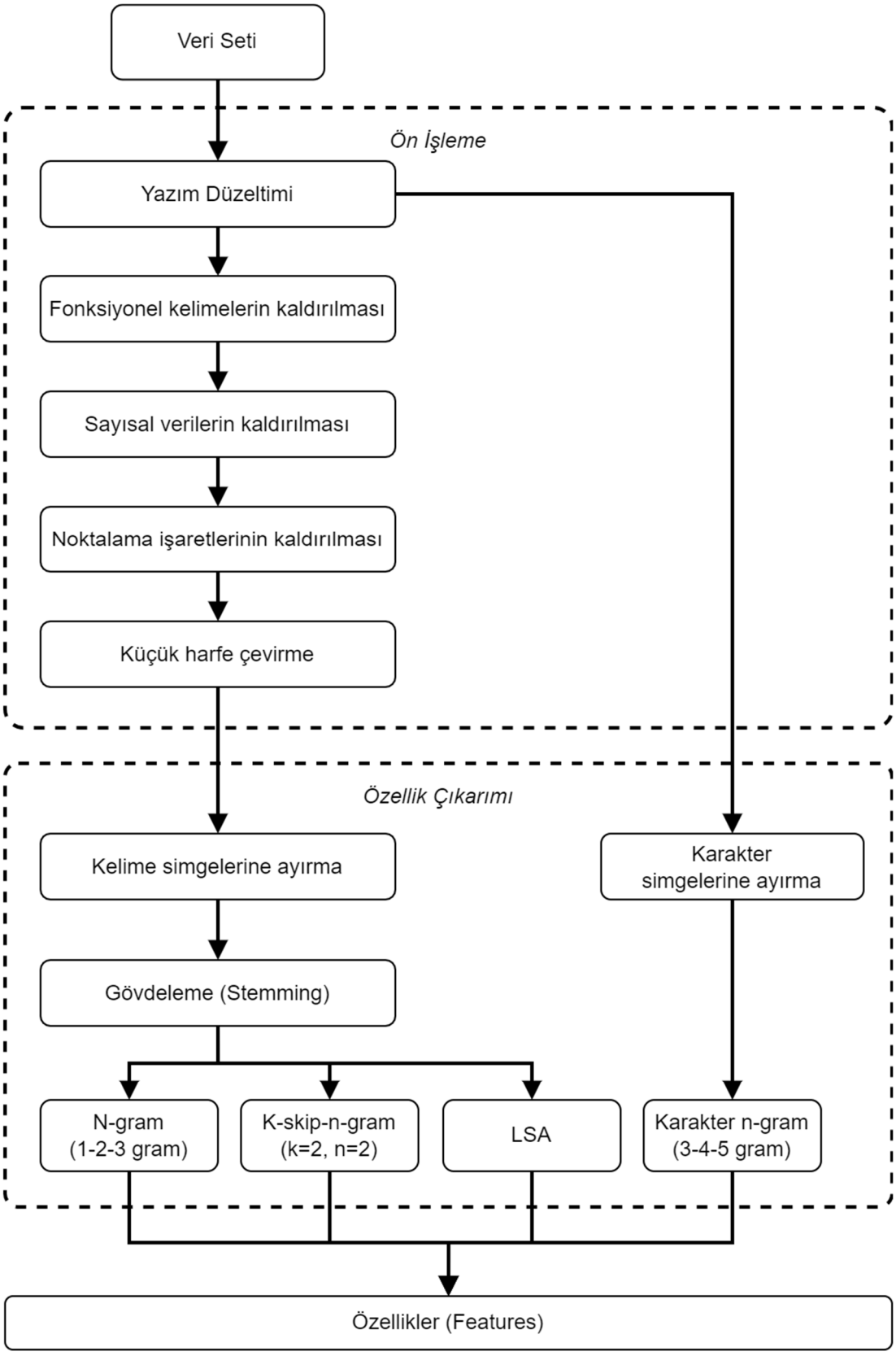
Frekansı (Term Frequency-Inverse Document Frequency – TF-IDF) yöntemi ile çıkarılan özelliklerin ilgili dokümanlarla ne derece ilişkilendirilebilecekleri belirlenmektedir. Bu aşamayı üç farklı sınıflandırıcının kullanıldığı sınıflandırma aşaması takip etmektedir. Üç sınıflandırıcının sonuçları oylamayı sınıflayıcı ile tek bir sınıflandırma sonucuna indirgenerek sınıflandırma modeli elde edilmektedir.

Algoritmanın test aşamasında ise hedef dokümanlar üzerinde ön işleme, özellik çıkarımı, özellik seçimi adımları benzer şekilde uygulandıktan sonra öğrenme aşamasında elde edilmiş sınıflandırma modeli de kullanılarak sınıflandırma gerçekleştirilmektedir.

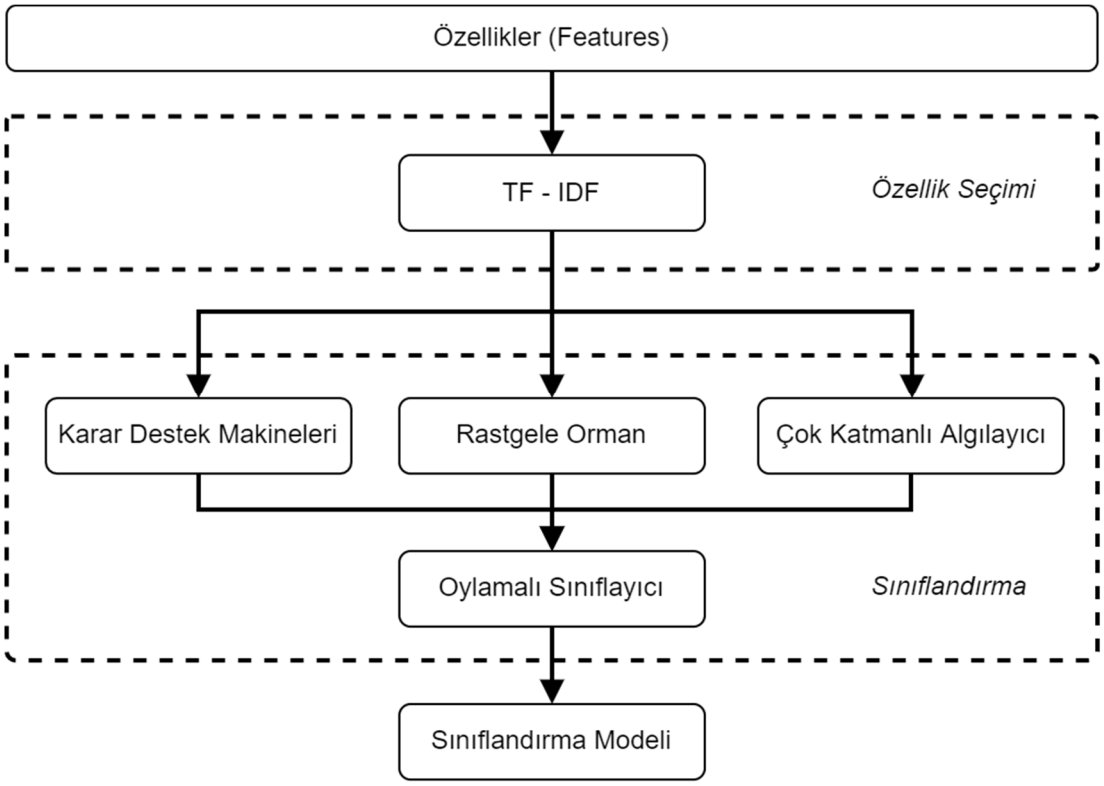
4.2.1. Önişleme

Bu aşamada öncelikle, metin üzerindeki olası harf bazlı saldırıları geri çevirmek amacıyla yazım düzeltimi algoritması uygulanmaktadır. (Mustafa, 2013) çalışmasında belirtildiği gibi, dokümanın tamamına yapılacak bir modifikasyon saldırısı durumunda sınıflandırıcı için ayırt edilebilir özellikler büyük oranda değişecek ve sınıflandırma başarısız olacaktır. Bu noktada, (Kesenek ve ark., 2021) çalışmasında ele alındığı üzere, metinde herhangi bir işlem yapmadan ve özellikler çıkarılmadan önce yazım düzeltimi yapılmasının başarımı arttırdığı gözlemlenmiştir.

Yazım düzeltimi işlemi için literatürde farklı yöntemler bulunmakla beraber, tez kapsamında sunulan sistemde başarılı sınıflandırma yapılmasının yanı sıra hızlı bir sınıflandırma sonucu alınması da önemlidir. Bunun için başarımı ve hızı (Chaabi & Ataa Allah, 2022; P. Gupta, 2020; Tolegenova, 2022) çalışmalarında ortaya konulmuş olan Damerau-Levenshtein algoritmasını hızlı çalışacak şekilde gerçekleyen Symspell (Garbe, 2022) kütüphanesinden faydalanılmıştır. Kütüphanenin çalışma aşamasında sözlük olarak İngilizce dilindeki 28765 kelime ve 44 kullanım sıklığı verilen Symspell Compound kütüphanesinin “defakto” sözlüğü kullanılmıştır.



Şekil 4.1. İçerik sınıflandırma özellik çıkarımı akışı.



Şekil 4.2. İçerik sınıflandırma özellik seçimi ve sınıflandırma akışı.

Yazım düzeltme algoritmalarının özellikle kullanılan kütüphane kaynaklı bazı sınırlamaları mevcuttur. Bunlardan birisi yanlış olduğu tespit edilen kelimenin yerine konulabilecek kelimenin birden fazla alternatifinin olmasıdır. Bu alternatifler arasında seçim yapılırken seçilen yanlış bir kelime ile metnin anlamı bozulabilmektedir. Bu tür sınırlamaların kütüphane kaynaklı olabileceğini göz önünde bulundurarak daha geniş bir kütüphane geliştirilmesi tez kapsamı dışında tutulmuştur.

Yazım düzeltimi aşamasından sonra, verilen dokümandaki sınıflandırmaya olumlu etki etmeyecek özellikler (features) metin içerisinden çıkarılmakta ve metinler arasında normalizasyon sağlanmaktadır. Metinlerdeki sayısal veriler, noktalama işaretleri ve fonksiyonel kelimeler, içerik hakkında ayırt edici bilgi sağlamadığı için metinden çıkarılmaktadır. Bütün harfler küçük karaktere dönüştürülerek ise normalizasyon sağlanmaktadır. Bu yaklaşım NLP aşamalarında genel olarak uygulanmakta ve böylece hem sınıflandırma başarımı arttırılmakta hem de boyut azaltımı (dimensionality reduction) ile veri miktarı düşürülmektedir.

4.2.2. Özellik çıkarımı

Bu aşamada öncelikle doküman hem kelime hem de karakter simgelerine (token) ayrılmaktadır. Kelime simge çıkarımı sonrasında özellik sayısının düşürülmesi için,

gövdeleme (stemming) yapılarak kelimeler eklerinden arındırılmaktadır. Gövdeleme ilse doküman sınıflamada boyut azaltımı sağlanmaktadır. Çünkü gövdeleme işlemi, doküman içerisinde ekler dolayısıyla farklı hallerde bulunması muhtemel aynı kelimeleri tek bir biçime indirger. Farklı çalışmalarda farklı gövdeleyiciler ile yapılan testlerde bu özellik ortaya konulmuştur (Haroon, 2018; Sharma, 2012). Gövdeleme sonrası elde edilen kelime simgelerinden farklı boyutlarda n-gram ve skip-gram'lar çıkarılmakta, ayrıca SVD dönüşümü yapılarak LSA özellikleri de çıkarılmaktadır. Karakter-gram'lar ile de benzer şekilde farklı boyutlarda n-gram'lar oluşturulmaktadır.

Farklı saldırı türlerine karşı başarılı bir algoritma olabilmesi amacıyla Özellik Çıkarımı aşamasında n-gram, skip-gram, LSA ve karakter-gram çıkarımları bir arada kullanılmıştır. Temel olarak n-gram bazlı olan algoritmanın başarımını sağlamak için modifikasyon saldırılarına karşı karakter-gram, eş anlamlı kelimelerle değiştirme ve karartma saldırılarına karşı LSA, yer değiştirme ve yerine koyma saldırılarına karşı ise skip-gram kullanımı tercih edilmiştir.

N-gram çıkarımı

Metin tabanlı sistemlerde çokça kullanılan bir yöntem olan n-gram çıkarımı, uzun bir metnin n sayıda kelime ya da karakterden oluşan kümelere bölünmesidir. N-gram çıkarımına örnek olarak "içerik atak tespiti" metnini ele alırsak,

1-gram çıkarıldığında: "içerik", "atak", "tespiti";

2-gram çıkarıldığında: "içerik atak", "atak tespiti", "tespiti içerik"

3-gram çıkarıldığında: "içerik atak tespiti"

kümeleri oluşacaktır. N-gram çıkarımı kelime grupları yerine hecelere ya da karakterlere de uygulanabilmektedir.

Tez kapsamında kelime n-gram uzunluğu 1, 2, ve 3; karakter n-gram'larda ise 3, 4 ve 5 olarak kullanılmıştır.

K-skip-n-gram çıkarımı

N-gram yönteminin genişletilmiş bir hali olan k-skip-n-gram da, içerik metin sınıflandırma algoritmalarında kullanılan bir yöntemdir. N-gramlar belli bir başarım sunsa da dilin yapısı gereği bir kelimenin hemen yanındaki kelimelerin bazen bağlam için gerekli bilgiyi taşımaması ve farklı cümleler içerisinde aranan kelime ile bir

önceki bağlamdaki kelimeler arasına tamlayıcı nitelikte kelimelerin girmesi nedeniyle her durumda istenilen başarım sağlanamamaktadır. Bu sebeple bu yaklaşım, n kelimeli diziyi arada k tane kelimenin atlanarak oluşturulduğu k-skip-n-gram yöntemi ile genişletilmiştir. Ayrıca k değerinin değiştirilmesi ile ilgili kelimenin içinde bulunduğu bağlam daha esnek olarak ele alınabilmektedir. Bu da uzun ve tamlamaların çok olduğu cümlelerdeki başarımı arttırmaktadır.

Bununla beraber, cümlenin yapısının değiştirilmesi, kelime ya da harf değiştirme/ekleme tarzı saldırılar bu algoritmalara karşı etkili olmakta ve veri sızıntısını engelleyebilmektedir.

Gizli anlamsal analiz (LSA)

Gizli anlamsal analiz ya da diğer adıyla gizli anlamsal indeksleme (Latent Semantic Indexing) yöntemi, büyük metin koleksiyonlarını analiz etmek ve bu metinler arasındaki anlamsal ilişkileri belirlemek için kullanılan bir tekniktir.

LSA, belgelerdeki terimlerin frekanslarını matematiksel olarak analiz eder ve belgeler arasındaki benzerlikleri bulur. Bu analizde, terimlerin sıklığı matris halinde temsil edilir. Satırlarda terimlerin sütunlarda dokümanların bulunduğu bu matris oldukça seyrektir. Bu sebeple Tekil Değer Ayrışımı (Singular Value Decomposition-SVD) yöntemi kullanılarak bu matrisin boyutu düşürülür. Bu dönüşüm ile belgeler ve terimler arasındaki gizli semantik ilişkileri ortaya çıkarır.

4.2.3. Özellik seçimi ve terim ağırlıklandırma

Bir terimin belirli bir dokümanda kaç kez geçtiği sayısının ilgili dokümandaki toplam terim sayısına bölünmesi ile elde edilen değer, o terimin doküman içerisindeki terim frekansı (Term Frequency - TF) olarak adlandırılır (Denklem 4.1). TF, dokümanın hangi kategoriye ait olduğunu gösteren önemli bir işaret olmaktadır.

$$TF(t,d) = \frac{f_{t,d}}{\sum f_{T,d}}, t \in d, T \in d \quad (4.1)$$

Bununla beraber, terimin farklı dokümanlardaki kullanım sıklığı fazla olduğunda, terimin ayrım gücü ve dolayısıyla sınıflandırmaya olan etkisi azalır. Benzer şekilde, doküman içinde az geçen bir terimin sınıflandırmadaki etkisinin olmadığını düşünmek yanıltıcı olabilir, çünkü bu terim diğer dokümanlarda geçmiyorsa esasında bu terim önemli bir ayırt edici özellik taşımaktadır. Bu durumda, ters belge frekansı (Inverse

Document Frequency-IDF) kullanılması gerekmektedir. IDF'nin formülü şu şekildedir:

$$IDF(t, d, D) = log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (4.2)$$

Bu denklemde *D, {d1, d2, ..., dn}* gibi dokümanlardan oluşan bir kümeyi, *t* ise bu dokümanlardaki bir terimi ifade etmektedir. Dolayısıyla formül, toplam doküman sayısının, terimin içinde geçtiği doküman sayısına bölünmesi sonucunun logaritmasının alınması olarak özetlenebilir. Burada olası sıfıra bölünme problemini engellemek için paydaya 1 eklenmektedir.

Denklem 4.1 ve 4.2'de hesaplanan TF ile IDF değeri çarpılarak TF-IDF değeri elde edilir:

$$TF\_IDF(t, d, D) = TF(t, d).IDF(t, d, D) \quad (4.3)$$

TF-IDF değerinin yüksek olması ilgili terimin verilen belgedeki frekansının yüksek ve tüm belgeler içerisinde ise düşük frekansta olmasını gösterir. Böylece belgeler içerisinde ortak ve yaygın olan kelimeler daha düşük TF-IDF değerine sahip olur ve ayırt edici olmadıkları modellenmiş olur.

Yaptığımız çalışmada özellik azaltımı için TF-IDF yöntemi uygulanırken, doküman frekansı (DF) değeri üçten küçük olan terimler kaldırılmıştır.

4.2.4. Sınıflandırma

Sınıflandırma işlemleri için farklı türde saldırılar olması durumunda çıkarılan özelliklerde oluşan değişimlerin sınıflandırmaya etkisini azaltmak için çoklu sınıflandırıcı yaklaşımı kullanılmıştır. (Catal & Nangir, 2017; Saha & Ekbal, 2013) çalışmalarında da uygulanan bu yöntem ile, farklı sınıflandırıcılardan gelen sonuçlar oylamalı sınıflandırma yöntemi ile değerlendirilerek sonuç elde edilmektedir. Literatürde başarımları kanıtlanmış olan, Karar Destek Makineleri, Rasgele Orman ve Çok Katmanlı Algılayıcıdan elde edilen sonuçlar Oylamalı Sınıflandırıcı'ya aktarılarak en çok oy alan sınıf seçilmektedir.

Rastgele Orman, bir grup ağaç tabanlı sınıflandırıcı olarak tanımlanabilir. Bu yöntemde her bir düğümde rastgele olarak seçilen değişkenler arasında en iyisini

kullanarak her bir düğümü dallara ayırır. Her veri seti, orijinal veri setinden üretilir. Ardından, rastgele özellik seçimi kullanılarak ağaçlar geliştirilir ve bu ağaçlar budanmaz.

Karar Destek Makineleri (Support Vector Machines-SVM), bir istatistiksel öğrenme yöntemidir ve sınıflandırma ile regresyon için kullanılabilir. Bu yöntem, n özellikli bir veri kümesini bir düzlem (hyper-plane) kullanarak ayırmaya çalışır. Düzlemdeki veri setlerini iki ayrı grup olarak varsayarsak bu grupları ayıran bir düzlem çizilebilir. Eğer veri kümesi doğrusal olarak ayrılamıyorsa, özellik uzayından yüksek boyutlu bir uzaya bir fonksiyon uygulanarak birçok noktadan oluşan bir düzlem elde edilebilir.

Yapay sinir ağları (Neural Network), insan beyninin çalışma şeklini örnek alarak geliştirilen bir yöntemdir. Birbirine bağlı nöronlardan oluşan bir ağ modellenir. Her bir nöron bir girdi değeri için bir çıktı değeri üreten fonksiyondan oluşur. Burada kullanılan fonksiyon aktivasyon fonksiyonu (activation function) olarak adlandırılır.

4.3. Gerçekleme

İçerik sınıflandırma algoritmasının gerçeklemesinde Python dili ve bu dilde geliştirilmiş olan scikit-learn (scikit-learn, 2020), nltk (nltk, 2021) ve symspell (Garbe, 2022) kütüphanelerinden faydalanılmıştır.

Özellik çıkarımı önişlemlerinde nltk ile fonksiyonel kelimelerin metinden atılması ve simgelerin (token) elde edilmesi sağlanmıştır. Gövdeleme işlemleri için ise symspell kütüphanesi kullanılmıştır. Yazım düzeltimi işlemleri için de symspell kütüphanesi, "defakto" sözlüğü ile kullanılmıştır.

Önişlemlerden sonra n-gram, k-skip-n-gram çıkarımı için nltk kütüphanesi kullanılmıştır. LSA analizi için gerekli model ise scikit-learn kütüphanesinde bulunan TruncatedSVD sınıfı aracılığıyla gerçekleştirilmiştir.

Özellikler elde edildikten sonra, özellik seçimi aşamasında TF-IDF dönüşümünde scikit-learn kütüphanesindeki TfidfVectorizer sınıfından yararlanılmıştır.

Sınıflandırma aşamasında ise, scikit-learn kütüphanesinde bulunan RandomForestClassifier, SVC ve MLPClassifier sınıfları ile sınıflandırma yapılarak aynı kütüphanede bulunan VotingClassifier ile oylamalı sınıflandırıcı gerçeklenmiştir.

4.4. İçerik Atak Tespiti

Ontolojinin detaylandırıldığı önceki bölümde belirtildiği gibi, APT'ler genellikle verileri olduğu gibi çıkarmaya çalışmak yerine DLP önlemlerini atlatmak için bu verileri değiştirme eğilimindedirler. Bu tür saldırılar ile, hassas içeriğin DLP sistemi tarafından hassas olarak algılanması engellenecek şekilde değiştirilmektedir. APT10 tarafından kullanılan Vigenère saldırısı bu amaçla kullanılmıştır (Suguru Ishimaru, 2022). Doküman içerisindeki ifadeleri ya da bölümleri farklı yerlere taşıyan dönüşüm saldırıları ve kelimeleri eşanlamlılarıyla değiştirildiği saldırılar diğer örnek yöntemlerdir. Bu saldırıların daha ayrıntılı sınıflandırması (Mustafa, 2013) tarafından sunulmuştur. Bu saldırıların (şifreleme tipi saldırılar hariç) ortak özelliği, sonuç olarak elde edilen içeriğin harf, sayı, kelime, ifade ve cümlelerden oluşmaya devam etmesidir. Dolayısıyla, sonuçta ortaya çıkan içerik, DLP sistemi için açıkça bozuk bir doküman görüntüsü sergilemez.

İçerik eşleştirme ve sınıflandırma algoritması sonuçlarında, metin içerisinde önceden var olan özelliklerin tamamen kaybolduğu bu tip ataklar sonrasında, sınıflandırma aşamasında başarılı sonuçlar alınmasının çok mümkün olmadığı görülmüştür. Bu tür atakların çözülebilmesi için, içerik üzerinde yapılan atak sonucu oluşan değişikliklerin belirlenerek bunların geri çevrilmesi gerekmektedir. Böyle bir geri çevrim işleminden sonra ancak bir sınıflandırma yapılabileceği değerlendirilmiştir. Örnek olarak şifrelenmiş bir metnin öncelikle şifresinin çözülmesi gerekmektedir. Bu ise, hem yapılan atağın algoritması bilinmediği için geri çevirmenin nasıl yapılacağının bilinememesi hem de ortak-özel anahtar şifreleme gibi yöntemlerde geri çevirmenin ancak özel anahtara sahip olunması sayesinde yapılabileceğinden ötürü mümkün olmayacaktır.

APT'lerin bu davranışı esasında veri çıkarma niyetinin varlığının önemli bir göstergesidir. Bu nedenle, sunulan sistemde içerik tabanlı saldırıların varlığının tespiti APT'lerin varlığı için de bir gösterge olarak ele alınmıştır. Bu sebeple, proseslerin eriştikleri dosyaların sürekli kontrol edilerek dosya üzerinde yapılan işlemlerin bir içerik saldırısı olup olmadığının tespitinin yapılmasının APT'lerin tespiti açısından faydalı olacağı sonucuna varılmıştır.

Bu kapsamda Harf Yer Değiştirme (Transposition) ve Vigenère saldırıları ele alınmıştır.

Harf Yer Değiştirme (Transposition) Saldırısı:

Harf yer değiştirme saldırısı basitçe şu şekildedir:

- Giriş = “Bugün okula gitmek istiyorum.”
- Cümle, matris olacak şekilde düzenlenir boş kalan yerlere x ya da 0 değeri yazılır.

B	U	G	Ü	N	
O	K	U	L	A	
G	I	T	M	E	K
	I	S	T	I	Y
O	R	U	M	.	0

- Matrisin transpozu alınır yani satırlar sütün sütunlar satır olacak şekilde tekrar düzenlenir.

B	O	G		O	B
U	K	I	I	R	U
G	U	T	S	U	G
Ü	L	M	T	M	Ü
N	A	E	I	.	N

- Oluşan yeni matriste satırlar sırayla okunarak şifreli mesaj elde edilir.
- Şifreli mesaj = “Bog okiirgutsuülmtmnaei. Ky0”

Basit bir saldırı olsa da bu tip saldırıların veriyi nasıl değiştirdiğini göstermesi açısından önemli bir saldırı türüdür. Bu tip saldırıların bir diğer önemli özelliği verinin entropi değerini neredeyse hiç değiştirmemesi sebebiyle şifreli bir veri olarak tespit edilebilmesidir. Yukarıdaki örnekte, ilk verinin Shannon entropi değeri, yaklaşık 4,10 iken, ikinci verinin ki yaklaşık 4,25’tir. Bu sebeple bazı DLP sistemlerinde kullanılan şifreli veri tespit yöntemlerini atlatabilecektir.

Bu saldırıyı tespit için şu yaklaşım başarılı olmaktadır:

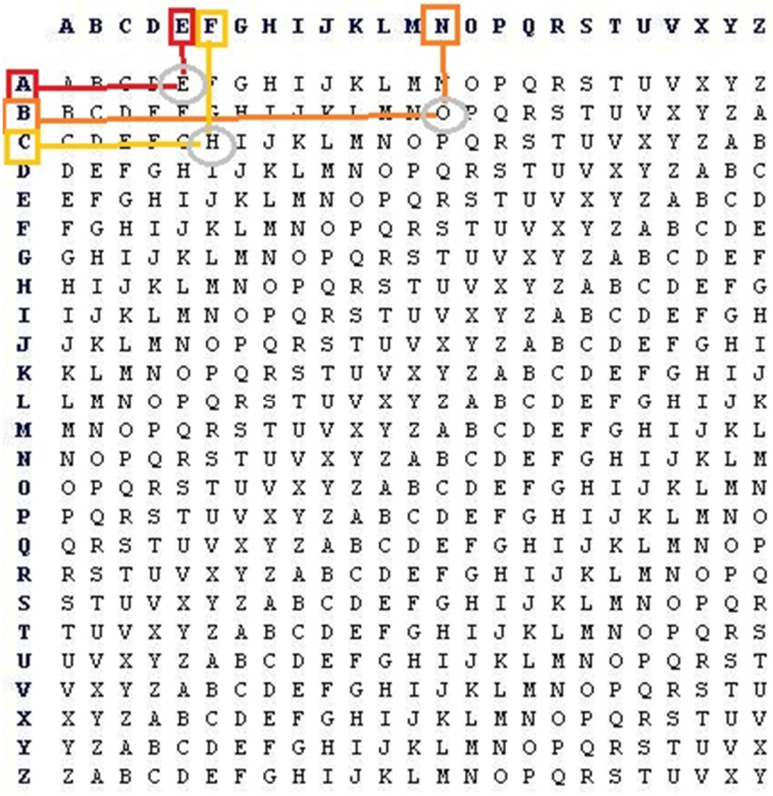
- Metnin önceki ve sonraki halinin karakter frekansı incelenmektedir.
- Karakter frekansları aynı olması durumunda; metnin önceki halinden bazı kelimeler seçerek sonraki halinde bu kelimelerin bütüncül bir şekilde bulunup bulunmadığı kontrol edilmektedir.
- Eğer bu kelimeler bütüncül bir şekilde bulunmuyor ise metin üzerinde yer değiştirme saldırısı yapıldığı sonucuna varılabilmektedir.

Vigenère:

Çoklu alfabe (poli-alfabetik) şifreleme olarak da adlandırılan Vigenère şifrelemede anahtara bağlı olarak her harf, birden fazla harfle eşleşmektedir. Vigenère, esasında

Sezar şifrelemesinin gelişmiş halidir. Sezar yönteminde harflerin değiştirilmesi için bir tek alfabe kullanılırken Vigenère şifrelemesinde birden fazla alfabe kullanılır. Böylece, mono alfabetik yöntemlerden farklı olarak, bir harf değiştirilince her seferinde aynı harfe dönülmez. Bu işlem, "Vigenere Tablosu" olarak bilinen bir tablo ile gerçekleştirilir (Şekil 4.3). Bu yaklaşımla bir mesajın şifrelenebilmesi için, bir anahtar kelimeye ihtiyaç vardır.

Örnek olarak, "bugün okula gitmek istiyorum" cümlesini "şifre" anahtarını kullanarak şifrelediğimizde "tfkns zpmpş kczğnp nleoöşkfs" sonucu alınmaktadır. Yer değiştirme saldırısında olduğu gibi, bu yöntemde de entropi değeri çok fazla değişmemektedir. Bu sebeple, basit ama etkili bir yöntemdir.



Şekil 4.3. Örnek Vigenere tablosu.

Bu saldırıyı tespit için ise şu yaklaşım başarılı olmaktadır:

- Saldırı sonrası, metnin içindeki her bir karakterin frekansı değişmektedir. Bu frekanslar hesaplanarak karşılaştırılmaktadır.
- Normal metin ve şifreli metinde boşluk, sayı, özel karakter gibi ifadeler aynı indis değerlerinde kalmaktadır, bu yüzden iki metin arasında bu değerlerin indis değerleri karşılaştırılarak saldırının tespiti sağlanabilmektedir.

5. APTONSYS: ETMEN TABANLI APT VERİ SIZINTISI TESPİT SİSTEMİ

APT'lere karşı etkili bir DLP sistemi için ontolojinin önemi, etkinliği ve sistemden toplanarak ontoloji karar mekanizmasına sunulması gereken bilgiler 3. bölümde detaylı olarak anlatılmıştır. Daha sonrasında DLP sistemlerini atlatmada etkili olan içerik tabanlı saldırılara karşı dayanıklı olarak geliştirilen içerik sınıflandırma algoritması 4. bölümde anlatılmıştır. Bu bölümde, bu özelliklere sahip bir sistemin nasıl gerçeklendiği anlatılmıştır.

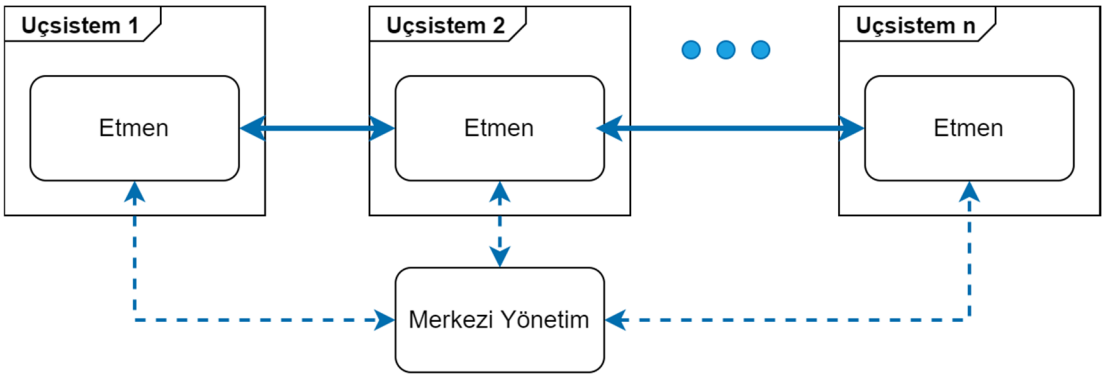
5.1. Motivasyon

Önceki bölümlerde anlatılan modeller ve çıkarım yöntemleri sistemdeki bütün bilgisayarlarda sistem çağrılarının toplanmasını gerektirmektedir. Çevrimiçi çalışan bir sistem oluşturmanın ilk kriteri, sistem çağrılarının toplanması ile beraber değerlendirmeye tabi tutulabilmesidir. Sistem çağrıları, işletim sistemine ait kütüphanelere yapılan çağrılar oldukları için onları toplamanın tek yolu, çağrıyı yapan prosese "enjekte" olarak çağrıyı yaptığı sırada bunu yakalamaktır. Bunu gerçeklemenin yolu ise, işlemleri izleyen uç noktalarda yazılımlara, ajanlara sahip olmaktır. Bu ajanlar, sistem çağrılarının bileşenlerini analiz etmeli ve sonuçları APT'lerin teknik ve taktikleri ile ilişkilendirmeye çalışmalıdır. Önceki bölümlerde anlatıldığı gibi APT'ler ağ üzerine yayılma eğilimindedir. Bu sebeple, ajanlar da sonuçları birbirleriyle paylaşarak sistem genelinde bir tespit gerçekleştirebilmelidir. Aksi halde atağın farklı aşamaları farklı bilgisayarlarda gerçekleştiği durumlarda tespit sağlanamayacaktır. Bunların yanında sistemin bütünlüğünü sağlamak, ajanları izlemek ve analiz sonuçlarını merkezi bir noktadan alabilmek için ise merkezi bir yönetim modülü faydalı olacaktır.

Etmen yazılımının farklı görevleri olması, farklı problemlere aynı anda çözüm sağlaması ve belli bir süre içerisinde çözüm üretecek performansı sergilemesi için çoklu modül yapısında olması gerekmektedir.

5.2. Genel Mimari ve Etmen Mimarisi

Belirtilen gereksinimleri sağlayan bir mimari, genel seviyede bakıldığında, uç noktalarda çalışan etmen yazılımları ve bir merkezi yönetim yazılımından meydana gelmektedir (Şekil 5.1). Etmenler, merkezi yönetimden bağımsız çalışabilmekte ve DLP işlevleri için gerekli analiz, iletişim ve karar verme yeteneklerine sahiptirler. Etmenler birbirleri ile haberleşerek yaptıkları tespitleri paylaşmaktadırlar. Merkezi yönetim yazılımı ise, sistemin takip ve yönetimi için yer almaktadır. Etmenler, sonuçları merkezi yönetim yazılımı ile de paylaşırlar.

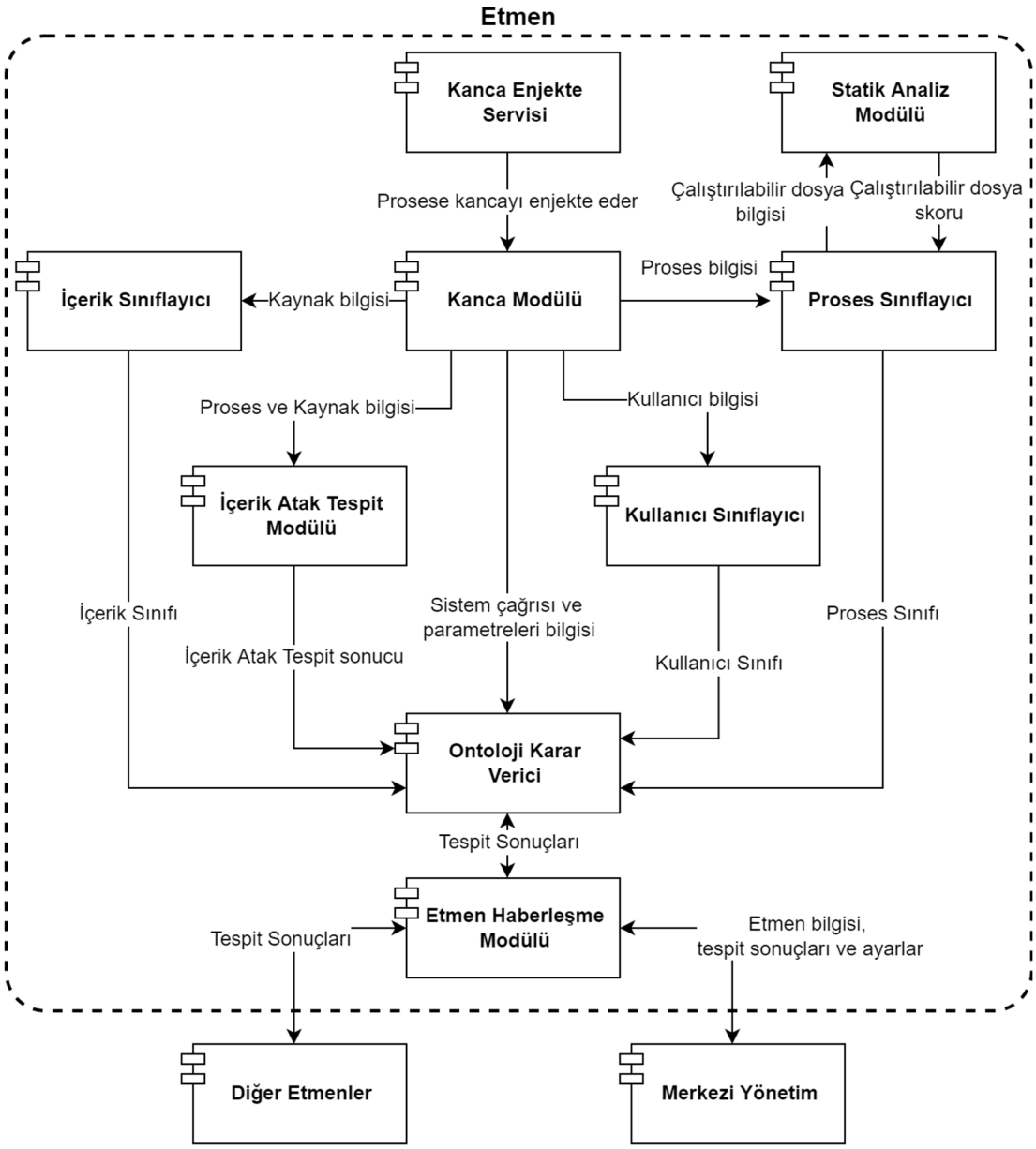


Şekil 5.1. APTONSYS sisteminin üst seviye mimarisi.

Etmen mimarisi ise farklı görevleri icra eden, birbirleri ile hızlı ve asenkron bir ağ protokolü (gRPC) ile haberleşen modüllerden meydana gelmektedir. Etmenin her bir modülü sistem çağrısının farklı bileşenlerini analiz ederek sonuçları ontoloji karar verme aşaması için toplamaktadırlar.

Şekil 5.2'de gösterilen etmen mimarisinde, öncelikle etmenin Kanca Enjekte Edici modülü ile yeni proseslerin yaratılmasını takip ettiği ve bunlara Kanca Modülü'nü enjekte ederek sistem çağrılarını yakaladığı belirtilmektedir. Kancalama işlemi tamamlandıktan sonra, proses tarafından yapılan bütün sistem çağrıları yakalanarak, sistem çağrısına ilişkin gerekli bilgiler toplanmakta ve detaylı analiz için diğer modüllere iletilmektedir. Bu modüllerdeki analiz tamamlandığında, sonuçlar Ontoloji Karar Verici'de (OKV) toplanmaktadır. OKV, bu sistem çağrısının bir APT saldırısının parçası olup olmadığına karar verir. Elde ettiği sonucu diğer etmenler ile paylaşarak sistem seviyesinde bir tespit sağlanır. Tespit bilgisini ayrıca Merkezi Yönetim Yazılımı ile de paylaşarak merkezi bir takip ve yönetim sağlanmış olur.

İlerleyen bölümlerde her bir modül detaylandırılmaktadır.



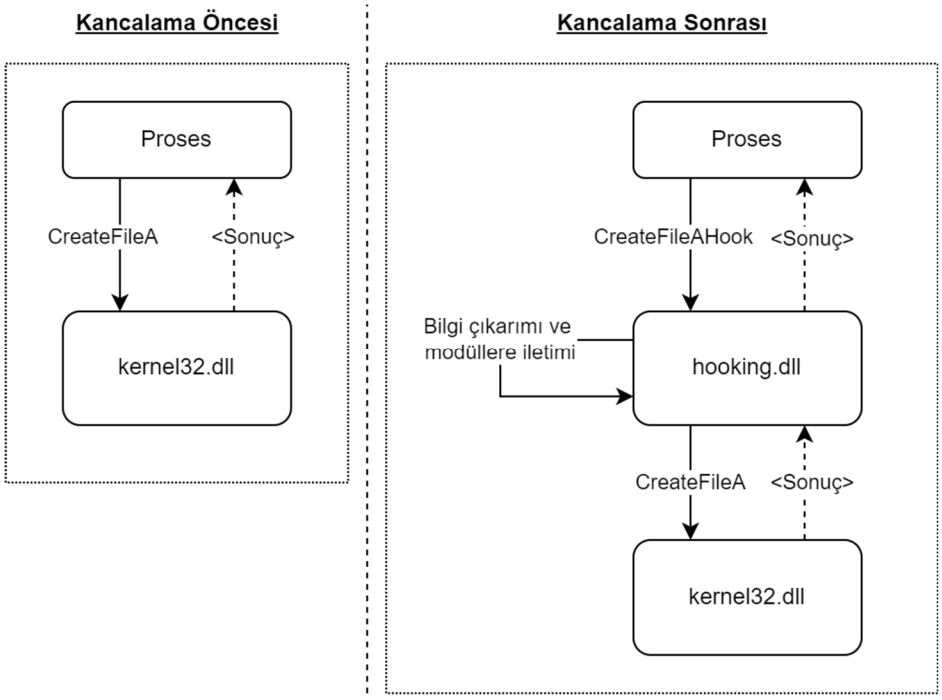
Şekil 5.2. APTONSYS Etmen mimarisi.

5.3. Modüller

5.3.1. Kancalama modülü

"Kancalama" terimi, bir sistem çağrısının yürütülmesi sırasında araya girip bu çağrıyı durdurarak, özel işlemleri gerçekleştirme ve bu özel işlemler tamamlandıktan sonra prosesin gerçek işletim sistemi API çağrısını yapmasına izin verme anlamına gelir. Şekil 5.3'te gösterildiği gibi, kancalama öncesinde hedef proses, bir işletim sistemi API çağrısını (CreateFileA) doğrudan yapabiliyordu. Kancalama sonrasında bu çağrı, Kancalama Modülünün özel işlevine yönlendirilir. Bilgi çıkarma ve iletimi

tamamlandığında, orijinal CreateFileA çağrısının çalışmasına izin verilir. Daha sonra orijinal çağrının sonucu işleme geri döndürülür.



Şekil 5.3. Kancalama öncesi ve sonrası.

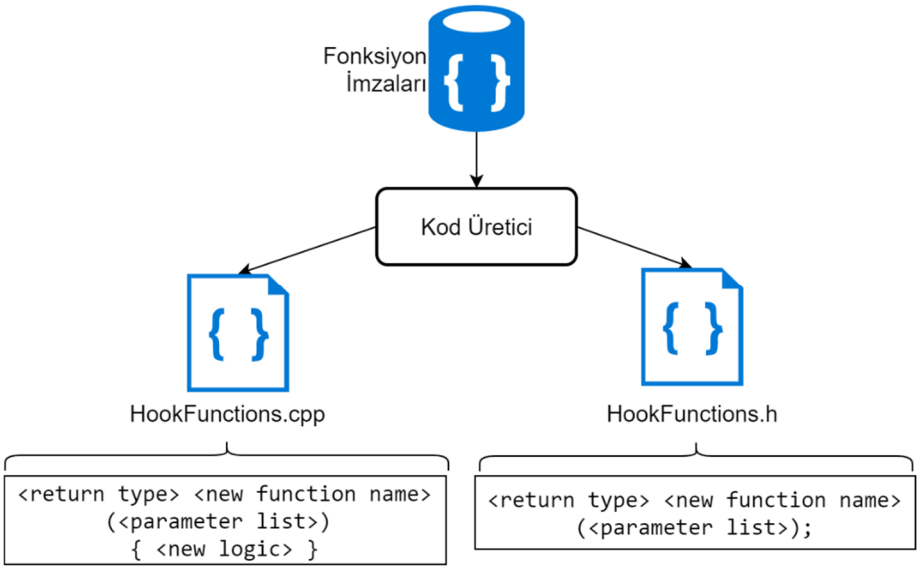
Tez kapsamında sunulan sistemde, sistem çağrısı bilgilerini toplamanın bir yöntemi olarak dinamik bağlanan kütüphane (Dynamic Link Library-DLL) enjeksiyonu ile kancalama yapılmıştır. Bu yöntem ile, sistem çağrısının öğeleri (proses, kullanıcı, sistem çağrısı parametreleri, vb.) hakkında daha fazla kontrol ve bilgi sağlanmaktadır. Ayrıca sistem çağrısı yakalama, işleme ve diğer modüllere gönderme işlemlerinin farklı prosesler içerisinde yapılarak paralelleştirme sağlanması da bu yöntemin önemli bir tercih sebebidir. Sistemin başarımını ve sonuçları göstermek için yeterli olması sebebiyle, kullanıcı seviyesi (user-level) sistem çağrıları seçilmiştir. Yöntemin gerçeklenmesinde EasyHook (EasyHook, 2023) kütüphanesinden faydalanılmış ve bu kütüphane üzerinde çeşitli iyileştirmeler yapılarak performans artışı sağlanmıştır.

EasyHook ve benzeri kütüphanelerde gerekli kancalama işleminin yapılması, ilgili sistem çağrısının birebir aynı imzasına (function signature) sahip bir fonksiyonun tanımlanması ile mümkün olmaktadır. Geliştirilen her bir fonksiyon içerisinde;

- İşletim sistemine ait gerçek fonksiyonun çağırılması
- Karar verme aşaması için gerekli bilgilerin çıkarılması ve diğer modüllere iletilmesi

- Gerçek fonksiyondan gelen sonucun geri döndürülmesi

için gerekli kodların bulunması gerekmektedir. Ayrıca, kancalanan prosesten yeni bir proses yaratılma durumunda özel olarak yeni prosesin de kancalanması gerekmektedir. Bu durum, yaklaşık 2600 sistem çağrısı için otomatik bir kod üretme altyapısının üretilmesini gerektirmiştir. Böylece, sistem çağrılarına ait hazırlanan imza veri tabanları kullanılarak gerekli başlık ve kod dosyaları oluşturulmaktadır (Şekil 5.4).

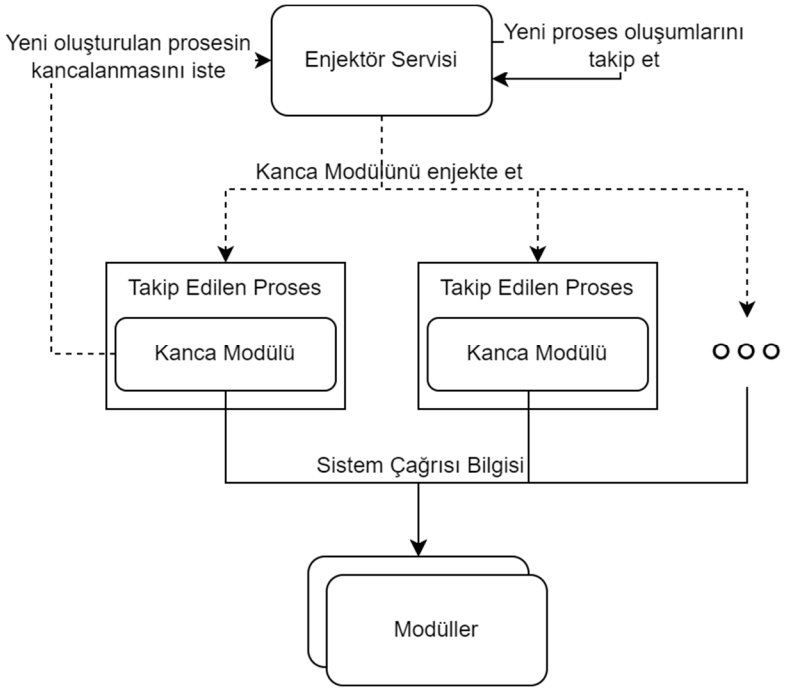


Şekil 5.4. Kod üretici çıktısı.

Sistem çağrısı yakalandığında, sistem çağrısının ayrıntıları ve parametreleri çıkarılır ve ilgili modüllere gönderilir. Daha önceki bölümlerde bahsedildiği gibi, bir sistem çağrısı, API fonksiyonunun adını, sistem çağrısının eriştiği kaynakların yolunu, sistem çağrısını yürüten işlemi ve işlemi sahiplenen kullanıcı hesabı bilgilerini içerir.

Kancalama modülü ayrıca "3.4.1 Sistem çağrılarından APT risk çıkarımı akışı" bölümünde anlatılan ilk seviye çıkarımı yaparak bazı parametrelerin statik eşlemelerini gerçekleştirir. Sistem çağrısının adını Sistem Çağrısı Kategorisi'ne eşler. Benzer şekilde, sistem çağrısının eriştiği kaynağın yolunu ve türünü, ilgili ObjectType ve DataLocation türüne çevirir.

Kancalama işleminin sistemde yaratılan her yeni proses için gerçekleştirilmesi gerekmektedir. Bu sebeple işletim sistemi seviyesinde, yeni oluşturulan prosesleri takip ederek onları henüz herhangi bir sistem çağrısı yapmadan kancalayan bir "Kanca Enjekte Servisi" geliştirilmiştir. Şekil 5.5'te gösterildiği üzere, yeni prosesleri takip ederek kancalama işlemini gerçekleştirmektedir.



Şekil 5.5. Kanca Enjekte Servisi çalışması.

5.3.2. Proses sınıflayıcı ve statik analiz modülü

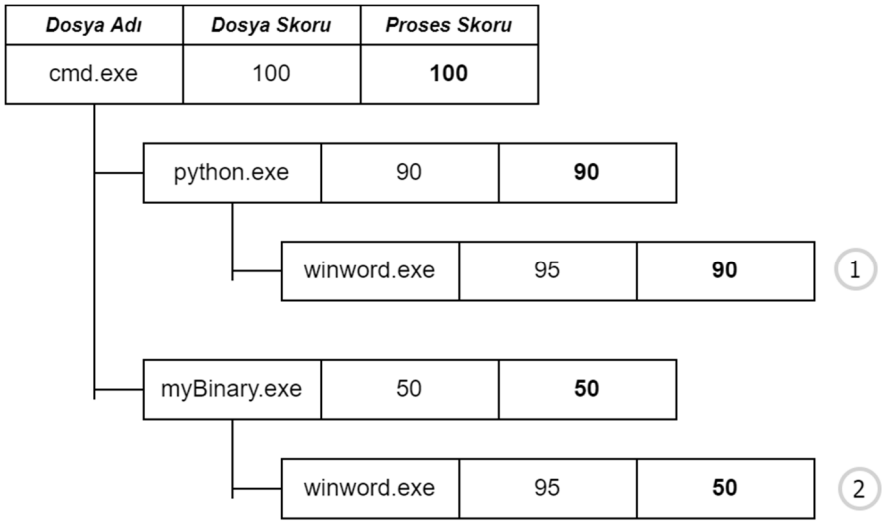
APT davranışı tespitinde, prosesin güvenirliğinin belirlenmesi kritik bir rol oynamaktadır. APT'ler hedef sistemdeki yerleşik programları, komut dosyalarını veya araçları kullanmaya çalışırlar. Ancak, sistemdeki ilk bulaşma genellikle kötü amaçlı yazılımlardan veya tehlikeli dosyalardan kaynaklanır (Han ve ark., 2021; Shabtai ve ark., 2012). Bu sebeple sistemde çalışacak her bir dosyanın bir ön değerlendirilmesinin yapılması gerekir. Bu değerlendirmenin, dosya çalıştırılmadan önce yapılması gerektiği için dosya bilgilerine dayalı statik bir analizin yapılması uygun görülmüştür. Bununla beraber, sistemde tespit edilen yeni bir yürütülebilir dosya kötü amaçlı yazılım olarak sınıflandırılmasa bile, güvenilmeyen bir uygulama olarak kabul edilmeli ve gerçekleştirdiği işlemler dikkatle izlenmelidir. APT'lerin sistemde hazırda var olan ve statik analiz sonucu güvenilir olarak değerlendirilebilecek "masum" uygulamalar üzerinden ataklarını gerçekleştirmeleri, sadece çalıştırılabilir dosyalara değil proseslerin atalık ilişkisi üzerinden takip edilmesi sonucu karar verilmesini gerektirmektedir.

Tez kapsamında sunulan sistemde, Proses Güven Seviyesi (ProcessTrustLevel) özelliği, sistemdeki her proses için hesaplanmaktadır. Bu işlem, iki adımda gerçekleştirilir:

İlk olarak, Statik Analiz Modülü tarafından "itibar puanı" Taşınabilir Yürütülebilir (Portable Executable File – PE File) başlıkları incelenerek statik olarak gerçekleştirilir (Garg & Yadav, 2019; S. Gupta ve ark., 2016; Hassen ve ark., 2017). Bunun sonucunda ilgili çalıştırılabilir dosya için bir skor üretilir.

Ardından bu puan, Proses Sınıflandırıcıya aktarılır. Burada bu ilk puan, ata prosesin itibar puanına göre azaltılır. Bu yaklaşıma göre bir prosesin skoru, kendi çalıştırılabilir dosyasının skoru ve atalarının skoru arasında en küçük olan skor olmaktadır. Bu sayede güvenilmeyen bir proses, güvenilir uygulamaları kullanarak yeni prosesler oluşturduğunda yanlışlıkla güvenilir proses olarak değerlendirilmemiş olur. Bu işlem Şekil 5.6'da gösterilmiştir. Bu şekilde, 1 ve 2 numaralı işlemler her ikisi de dosya puanı 95 olan Winword yürütülebilir dosyasını kullanarak oluşturulmuştur. Ancak bu dosya puanları kendi ata proseslerinin puanlarına düşürülür. Bu sayede güvenilmeyen myBinary.exe'den kaynaklanan Winword prosesi de güvenilmeyen bir proses olarak değerlendirilerek bu prosesin erişimleri kısıtlanabilir.

En son olarak, elde edilen proses skoru, ontoloji değerlendirmesine uygun olarak ilgili ProcessTrustLevel alt-sınıfına çevrilmektedir.



Şekil 5.6. Proses skoru oluşturma akışı.

5.3.3. İçerik sınıflandırma modülü

Sistem çağrısının erişim sağladığı dosya yolları Kancalama Modülü tarafından belirlendikten sonra İçerik Sınıflandırma Modülü'ne iletilir. Bu modül, 4. bölümde anlatılan yöntemi içeren bir içerik sınıflandırma algoritması kullanarak içeriğin gizlilik seviyesine karar verir. Elde edilen sonuç ContentPrivacyLevel sınıfının alt

sınıflarından olan Kurumsal Gizli (G), Hizmete Özel (Ö) veya Tasnif Dışı (TD) olmak üzere 3 sınıftan birisi olarak OKV'ye iletilir.

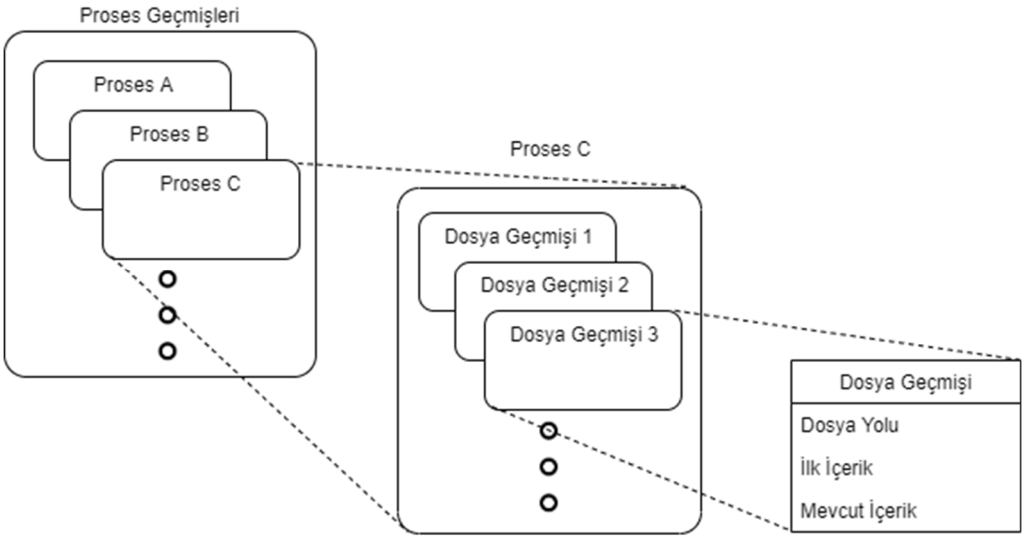
5.3.4. İçerik saldırısı tespit modülü

Bu modül 4.3 bölümünde detaylandırılan İçerik Atak Tespiti yöntemini uygulayarak ilgili prosesin bir içerik atağı yapıp yapmadığını ve dolayısıyla sistem çağrısının bir içerik atağının parçası olup olmadığını tespit eder. Elde edilen sonuç ContentModification sınıfının alt sınıflarından birisi olarak OKV'ye iletilir.

İçerik saldırısı tespit modülü, dosya operasyonları ile ilgilendiği için üç kategorideki sistem çağrılarını dikkate almaktadır:

- CREATE_FILE: Yeni dosya oluşturma, dosyaya okumak ya da yazmak için erişim sağlanırken ve dosyaların işletim sistemi bazında tutulan üstveri bilgilerinin okunması gibi işlemlerde kullanılmaktadır.
- READ_FILE: Var olan bir dosyanın içeriğinin getirilmesini sağlamaktadır.
- WRITE_FILE: Dosyaya yazma işlemi yapılırken kullanılmaktadır.

Bu modül gerçeklenirken, APT'lerin verileri Prosesler Arası Haberleşme (Inter-process Communication-IPC) ile veri kaçırabilmesi de göz önünde bulundurulmuştur. Bu tip bir atakta A prosesi veriyi okumakta, içeriği değiştirerek ya da değiştirmeden IPC ile B prosesine aktarmakta, B prosesi de içeriği hedefe yazmaktadır. Buna karşı önlem olarak İçerik saldırısı tespit modülünün sistemde takip edilen bütün proseslerin geçmiş dosya erişimleri ve dosyaların da mevcut içerikleri saklanmaktadır (Şekil 5.7). Herhangi bir proses, bir dosyaya yazma yapmak istediği zaman bütün proseslerin açık olan dosyalarının hem mevcut içerikleri hem de ilk içerikleri ile yazılmak istenen içerik karşılaştırılarak 4.3 bölümünde anlatılan yöntemler kullanılarak atak tespiti yapılmaktadır. Sadece mevcut içeriklerin değil ilk içeriklerin de takip edilmesinin sebebi, zararlının değişikliği bir kerede değil de adım adım yazması durumunda da tespit sağlayabilmektir.



Şekil 5.7. Proses içerik takibi öğeleri.

5.3.5. Kullanıcı sınıflandırma modülü

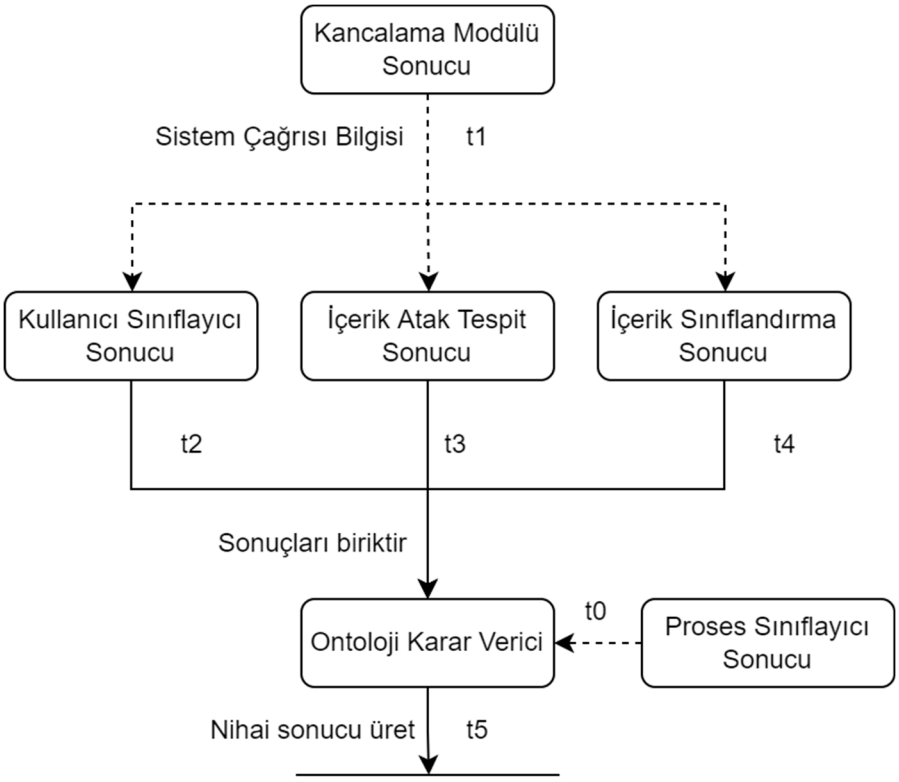
İşlemlerde kullanılan kullanıcı hesabı, APT davranışını tespit etmede başka bir önemli bir göstergedir. Bazı APT teknikleri belirli kullanıcı hesaplarının kullanılmasını gerektirir. Örneğin, Windows işletim sisteminde Erişilebilirlik Özellikleri (T1015) tekniği Sistem kullanıcısı kullanılarak çalıştırılır (MITRE, 2019a).

Bu amaçla, sunulan sistemde kullanıcı hesabı, APT davranışını belirlemede bir özellik olarak kabul edilir. Sistemdeki diğer özellikler gibi, kullanıcı hesapları, APT saldırılarında kullanılan bilinen türler halinde gruplandırılır.

5.3.6. Ontolojik karar verici modül

Sistemin nihai karar işleminin gerçekleştiği modüldür. Sınıflandırıcılardan gelen sonuçlar bir araya getirilerek ontoloji için gerekli öğeler oluşturulmakta ve ontoloji içerisine tanımlanmış kurallara göre olası APT etkinliğinin tespiti sağlanmaktadır.

Etmen, dağıtık bir sistem olduğu için ve farklı sınıflandırıcıların sonuçları farklı zamanlarda OKV'ye ulaştığı için OKV içerisinde gelen verilerin senkronizasyonunun sağlanması gerekmektedir. Şekil 5.8'de gösterildiği üzere, bir proses kancalandığı zaman öncelikle proses sınıflayıcıdan ve bir seferlik sonuç gelmektedir. Daha sonrasında her yeni sistem çağrısı ile farklı sınıflayıcılar değerlendirmelerini yapmakta ve OKV'ye sonuçlarını iletmektedirler. OKV, gelen sonuçları biriktirmekte ve bütün modüllerden sonuçlar geldikten sonra karar aşamasına geçmektedir.



Şekil 5.8. OKV'de modüllerden gelen sonuçların senkronize edilmesi.

Karar aşamasında ilk olarak değerlendirilen özellik, herhangi bir APT tekniğini olup olmadığıdır. Teknik Tespit Kuralları bölümünde detaylandırılan kurallar çerçevesinde verilen karar neticesinde bir APT tekniği tespit edilirse Şekil 5.9'da gösterildiği gibi farklı işlem aşamaları bir arada çalıştırılmaktadır.

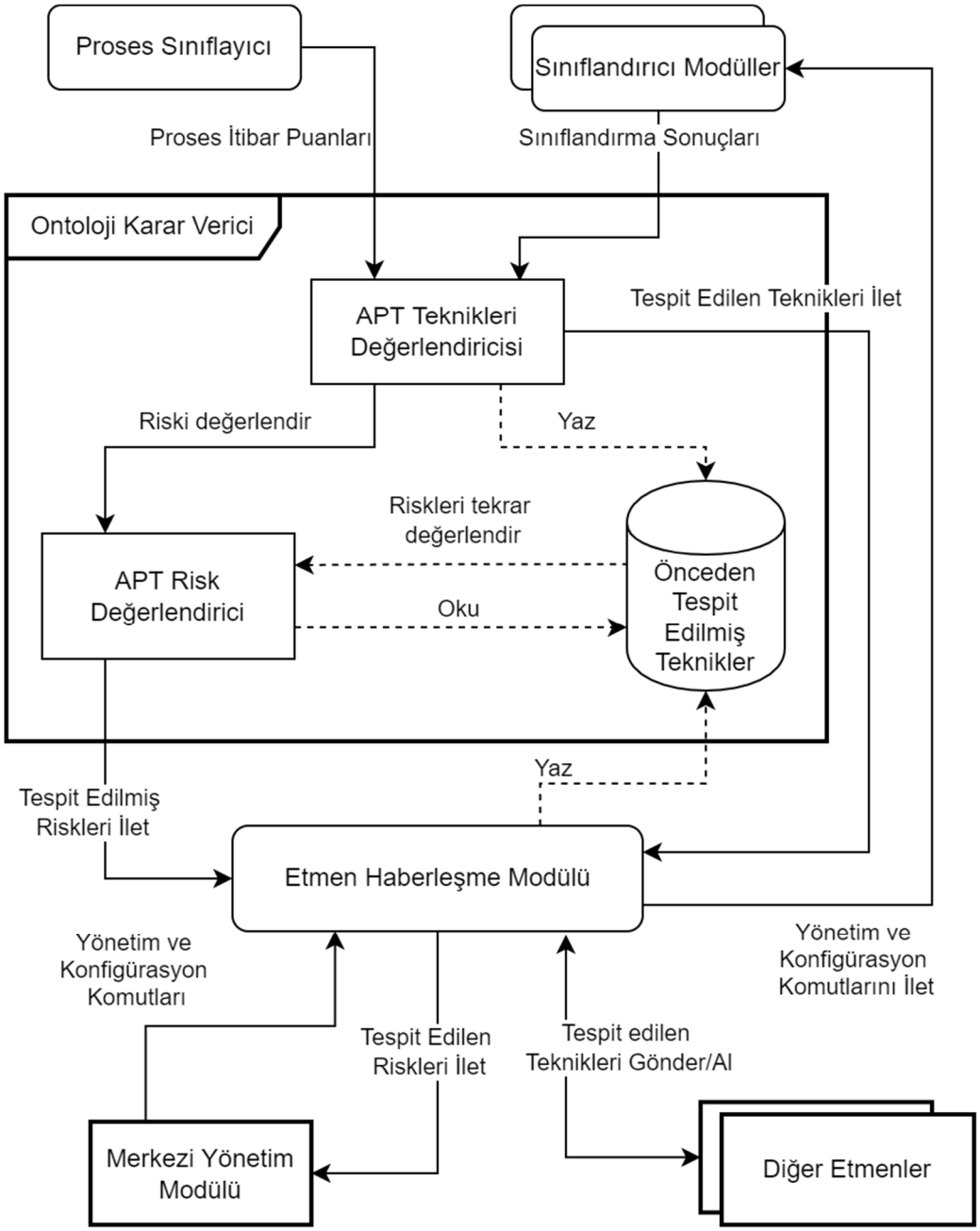
APT tekniği tespiti durumunda, tespit edilen teknikler APT Risk değerlendirici içerisinde değerlendirilerek sistemdeki APT riski ve APT veri sızdırma olasılığı değerlendirilmektedir. Tespit edilen teknikler, ayrıca diğer etmenlere de iletilerek onların da sistem genelindeki tehdit durumunu öğrenmeleri ve risk değerlendirmesini yapmaları sağlanmaktadır. APT Risk değerlendirmesi sonucu risk durumunda bir değişiklik olursa sistem yöneticisinin durum hakkında bilgilendirilmesi için Merkezi Kontrol Yazılımına bilgi aktarılmaktadır.

5.4. Modüller Arası Haberleşme

Sunulan sistemin farklı modüllerden oluşması, gerek birden çok proses ile sınıflandırma yapılabilmesine imkân sunarak performans kazancı sağlamakta, gerekse de farklı sınıflayıcıların farklı yazılım dillerinde gerçeklenebilmesine imkân sunmaktadır. Ayrıca, kancalama modülünün kancalanan hedef proseslerin altında

çalışması gerektiğinden en azından kancalama modülünün ayrı bir modül olmasını gerektirmiştir. Bu sebeple, sunulan sistemin makul sürede karar verebilmesi için modüller arası iletişimin etkin ve hızlı olarak sağlanması gerekmiştir. Ayrıca, sadece etmen modülleri arasında değil, etmenler arasında da benzer bir iletişim çözümü sunulması gerekmektedir. Bu sebeple seçilecek yöntemin sadece aynı bilgisayar içerisinde değil, farklı bilgisayarlar arasında da iletişime imkân sunması önemli olmuştur.

Bu sebeplerle, yapılan değerlendirme sonucu gRPC (gRPC, 2023) protokolü modüller ve etmenler arası iletişim için tercih edilmiştir. Performans açısından, HTTP/2'yi kullanan gPRC, birden fazla isteği tek bir bağlantı üzerinden çoğaltmayı desteklemekte ve başlık sıkıştırma gibi özellikleri kullanarak verimi arttırmaktadır. Bunun yanında iki taraflı iletişimi desteklemesi ve asenkron iletişime imkân sunması da önemli bir tercih sebebi olmuştur. Farklı yazılım dillerini desteklemesi sayesinde ise modüllerin farklı yazılım dillerinde geliştirilmesine imkân sağlamıştır.



Şekil 5.9. Ontoloji Karar Verici, teknik, risk ve kural akışı.

5.5. Etmenler Arası Veri Paylaşımı ve Bütüncül Karar Mekanizması

Etmenlerin birbirleri ile veri paylaşımı yapabilmelerinin ilk adımı birbirlerini keşfetmeleridir. Bu konuda farklı yöntemler mevcuttur (Falco & Robiolo, 2019):

Merkezi Dizin Hizmeti: Bu yaklaşımda, tüm mevcut etmenleri ve yeteneklerini kaydeden merkezi bir dizin hizmeti bulunur. Etmenler, kendilerini bu dizine kaydedebilir ve diğerlerini arayabilirler. Bu yöntem, göreceli olarak küçük ölçekli sistemler için merkezi bir hizmetin yükü kaldırabileceği durumlar için uygundur.

Etmen İlanı: Etmenler periyodik olarak varlıklarını ve yeteneklerini ağa yayınlayabilirler. Diğer etmenler bu ilanları dinler ve mevcut etmenlerin bir listesini tutarlar. Bu yaklaşım, etmenlerin sık sık sisteme katıldığı veya ayrıldığı dinamik ortamlar için uygundur.

Hizmet Keşif Protokolleri: Birçok çoklu etmen sistemi, yerel ağdaki hizmetleri ve etmenleri bulmak için DNS-SD (DNS Hizmet Keşfi) veya mDNS (Çoklu DNS) gibi standartlaştırılmış hizmet keşif protokollerini kullanır. Bu protokoller özellikle IoT ve ağlı ortamlarda yerel keşif için kullanışlıdır.

Noktadan Noktaya Keşif: Etmenler, birbirlerini doğrudan noktadan noktaya iletişim yoluyla bulabilirler. Bu yaklaşım, merkezi bir dizinin bulunmadığı dağıtılmış sistemlerde yaygındır. Etmenler bilgi alışverişi yapar veya yeteneklerini sorgularlar.

Hiyerarşik Keşif: Bazı sistemlerde, daha yüksek seviyedeki etmenler, daha düşük seviyedeki etmenleri bulma ve bu bilgiyi sürdürme sorumluluğuna sahiptir. Bu yaklaşım, büyük ölçekli, düzenli sistemler için uygundur.

Üstüne Geçme Ağları: Etmenler, keşifi kolaylaştırmak için üstüne geçme ağları oluşturabilirler. Bu ağlar, coğrafi yakınlık, yetenekler veya paylaşılan amaçlar gibi belirli kriterlere dayalı olarak yapılandırılabilir. Etmenler aynı üstüne geçme ağı içinde diğerlerini keşfedebilirler.

Etmen Tanıtımı: Etmenler, birbirlerini güvendiğimiz bir üçüncü taraf veya bilinen bir aracı aracılığıyla birbirlerine tanıtabilirler. Bu yaklaşım, etmenlerin önerilere veya tavsiyelere dayalı olarak belirli diğer etmenlerle bağlantı kurmaları gerektiğinde kullanışlıdır.

Coğrafi veya Yakınlık Temelli Keşif: Bazı durumlarda, etmenler, Bluetooth, Wi-Fi veya RFID gibi teknolojileri kullanarak fiziksel yakınlığa dayalı olarak birbirlerini bulabilirler. Bu, IoT ve mobil uygulamalarda yaygındır.

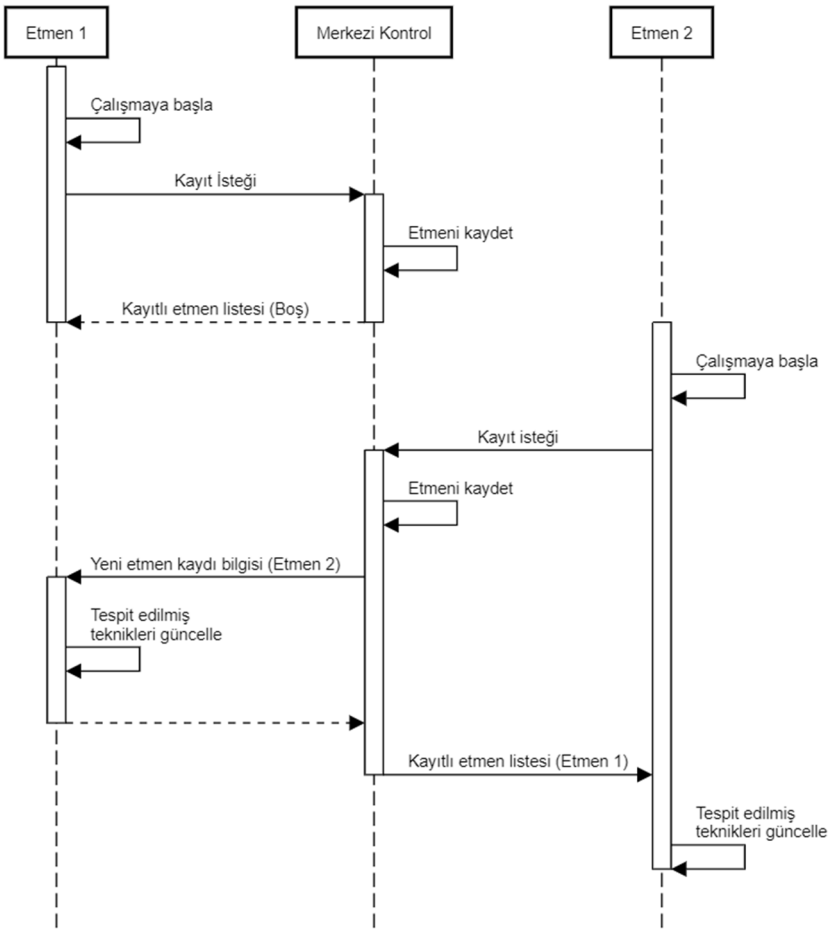
İçerik Temelli Keşif: Etmenler, yetenekleri veya hizmetleri ile ilişkilendirilen belirli içerik, veri veya anahtar kelimelere dayalı olarak diğer etmenleri arayabilirler. Bu, bilgi alımı sistemleri için kullanışlıdır.

Tez kapsamında kurumsal bir sistemin odağa alınması ve takip için merkezi yönetim yazılımının halihazırda bulunması sebebiyle etmenlerin birbirlerini merkezi yönetim sistemi üzerinden keşfetmelerinin uygun olacağı değerlendirilmiştir. Ancak

etmenlerin merkezi yazılıma tamamen bağımlı olmalarını önlemek için etmen kaydının ve başlatılmasının etmenler tarafından gerçekleştirilmesi sağlanmıştır. Bu yönüyle, sunulan sistem literatürdeki Merkezi Dizin Hizmeti ve Etmen İlanı yöntemlerinin bir birleşimidir.

Buna göre etmen kayıt ve keşif akışı Şekil 5.10'daki gibi gerçeklenmiştir:

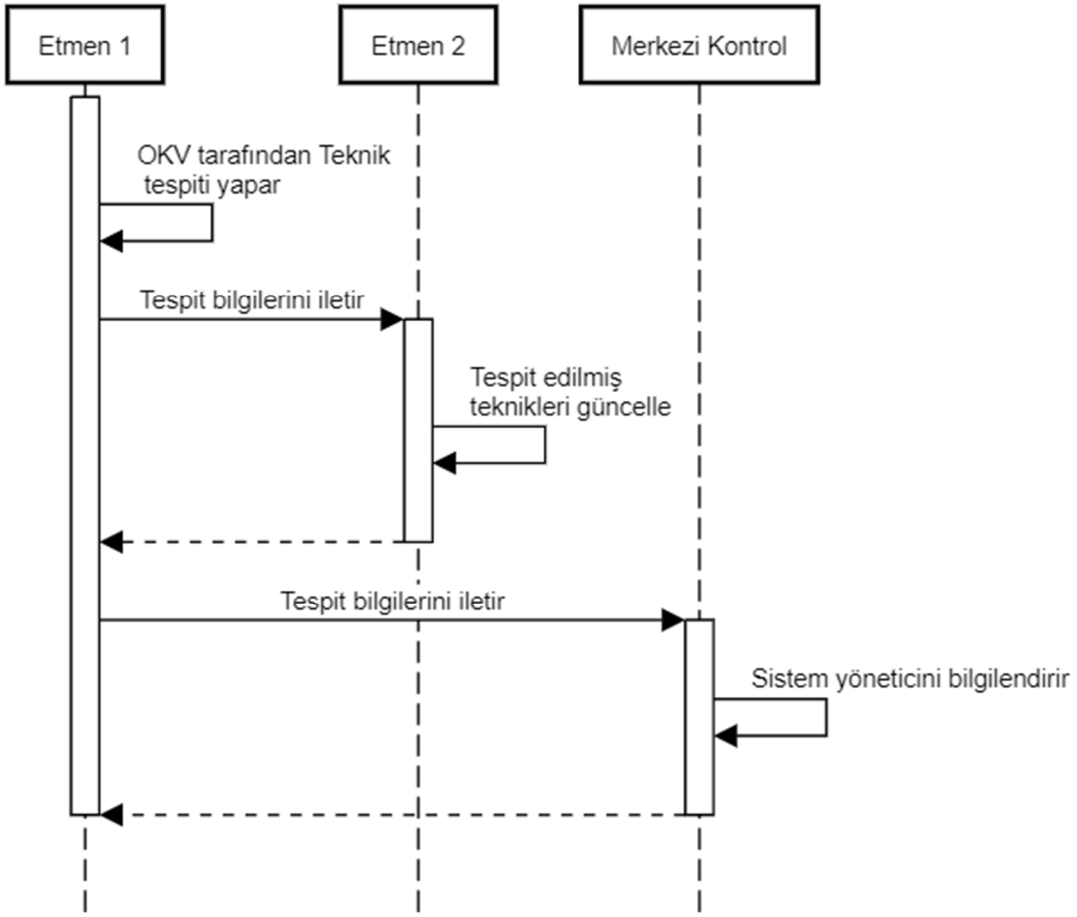
- Bir etmen çalışmaya başladığı zaman MKY'ye Kayıt mesajı gönderir.
- MKY, Kayıt mesajını aldığında
    - Mesajı yollayan etmene, sistemde var olan diğer etmenlerin iletişim bilgilerini (Host ve port) ve onlar tarafından daha önceden tespit edilmiş Teknikleri yollar.
    - Diğer etmenlere, bu yeni kaydolan etmenin iletişim bilgilerini yollar.
- Bu aşamadan sonra etmenin MKY'ye bağımlılığı kalmamış olur.



Şekil 5.10. Etmen Kayıt Akışı.

Etmenler, çalışma zamanında ise bir APT tekniği tespit ettikleri zaman birbirlerine bu bilgiyi gönderirler (Şekil 5.11). Bu sayede sistem genelinde bütüncül bir karar

mekanizması çalışmış olur ve bir bilgisayarda başlayan APT saldırısının başla bir bilgisayarda veri kaçırma aşamasına geçmesi durumunda APT'nin bu davranıii tespit edilebilmiş olur. Etmenler benzer şekilde APT risk tespitini de elde ettikleri ortak bilgileri kullanarak bağımsız olarak yaparlar. Sistem Yöneticisini bilgilendirme amaçlı olarak da MKY'ye tespit bilgilerini gönderirler.



Şekil 5.11. Etmenlerin tespit bilgilerini paylaşımı.

6. DENEYSEL ÇALIŞMALAR VE BULGULAR

Bu tez çalışmasında önerilen modeller ve sistem iki aşamalı olarak test edilerek başarımı gösterilmiştir. Öncelikle önerilen ontoloji modeli sistemden toplanan verilerle bağımsız olarak test edilmiştir. İçerik sınıflandırma algoritması da veri kümeleri ile bağımsız olarak test edilmiştir. Nihai olarak, kurulan ağ üzerinde APT ataklarının icra edilmesi ve bunun neticesinde sistem tarafından tespitlerin yapılması ile tez kapsamında sunulan etmen tabanlı APT veri sızıntısı tespit sisteminin başarımı bütüncül olarak sergilenmiştir.

6.1. Ontoloji Karar Verici Bulguları

OKV deneysel çalışmaları, OKV'nin sistemden bağımsız ve tekil olarak, çevrimdışı test edilmesi ile gerçekleştirilmiştir. Bu noktada uygulanan yöntem ve sonuçlar aşağıdaki bölümlerde detaylandırılmıştır.

6.1.1. Veri kümesi

Çevrimdışı testler, genel olarak literatürde var olan veri kümelerinin ilgili çözüme sunulması ile gerçekleştirilir. İçerik sınıflandırma algoritmasının testlerinde de takip edilen bu yöntemi OKV için uygulamak mümkün olmamıştır. Literatürde yer alan, zararlı yazılımların saldırı ve sistemde bıraktıkları izleri barındıran (ACM SIGCOMM, 2008; DEF CON, 2022; MIT Lincoln Laboratory, 2000; The UCI KDD Archive, 1999; Turcotte ve ark., 2018) bulunmaktadır. Ancak, bu veri kümeleri ağ paketlerini ve bazı durumlarda sistem günlüklerini ve olayları içermekle beraber, sistem çağrılarını içermemektedir. Bu sebeple sistem çağrılarına dayalı olan OKV'nin testleri için uygun değillerdir.

Sistem çağrılarını içeren (AZSecure-data, 2017; Catak ve ark., 2020) gibi veri kümeleri ise yalnızca sistem çağrılarını içermekle beraber, bu çağrıların proses, atak gibi bağlam bilgilerini içermemektedir. Bunun yanında tez kapsamına uygun olarak APT'lerin gerçekleştirdiği atak senaryolarını sunmamaktadır.

Bu nedenlerle, APT'lerin gerçekleştirdiği saldırılara ait sistem çağrıları, işlem ve içerik bilgilerini elde edebilmek için atakların gerçekleştirilmesi ve bu sırada gerekli

verilerin toplanması yöntemi ile gerekli veriler oluşturulmuştur. Bu noktada, mevcut APT araştırmalarına daha iyi uyum sağlamak amacıyla, test senaryoları için APT analiz raporları göz önünde bulundurulmuştur. Bu raporlar, APT'lerin saldırı senaryolarını ve APT'ler tarafından kullanılan yöntemleri açıklamaktadır. Literatürü takip ve katkı amacıyla, bu saldırı senaryolarının açık kaynaklı APT simülasyon araçları kullanarak gerçekleştirilmesi yöntemi, en uygun yaklaşım olarak kabul edilmiştir.

Veri kümesi oluşturma yöntemi belirlendikten sonra 3.4 Ontoloji Tespit Kuralları bölümünde belirtildiği üzere tez kapsamında örnek olarak ele alınan APT3 için test senaryosu oluşturulmuştur. Bu noktada aynı bölümde bahsi geçen analiz raporları dikkate alınarak MITRE'nin APT3 Saldırı Planı'na benzer bir senaryo oluşturulmuştur. Tablo 6.1'deki APT teknikleri sistem üzerinde uygulanarak sistem çağrıları ve ilişkili veriler Kancalama Modülü yardımıyla toplanmıştır. Tablo 6.1'de gösterildiği üzere, APT3'ün uyguladığı tekniklerden belli bir kısmı açık kaynaklı olan Atomic Red Team - ART (*Atomic Red Team*, 2020) ve Purple Team Automation - PTA (*Purple Team ATT&CK Automation*, 2021) sistemlerinden faydalanılarak gerçekleştirilmiştir. T1022 için ise içerik tabanlı bir saldırıyı modelleyebilmek için kendimizin geliştirdiği bir araç ile 4.3 bölümünde anlatılan saldırılardan Vigenère saldırısı gerçekleştirilmiştir.

APT teknikleri uygulanarak sistem çağrıları ve ilgili bilgiler toplandıktan sonra elde edilen sistem çağrısı verileri, proses ve kullanıcı hesabı ayrıntıları ile zenginleştirilmiştir. Ayrıca, sistem çağrısının parametrelerinden olan hedef dosyaların yolları da kayıt verilerine dahil edilmiştir. Sonuç olarak oluşan kayıt verilerinden bir örnek Şekil 6.1'de sergilenmiştir. Bu verideki bilgiler kullanılarak ontolojinin çıkarım yapması için gerekli öğeler elde edilmektedir. Şekil 6.1'deki veride, SystemCallInfo alanının Type parametresi ontoloji içerisindeki SystemCallCategory sınıfını belirler. ProcessInfo alanının FilePath parametresi ontolojideki Executable sınıfının özelliğini belirler. PID ve CalledFunction özellikleri ise SystemCall isCalledByProcess Process ilişkisinin oluşturulmasını sağlar. Verilerde, aynı sistem çağrısının farklı zamanlarda ve farklı işlemler tarafından çağrılması durumunu ayırt edebilmek için ise Index ve PID parametreleri verilere eklenmiştir.

Tablo 6.1. OKV testleri için uygulanan APT teknikleri.

APT Teknik Kodu	APT Teknik Adı	APT Taktik Adı	Atak Gerçeklemesi
T1136	Create Account	Persistence	ART
T1015	Accessibility Features	Persistence	PTA
T1055	Process Injection	Privilege Escalation	ART
T1015	Accessibility Features	Privilege Escalation	PTA
T1003	Credential Dumping	Credential Access	PTA
T1033	System Owner / User Discovery	Discovery	ART
T1016	System Network Configuration Discovery	Discovery	ART
T1076	Remote Desktop Protocol	Lateral Movement	ART
T1074	Data Staged	Collection	ART
T1041	Exfiltration Over Command-and-Control Channel	Exfiltration	ART
T1022	Data Encrypted	Exfiltration	Özel


```
{
  "SystemCallInfo": {
    "Index": 18,
    "HostId": "VICTIM.sakaryadlp.local",
    "UserId": "staff",
    "Time": 1605444198957961000,
    "Pid": 7664,
    "Tid": 5360,
    "Type": "199",
    "CalledFunction": "CreateProcessW",
    "Parameter1": "C:\\Windows\\System32\\chcp.com",
    "Parameter2": "\"C:\\Windows\\System32\\chcp.com\"   437 "
  },
  "ProcessInfo": {
    "PID": 7664,
    "FilePath": "C:\\Windows\\system32\\cmd.exe",
    "Arguments": "\"cmd.exe\"",
    "$type": "ProcessInfo"
  },
}
```

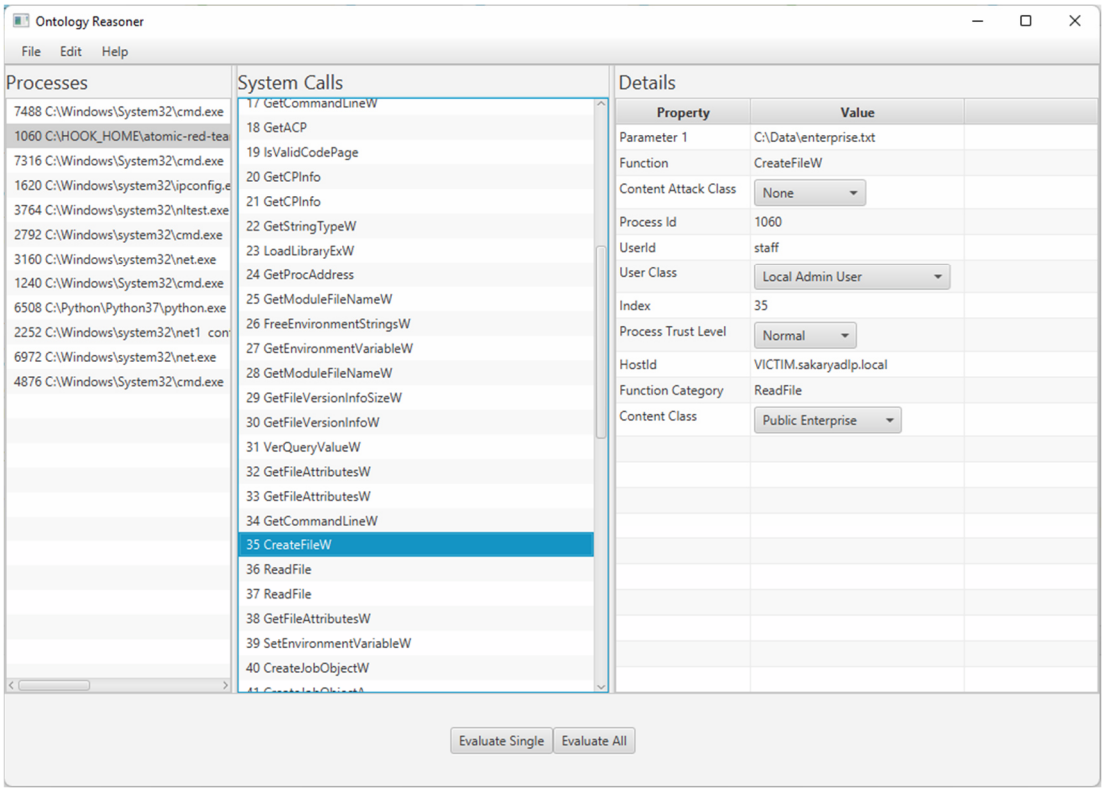

Şekil 6.1. OKV test verisi kayıt örneği.

6.1.2. Testlerin gerçekleştirilmesi ve sonuçlar

Veri kümesi oluşturulduktan sonra, toplanan veriler OKV'ye beslenerek sonuçlar bir arayüz üzerinde sergilenmiştir. Bu test sürecine ilişkin akış Şekil 6.2'de gösterilmiştir.

Şekil 6.2. OKV bağımsız test akışı.

OKV'nin testlerinin bağımsız gerçekleştirilebilmesi için bir arayüz tasarlanmış ve okunan veriler bu arayüz üzerinde sergilenerek proses sınıfı, içerik sınıfı gibi verilerde olmayan detayların kullanıcı tarafından eklenmesi sağlanmıştır. Şekil 6.3'te sergilenen arayüzde, kayıtların proseslere göre gruplandığı ve her bir sistem çağrısının seçilebildiği görülmektedir. Arayüz üzerinde birden çok sistem çağrısı için bahsi geçen özelliklerin ayarlanabilmesine imkân sunulmaktadır.



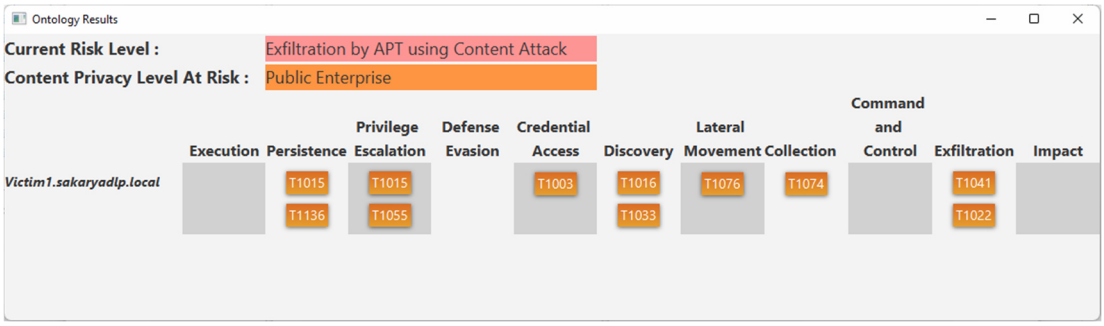
Şekil 6.3. OKV test arayüzü.

Arayüz üzerinde ilgili seçimler yapıldıktan sonra "Evaluate Single/All" seçeneği seçilir. Bu seçimle beraber OKV ile çıkarım yapılması sağlanır. Çıkarım aşaması, öncelikle ilgili sistem çağrısı bilgilerinin ontoloji bireylerine çevrilmesi ile başlar. Şekil 6.3'te seçili olan CreateFileW sistem çağrısı için Tablo 6.2'deki bireyler otomatik olarak yaratılmaktadır. Sonrasında bu bireyler ile 3 bölümünde anlatılan yönteme göre ontoloji çıkarım yapma süreci çalışmaktadır. Bu işlem bütün sistem çağrıları için ayrı ayrı gerçekleştirilerek sonuçta elde edilen bütün AptTechnique, AptTactic ve AptRisk sınıfları kullanılarak yine arayüz üzerinde sergilenmektedir. APT3'ün Tablo 6.1'de gösterilen teknikleri neticesinde ilgili bütün APT teknikleri tespit edilerek APT risk seviyesi en üst seviye olan "Exfiltration by APT using Content Attack" seviyesi olarak belirlenmiştir. İlgili sonuç ekranı Şekil 6.4'te gösterilmiştir.

Benzer testler APT3'ün 0 bölümünde bahsedilen bütün APT teknikleri için gerçekleştirilerek tespitlerin başarılı olarak yapıldığı gözlemlenmiştir.

Tablo 6.2. OKV testlerinde otomatik oluşturulan birey örnekleri.

Birey	Sınıf	Nesne Özelliği	Sınır
SystemCall_35	SystemCall	isCalledByProcess	Process_1060
		systemCallOriginatedAtHost	Host_victim
		hasSystemCallCategory	SysCallCat_CreateFile
		hasContent	Content_Ent
		hasTargetObjectType	TargetObjType_DocFile
		isCalledByUser	User_staff
Process_1060	NormalRepProcess	hasProcessTrustLevel	ProcTrustLev_Norm
ProcTrustLev_Norm	ProcessTrustLevelNormal		
Host_victim	Host		
SysCallCat_CreateFile	CreateFile		
Content_Ent	Content	hasContentPrivacyLevel	ContentPrivLev_PubEnt
		hasContentModification	AllowedModification
ContentPrivLev_PubEnt	EnterprisePublic		
TargetObjType_DocFile	DocumentFile		
User_staff	LocalAdminUser		



Şekil 6.4. OKV sonuç arayüzü.

6.2. İçerik Sınıflandırma Algoritması Bulguları

İçerik sınıflandırma algoritmasının testlerinde tez kapsamında kurulan sisteme uygun olarak verilen metinlerin "Kurumsal Gizli", "Hizmete Özel" ve "Tasnif Dışı" olmak üzere 3 farklı sınıfta sınıflandırılmasındaki başarım test edilmiştir. Literatürde DLP alanında kullanılan veri kümelerindeki metinler kullanılarak, saldırı öncesi ve sonrası sınıflandırma başarımı karşılaştırılmıştır. İlerleyen bölümlerde bu aşamalar detaylandırılarak sonuçlar sergilenmektedir.

6.2.1. Veri kümesi

Eğitim aşamasının ilk basamağı, uygun veri kümesini (veri setini) oluşturmaktır. Çalışmanın güvenirliğini göstermesi açısından DLP alanında kullanılan bir veri kümesi seçilmesi önemli bir kriter olmuştur. Bu sebeple (Hart ve ark., 2011) çalışmasında kullanılan veri kümeleri kullanılmıştır. Bu kümelerden Dyncorp, Mormon ve Transcendental Meditation (TM) kümeleri, ilgili kurumların Wikileaks'e sızmış olan ve bunun yanında bu kurumların resmi sitelerinden elde edilen dokümanlardan oluşmaktadır.

Dyncorp'a ait gizli nitelikteki dokümanlardan 2 tanesi okunabilmiş ve "Kurumsal Gizli" kategorisine eklenmiştir. Resmi sitelerinden elde edilen 198 PDF dosyası ise "Hizmete Özel" grubuna eklenmiştir.

Mormon'a ait kitaplardan bazıları seçilerek sınıflama aşamasında yeterli örnek oluşturması için 1000 karakterlik parçalara ayrılmıştır. Bu metinler "Kurumsal Gizli" olarak etiketlenmiştir. Ayrıca kendi internet sitelerinden 3 adet pdf indirilmiş ve bunlar da 1000 karakterlik parçalara ayrılarak "Hizmete Özel" olarak etiketlenmiştir. Bunun sonucunda 593 "Kurumsal Gizli", 2541 adet "Hizmete Özel" metin elde edilmiştir.

TM'ye ait 85 adet doküman "Kurumsal Gizli" grubuna eklenmiştir. İnternet sitelerinden alınan 120 adet sayfa da "Hizmete Özel" grubuna eklenmiştir.

Bunlara ek olarak DBpedia veri kümesi de kullanılmıştır. DBpedia, Wikipedia'dan yapılandırılmış bilgi çıkarmak için topluluk kaynaklı bir çabadır (Lehmann ve ark., 2015). (Zhang ve ark., 2015) çalışmasında DBpedia'nın 2014 versiyonundan seçim yaparak etiketleme gerçekleştirmiştir. (Xiang Zhang, 2023) çalışmasında sunulan bu veri kümesinden 2000 adet doküman alınarak "Kurumsal Olmayan" olarak etiketlenmiştir.Bu işlemler sonucu oluşturulan veri kümeleri bilgileri Tablo 6.3'te sergilenmiştir.

Tablo 6.3. İçerik sınıflandırma testleri veri kümeleri bilgileri.

Etiket	DynCorp	Mormon	TM	DBPedia	Toplam
Kurumsal Gizli (G)	2	593	85	-	680
Hizmete Özel (Ö)	198	2541	120	-	2859
Tasnif Dışı (TD)	-	-	-	2000	2000

Test işlemleri için öncelikle modelin eğitim aşamasının gerçekleşmesi gerekmektedir. Bu sebeple, veri kümelerinin yaklaşık %70'lik kısmı eğitim, %30'luk kısmı test için ayrılmıştır. Buna göre oluşan kümeler Tablo 6.4'te sergilenmiştir.

Tablo 6.4. İçerik sınıflandırma testleri eğitim ve test veri kümeleri sayıları.

Etiket	Eğitim	Test	Toplam
Kurumsal Gizli (G)	466	214	680
Hizmete Özel (Ö)	2023	836	2859
Tasnif Dışı (TD)	1387	613	2000

6.2.2. Değerlendirme kriterleri

Sınıflandırma sonuçlarının değerlendirilmesinde, bu alanda yaygın olarak kullanılan f1-skor hesaplaması ve doğruluk (accuracy) değeri ölçüt olarak alınmıştır. Bu değerlerin hesaplamasında sınıflandırma sonuçlarının başarımı kullanılmaktadır. Bir sınıflandırma sonucunda 4 farklı sonuç elde edilebilir:

1. Gerçek Pozitif (True Positive, TP)
   Sınıflandırma işlemi, sonucu pozitif olarak tahmin etmiştir ve gerçekte de sonuç pozitiftir. Bu yönüyle başarılı bir sınıflandırma yapılmıştır.
2. Gerçek Negatif (True Negative, TN)

Sınıflandırma işlemi, sonucu negatif olarak tahmin etmiştir ve gerçekte de sonuç negatiftir. Bu yönüyle başarılı bir sınıflandırma yapılmıştır.

3. Yanlış Pozitif (False Positive, FP)
   Sınıflandırma işlemi, sonucu pozitif olarak tahmin etmiştir, ancak gerçekte sonuç negatiftir. Bu yönüyle başarısız bir sınıflandırma yapılmıştır.
4. Yanlış Negatif (False Negative, FN)
   Sonuç sınıflandırma işlemi tarafında negatif olarak tahmin edilmiştir, ancak gerçekte sonuç pozitiftir. Bu yönüyle başarısız bir sınıflandırma yapılmıştır.

Tez kapsamında yapılan sınıflandırmada 3 farklı sonuç olduğu için hata matrisi (confusion matrix) Tablo 6.5'te gösterilmiştir.

Tablo 6.5. İçerik sınıflandırma testleri hata matrisi.

		Gerçek değer		
		Kurumsal Gizli (G)	Hizmete Özel (Ö)	Tasnif Dışı (TD)
Tahmin edilen değer	Kurumsal Gizli (G)	TP	FP	FP
	Hizmete Özel (Ö)	FN	TP	FP
	Tasnif Dışı (TD)	FN	FN	TN

Buna göre, doğru tahminlerin toplam tahminlere oranı olan "doğruluk" değerinin formülü şu şekildedir:

$$accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (6.1)$$

F_{β}-skor hesaplamasının ß değerinin 1 olarak alındığı f1-skor formülü ise denklem 6.2'de gösterildiği gibidir.

$$f - skor = \frac{2pr}{p + r} \quad (6.2)$$

Bu formüldeki p değeri, sınıflandırma sonuçlarının precision (kesinlik) değerini, r değeri ise recall (duyarlılık) değerini göstermektedir. Bu değerlerin ise denklem 6.3 ve 6.4 gösterildiği şekilde hesaplanır.

$$r = \frac{TP}{TP + FN} \quad (6.3)$$

$$p = \frac{TP}{TP + FP} \quad (6.4)$$

Formüllerden görülebildiği üzere f1-skor değeri duyarlılık ile kesinlik değerlerinin harmonik ortalamasıdır. Duyarlılık değeri (r) pozitif olarak sınıflandırılması gereken işlemlerin hangi oranda pozitif olarak sınıflandırıldığını göstermektedir. Kesinlik değeri (p) ise pozitif olarak sınıflandırılanların doğru olarak tespit edilme oranını göstermektedir. Bu iki metriğin bir arada değerlendirilmesini sağlamak için f-skor hesaplaması önemli bir gösterge olmaktadır. 0 ile 1 arasında değerler alabilmektedir ve 1'e yakın değerler alması başarılı kabul edilmektedir.

Diğer yandan gerçekleme aşamasında karşılaştırma yapılan CNN modeli f-skor değeri üretmek için gerekli olan sonuçları vermediğinden bu yöntemin sonuçları ile doğruluk (accuracy) değeri kullanılarak karşılaştırma yapılmıştır. Bu değer de aynı şekilde 0-1 arası değer alabilmekte ve 1 mutlak başarıyı ifade etmektedir.

6.2.3. Testler ve sonuçlar

Test akışında, öncelikle eğitim için ayrılmış dokümanlar ile 4. Bölümde anlatılan adımlar takip edilerek eğitim gerçekleştirilmiş ve sınıflandırma modelleri oluşturulmuştur. Sonrasında, test işlemleri için ayrılmış metinler ile içerik tabanlı saldırı olduğu ve olmadığı durumlarda sınıflandırma yapılmıştır.

Sonuçların literatürde var olan önceki çalışmalarla karşılaştırılması için ise (Tripathy ve ark., 2016) çalışmasında kullanılan Kategori Profilleri, n-gram temelli kullanılabilen Stokastik Gradyan İnişi (Stochastic Gradient Descent-SGD) (Ruder, 2017) ve n-gram temelli olmayan bir yöntem ile de test karşılaştırmak için (Maheshwari, 2018) çalışmasında anlatılan yönteme bağlı kalarak Evrişimli Sinir Ağı (Convolutional Neural Network-CNN) ile sınıflandırma yapılarak sonuçlar karşılaştırılmıştır.

Testlerde farklı atak türlerindeki sonuçların değerlendirilmesi sağlanmıştır. Bu sebeple Tablo 6.6'da gösterilen içerik tabanlı ataklar gerçekleştirilmiştir. Sonuç tablolarında bütüncül bir sonuç göstermek için, atak olmadığı durum bu tabloda A0 ile belirtilmiştir.

Yapılan ataklara göre oluşan test sonuçları Tablo 6.7 ve Tablo 6.8'de gösterilmiştir. Ayrıca Şekil 6.5 ve Şekil 6.6'da sonuçlardaki başarımın grafiksel olarak karşılaştırılması sunulmuştur.

Tablo 6.6. İçerik sınıflandırma testlerinde kullanılan ataklar.

		Atak Türü	**Atak Kodu**
		Atak yok	A0
Yapısal Ataklar	Modifikasyon - Kelimeleri değiştirme	Dokümandaki kelimelerin sonuna harf eklenmesi	A1
		Dokümandaki kelimelerin başına harf eklenmesi	A2
		Dokümandaki kelimelerin ortasına harf eklenmesi	A3
		Dokümandaki boşlukların kaldırılması	A4
		Dokümandaki boşluklar yerine harf konulması	A5
		Dokümandaki boşluklar yerine + konulması	A6
		Dokümandaki kelimelerden bir harf çıkarılması	A7
		Dokümandaki E ve e harfi yerine 1 sayısının konulması	A8
	Yer değiştirme	Paragrafların yerinin değiştirmesi	A9
		Kelimelerin yerinin değiştirmesi	A10
	Yerine koyma	Summarizer özetlemesi	A11
		LSASumarizer özetlemesi	A12
		LEXMark özetlemesi	A13
		TEXRank özetlemesi	A14
		SUMBasic özetlemesi	A15
		KLSummarizer özetlemesi	A16
		ReductionSummarizer özetlemesi	A17
Karartma Atakları		Kelimelerin eş anlamlı kelimeler ile değiştirilmesi	A18

Tablo 6.7. İçerik sınıflandırma testlerinde kullanılan ataklar.

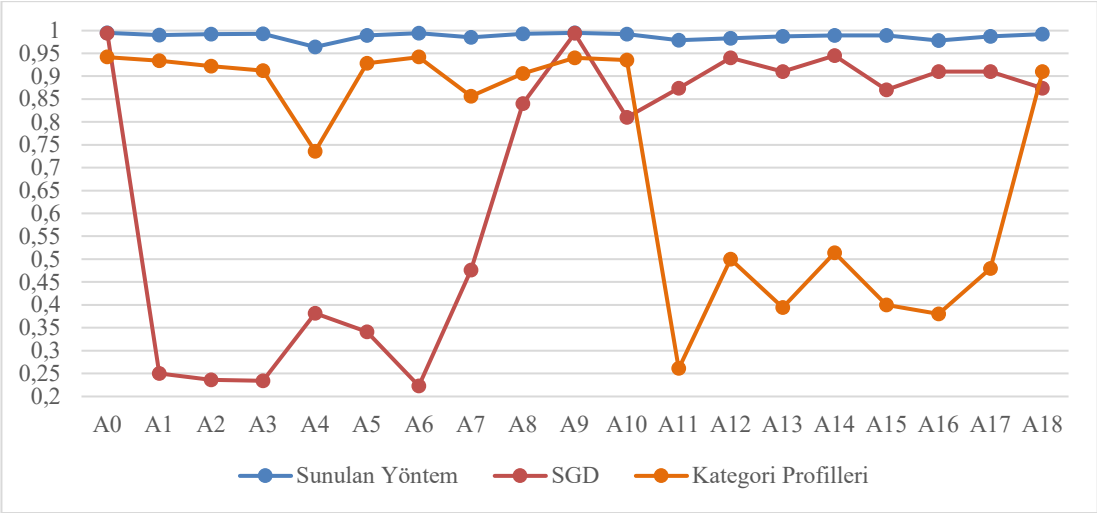
		Sunulan Yöntem			Kategori Profilleri			SGD		
Atak yok A0		G	Ö	TD	G	Ö	TD	G	G	TD
	G	205	0	0	149	20	0	206	0	0
	Ö	9	0	0	64	810	1	4	833	0
	TD	0	836	612	1	6	611	4	3	612
	F-skor	0,98	0,99	1,00	0,78	0,95	0,99	0,98	1,00	0,99
	F-skor(μ)	0,99			0,94			0,99		
Dokümandaki kelimelerin sonuna harf eklenmesi A1		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	201	0	0	145	0	2	1	3	0
	Ö	130	834	0	9	810	8	1	39	1
	TD	0	2	612	60	26	602	212	794	611
	F-skor	0,76	0,93	1,00	0,80	0,97	0,93	0,01	0,09	0,55
	F-skor(μ)	0,99			0,93			0,25		

Tablo 6.7. (Devamı) İçerik sınıflandırma testleri KP ve SGD karşılaştırma sonuçları.

		Sunulan Yöntem			Kategori Profilleri			SGD		
Dokümandaki kelimelerin başına harf eklenmesi A2		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	203	0	0	143	0	0	0	1	0
	Ö	11	836	0	1	780	0	0	28	0
	TD	0	0	612	70	56	612	214	807	612
	F-skor	0,97	0,99	1,00	0,80	0,96	0,91	0,00	0,06	0,55
	F-skor(μ)	0,99			0,92			0,23		
Dokümandaki kelimelerin ortasına harf eklenmesi A3		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	201	0	0	129	10	0	0	10	0
	Ö	12	835	0	10	783	1	1	28	2
	TD	0	1	612	75	43	611	213	808	610
	F-skor	0,97	0,99	1,00	0,73	0,96	0,91	0,00	0,06	0,50
	F-skor(μ)	0,99			0,91			0,234		
Dokümandaki boşlukların kaldırması A4		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	185	1	0	50	0	0	14	0	0
	Ö	20	805	0	37	587	0	0	142	0
	TD	9	30	612	127	249	612	200	694	612
	F-skor	0,97	0,99	1,00	0,38	0,80	0,77	0,12	0,29	0,58
	F-skor(μ)	0,99			0,73			0,37		
Dokümandaki boşlukların yerine harf konulması A5		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	197	0	0	138	1	5	3	1	0
	Ö	17	835	0	6	787	5	4	111	0
	TD	0	1	612	60	40	602	207	724	612
	F-skor	0,96	0,99	1,00	0,79	0,97	0,92	0,03	0,23	0,57
	F-skor(μ)	0,99			0,93			0,33		
Dokümandaki boşlukların yerine simge konulması A6		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	205	0	0	147	18	0	0	0	0
	Ö	9	836	0	60	810	1	0	20	0
	TD	0	0	612	5	8	611	214	816	612
	F-skor	0,98	0,99	1,00	0,78	0,95	0,99	0,38	0,20	0,22
	F-skor(μ)	0,99			0,94			0,22		
Dokümandaki kelimelerden bir harf çıkarılması A7		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	191	0	0	123	16	0	3	0	0
	Ö	23	835	1	11	680	1	0	268	2
	TD	0	1	611	70	140	611	211	568	610
	F-skor	0,94	0,99	1,00	0,72	0,89	0,85	0,03	0,48	0,61
	F-skor(μ)	0,99			0,86			0,48		
Dokümandaki E ve e harfi yerine 1 sayısı konulması A8		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	203	0	0	107	1	0	58	0	0
	Ö	11	836	0	2	788	1	25	760	2
	TD	0	0	612	95	47	611	131	76	610
	F-skor	0,97	0,99	1,00	0,69	0,97	0,90	0,43	0,94	0,85
	F-skor(μ)	0,99			0,91			0,84		
Paragrafların yerinin değiştirmesi A9		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	205	0	0	147	27	0	206	0	0
	Ö	9	836	0	66	808	1	4	833	0
	TD	0	0	612	1	1	611	4	3	612
	F-skor	0,98	0,99	1,00	0,76	0,94	1,00	0,98	1,00	0,99
	F-skor(μ)	0,99			0,94			0,99		

Tablo 6.7. (Devamı) İçerik sınıflandırma testleri KP ve SGD karşılaştırma sonuçları.

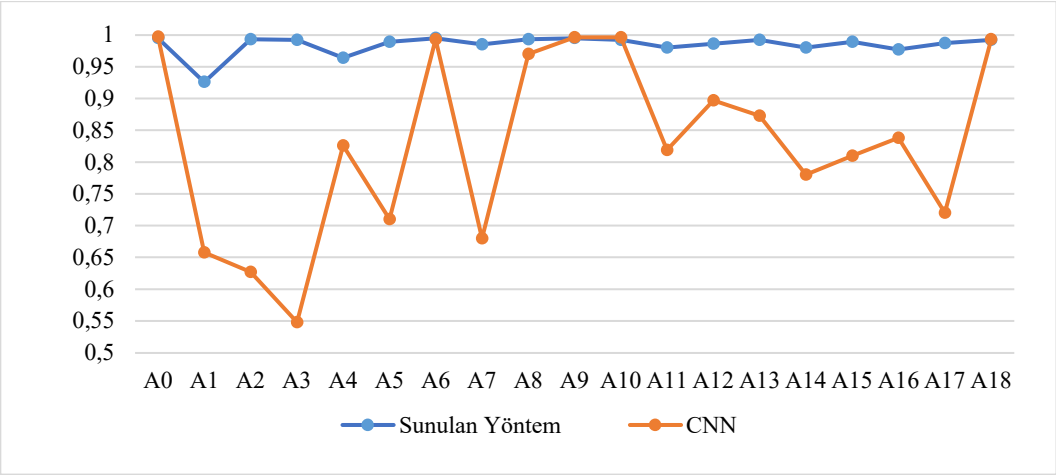
		Sunulan Yöntem			Kategori Profilleri			SGD		
Kelimelerin yerinin değiştirmesi A10		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	202	0	0	143	39	0	70	0	0
	Ö	12	836	1	70	805	2	2	674	1
	TD	0	0	611	1	2	610	142	162	611
	F-skor	0,97	0,99	1,00	0,72	0,93	1,00	0,49	0,89	0,80
	F-skor(µ)	0,99			0,93			0,81		
Summarizer özetlemesi A11		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	199	6	0	5	1	0	150	2	0
	Ö	8	818	1	11	47	1	6	685	0
	TD	7	12	611	198	788	612	58	149	612
	F-skor	0,95	0,98	0,98	0,05	0,11	0,55	0,82	0,90	0,86
	F-skor(µ)	0,98			0,26			0,87		
LSASummarize özetlemesi A12		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	202	3	0	9	9	0	166	0	0
	Ö	8	826	0	19	300	1	5	785	0
	TD	4	7	612	186	527	611	43	51	612
	F-skor	0,94	0,99	0,99	0,08	0,52	0,63	0,87	0,97	0,93
	F-skor(µ)	0,98			0,50			0,94		
LEXMark ile özetlemesi A13		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	200	0	0	12	13	0	161	1	0
	Ö	9	832	0	12	165	1	6	733	0
	TD	0	4	612	190	658	611	47	102	612
	F-skor	0,96	0,99	0,99	0,10	0,33	0,59	0,86	0,93	0,89
	F-skor(µ)	0,99			0,39			0,91		
TEXMark ile özetlemesi A14		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	191	1	0	18	28	0	185	0	0
	Ö	20	831	0	28	300	1	6	773	0
	TD	3	4	612	168	508	611	23	63	612
	F-skor	0,98	0,99	1,00	0,14	0,52	0,64	0,93	0,96	0,93
	F-skor(µ)	0,99			0,51			0,95		
SUMBasic ile özetlemesi A15		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	206	0	0	5	6	0	143	1	0
	Ö	4	826	0	6	136	1	2	684	0
	TD	4	10	612	203	694	611	69	151	612
	F-skor	0,98	0,99	0,99	0,04	0,28	0,58	0,80	0,90	0,85
	F-skor(µ)	0,99			0,40			0,87		
KLSummarizer ile özetlemesi A16		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	194	1	0	4	10	0	155	0	0
	Ö	15	820	1	18	157	1	6	742	0
	TD	5	15	611	192	669	611	53	94	612
	F-skor	0,95	0,98	0,98	0,04	0,31	1,00	0,84	0,94	0,89
	F-skor(µ)	0,98			0,38			0,91		
ReductionSummarize özetlemesi A17		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	201	3	0	13	18	0	173	1	0
	Ö	10	829	1	27	269	1	5	723	0
	TD	3	4	611	174	549	611	36	112	612
	F-skor	0,96	0,99	1,00	0,11	0,47	0,63	0,89	0,92	0,89
	F-skor(µ)	0,99			0,48			0,91		
Kelimelerin eş anlamlı kelimeler ile değiştirilmesi A18		G	Ö	TD	G	Ö	TD	G	Ö	TD
	G	203	0	0	130	0	0	193	1	0
	Ö	11	834	0	2	776	1	3	817	1
	TD	0	2	612	82	60	611	18	18	612
	F-skor	0,97	0,99	1,00	0,76	0,96	0,90	0,95	0,99	0,97
	F-skor(µ)	0,99			0,91			0,98		



Şekil 6.5. İçerik sınıflandırma testleri KP ve SGD karşılaştırma grafiği.

Tablo 6.8. İçerik sınıflandırma testleri CNN karşılaştırmalı sonuçları.

	Sunulan Yöntem	CNN
Atak yok – A0	0,995	0,997
Dokümandaki kelimelerin sonuna harf ekleme – A1	0,926	0,658
Dokümandaki kelimelerin başına rasgele harf ekleme – A2	0,993	0,627
Dokümandaki kelimelerin ortasına rasgele harf ekleme – A3	0,992	0,548
Dokümandaki boşlukları kaldırma – A4	0,964	0,826
Dokümandaki boşluklar yerine rasgele harf konulması – A5	0,989	0,71
Dokümandaki boşluklar yerine + konulması – A6	0,995	0,993
Dokümandaki kelimelerden rasgele bir harf çıkarılması – A7	0,985	0,68
Dokümandaki E ve e harfi yerine 1 sayısı konulması – A8	0,993	0,97
Cümlenin yerini değiştirme – A9	0,995	0,996
Kelimelerin yer değiştirmesi – A10	0,992	0,996
Summarizer ile özet – A11	0,98	0,819
LSASumarizer ile özet – A12	0,986	0,897
LEXMark ile özet – A13	0,992	0,873
TEXRank ile özet – A14	0,98	0,78
SUMBasic ile özet – A15	0,989	0,81
KLSummarizer ile özet – A16	0,977	0,838
ReductionSummarizer ile özet – A17	0,987	0,72
Eş anlamlı kelime atakları – A18	0,992	0,993



Şekil 6.6. İçerik sınıflandırma testleri CNN karşılaştırma grafiği.

Sonuçlar incelendiğinde gerek sunulan çözümün gerekse de karşılaştırma yapılan yöntemlerin atak olmadığı durumda (A0) başarılı sonuçlar ürettiği görülmektedir. Atak altında ise sunulan yöntemin her tür atak türünde yüksek bir başarım sunduğu görülmüştür. Karşılaştırılan diğer yöntemlerin başarımında ise ciddi kayıplar oluşmuştur.

KP yönteminin modifikasyon ve yer değiştirme ataklarında (A1-A10) nispeten başarımını koruduğu, ancak özetleme türünden yerine koyma ataklarında (A11-A17) başarısız kaldığı gözlemlenmiştir. Eş anlamlı kelimeler kullanma atağının (A18) ise başarımına etkisi olmadığı belirlenmiştir.

SGD yönteminin ise modifikasyon ataklarında (A1-A8) genelde başarısız olduğu, yer değiştirme ataklarında (A9-A10), özetleme ataklarında (A11-A17) ve eş anlamlı kelimeler kullanma atağında (A18) başarımını nispeten daha iyi koruduğu gözlemlenmiştir.

CNN yönteminin ise genel olarak başarım kaybı yaşadığı, harf tabanlı bazı ataklarda (A6, A8), yer değiştirme ataklarında (A9-A10) ve eş anlamlı kelimeler kullanma atağında (A18) başarımını nispeten daha iyi koruduğu görülmüştür.

Sonuçlar bir arada değerlendirildiğinde sunulan yöntemde yer alan adımların farklı atak türlerine karşı başarımı arttırdığı görülmektedir. Buna göre içerik sınıflandırma algoritması testleri ile ortaya konulan sonuçlar şu şekildedir:

- Kelimelerin başına ya da sonuna harf eklenmesi, kelimelerdeki bazı harflerin değiştirilmesi ve boşlukların başka simgelerle değiştirilmesi gibi ataklara karşı yazım düzeltimi işlemi başarımı arttırmaktadır.

- Kelime bazlı ataklara karşı kullanılan n-gram ve LSA özellikleri başarımı arttırmıştır.
- Özetleme ataklarına karşı k-skip n-gram kullanımı başarımı arttırmıştır.
- Dokümandaki boşlukların tamamen kaldırılması ya da bir harf ile değiştirilmesi şeklindeki kelimelerin tanınabilirliğini bozan ataklara karşı karakter-gram kullanımı başarımı arttırmıştır.

6.3. APTONSYS Bulguları

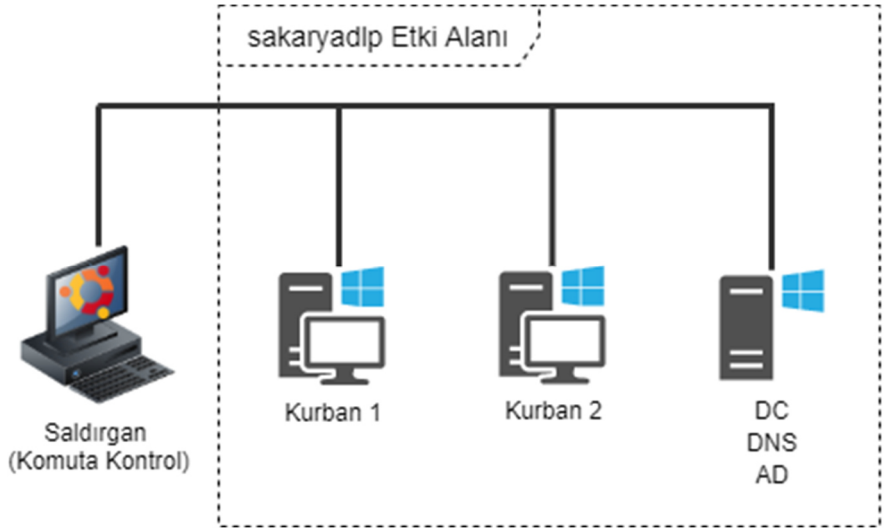
6.3.1. Test yöntemi ve mimarisi

Sistemin entegre olarak test edilmesi, test ağının kurularak sistemdeki bileşenlerin ve atakların modellenmesi, sonrasında da bu ağ üzerindeki bilgisayarlarda atakların icrası ile gerçekleştirilmiştir. Bu işlemler neticesinde APT teknik, taktik ve risklerinin tespit edildiği ve Merkezi Yönetim Yazılımı'nda toplanarak sergilendiği gözlemlenmiştir.

İlerleyen bölümlerde bu aşamalar detaylandırılmıştır.

6.3.1.1. Kurumsal ağın modellenmesi

Kurumsal bir ağın tez açısından asgari gereksinimlerini karşılamak için bir etki alanı (domain) içerisinde yer alan 3+1 bilgisayarın yer aldığı bir ağ kurulmuştur. Bunlardan birisi etki alanı sunucusu, diğeri komuta kontrol merkezi görevini görecek saldırgan bilgisayar, diğer ikisi ise etki alanına dahil olan ve testte kurban olarak seçilen bilgisayarlardır. Bu bilgisayarların yerleşimleri Şekil 6.7'de gösterilmiştir.



Şekil 6.7. Sistem testleri ağ yerleşimi.

Oluşturulan topoloji, kurumsal birçok sistemde olduğu gibi bir etki alanı içerisinde tanımlanmıştır. Bunun için bir **sakaryadlp.local** isimli bir etki alanı oluşturulmuş ve tüm topoloji bu etki alanı içerisinde modellenmiştir. Tablo 6.9'da etki alanı içerisinde kurulan bilgisayarlar, işletim sistemleri, üzerlerinde kurulan servisler ve topoloji ile arasındaki ilişkileri verilmiştir. Bilgisayarlar bir sanal ağ üzerinde yer aldıkları ve internet erişimleri gerektiği için sunucu ve saldırgan 192.168.80.1 ile belirtilen gerçek makine üzerindeki sanal ağ arayüzünü ağ geçidi olarak kullanmaktadırlar. Kurban makineler ise etki alanı sunucusu üzerinden bağlantı sağlamaktadırlar.

Tablo 6.9. Sistem testleri bilgisayarların konfigürasyon bilgileri.

Sanal Makina	Bilgisayar İsmi	IP Adresi	İşletim Sistemi
Etki Alanı Sunucusu	domaincontroller.saudlp.local	192.168.80.21	Windows Server 2019
Kurban 1	victim1.saudlp.local	192.168.80.22	Windows 10
Kurban 2	victim2.saudlp.local	192.168.80.23	Windows 10
Saldırgan	attacker	192.168.80.24	Ubuntu 18.02

6.3.1.2. Kullanıcıların modellenmesi

Kullanıcıların modellemesinde kurumsal bir Windows sistemine ve ontoloji sınıflarına uygun olarak belirlenmiştir:

1. Yönetici yetkili kullanıcılar
    a. Etki alanı yöneticileri
    b. Yerel bilgisayar yöneticileri
    c. Sistem kullanıcısı (SYSTEM)
2. Normal kullanıcılar
    a. Etki alanı kullanıcıları
    b. Yerel bilgisayar kullanıcıları

6.3.1.3. Korunacak içeriklerin modellenmesi

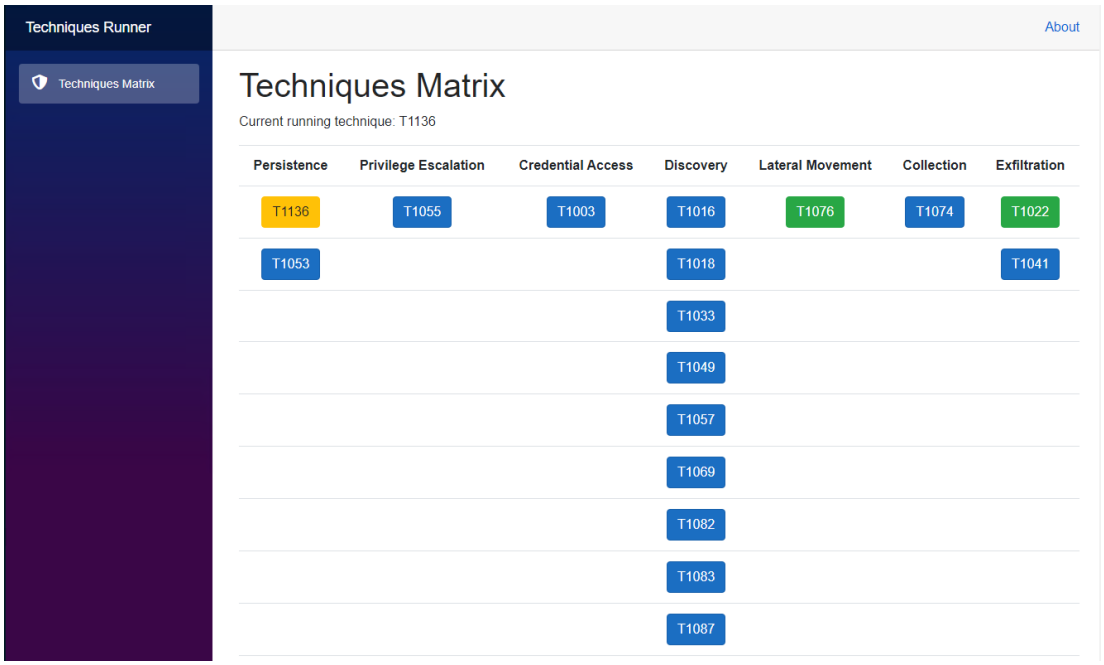
İçerik gizlilik seviyeleri İçerik Sınıflandırma Algoritması ve ontoloji sınıflarına uygun olarak 3 türde belirlenmiştir:

1. Kurumsal Gizli
2. Hizmete Özel
3. Tasnif Dışı

Doküman örnekleri, İçerik Sınıflandırma Algoritması kapsamında kullanılan test verilerinden seçilerek kurban bilgisayarlara ayrı dokümanlar olarak konulmuştur.

6.3.1.4. Atakların modellenmesi

6.1.1 bölümünde belirtildiği üzere, saldırı modellemesi için en uygun APT'nin APT3 olduğu değerlendirilmiştir. Sistemin entegre testlerinde bu APT'ye ait tekniklerin uygulanabilmesi, bir komuta kontrol bilgisayarı aracılığıyla kurban bilgisayarlarda APT tekniklerinin uygulanmasını gerektirmektedir. APT tekniklerinin uygulanması diğer bölümlerde anlatıldığı gibi ART, RTA, PTA ve Caldera araçları ile yapılabilmektedir. Bu atakların icrası ve özellikle T1041 (Exfiltration over Command and Control Channel) gibi tekniklerde kaçırılmak istenen dosyaların yüklenebileceği bir komuta kontrol sunucusunun varlığı gerektiği için Komuta Kontrol Yazılımı ve kurban bilgisayarlarda çalışan Saldırgan Ajan Yazılımı gerçeklenmiştir. İlk Bulaşma (Initial Access) aşamasının geçilmiş olduğu kabul edilerek zararlı yazılımın kurban bilgisayarlara bulaşmış olduğu varsayılmıştır. Bu sebeple kurban bilgisayarlarında saldırıları gerçekleştiren Saldırgan Ajan Yazılımı test başlangıcında çalışır durumdadır. Ataklar bu ajan yazılımlarının sağladığı HTTP arayüzleri üzerinden başlatılarak sonuçları Merkezi Kontrol Yazılımı üzerinden takip edilmiştir. Ajan yazılımları arayüzlerinden seçilen APT tekniğine ait işlemler, kurban makine üzerinde icra edilmektedir. Şekil 6.8'de Saldırgan Ajan Yazılımı arayüzü sergilenmektedir. Bu şekildeki mavi ile gösterilen teknikler, henüz icra edilmemiş olanları; sarı ile gösterilen şu anda uygulanan tekniği; yeşil olanlar ise başarılı olarak uygulanmış teknikleri göstermektedir.



Şekil 6.8. Saldırgan Ajan Yazılımı arayüzü.

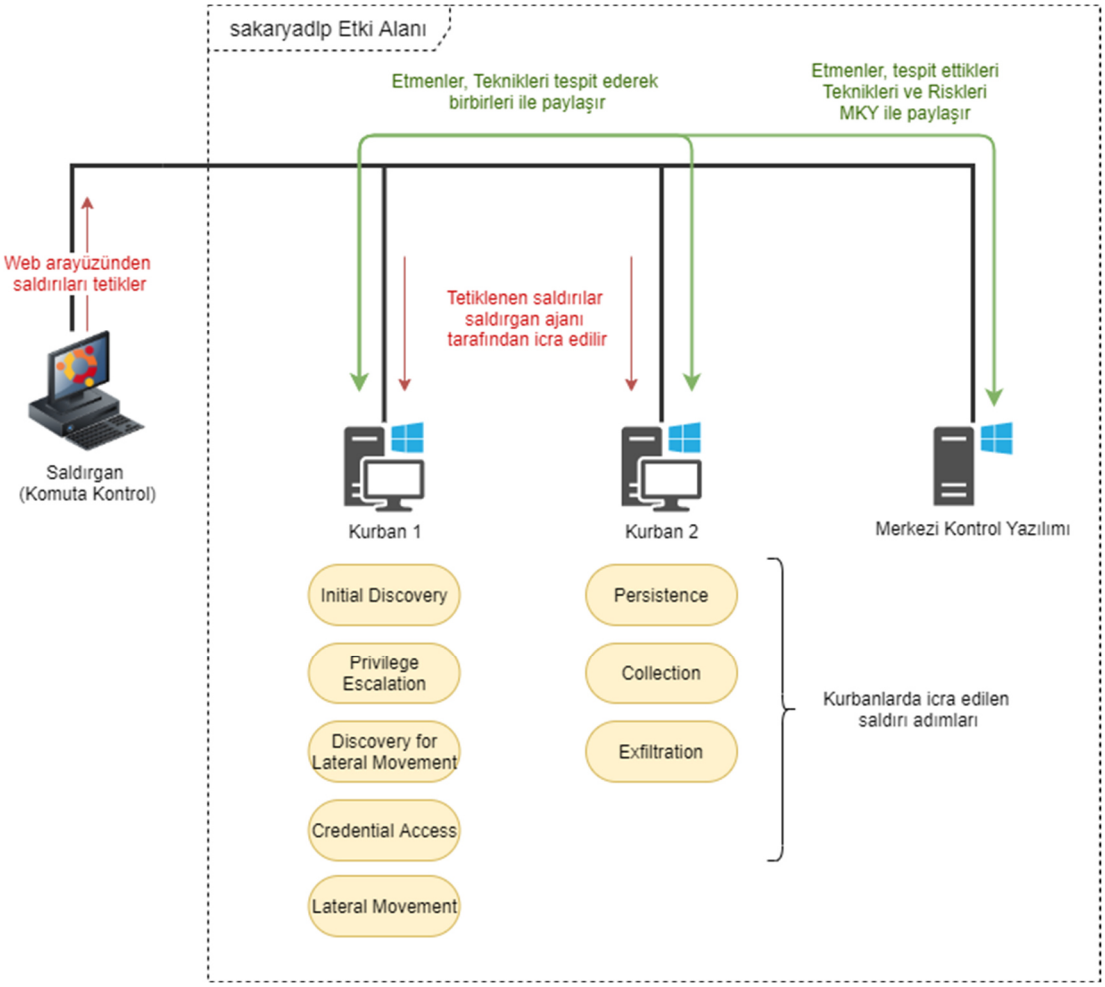
Sistem testlerinde uygulanan APT teknikleri ve hangi gerçekleme kullanılarak uygulandığı bilgileri Tablo 6.10'da belirtilmiştir. T1022 için kendimizin geliştirdiği bir araç ile 4.3 bölümünde anlatılan yer değiştirme (transposition) saldırısı ile içerik tabanlı bir saldırı modellenmiştir. T1041 için ise kurban bilgisayardan Komuta Kontrol bilgisayarına dosya içeriğini gönderen bir araç gerçeklenmiştir. Diğer teknikler için tabloda belirtilen açık kaynaklı atak gerçeklemeleri kullanılmıştır.

6.3.2. Saldırılar ve tespit sonuçları

Bir önceki bölümde anlatıldığı üzere belirlenen APT teknikleri Saldırgan Ajan Yazılımı kullanılarak sıra ile kurban bilgisayarlarda icra edilmiştir. Saldırıların uygulanma planı hem MITRE'nin APT3 Saldırı Planı'na (Korban ve ark., 2017) uygun olması, hem kurumsal bir sistemin daha iyi modellenebilmesi, hem de APT risk seviyesinin gittikçe artacağı bir durum oluşturulabilmesi dikkate alınarak belirlenmiştir. Buna göre APT tekniklerinin bir kısmı 1. kurban bilgisayarda gerçekleştirilmiş, atağın yanal hareketi (Lateral Movement) sonrasında ise diğer ataklar 2. kurban bilgisayarda gerçekleştirilmiştir. Böylece sistemin etmenler arası iletişim ile ağ genelinde bir tespit yapabildiği gösterilmiştir. Şekil 6.9'da atakların nasıl gerçekleştirildiği ve her bir kurban bilgisayarda uygulanan saldırı adımları gösterilmiştir. Ayrıca etmenlerin tespit sonrası bilgi paylaşımı ile MKY üzerinde sonuçların toplandığı betimlenmiştir.

Tablo 6.10. Sistem testleri için uygulanan APT teknikleri.

APT Teknik Kodu	APT Teknik Adı	APT Taktik Adı	Atak Gerçeklemesi
T1136	Create Account	Persistence	ART
T1053	Scheduled Task	Persistence	ART
T1055	Process Injection	Privilege Escalation	ART
T1003	Credential Dumping	Credential Access	PTA
T1033	System Owner / User Discovery	Discovery	ART
T1016	System Network Configuration Discovery	Discovery	ART
T1057	Process Discovery	Discovery	ART
T1082	System Information Discovery	Discovery	ART
T1069	Permission Groups Discovery	Discovery	ART
T1087	Account Discovery	Discovery	ART
T1018	Remote System Discovery	Discovery	ART
T1049	System Network Connections Discovery	Discovery	ART
T1083	File and Directory Discovery	Discovery	ART
T1076	Remote Desktop Protocol	Lateral Movement	ART
T1074	Data Staged	Collection	ART
T1041	Exfiltration Over Command-and-Control Channel	Exfiltration	Özel
T1022	Data Encrypted	Exfiltration	Özel



Şekil 6.9. Sistem testleri uygulama akışı.

İlerleyen bölümlerde sıra ile uygulanan APT saldırıları ve sonuçları detaylandırılmıştır.

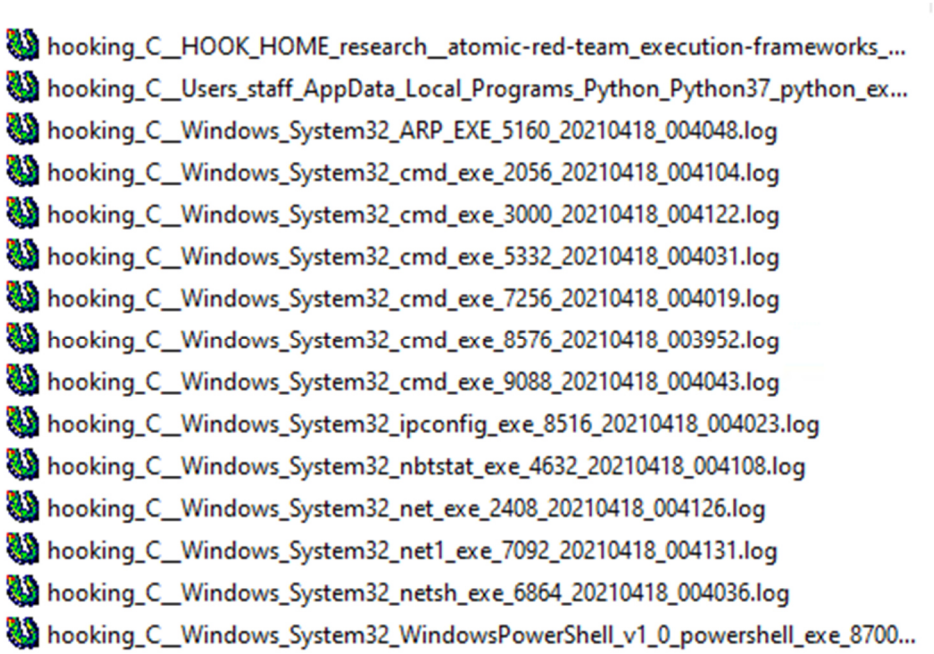
6.3.2.1. İlk keşif saldırısı

Saldırgan Ajan Yazılımı üzerinden T1016 seçilerek Kurban 1 bilgisayarında T1016 işlemlerinin icrası başlatılmıştır. T1016 için kullanılan ART'nin gerçeklemesi içerisinde yeni bir cmd.exe prosesi başlatılarak sıra ile şu komutlar çalıştırılmaktadır:

ipconfig /all
netsh interface show
arp -a
nbtstat -n
net config

Bu komutların her birisi yeni bir ya da daha fazla proses yaratılmasına neden olmaktadır. Öncelikle ART komutlarını çalıştıran python.exe prosesi ve komutları

başlatan ilk cmd.exe'ye ait proses olmak üzere bütün bu "çocuk" prosesler Kancalama Modülü tarafından yakalanmaktadır. Bu proseslere ilişkin oluşan kayıt dosyaları Şekil 6.10'da gösterilmiştir.



Şekil 6.10. T1016 sonucu oluşan proseslerin Kancalama modülü log kayıt dosyaları.

Burada görülen her bir proses için Statik Analiz Modülü ve Proses Sınıflayıcı tarafından değerlendirme yapılarak skor üretilmektedir. Bu programların hepsi, esasında sistem içerisinde standart olarak var olan programlar oldukları için Statik Analiz tarafından en yüksek puan ile değerlendirilmiştir. Şekil 6.11'de Statik Analiz modülü kayıt dosyalarında ilgili prosesler için oluşturulan skorlar gösterilmiştir.

```
The prediction for C:\WINDOWS\system32\ipconfig.exe is : MaliciousnessEnum.Benign
With the probability of : 1.0
The prediction for C:\WINDOWS\system32\netsh.exe is : MaliciousnessEnum.Benign
With the probability of : 1.0
The prediction for C:\WINDOWS\system32\ARP.EXE is : MaliciousnessEnum.Benign
With the probability of : 1.0
The prediction for C:\WINDOWS\system32\nbtstat.exe is : MaliciousnessEnum.Benign
With the probability of : 1.0
The prediction for C:\WINDOWS\system32\net.exe is : MaliciousnessEnum.Benign
With the probability of : 1.0
The prediction for C:\Windows\System32\net1.exe is : MaliciousnessEnum.Benign
With the probability of : 1.0
```

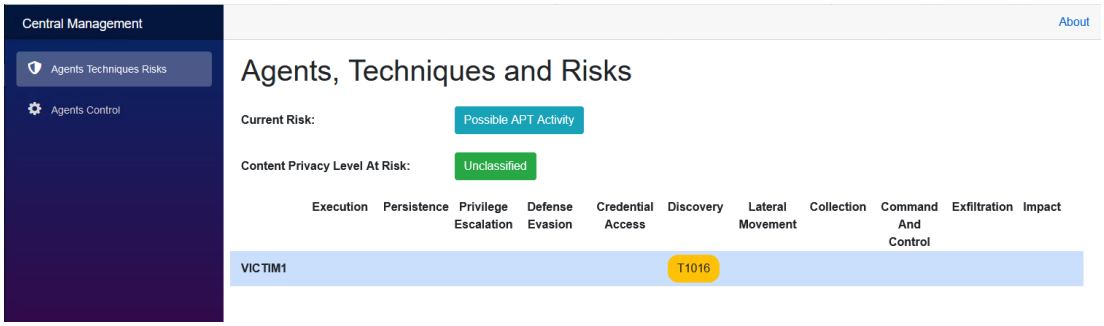
Şekil 6.11. T1016 sonucu oluşan proseslerin Statik Analiz değerlendirme sonuçları.

Daha sonrasında ilgili prosesler çalışmaya başlayarak sistem çağrıları oluşturmuş, bunlar Kancalama modülü ile yakalanarak OKV'ye iletilmiştir. Bu sistem

çağrılarından olan GetNetworkParams gerçekleştiğinden T1016'ya ilişkin aşağıdaki kural tarafından tespit sağlanmıştır.

$$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p), GetNetworkConfigInfo(?\,sysCallCat),$$
$$hasSystemCall(?\,apt, ?\,sysCall), hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$$
$$hasTargetObjectType(?\,sysCall, ?\,targetObjType),$$
$$isCalledByProcess(?\,sysCall, ?\,p)$$
$$\rightarrow SystemNetworkConfigurationDiscovery(?\,apt)$$

Bu tespit bilgisi MKY'ye iletilerek orada sergilenmesi sağlanmıştır. Sistemde bir "Discovery" tekniğinin varlığı "Olası APT Aktivitesi Riski" olarak tanımlandığı için bu yönde bir Risk de belirlenerek MKY'ye iletilmiştir. İşlemler neticesinde MKY ekranında beliren tespit durumları Şekil 6.12'de gösterilmiştir.



Şekil 6.12. T1016 tespiti sonrası MKY ekranı.

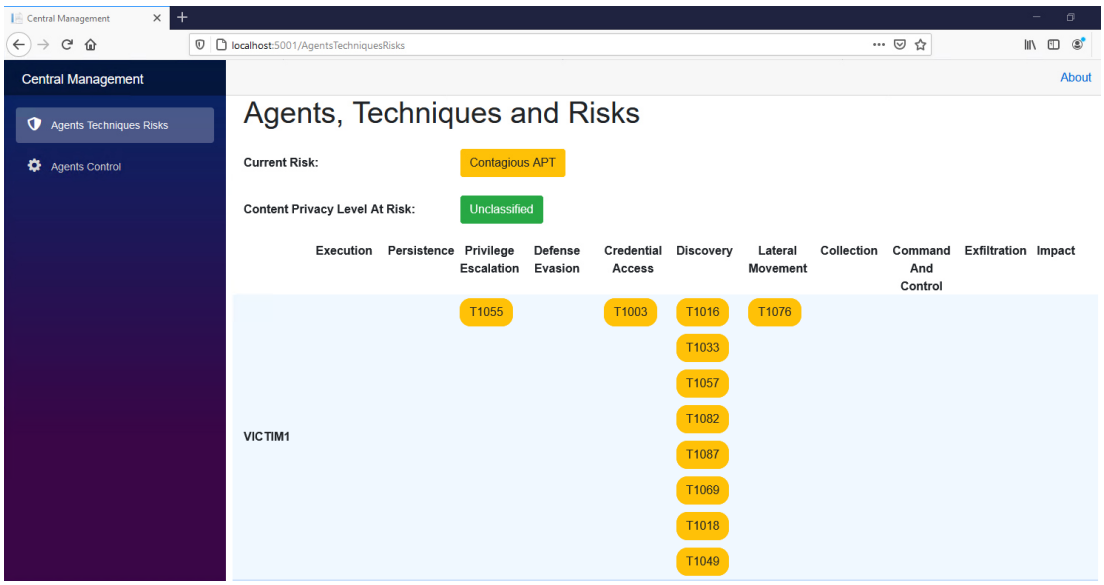
6.3.2.2. Detaylı keşif, kimlik bilgisi erişimi, hak yükseltme ve yanal hareket saldırıları

İlk keşif aşamasından sonra, emülasyon planına uygun olarak Yanal Hareket adımı dahil ilgili teknikler Kurban 1 bilgisayarında uygulanmış ve tespitleri sağlanmıştır. Bu aşamalarda sırası ile şu Teknikler uygulanmıştır:

- İlk Keşif (Initial Access): T1033, T1082, T1069, T1087
- Hak Yükseltme (Priviledge Escalation): T1055
- Yanal Hareket için Keşif (Discovery): T1018, T1016, T1049
- Kimlik Bilgisi Erişimi (Credential Access): T1003
- Yanal Hareket (Lateral Movement): T1076

Bu tekniklerin icrası ve tespiti ilk keşif saldırısında olduğu gibi Saldırgan Ajan Yazılımı üzerinden gerçekleştirilmiştir. Tespitler de 0 bölümündeki kurallara uygun olarak sağlanmıştır. Bunun neticesinde risk tanımlarına uygun olarak Yanal Hareket Taktiğine ilişkin bir teknik tespit edildiği için risk seviyesi "APT Yayılımı Riski

(Contagious APT)” olarak arttırılmıştır. MKY ekranında bu tespitlerin yapıldığı Şekil 6.13’te gösterilmiştir.



Şekil 6.13. Yanal hareket taktiğine kadar gerçekleştirilen saldırılar sonucu oluşan MKY ekranı.

6.3.2.3. Veri toplama saldırısı

Yanal Hareket saldırısından sonra gerçekçi bir senaryoya uygun olarak, sıradaki ataklar Kurban 2 bilgisayarında icra edilmiştir. Böylece sistemin birden çok bilgisayarda çalışan etmenin birbirleri ile haberleşmeleri ve kolektif sonuç üretmeleri test edilmiştir. Emülasyon planına uygun olarak öncelikle İz Bırakma saldırısı için T1136 ve T1053 teknikleri uygulanmış ve tespit edilmiştir. Bunlardan sonra da İçerik Sınıflandırma modülünün sonuçlarının değerlendirilebildiği T1074 tekniği icra edilmiştir.

T1074 tekniği, önemli dosyaların ileride kaçırmak amaçlı olarak farklı konuma kopyalanmasını içermektedir. Tez kapsamındaki uygulamada da önceden önemli verilerin olduğu bir konum belirlenmiş ve bu konumdaki dosyalar sistemin geçici dosyalar klasörüne (TEMP) kopyalanmıştır.

Bunun neticesinde, İçerik Sınıflandırıcı kopyalanan dosyaların içeriklerini tek tek sınıflandırmış; bunlar arasında yer alan Kurumsal Gizli bir dokümanı da tespit ederek sonucu OKV’ye iletmiştir. Şekil 6.14’te İçerik Sınıflandırıcının ilgili log kayıtlarından bir kısım görülmektedir (Kayıtlarda Kurumsal gizli G ile ifade edilmiştir):

```
Classification of C:\Veriler\Kurumsal Gizli\a.txt with is G
Sending result [3] of system Call with type 59 to Ontology Reasoner.
Classification of C:\Veriler\Kurumsal Gizli\a.txt with is G
Sending result [3] of system Call with type 86 to Ontology Reasoner.
```

Şekil 6.14. T1074 saldırısı sırasında oluşan İçerik Sınıflandırıcı kayıtları.

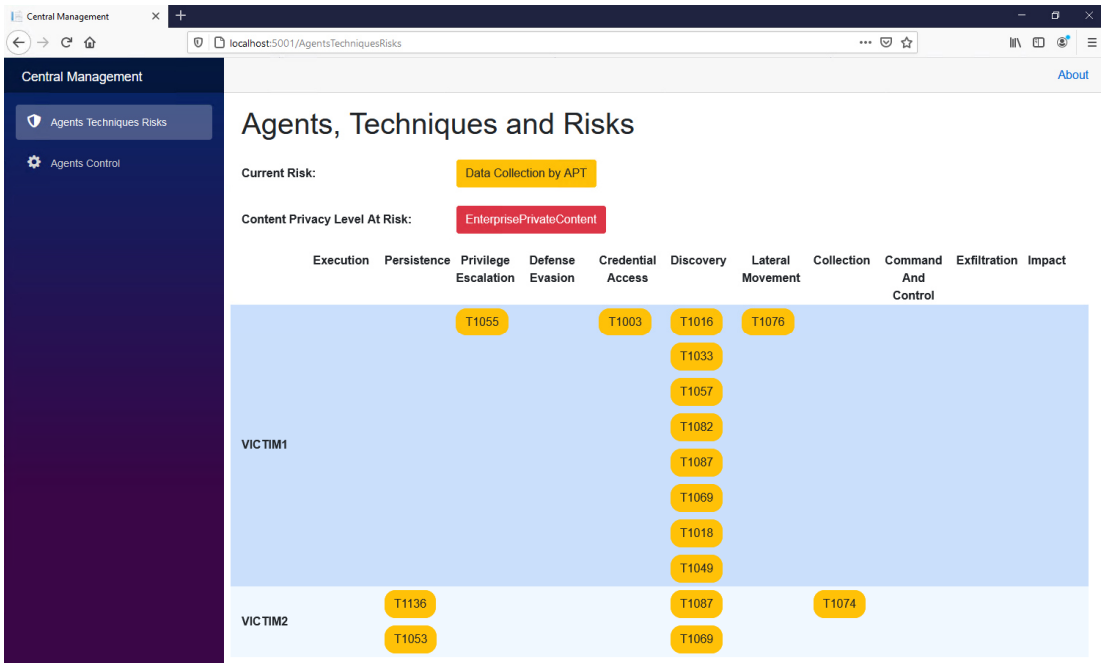
Kayıtlarda geçen 59 ve 86 numaraları, sırası ile GetFileAttributes ve CopyFile sistem çağrılarını ifade etmektedir.

Bu bilgi, diğer modüllerden gelen bilgilerle beraber OKV'ye iletilmiştir. OKV içerisinde, öncelikle aşağıdaki kural işletilerek T1074 tespiti sağlanmıştır:

$owl{:}\,Thing(?\,apt), NotTrustedProcess(?\,p), ReadFile(?\,sysCallCat),$
$Content(?\,content),$
$(EnterprisePublicContent\ OR\ EnterprisePrivateContent)(?\,privacyLevel),$
$hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$
$hasContent(?\,sysCall, ?\,content), hasContentPrivacyLevel(?\,content, ?\,privacyLevel),$
$hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$
$\rightarrow\ DataStaged(?\,apt)$

Bu kurala göre, dosya araştırma ya da okumaya ilişkin bir sistem çağrısı gerçekleşirse ve ilgili dosya İçerik Sınıflandırıcı tarafından herhangi bir kurumsal gizlilik seviyesine sahip olarak sınıflandırılmışsa ve de ilgili proses güvenilir bir proses değilse T1074 (Data Staged) tekniği tespit edilmektedir. Buradaki örnekte ilgili dosyaya ilişkin GetFileAttributes ve CopyFile sistem çağrıları ile erişim sağlanmış; bu sistem çağrılarının erişim yaptığı dosyanın içeriği de İçerik Sınıflandırıcı tarafından Kurumsal Gizli (EnterprisePrivateContent) olarak sınıflandırılmış ve bu sebeplerle tespit sağlanabilmiştir.

Tekniğin tespitinden sonra OKV, risk tespitini de yapmıştır. Bu aşamada "Collection" taktiğine ilişkin bir Teknik de tespit edilmiş olduğu için risk seviyesi "APT Veri Toplama Riski" seviyesine çıkarılmıştır. Ayrıca, İçerik Sınıflandırıcı'nın sonucu da dikkate alınarak MKY'ye hangi seviyede bir içeriğin risk altında olduğu iletilmiş ve "Risk altındaki içerik seviyesi" olarak "Kurumsal Gizli" değeri belirtilmiştir (Şekil 6.15).



Şekil 6.15. Veri Toplama Tekniği uygulandıktan sonra MKY arayüzü.

6.3.2.4. Veri kaçırma saldırısı

Veri toplama adımından sonra, yine Kurban 2 bilgisayarı üzerinde T1041 tekniği uygulanmıştır. Önceki adımlarda elde edilen gizli veriler, Saldırgan Komuta Kontrol bilgisayarına bir HTTP-POST isteği ile gönderilmiştir. T1041 için belirlenmiş, aşağıdaki kuralın tespiti sağlaması ile veri sızdırma girişimi tespit edilmiştir:

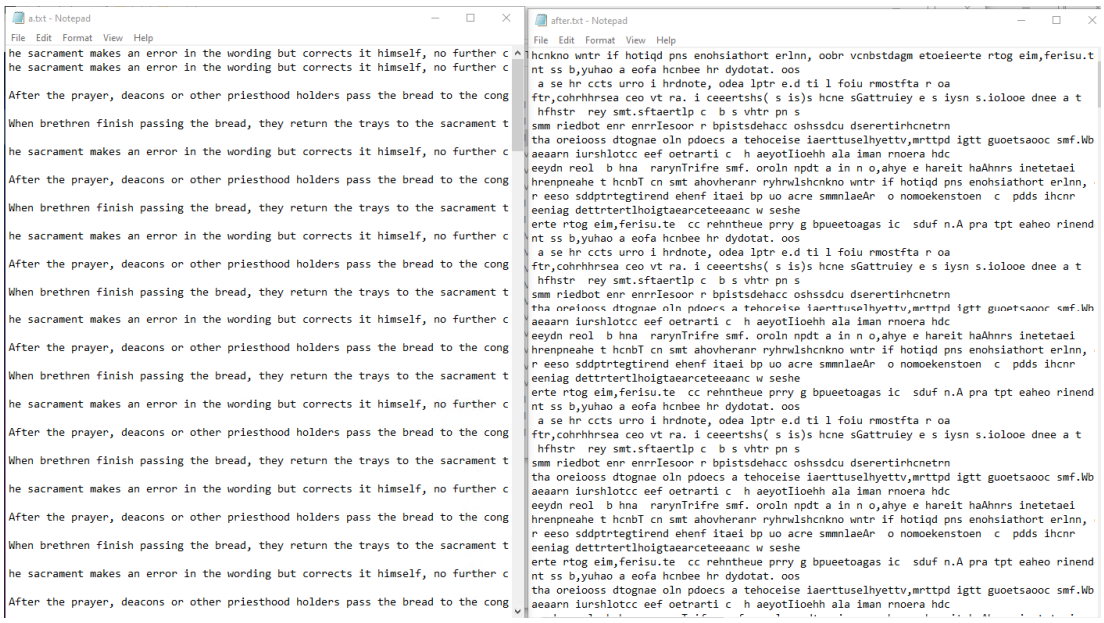
$$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$$
$$InitializeNetworkConnection(?\,sysCallCat),$$
$$Collection(?\,preDetectedCollectionTactic),$$
$$hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$$
$$hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$$
$$\rightarrow ExfiltrationOverCommandControlChannel(?\,apt)$$

Bu kuralda önemli bir detay, daha önceden bir APT Veri Toplama Taktiği'nin uygulanıp uygulanmadığının denetlenmesidir. Aksi durumda herhangi bir ağ bağlantısını veri sızdırma girişimi olarak değerlendirmeye sebep olunacaktı.

Tekniğin tespitinden sonra OKV, Risk tespitini yaparak, "Exfiltration" Taktiğine ilişkin bir Teknik tespit edilmiş olduğu için risk seviyesini "APT Veri Kaçırma Riski" seviyesine çıkarmıştır.

6.3.2.5. İçerik atağı ile veri kaçırma saldırısı

APT Saldırı Planı'na ek olarak, sistemde geliştirilen İçerik Atak Tespiti modülünün etkisini ve başarımını ölçmek için, yine Saldırgan Ajan Yazılımı aracılığıyla tetiklenen kendi geliştirdiğimiz bir yazılım ile içerik atağı uygulanmıştır. "Kurumsal Gizli" olarak belirlenmiş dosyalardan birisi ilgili program ile okunarak içeriği üzerinde Yer Değiştirme (Transposition) saldırısı uygulanmış ve sonuçta elde edilen içerik başka bir klasöre yazılmıştır (Şekil 6.16).



Şekil 6.16. Yer değiştirme içerik atağı öncesi ve sonrası.

Diğer proseslerde olduğu gibi bu içerik atağını gerçekleştiren yazılım da kancalanmıştır. Uygulamadan elde edilen sistem çağrıları ve parametreleri diğer modüllerle beraber İçerik Atak Sınıflandırıcı'ya da iletilmiştir. İçerik Atak Sınıflandırıcı, sistem çağrıları üzerinden prosesleri ve eriştiği dosyaları ilişkilendirerek erişim sağlanmış dosyaların önceki hallerini ve dosyaya yazılan içerikleri takip ederek karşılaştırma yapmaktadır. Bu testte de benzer şekilde çalışarak, yeni dosyaya yazılan içeriğin ilk okunan içeriğin saldırı yapılmış bir hali olduğunu tespit ederek bu bilgiyi OKV'ye iletmektedir.

OKV içerisinde bu durum, T1022 – Veri Şifrelemesi (Data Encrypted) tekniği altındaki kurallarda ele alınmaktadır. Bir dosyaya yazma işlemi olması ve İçerik Atak Sınıflandırıcıdan bir atak tespiti gelmesi durumunda, güvenilir bir proses değilse Tekniğin tespiti sağlanmaktadır:

$owl{:}Thing(?\,apt),$

$NotTrustedProcess(?\,p),$

$Content(?\,content),$

$ContentAttack(?\,contentMod),$

$hasContent(?\,sysCall, ?\,content),$

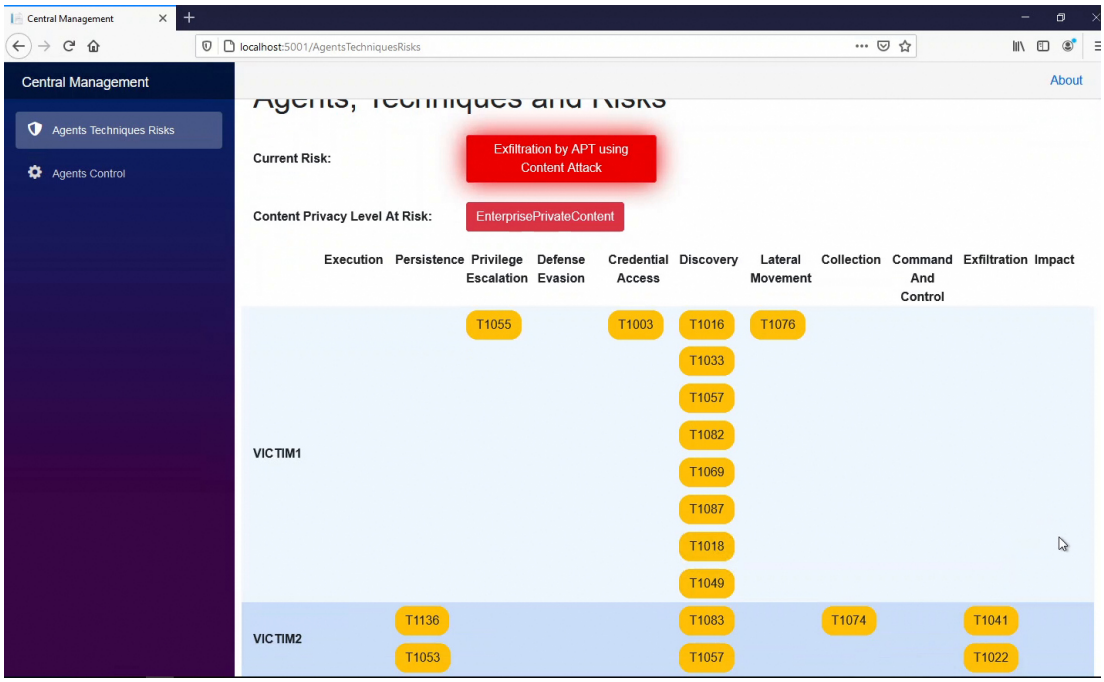
$hasContentPrivacyLevel(?\,content, ?\,privacyLevel),$

$hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$

$isCalledByProcess(?\,sysCall, ?\,p),$

$\rightarrow\ DataStaged(?\,apt)$

Bu tekniğin tespit edilmesi ile beraber, Risk seviyesi arttırılarak "APT Kaynaklı İçerik Atağı ile Veri Kaçırma Riski" seviyesine çıkarmıştır. Diğer tespitlerde olduğu gibi bu bilgi de MKY'ye iletilerek sistem yöneticisinin bilgisine sunulmuştur (Şekil 6.17):



Şekil 6.17. APT Kaynaklı İçerik Atağı ile Veri Kaçırma Riski tespiti sonrasındaki MKY arayüzü.

Sistemin iki farklı bilgisayarda uygulanan APT tekniklerini birleştirmesi, içerik ve içerik atak sınıflandırmasını ve proses bilgilerini dikkate alarak "APT Kaynaklı İçerik Atağı ile Veri Kaçırma Riski" tespiti yapmasıyla tez kapsamında sunulan çözümün başarılı olarak çalıştığı gösterilmiştir.

7. SONUÇ VE TARTIŞMA

7.1. Sonuç ve Değerlendirme

APT kavramının ortaya çıkması ve her geçen sene etkisini arttırması, DLP sistemlerinde var olan zafiyetlerin daha çok kullanılmasına ve artan veri sızıntısı olaylarına neden olmaktadır. DLP sistemlerinin özellikle kullanımdaki verinin (DIU) korunması açısından yetersiz kalması, içerik eşleştirme ve sınıflandırma algoritmaların APT'lerin kullandığı içerik tabanlı ataklara karşı dayanıksız olması, APT'lerin uzun süreye yayılan hedef odaklı atakları ile sistem üzerindeki masum uygulamaları amaçları için kullanabilmesi ve DLP sistemlerinin sistem genelinde bir denetim yapmıyor olmaları bu veri sızıntısı olaylarının artarak devam etmesinde önemli bir etkendir.

Tez kapsamında, geçmiş çalışmalar değerlendirilerek bu problemlere farklı aşamalarda çözümler sunulmuştur:

1. APT davranışının ve sistem üzerinde oluşturduğu risk seviyesinin tespiti sağlanarak sistemin "durumsal farkındalığa" sahip olması sağlamıştır. Bu sayede tespit edilen işlemler bağımsız birer olay olarak değerlendirilmemekte, bütüncül bir yaklaşım ile veri sızdırma olasılığı ortaya konulmaktadır.
2. Yapılan APT davranış tespiti, MITRE ATT&CK ile entegre edilerek literatürde ve endüstride kabul görmüş olan bir bilgi tabanı ile ilişkilendirilmiştir. Bu sayede, sunulan çözümün bilime katkısı arttırılmış ve yapılan tespitlerin APT'ler ile ilişkilendirilebilmesi sağlanmıştır.
3. DLP sistemlerinin içerik tabanlı saldırılara karşı zayıflıkları dikkate alınarak bu saldırılar altında bile başarımını koruyan yeni bir içerik sınıflandırma algoritması sunulmuştur.
4. İçerik tabanlı saldırıların içeriği henüz değiştirmeden tespit edilmesini sağlayan bir içerik atak tespit modülü sunularak bu durum veri sızdırma davranışı ile ilişkilendirilmiştir.

5. Önerilen çözümün gerçek bir sistem üzerinde çevrimiçi olarak çalışmasına imkân sunan bir etmen mimarisi sunulmuştur. Bu mimarinin gerçeklenme detayları ve işlem akışı verilmiştir.
6. Etmenler arası veri paylaşımı yapılarak sistem genelinde bütüncül bir APT risk tespit sistemi sağlanmıştır.
7. Oluşturulan sistem MITRE APT3 saldırı planına sadık kalarak açık kaynaklı APT saldırı tespit yöntemleri ile çevrimiçi olarak test edilmiştir.

Bu tez çalışması, özellikle APT kaynaklı veri sızıntılarına karşı DLP sistemlerinde anlamsal analiz ve ontolojinin potansiyelini göstermiştir. APT'ler, hedef işletim sisteminde yer alan standart yazılımları ve asıl amaçlarını gizleyen araçları ve teknikleri kullanırlar. Çoğu durumda niyetlerini açığa çıkarmak için saldırıyı oluşturan karmaşık detayları ilişkilendirerek karar verebilen bir güvenlik analistinin kapsamlı bir incelemesi gerekir. Bu uzman bilgisini yansıtabilen iyi yapılandırılmış bir ontoloji ve Semantik Web Kural Dili (SWRL) kuralları ile hedef sistemdeki varlıklar ve olaylar arasındaki ilişkiler ortaya çıkarılarak APT saldırılarının detayları tespit edilebilir. Tez kapsamında sunulan APTON ontolojisi ile bu hedef gerçekleştirilmiş ve APT atağının bütün aşamaları tespit edilerek risk seviyesi ortaya konulmuştur.

APT'lerin veri sızdırma çabaları en önemli özelliklerindendir. Bununla birlikte mevcut DLP sistemleri olayları tekil olarak değerlendirmekte ve durum farkındalığı taşımamaktadırlar. Bu sebeple, APT'nin sistemde yayıldığı ve yeterli yetkilere ulaştığı durumlarda gerçekleştirdiği işlemleri normal yetkili bir kullanıcının işlemleri olarak değerlendirme problemi taşımaktadırlar. Tez kapsamında sunulan etmen tabanlı sistem ile bütün ağ genelinde bütünleşik bir tespit ve değerlendirme imkânı ortaya konulmuştur. Yapılan tespitlerin MITRE ATT&CK'ta yer alan teknik ve taktiklerle ilişkilendirilmesi ve bunun sonucunda sistem genelindeki APT risk seviyesinin tespit edilebilmesi, DLP sisteminin durum farkındalığını gerçekleştirmektedir. Bu sayede DLP sisteminin veri sızıntısı bilgisi ile beraber APT riskini de bildirmesi sağlanabilmiştir.

Tez kapsamında yapılan araştırmalar neticesinde içerik tabanlı atakların veriyi DLP sisteminin eşleştirme ve sınıflandırma yapamayacağı seviyede değiştirebildiği görülmüştür. Bu sebeple sistem içerisinde metnin önceki ve sonraki halini sürekli karşılaştıran İçerik Atak Tespit modülü eklenmiştir. Yapılan testlerde bu yaklaşımın

hem APT davranış tespitine katkıda bulunduğu hem de içerik sızıntısını önlemek için faydalı olduğu görülmüştür.

Etmen içerisinde birden çok modülün farklı prosesler olarak çalışabilmesi ve asenkron iletişim sağlayabilmeleri ile normal özelliklerde bir bilgisayarda bile saniye mertebelerinde sonuçların alınabildiği, bu süreçte hedef proseslerin DLP sisteminden etkilenmeden işlemlerine normal devam edebildiği görülmüştür. Ayrıca, farklı modüllerin farklı yazılım dillerinde yazılmasına imkân sağlanmıştır. İçerik sınıflandırma modülünde Python dilindeki kütüphanelerden, ontoloji karar verici içerisinde Java tabanlı kütüphanelerden, etmen haberleşme modülünde .Net tarafından sunulan ağ iletişim imkânlarından faydalanılmıştır.

Sonuç olarak, bu tezin bulguları, ontoloji temelli yaklaşımların veri sızıntısı önleme yeteneklerini artırmada ve APT davranışlarını tespit etmede etkinliğini göstermektedir. Düşük seviye sistem detaylarını iyi tanımlanmış kurallar ve risk değerlendirmeleri ile birleştirerek, APT saldırılarına karşı savunma imkanları arttırılacaktır.

7.2. Çalışmanın Bilime Katkıları

- Sistem çağrısı, proses, kaynak, hedef, kullanıcı hesabı, içerik, içerik üzerindeki değişiklikler ve önceki tespit sonuçlarının bir arada değerlendirilebilmesine imkân sunan bir sistem önerisi ve sistemin merkezinde yer alan ontoloji tabanlı karar verme mekanizması sunularak DLP alanında yeni bir metot ortaya konulmuştur.
- Veri sızıntısını önlemek için APT davranışlarını tespit etmek amacıyla yeni bir anlamsal analiz modeli sunulmuştur. MITRE ATT&CK çerçevesi içerisinde tanımlanmış APT teknik ve taktiklerinin sistem çağrısı bileşenleri, proses, içerik sınıflandırması ve içerik atak durumu ile ilişkilendiren özgün APTON ontolojisi ve tespit kuralları yeni yaklaşımlara yol gösterici olmaktadır.
- APT'lerin veri kaçırma amaçlı içerik tabanlı saldırılarına karşı dayanıklı özgün bir içerik sınıflandırma yöntemi sunulmuştur.
- İçerik tabanlı saldırıların henüz uygulanma aşamasında iken tespit edilmesini sağlayan bir yaklaşım sunulmuş, bu tespitin APT davranış ve risk tespiti için nasıl kullanılabileceğine dair geçerli bir öneri ortaya konulmuştur.

- MITRE teknik ve taktiklerinin, içerik tabanlı atakların tespiti ile birleştirilmesi sonucu sistem üzerindeki APT risk değerlendirilmesine yönelik bir sınıflandırma yöntemi sunulmuştur.

7.3. Gelecek Çalışmalar

APT tekniklerini tespit etmek için SWRL kuralları geliştirmek, işletim sisteminin ve APT'lerin kullandığı araçların derinlemesine anlaşılmasını gerektirmektedir. Bu tez kapsamında, MITRE matrisinin APT3 özelindeki bir alt kümesi için SWRL kuralları sunulmuştur. Bununla birlikte, ontolojideki aynı sınıf ve özellikleri kullanarak MITRE matrisindeki tüm APT teknikleri için SWRL kuralları tanımlamanın mümkün olduğu değerlendirilmektedir.

Tez kapsamında, tespit sonucunda durum bildirimi seviyesinde bırakılan aksiyon adımları, APT risk durumuna göre DLP sisteminin yüksek hassasiyetteki verilere erişimi tamamen engellemesi, risk tespit edilen bilgisayarlarda ağ erişiminin engellenmesi, işleme neden olan kullanıcı hesaplarının pasifleştirilmesi, ilgili proseslerin ve çalıştırılabilir dosyaların skor değerlerinin dinamik olarak düşürülmesi gibi çok farklı aksiyon adımları ile daha etkin bir DLP sistemi oluşturulabilir.

İçerik sınıflandırma algoritması içerisinde metin bazlı içerikler ele alınmıştır. Bu özelliklere resimler içerisinde karakter tespiti yeteneklerinin eklenmesi ile daha geniş bir atak spektrumunun ele alınabilmesine imkân sunulacaktır.

KAYNAKLAR

ACM SIGCOMM. (2008, Nisan 9). *Traces In The Internet Traffic Archive*. Traces In The Internet Traffic Archive. https://ita.ee.lbl.gov/html/traces.html

Ahmad, A., Webb, J., Desouza, K. C., & Boorman, J. (2019). Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack. *Computers & Security*, *86*, 402-418. https://doi.org/10.1016/j.cose.2019.07.001

Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2016). A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, *62*, 137-152. https://doi.org/10.1016/j.jnca.2016.01.008

Amer, E., & Zelinka, I. (2020). A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence. *Computers & Security*, *92*, 101760. https://doi.org/10.1016/j.cose.2020.101760

Amoroso, E. G. (1994). *Fundamentals of Computer Security Technology*. PTR Prentice Hall. https://books.google.com.tr/books?id=f95QAAAAMAAJ

Apache. (2023, Ekim 20). *Apache Tika*. https://tika.apache.org/

Atapour, C., Agrafiotis, I., & Creese, S. (2018). Modeling Advanced Persistent Threats to enhance anomaly detection techniques. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *9*(4), 71-102. https://doi.org/10.22667/JOWUA.2018.12.31.071

Atomic Red Team. (2020). https://github.com/redcanaryco/atomic-red-team

AZSecure-data. (2017, Ocak). *CDMC2010 Malware API Sequence Dataset*. https://dibbs.ai.arizona.edu/dibbs/csdmc2010/CSDMC2010.zip

Bai, J., Wang, J., & Zou, G. (2014). A Malware Detection Scheme Based on Mining Format Information. *The Scientific World Journal*, *2014*, 1-11. https://doi.org/10.1155/2014/260905

Bianco, D. J. (2014). The Pyramid of Pain | Enterprise Detection & Response. *17-01-2014*, 11.

Blasco, J., Hernandez-Castro, J. C., Tapiador, J. E., & Ribagorda, A. (2012). Bypassing information leakage protection with trusted applications. *Computers & Security*, *31*(4), 557-568. https://doi.org/10.1016/j.cose.2012.01.008

Brewer, R. (2014). Advanced persistent threats: Minimising the damage. *Network Security*, *2014*(4), 5-9. https://doi.org/10.1016/S1353-4858(14)70040-6

Bryant, B. D., & Saiedian, H. (2020). Improving SIEM alert metadata aggregation with a novel kill-chain based classification model. *Computers & Security*, *94*, 101817. https://doi.org/10.1016/j.cose.2020.101817

Canbay, Y., Yazici, H., & Sagiroglu, S. (2017). A Turkish language based data leakage prevention system. *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*, 1-6. https://doi.org/10.1109/ISDFS.2017.7916514

Canfora, G., Medvet, E., Mercaldo, F., & Visaggio, C. A. (2015). Detecting Android Malware Using Sequences of System Calls. *Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile*, 13-20. https://doi.org/10.1145/2804345.2804349

Canzanese, R., Mancoridis, S., & Kam, M. (2015). System Call-Based Detection of Malicious Processes. *2015 IEEE International Conference on Software Quality, Reliability and Security*, 119-124. https://doi.org/10.1109/QRS.2015.26

Catak, F. O., Javed Ahmed, Kevser Sahinbas, & Zahid Hussain Khand. (2020). *Mal-API-2019*. Windows Malware Dataset with PE API Calls. https://github.com/ocatak/malware_api_class

Catal, C., & Nangir, M. (2017). A sentiment classification model based on multiple classifiers. *Applied Soft Computing*, *50*, 135-141. https://doi.org/10.1016/j.asoc.2016.11.022

Chaabi, Y., & Ataa Allah, F. (2022). Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram. *Journal of King Saud University - Computer and Information Sciences*, *34*(8), 6116-6124. https://doi.org/10.1016/j.jksuci.2021.07.015

Chen, P., Desmet, L., & Huygens, C. (2014). A Study on Advanced Persistent Threats. İçinde C. Salinesi, M. C. Norrie, & Ó. Pastor (Ed.), *Advanced Information Systems Engineering* (C. 7908, ss. 63-72). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44885-4_5

Choi, J., Choi, C., Lynn, H. M., & Kim, P. (2015). Ontology Based APT Attack Behavior Analysis in Cloud Computing. *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, 375-379. https://doi.org/10.1109/BWCCA.2015.69

CISA. (2020, Haziran 30). *Advanced Persistent Threat Activity Exploiting Managed Service Providers*. https://www.cisa.gov/news-events/alerts/2018/10/03/advanced-persistent-threat-activity-exploiting-managed-service

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., & Stein, Clifford. (2022). The Rabin–Karp algorithm. İçinde *Introduction to Algorithms* (4. bs, ss. 962-966). MIT Press. https://courses.csail.mit.edu/6.006/spring11/rec/rec06.pdf

Cosma, G., & Joy, M. (2012). An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis. *IEEE Transactions on Computers*, *61*(3), 379-394. https://doi.org/10.1109/TC.2011.223

Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2022). Comparison of Entropy Calculation Methods for Ransomware Encrypted File Identification. *Entropy*, *24*(10), 1503. https://doi.org/10.3390/e24101503

DEF CON. (2022). *DEF CON Capture the Flag Archive*. DEF CON Capture the Flag Archive. https://defcon.org/html/links/dc-ctf.html

Du, D., Yu, L., & Brooks, R. R. (2015). Semantic Similarity Detection For Data Leak Prevention. *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 1-6. https://doi.org/10.1145/2746266.2746270

EasyHook. (2023, Ekim). *EasyHook*. https://easyhook.github.io/

Falco, M., & Robiolo, G. (2019). A Systematic Literature Review in Multi-Agent Systems: Patterns and Trends. *2019 XLV Latin American Computing Conference (CLEI)*, 1-10. https://doi.org/10.1109/CLEI47609.2019.235098

FireEye. (2019). *Double Dragon: APT41, a Dual Espionage and Cyber Crime Operation*. 1-68.

FireEye. (2015, Haziran 23). Operation Clandestine Wolf – Adobe Flash Zero-Day in APT3 Phishing Campaign. *Operation Clandestine Wolf – Adobe Flash Zero-Day in APT3 Phishing Campaign*. https://www.fireeye.com/blog/threat-research/2015/06/operation-clandestine-wolf-adobe-flash-zero-day.html

Forcepoint. (2016). *TRITON AP-DATA and TRITON AP-ENDPOINT*. https://www.forcepoint.com/sites/default/files/resources/files/brochure_triton_ap_data_en.pdf

Forcepoint. (2022). *Forcepoint F1E End User Guide*. https://help.forcepoint.com/F1E/en-us/v21/ep_end_user/ep_end_user.pdf

Garbe, W. (2022). *SymSpell* [C#]. https://github.com/wolfgarbe/SymSpell

Garg, V., & Yadav, R. K. (2019). Malware Detection based on API Calls Frequency. *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 400-404. https://doi.org/10.1109/ISCON47742.2019.9036219

Gessiou, E., Vu, Q. H., & Ioannidis, S. (2011). IRILD: An Information Retrieval Based Method for Information Leak Detection. *2011 Seventh European Conference on Computer Network Defense*, 33-40. https://doi.org/10.1109/EC2ND.2011.21

Grégio, A., Bonacin, R., De Marchi, A. C., Nabuco, O. F., & De Geus, P. L. (2016). An ontology of suspicious software behavior. *Applied Ontology*, *11*(1), 29-49. https://doi.org/10.3233/AO-160163

gRPC. (2023, Ekim). *gRPC*. gRPC. https://grpc.io/

Gupta, P. (2020). A Context-Sensitive Real-Time Spell Checker with Language Adaptability. *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 116-122. https://doi.org/10.1109/ICSC.2020.00023

Gupta, S., Sharma, H., & Kaur, S. (2016). Malware Characterization Using Windows API Call Sequences. Içinde C. Carlet, M. A. Hasan, & V. Saraswat (Ed.), *Security, Privacy, and Applied Cryptography Engineering* (C. 10076, ss. 271-280). Springer International Publishing. https://doi.org/10.1007/978-3-319-49445-6_15

Han, W., Xue, J., Wang, Y., Zhang, F., & Gao, X. (2021). APTMalInsight: Identify and cognize APT malware based on system call information and ontology knowledge framework. *Information Sciences*, *546*, 633-664. https://doi.org/10.1016/j.ins.2020.08.095

Haroon, M. (2018). Comparative Analysis of Stemming Algorithms for Web Text Mining. *International Journal of Modern Education and Computer Science*, *10*(9), 20-25. https://doi.org/10.5815/ijmecs.2018.09.03

Hart, M., Manadhata, P., & Johnson, R. (2011). Text Classification for Data Loss Prevention. Içinde S. Fischer-Hübner & N. Hopper (Ed.), *Privacy Enhancing Technologies* (C. 6794, ss. 18-37). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22263-4_2

Hassen, M., Carvalho, M. M., & Chan, P. K. (2017). Malware classification using static analysis based features. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1-7. https://doi.org/10.1109/SSCI.2017.8285426

Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*. 14.

InfoWatch Analytics Center. (2018). *Data Breach Report: A Study on Global Data Leaks in H1 2018*. https://infowatch.com/sites/default/files/report/analytics/Data_Breach_Report_Global_Data_Leaks_H1_2018.pdf

Jacob, G., Debar, H., & Filiol, E. (2008). Behavioral detection of malware: From a survey towards an established taxonomy. *Journal in Computer Virology*, *4*(3), 251-266. https://doi.org/10.1007/s11416-008-0086-0

Kafka, F. (2018). *Eset's Guide to Deobfuscating and Devirtualizing Finfisher* (January). ESET. https://www.eset.com/me/whitepapers/wp-finfisher/

Karlberger, C., Bayler, G., Kruegel, C., & Kirda, E. (2007). Exploiting redundancy in natural language to penetrate Bayesian spam filters. *Proceedings of the first USENIX workshop on Offensive Technologies*, 9.

Katz, G., Elovici, Y., & Shapira, B. (2014). CoBAn: A context based model for data leakage prevention. *Information Sciences*, *262*(June 2002), 137-158. https://doi.org/10.1016/j.ins.2013.10.005

Kaya, E., Özçelik, İ., & Can, Ö. (2019). An Ontology Based Approach for Data Leakage Prevention Against Advanced Persistent Threats. *Research Conference on Metadata and Semantics Research*, 115-125.

Kesenek, Y. (2019). *Zararlı Yazılım Kaynaklı Veri Kaçırma Ataklarına Karşı Doküman Sınıflandırma Algoritması Geliştirme* [Master Thesis]. Sakarya University.

Kesenek, Y., Özçelik, İ., & Kaya, E. (2021). Zararlı yazılım kaynaklı veri kaçırma ataklarına karşı yeni bir doküman sınıflandırma algoritması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*. https://doi.org/10.17341/gazimmfd.641580

Ki, Y., Kim, E., & Kim, H. K. (2015). A Novel Approach to Detect Malware Based on API Call Sequence Analysis. *International Journal of Distributed Sensor Networks*, *11*(6), 1-9. https://doi.org/10.1155/2015/659101

Kim, M., Dey, S., & Lee, S.-W. (2019). Ontology-Driven Security Requirements Recommendation for APT Attack. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 150-156. https://doi.org/10.1109/REW.2019.00032

Korban, C. A., Miller, D. P., Pennington, A., & Thomas, C. B. (2017). *APT3 Adversary Emulation Plan*. MITRE.

Kozachok, A. V., & Kozachok, V. I. (2018). Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis. *Journal of Computer Virology and Hacking Techniques*, *14*(3), 225-231. https://doi.org/10.1007/s11416-017-0309-3

Lajevardi, A. M., & Amini, M. (2019). A semantic-based correlation approach for detecting hybrid and low-level APTs. *Future Generation Computer Systems*, *96*, 64-88. https://doi.org/10.1016/j.future.2019.01.056

Landwehr, C., BULL, A., Mcdermott, J., & CHOI, W. (1994). A Taxonomy of Computer Program Security Flaws. *ACM Computing Surveys*, *26*(3). https://doi.org/10.1145/185403.185412

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., & Bizer, C. (2015). DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, *6*(2), 167-195. https://doi.org/10.3233/SW-140134

Lemay, A., Calvet, J., Menet, F., & Fernandez, J. M. (2018). Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, *72*, 26-59. https://doi.org/10.1016/j.cose.2017.08.005

Li, M., Huang, W., Wang, Y., Fan, W., & Li, J. (2016). The study of APT attack stage model. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 1-5. https://doi.org/10.1109/ICIS.2016.7550947

Lindqvist, U., & Jonsson, E. (1997). How to systematically classify computer security intrusions. *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*, 154-163. https://doi.org/10.1109/SECPRI.1997.601330

Luh, R., Schrittwieser, S., & Marschalek, S. (2016). TAON: An ontology-based approach to mitigating targeted attacks. *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*, 303-312. https://doi.org/10.1145/3011141.3011157

Maheshwari, A. (2018, Temmuz 17). Report on Text Classification using CNN, RNN & HAN. *Jatana*. https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f

Mandiant Intelligence Center. (2013). *APT1 Exposing One of China's Cyber Espionage Units* (s. 74). Mandiant Intelligence Center. https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

Martins, B., & Silva, M. J. (2004). Spelling Correction for Search Engine Queries. Içinde J. L. Vicedo, P. Martínez-Barco, R. Muńoz, & M. Saiz Noeda (Ed.), *Advances in Natural Language Processing* (C. 3230, ss. 372-383). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30228-5_33

Matasano. (2007). *Defeating Extrusion Detection*. ttp://www.blackhat.com/ presentations/bh-usa-07/Monti_and_Moniz/ Presentation/bh-07-monti_and_moniz.pdf

McAfee. (2022). *McAfee Data Loss Prevention Endpoint*. https://www.mcafee.com/enterprise/en-us/assets/data-sheets/ds-dlp-endpoint.pdf

Micah Yates. (2017). *APT3 Uncovered: The code evolution of Pirpi*. https://recon.cx/2017/montreal/resources/slides/RECON-MTL-2017-evolution_of_pirpi.pdf

Micro Focus & OpenText. (2021). *KeyView*. https://www.microfocus.com/en-us/products/file-viewer-filter/overview

Microsoft. (2002, Ocak 16). Detours. *Microsoft Research*. https://www.microsoft.com/en-us/research/project/detours/

Microsoft. (2021). *Windows API*. https://docs.microsoft.com/en-us/windows/win32/api/

Mirza, N. A. S., Abbas, H., Khan, F. A., & Al Muhtadi, J. (2014). Anticipating Advanced Persistent Threat (APT) countermeasures using collaborative security mechanisms. *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, 129-132. https://doi.org/10.1109/ISBAST.2014.7013108

MIT Lincoln Laboratory. (2000). *2000 DARPA Intrusion Detection Evaluation Data Set*. 2000 DARPA Intrusion Detection Evaluation Data Set. https://archive.ll.mit.edu/ideval/data/2000data.html

MITRE. (2018). *APT3 Enterprise Evaluation*. https://attackevals.mitre-engenuity.org/enterprise/apt3/

MITRE. (2019a, Temmuz 16). *Accessibility Features, Technique T1015*. https://attack.mitre.org/versions/v6/techniques/T1015/

MITRE. (2019b, Temmuz 18). *Process Injection, Technique 1055*. Process Injection. https://attack.mitre.org/versions/v6/techniques/T1055/

MITRE. (2019c, Ekim). *Data Encrypted, Technique T1022*. https://attack.mitre.org/versions/v6/techniques/T1022/

MITRE. (2019d, Ekim). *Data Staged, Technique T1074*. https://attack.mitre.org/versions/v6/techniques/T1074/

MITRE. (2019e, Ekim). *Exfiltration Over Alternative Protocol, Technique T1048*. Exfiltration Over Alternative Protocol. https://attack.mitre.org/versions/v6/techniques/T1048/

MITRE. (2019f, Ekim). *PowerShell, Technique T1086*. https://attack.mitre.org/versions/v6/techniques/T1086/

MITRE. (2020a). *MITRE ATT&CK*. https://attack.mitre.org/versions/v6/

MITRE. (2020b, Haziran 9). *MITRE APT3*. APT3. https://attack.mitre.org/versions/v6/groups/G0022/

MITRE. (2021). *MITRE Caldera*. https://github.com/mitre/caldera

Mogull, R. (2010). *Understanding and Selecting a Data Loss Prevention Solution*. Securosis. https://securosis.com/assets/library/publications/DLP-Whitepaper.pdf

Moon, D., Im, H., Kim, I., & Park, J. H. (2017). DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of Supercomputing*, *73*(7), 2881-2895. https://doi.org/10.1007/s11227-015-1604-8

Moon, Daesung, Lee, H., & Kim, Ikkyun. (2014). Host based Feature Description Method for Detecting APT Attack. *Journal of The Korea Institute of Information Security & Cryptology*, *24*(5), 839-850. https://doi.org/10.13089/JKIISC.2014.24.5.839

Mustafa, T. (2013). Malicious Data Leak Prevention and Purposeful Evasion Attacks: An approach to Advanced Persistent Threat (APT) management. *2013 Saudi International Electronics, Communications and Photonics Conference*, 1-5. https://doi.org/10.1109/SIECPC.2013.6551028

Ned Moran, Mike Scott, Mike Oppenheim, & Joshua Homan. (2015). *Operation Double Tap*. Mandiant. https://www.mandiant.com/resources/blog/operation-doubletap

Nguyen, N., Reiher, P., & Kuenning, G. H. (2003). Detecting insider threats by monitoring system call activity. *IEEE Systems, Man and Cybernetics SocietyInformation Assurance Workshop, 2003.*, 45-52. https://doi.org/10.1109/SMCSIA.2003.1232400

Nissim, N., Lapidot, Y., Cohen, A., & Elovici, Y. (2018). Trusted system-calls analysis methodology aimed at detection of compromised virtual machines using sequential mining. *Knowledge-Based Systems*, *153*, 147-175. https://doi.org/10.1016/j.knosys.2018.04.033

nltk. (2021). *NLTK-Natural Language Toolkit*. https://www.nltk.org/

Ora Lassila & Ralph R. Swick. (1999, Şubat 22). *Resource Description Framework (RDF) Model and Syntax Specification*. Resource Description Framework (RDF) Model and Syntax Specification. https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Orgill, G. L., Romney, G. W., Bailey, M. G., & Orgill, P. M. (2004). The Urgency for Effective User Privacy-Education to Counter Social Engineering Attacks on Secure Computer Systems. *Proceedings of the 5th Conference on Information Technology Education*, 177-181. https://doi.org/10.1145/1029533.1029577

Ponemon Institute LLC & IBM. (2023). *Cost of a Data Breach Report 2023*.

Prachi., Dabas, N., & Sharma, P. (2023). MalAnalyser: An effective and efficient Windows malware detection method based on API call sequences. *Expert Systems with Applications*, *230*, 120756. https://doi.org/10.1016/j.eswa.2023.120756

Priya, M., Kalpana, R., & Srisupriya, T. (2017). Hybrid optimization algorithm using N gram based edit distance. *2017 International Conference on Communication and Signal Processing (ICCSP)*, 0216-0221. https://doi.org/10.1109/ICCSP.2017.8286823

Purple Team ATT&CK Automation. (2021). https://github.com/praetorian-code/purple-team-attack-automation

PwC. (2017). *Operation Cloud Hopper* (s. 25). PwC.

Rashid, A., Ramdhany, R., Edwards, M., Kibirige Mukisa, S., Ali Babar, M., Hutchison, D., & Chitchyan, R. (2014). *Detecting and Preventing Data Exfiltration*. Lancaster University. https://web.archive.org/web/20160807050730/http://www.lancaster.ac.uk/media/lancaster-university/content-assets/images/security-lancaster/seculanc_data_exfil_report.pdf

Ravi, C., & Manoharan, R. (2012). Malware Detection using Windows API Sequence and Machine Learning. *International Journal of Computer Applications*, *43*(17), 12-16. https://doi.org/10.5120/6194-8715

Red Team Automation. (2021). https://github.com/endgameinc/RTA

Ruder, S. (2017). *An overview of gradient descent optimization algorithms* (arXiv:1609.04747). arXiv. http://arxiv.org/abs/1609.04747

Saha, S., & Ekbal, A. (2013). Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, *85*, 15-39. https://doi.org/10.1016/j.datak.2012.06.003

Schultz, M. G., Eskin, E., Zadok, F., & Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, 38-49. https://doi.org/10.1109/SECPRI.2001.924286

scikit-learn. (2020, Ağustos 4). *scikit-learn: Machine learning in Python*. https://scikit-learn.org/0.23/index.html

Sealpath. (2020, Haziran 23). ➤The Three States of Data Guide—Description and How to Secure them. *Sealpath*. https://www.sealpath.com/blog/protecting-the-three-states-of-data/

Shabtai, A., Elovici, Y., & Rokach, L. (2012). *A Survey of Data Leakage Detection and Prevention Solutions*. Springer US. https://doi.org/10.1007/978-1-4614-2053-8

Shapira, Y., Shapira, B., & Shabtai, A. (2013). *Content-based data leakage detection using extended fingerprinting*.

Sharma, D. (2012). Stemming Algorithms: A Comparative Study and their Analysis. *International Journal of Applied Information Systems*, *4*(3), 7-12. https://doi.org/10.5120/ijais12-450655

Shu, X., & Yao, D. (Daphne). (2013). Data Leak Detection as a Service. İçinde A. D. Keromytis & R. Di Pietro (Ed.), *Security and Privacy in Communication Networks* (ss. 222-240). Springer Berlin Heidelberg.

Simpson, G. G. (1961). *Principles of Animal Taxonomy*. Columbia University Press. https://books.google.com.tr/books?id=U-muavt8MnEC

Singh, S., Sharma, P. K., Moon, S. Y., Moon, D., & Park, J. H. (2019). A comprehensive study on APT attacks and countermeasures for future networks and communications: Challenges and solutions. *The Journal of Supercomputing*, *75*(8), 4543-4574. https://doi.org/10.1007/s11227-016-1850-4

Suguru Ishimaru. (2022, Ekim 31). *APT10: Tracking down LODEINFO 2022, part II*. https://securelist.com/apt10-tracking-down-lodeinfo-2022-part-ii/107745/

Symantec. (2021). *Symantec Data Loss Prevention*. https://docs.broadcom.com/doc/data-loss-prevention-family-en

Symantec. (2022). *Symantec Data Loss Prevention Product Brief*.

Symantec. (2016, Eylül 6). *Symantec Security Response*. Buckeye cyberespionage group shifts gaze from US to Hong Kong. https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=92a4528c-2bdb-498f-85c8-4273bfdc66aa

The UCI KDD Archive. (1999). *KDD Cup 1999 Data*. KDD Cup 1999 Data. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Tok, M. S., & CeliKtas, B. (2019). MuddyWater APT Grubu ve Makro Zararlı Yazılım Analizi Metodolojisi Önerisi. *Bilişim Teknolojileri Dergisi*, 253-263. https://doi.org/10.17671/gazibtd.512800

Tolegenova, A. (2022). Automatic Error Correction: Evaluating Performance of Spell Checker Tools. *Suleyman Demirel University Bulletin: Natural and Technical Sciences*, *58*(1), 15-21. https://doi.org/10.47344/sdubnts.v58i1.690

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117-126. https://doi.org/10.1016/j.eswa.2016.03.028

Turcotte, M. J. M., Kent, A. D., & Hash, C. (2018, Kasım). *Unified Host and Network Data Set*. Unified Host and Network Data Set. https://doi.org/10.1142/9781786345646_001

Väisänen, T., Trinberg, L., & Pissanidis, N. (2016). I accidentally malware—What should I do... İs this dangerous? Overcoming inevitable risks of electronic communication. *NATO CCD COE*.

Verizon. (2023). *2022 Data Breach Investigations Report*. https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf

W3C. (2012, Aralık 11). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. https://www.w3.org/TR/owl2-overview/

W3C. (2014, Şubat 25). *RDF Schema 1.1*. https://www.w3.org/TR/rdf-schema/

Weeks, M. (2013). *Ambush* [Ruby]. https://github.com/scriptjunkie/Ambush

Woo, S., On, J., & Lee, M. (2013). Behavior ontology: A framework to detect attack patterns for security. *Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013*, 738-743. https://doi.org/10.1109/WAINA.2013.42

Wüest, C. (2013). *Targeted attacks: How sophisticated are they really?* https://2013.swisscyberstorm.com/files/Targeted-Attacks_Candid-Wueest.pdf

Xiang Zhang. (2023, Kasım 2). *Dbpedia_14*. https://huggingface.co/datasets/dbpedia_14

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-Level Convolutional Networks for Text Classification. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 649-657.

EKLER

EK 1. APTON SWRL Kuralları

Bu bölümde tez kapsamında sunulan APTON ontolojisi içinde APT teknik ve risklerinin tespit edilmesini sağlayan SWRL kuralları listelenmektedir. Tablolardaki teknikler MITRE ATT&CK çerçevesindeki teknik numaraları esas alınarak sıralanmışlardır. Tekniğin MITRE ATT&CK içerisindeki kodu, ismi ve sonrasında tespit için kullanılan kurallar gösterilmektedir.

Tablo A.1. APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1002 Data Compressed	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CompressionLibrary(?\,targetObjType), LoadLibrary(?\,sysCallCat)$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow DataCompressed(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CompressedFile(?\,targetObjType), WriteFile(?\,sysCallCat),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow DataCompressed(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CompressionExecutable(?\,targetObjType), CreateProcess(?\,sysCallCat),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow DataCompressed(?\,apt)$
T1003 Credential Dumping	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $ReadProcessMemory(?\,sysCallCat),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow CredentialDumping(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $ReadRegistry(?\,sysCallCat), LocalSystemRegistrySecurity(?\,targetDataLocation),$ $hasTargetDataLocation(?\,sysCall, ?\,targetDataLocation),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow CredentialDumping(?\,apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1003 Credential Dumping	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadFile(?sysCallCat),$ $TrustedRemoteLocationDomServerSysVol(?targetDataLocation),$ $hasTargetDataLocation(?sysCall, ?targetDataLocation),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow CredentialDumping(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadFile(?sysCallCat), LocalSystemFileSecurity(?targetDataLocation),$ $hasTargetDataLocation(?sysCall, ?targetDataLocation),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow CredentialDumping(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadFile(?sysCallCat), LocalSystemFileUsers(?targetDataLocation),$ $hasTargetDataLocation(?sysCall, ?targetDataLocation2),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow CredentialDumping(?apt)$
T1015 Accessibility Features	$owl{:}Thing(?aptTechnique), owl{:}Thing(?aptTactic), NotTrustedProcess(?p),$ $PrivilegedUser(?user), AccessibilityExecutable(?fileType),$ $WriteFile(?sysCallCategory), Executable(?exe)$ $LocalSystemFile(?dataLoc),$ $hasTargetObjectType(?sysCall, ?fileType),$ $hasSystemCallCategory(?sysCall, ?sysCallCategory),$ $isCalledByUser(?sysCall, ?user),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow isUtilizedByAptTechnique(?exe, ?aptTechnique),$ $AccessibilityFeatures(?aptTechnique),$ $Persistence(?aptTactic)$
T1016 System Network Configuration Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), NetworkConfigurationTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow SystemNetworkConfigurationDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkParameters(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConfigurationDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkConfigInfo(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConfigurationDiscovery(?apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat),$ $LocalSystemRegistryNetworkConfig(?targetDataLocation)$ $hasTargetDataLocation(?sysCall, ?targetDataLocation),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConfigurationDiscovery(?apt)$
T1018 Remote System Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkDiscoveryInfo(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow RemoteSystemDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkShareInfo(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow RemoteSystemDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), NetworkDiscoveryTool(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow RemoteSystemDiscovery(?apt)$
T1022 Data Encrypted	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $Content(?content), ContentAttack(?contentMod),$ $hasContent(?sysCall, ?content), hasContentModification(?content, ?contentMod)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow DataEncrypted(?apt)$
T1033 System Owner/User Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p), ReadRegistry(?sysCallCat),$ $LocalUserRegistryVolatileEnv(?targetDataLocation),$ $hasSystemCall(?apt, ?sysCall), hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasTargetDataLocation(?sysCall, ?targetDataLocation),$ $isCalledByProcess(?sysCall, ?p)$ $\rightarrow SystemOwnerUserDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat), LocalSystemRegistryUserDetails(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $isCalledByProcess(?sysCall, ?p), hasSystemCall(?apt, ?sysCall),$ $\rightarrow SystemOwnerUserDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), OwnerDiscoveryTool(?sysCallCat)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemOwnerUserDiscovery(?apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1041 Exfiltration Over Command-and-Control Channel	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $InitializeNetworkConnection(?sysCallCat),$ $Collection(?preDetectedCollectionTactic)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow ExfiltrationOverCommandControlChannel(?apt)$
T1049 System Network Connections Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), NetworkStatusDiscoveryTool(?targetObjType)$ $hasTargetObjectType(?sysCall, ?targetObjType),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConnectionsDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkResourcesInfo(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConnectionsDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $GetNetworkDiscoveryInfo(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemNetworkConnectionsDiscovery(?apt)$
T1053 Scheduled Task	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $WriteFile(?sysCallCat), LocalSystemFileScheduledTasks(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow ScheduledTask(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ComObjectScheduledTask(?targetObjType), ComCreateInstance(?sysCallCat)$ $hasTargetObjectType(?sysCall, ?targetObjType),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow ScheduledTask(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateFile(?sysCallCat), LocalSystemFileScheduledTasks(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $\rightarrow ScheduledTask(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ConfigScheduledTask(?sysCallCat),$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow ScheduledTask(?apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1053 Scheduled Task	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CreateProcess(?\,sysCallCat), TaskSchedulerTool(?\,targetObjType),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ScheduledTask(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $WriteRegistry(?\,sysCallCat), LocalSystemRegistryScheduledTasks(?\,targetDataLoc),$ $hasTargetDataLocation(?\,sysCall, ?\,targetDataLoc)$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ScheduledTask(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CreateScheduledTask(?\,sysCallCat),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat)$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ScheduledTask(?\,apt)$
T1055 Process Injection	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $WriteProcessMemory(?\,sysCallCat),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow ProcessInjection(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CreateProcess(?\,sysCallCat), DllInjectionTool(?\,targetObjType),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType)$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ProcessInjection(?\,apt)$
T1057 Process Discovery	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CreateProcess(?\,sysCallCat), ServiceListingTool(?\,targetObjType),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType)$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ProcessDiscovery(?\,apt)$
	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $CreateProcess(?\,sysCallCat), ProcessDiscoveryTool(?\,targetObjType),$ $hasTargetObjectType(?\,sysCall, ?\,targetObjType)$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow ProcessDiscovery(?\,apt)$
T1069 Permission Groups Discovery	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $GetAccountsInfo(?\,sysCallCat),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat)$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p),$ $\rightarrow PermissionGroupsDiscovery(?\,apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1069 Permission Groups Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), GroupsDiscoveryTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow PermissionGroupsDiscovery(?apt)$
T1074 Data Staged	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $FileDiscoveryCall(?sysCallCat), Content(?content),$ $UnstructuredContent(?privacyLevel),$ $hasContent(?sysCall, ?content), hasContentPrivacyLevel(?content, ?privacyLevel)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow DataStaged(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadFile(?sysCallCat), Content(?content), UnstructuredContent(?privacyLevel),$ $hasContent(?sysCall, ?content), hasContentPrivacyLevel(?content, ?privacyLevel)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow DataStaged(?apt)$
T1076 Remote Desktop Protocol	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), RemoteDesktopTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow RemoteDesktopProtocol(?apt)$
T1082 System Information Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat), LocalSystemRegistryNetworkConfig(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemInformationDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), SystemInformationCollectionTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemInformationDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat), LocalSystemRegistrySystemInfo(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow SystemInformationDiscovery(?apt)$
T1083 File and Directory Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), FileDiscoveryTool(?targetObjType),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasTargetObjectType(?sysCall, ?targetObjType),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow FileDirectoryDiscovery(?apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1087 Account Discovery	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), GroupsDiscoveryTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow AccountDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), AccountDiscoveryTool(?targetObjType),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow AccountDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat), LocalSystemRegistryUserDetails(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow AccountDiscovery(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $ReadRegistry(?sysCallCat), LocalSystemRegistryNetworkConfig(?targetDataLoc),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow AccountDiscovery(?apt)$
T1136 Create Account	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $WriteRegistry(?sysCallCat), LocalSystemRegistrySecurity(?targetDataLocation),$ $PrivilegedUser(?user)$ $isCalledByUser(?sysCall, ?user),$ $hasTargetDataLocation(?sysCall, ?targetDataLocation),$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p)$ $\rightarrow CreateAccount(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $WriteFile(?sysCallCat), LocalSystemFileSecurity(?targetDataLoc),$ $PrivilegedUser(?userClass),$ $isCalledByUser(?sysCall, ?userClass),$ $hasTargetDataLocation(?sysCall, ?targetDataLoc)$ $hasSystemCallCategory(?sysCall, ?sysCallCat)$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow CreateAccount(?apt)$
	$owl{:}Thing(?apt), NotTrustedProcess(?p),$ $CreateProcess(?sysCallCat), UserCreationTool(?targetObjType),$ $LocalAdminUser(?userClass),$ $isCalledByUser(?sysCall, ?userClass),$ $hasTargetObjectType(?sysCall, ?targetObjType)$ $hasSystemCallCategory(?sysCall, ?sysCallCat),$ $hasSystemCall(?apt, ?sysCall), isCalledByProcess(?sysCall, ?p),$ $\rightarrow CreateAccount(?apt)$

Tablo A.1 (Devamı) APT Teknik Tespit Kuralları.

APT Tekniği	SWRL Kuralları
T1136 Create Account	$owl{:}Thing(?\,apt), NotTrustedProcess(?\,p),$ $AddUser(?\,sysCallCat),$ $hasSystemCallCategory(?\,sysCall, ?\,sysCallCat),$ $hasSystemCall(?\,apt, ?\,sysCall), isCalledByProcess(?\,sysCall, ?\,p)$ $\rightarrow CreateAccount(?\,apt)$

Tablo A.2. APT Risk Tespit Kuralları.

APT Riski	SWRL Kuralları
Muhtemel APT Riski (Possible APT Risk)	$owl{:}Thing(?\,aptRisk)$ $Discovery(?\,aptTacDiscovery),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery),$ $\rightarrow PossibleAptActivityRisk(?\,aptRisk)$
Kalıcı APT Riski (Persistent APT Risk)	$owl{:}Thing(?\,aptRisk),$ $Discovery(?\,aptTacDiscovery), Persistence(?\,aptTacPersistence)$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacPersistence)$ $\rightarrow PersistentAptRisk(?\,aptRisk)$
Bulaşıcı APT Riski (Contagious APT Risk)	$owl{:}Thing(?\,aptRisk),$ $Discovery(?\,aptTacDiscovery),$ $LateralMovement(?\,aptTacLateralMovement),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacLateralMovement)$ $\rightarrow ContagiousAptRisk(?\,aptRisk)$
APT Veri Toplama Riski (APT Data Collection Risk)	$owl{:}Thing(?\,aptRisk),$ $Discovery(?\,aptTacDiscovery),$ $LateralMovement(?\,aptTacLateralMovement),$ $Collection(?\,aptTacCollection),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacLateralMovement)$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacCollection),$ $\rightarrow AptDataCollectionRisk(?\,aptRisk)$
APT Veri Sızdırma Riski (APT Data Exfiltration Risk)	$owl{:}Thing(?\,aptRisk),$ $Discovery(?\,aptTacDiscovery),$ $LateralMovement(?\,aptTacLateralMovement),$ $Collection(?\,aptTacCollection),$ $Exfiltration(?\,aptTacExfiltration)$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacLateralMovement),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacCollection),$ $containsTacticForRisk(?\,aptRisk, ?\,aptTacExfiltration),$ $\rightarrow AptDataExfiltrationRisk(?\,aptRisk)$

Tablo A.2. (Devamı) APT Risk Tespit Kuralları.

APT Riski	SWRL Kuralları
İçerik-tabanlı atak içeren APT Veri Sızdırma Riski (APT Data Exfiltration Using Content Attack Risk)	$owl{:}Thing(?\,aptRisk)$, $Discovery(?\,aptTacDiscovery)$, $LateralMovement(?\,aptTacLateralMovement)$, $Collection(?\,aptTacCollection)$, $Exfiltration(?\,aptTacExfiltration)$ $DataEncrypted(?\,aptTechDataEncrypted)$, $containsTacticForRisk(?\,aptRisk, ?\,aptTacDiscovery)$, $containsTacticForRisk(?\,aptRisk, ?\,aptTacLateralMovement)$, $containsTacticForRisk(?\,aptRisk, ?\,aptTacCollection)$, $containsTacticForRisk(?\,aptRisk, ?\,aptTacExfiltration)$, $containsTechniqueForRisk(?\,aptRisk, ?\,aptTechDataEncrypted)$, $\rightarrow AptDataExfiltrationUsingContentAttackRisk(?\,aptRisk)$

ÖZGEÇMİŞ

Ad-Soyad **:** Emrah KAYA

Erişim Adresi **:** emrah.kaya at gmail dot com

ÖĞRENİM DURUMU:

- **Yüksek lisans** **:** 2009, Gebze Yüksek Teknoloji Enstitüsü (Gebze Teknik Üniversitesi), Bilgisayar Mühendisliği Anabilim Dalı
- **Lisans** **:** 2006, Marmara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM:

- 2006-2018 yılları arasında TÜBİTAK'ta araştırmacı olarak dağıtık sistemler ve simülasyonlar üzerinde çalıştı.
- 2018 yılından bu yana uluslararası alanda bulut bilişim konularında çalışmaya devam etmektedir.

YAYINLAR:

- Kesenek, Y., Özçelik, İ., & Kaya, E. (2021). Zararlı yazılım kaynaklı veri kaçırma ataklarına karşı yeni bir doküman sınıflandırma algoritması. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi. https://doi.org/10.17341/gazimmfd.641580
- Kaya E., Özçelik İ., Can Ö. (2019) An Ontology Based Approach for Data Leakage Prevention Against Advanced Persistent Threats. In: Garoufallou E., Fallucchi F., William De Luca E. (eds) Metadata and Semantic Research. MTSR 2019. Communications in Computer and Information
- Emrah Kaya and Fatih Erdoğan Sevilgen. 2009. A fully distributed data collection method for HLA based distributed simulations. In Proceedings of the 2009 Summer Computer Simulation Conference (SCSC '09). Society for Modeling & Simulation International, Vista, CA, 337–347.