T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ANALİZ SÜRECİNİ ATLATMAYA ÇALIŞAN ZARARLI YAZILIMLAR VE DERİN ÖĞRENME TEMELLİ ZARARLI YAZILIM TESPİTİ

İRFAN BULUT

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
DOÇ. DR. ALİ GÖKHAN YAVUZ

İSTANBUL, 2017

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ANALİZ SÜRECİNİ ATLATMAYA ÇALIŞAN ZARARLI YAZILIMLAR VE DERİN ÖĞRENME TEMELLİ ZARARLI YAZILIM TESPİTİ

ANALİZ SÜRECİNİ ATLATMAYA ÇALIŞAN ZARARLI YAZILIMLAR VE DERİN ÖĞRENME TEMELLİ ZARARLI YAZILIM TESPİTİ

İrfan BULUT tarafından hazırlanan tez çalışması 05.06.2017 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Ali Gökhan Yavuz

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Ali Gökhan Yavuz

Yıldız Teknik Üniversitesi _____

Prof. Dr. Selim Akyokuş

Doğuş Üniversitesi _____

Doç. Dr. Songül Varlı Albayrak

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

"*Yakın gelecekte çıkabilecek büyük bir savaşta ilk mermi internette atılacaktır*."

(Rex Hughes – **NATO Güvenlik Danışmanı**)

Zararlı yazılımlar, günümüzde artık sadece meraklılar, amatörler ya da sıradan siber korsanlar tarafından değil, belirli ve özel amaçlara ulaşmak için bilgili ve uzman siber suçlular ya da devlet çalışanları tarafından tasarlanmakta ve geliştirilmektedir. Günümüzde zararlı yazılımları tasarlayan, *geliştiren* ve *bundan fayda sağlayanlar* bireysel siber saldırganlar, organize olmuş siber suç grupları ya da devlet kurumları arasında dağılım göstermektedir. Siber uzaydaki zararlı yazılımların sayısının, kabiliyetlerinin gün geçtikçe artması ve yapılarının daha da karmaşıklaşmasıyla bir cihaza bulaşması, etkilerinin herkesi içine alacak kadar genişlemesi ve insan hayatına etki edecek noktaya gelmesi kaçınılmaz olarak görülmektedir. Bu nedenle zararlı yazılımlarla mücadele siber güvenlik için çok kritik bir yerde durmaktadır. Umarım ülkemiz adına farkındalık oluşturacak bir tez çalışması olmuştur.

Bu çalışmanın gerçekleştirilmesinde, iki yıl boyunca değerli bilgilerini benimle paylaşan danışman hocam; Doç. Dr. Ali Gökhan YAVUZ'a, çalışmam boyunca yardımlarını esirgemeyen Yrd. Doç. Dr. M. Amaç Güvensan'a ve çalışma süresince tüm zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan değerli aileme sonsuz teşekkürlerimi sunarım.

Mayıs, 2017

İrfan BULUT

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ

AV	Anti-virüs
DDoS	Distributed Denial of Service attack
MD5	Message Digest algorithm 5
APK	Android Package Kit
APT	Advanced Persistent Threat
API	Application Programming Interface

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

ANALİZ SÜRECİNİ ATLATMAYA ÇALIŞAN ZARARLI YAZILIMLAR VE DERİN ÖĞRENME TEMELLİ ZARARLI YAZILIM TESPİTİ

İrfan BULUT

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Ali Gökhan YAVUZ

Yaşadığımız çağ, bilgiyi üretme ve ulaşma hızının her geçen gün artmasıyla, bilişim çağı olarak adlandırılmaktadır. Sürekli bir gelişim içerisinde olan bilişim teknolojileri, ihtiyacımız olan bilgiye daha hızlı ve daha kolay bir şekilde ulaşmamıza imkân sağlamaktadır. Türkiye İstatistik Kurumu'nun 2015 yılı nisan ayına ait verilerinde, Türkiye'de internet kullanım oranı %55,9 seviyesine ulaşmıştır. Geniş bant internet erişim imkânına sahip olan hanelerin oranı ise %67,8'dir. Aynı araştırma sonuçlarına göre cep telefonu veya akıllı telefona sahip olma oranı ise %96,8'e kadar yükselmiştir. İstatistiklerden de anlaşıldığı üzere, kullanıcı sayısının artması yanında internete bağlı cihazların çeşitlenmesi de üzerinde durulması gereken bir noktadır. Bilgi ve iletişim sistemleri sadece son kullanıcı için genişlememekte aynı zamanda enerji, haberleşme, su kaynakları, tarım, sağlık, ulaşım, eğitim ve finansal hizmetler gibi kritik altyapı sektörlerinde faaliyet gösteren kurum ve kuruluşlarda da etkin bir biçimde kullanılmaktadır. Dolayısıyla bu sistemlere yapılan siber saldırılar, bu sistemlerin hizmet dışı kalmasına, kötüye kullanılmasına, büyük ölçekli ekonomik zarara, kamu düzeninin bozulmasına, can kaybına ve hatta ulusal güvenliğin zedelenmesine neden olma riskini taşımaktadır. 2007 yılında, Estonya'da meydana gelen siber saldırı bir ülkenin kritik altyapılarının siber saldırılar karşısında ne kadar kırılgan olabileceğini gözler önüne sermiştir. Ve bu olay tüm dünya da devletler nezdinde siber güvenliğin daha bilincli olarak ele alınmasını sağlayan bir olay olarak kayıtlara geçmiştir. Bu saldırı siber güvenlik anlamında bir kırılma noktası oluşturmuştur. Bir başka deyişle, siber saldırı/savaş senaryolarının yalnızca birer komplo teorisi olmadığı görülmüş ve başta

NATO olmak üzere BM, AGİT gibi birçok kurum askeri güvenlik politikalarını siber dünyayı da dikkate alarak revize etmeye başlamışlardır. Kritik altyapılar ve bu altyapıların korunması konuları ülkemizde ulusal bilgi güvenliği başlığı altında değerlendirilmiştir. Ulusal bilgi güvenliği konusunda ülkemizde yürütülen çalışmaların geçmişi ise gelişmiş ülkelerin bu konu üzerinde çalışmalar yaptığı 1990'ların sonlarına kadar gitmektedir. Şu ana kadar ortaya çıkan en önemli yasal düzenleme 20 Ekim 2012 tarihli Resmi Gazetede "Ulusal Siber Güvenlik Çalışmalarının Yürütülmesi ve Koordinasyonuna ilişkin Bakanlar Kurulu Kararı"dır. Bu önemli kararla ülkemizin siber güvenliğinin sağlanması konusunda idari, teknik ve hukuki yapıların oluşturulması hız kazanmış, siber güvenliğe ilişkin koordinasyonun sağlaması amacıyla, "Siber Güvenlik Kurulu" oluşturulmuştur. Bu kurulun ilk toplantısında "Ulusal Siber Güvenlik Stratejisi ve 2013-2014 Eylem Planı " kabul edilmiş ve alınan kararlar 20 Haziran 2013 tarihinde Bakanlar Kurulu Kararı olarak yayımlanmıştır. Bu eylem planının 5 numaralı maddesinde Siber Güvenlik Kurulu'nca ülkemizin kritik altyapıları bilgi güvenliği kapsamında ilk etapta ulaşım, enerji, elektronik haberleşme, finans ve kritik kamu hizmetleri gibi alanlar öncelikli olarak belirlenmiştir.

Tarihteki en ünlü başkomutanlardan Sun Tzu 'nun M.Ö. 5. yüzyılda yazmış olduğu Savaş Sanatı adlı kitapta maalesef günümüzün modern savaşı olan siber savaşlarda neler yapılması gerektiğine ilişkin bir tavsiye yer almamaktadır. Siber savaşlar geleneksel savaşlardan farklı da olsa bir analoji yapılabilir ve siber savaşların mermisi olarak zararlı yazılımları gösterebiliriz. Bu nedenler zararlı yazılımlarla mücadele ve bilgi birikimi önemli bir üstünlük noktası ve savunma stratejisidir. Zararlı yazılımlarla mücadalede alanında hazırlanan bu tezle ulusal siber güvenlik stratejisinin bir parçası olarak katkı sağlamak hedeflenmiştir. Bu kapsamda analiz sürecini atlatmaya çalışan modern zararlı yazılımlar incelenmiş ve derin öğrenme üzerinden probleme çözüm getirilmiştir.

Anahtar Kelimeler: Zararlı yazılım, mobil zararlı yazılım, sandbox, zararlı yazılım analizi, anti-analiz, derin öğrenme, android

ABSTRACT

EVASIVE MALWARE AND DEEP NEURAL NETWORK BASED MALWARE DETECTION

İrfan BULUT

Department of Computer Engineering

MSc. Thesis

Advisor: Assoc. Prof. Dr. Ali Gökhan YAVUZ

The age we live in is called as the age of information, with the speed of producing and reaching information increasing day by day. Information technologies that are constantly evolving allow us to reach the information we need faster and more easily. In the statistics of the Turkish Statistical Institute for April 2015, the Internet usage rate in Turkey has reached 55.9%. The rate of the households with broadband Internet access is 67.8%. According to the same research results, the rate of having a mobile phone or smartphone has increased to 96.8%. As it can be understood from the statistics, the number of users connected to the Internet is also a point that should be emphasized. Information and communication systems are used not only for the end user but also for institutions and organizations active in the critical infrastructure sectors such as energy, communication, water resources, agriculture, health, transportation, education and financial services. Therefore, the cyber attacks on these systems carry the risk of causing these systems to become out of service, abuse, large-scale economic damage, deterioration of public order, loss of life and even national security. The 2007 cyber attack, which took place in Estonia, showed how fragile the critical infrastructures could be in the face of cyber attacks. And this event has entered the record as an event that allows the whole world to treat cyber security more consciously. This attack created a breaking point in the sense of security of cyberspace. In other words, cyber attack / war scenarios were seen as not only conspiracy theories,

but many organizations such as NATO, UN, and OSCE have begun to revise their military security policies taking into account the cyber world. Critical infrastructures and the protection of these infrastructures have been assessed under the title of national information security in our country. The history of the work carried out in our country on national information security goes back to the late 1990s when developed countries were working on this issue. The most important legal regulation that has arisen so far is the Official Gazette dated October 20, 2012 "Decision of the Council of Ministers on the Execution and Coordination of National Cyber Security Work". With this important decision, the establishment of administrative, technical and legal structures for the provision of cyber security of our country has gained momentum and the "Cyber Security Council" has been established in order to ensure coordination of cyber security. The "National Cyber Security Strategy and 2013-2014 Action Plan" was adopted at the first meeting of this committee and the decisions were published as a decision of the Council of Ministers on 20 June 2013. In the article 5 of this action plan, priority areas such as transportation, energy, electronic communication, finance and critical public services have been determined in the first stage within the information infrastructure of the critical infrastructure of our country by the Cyber Security Council.

One of the most famous commanders in history is Sun Tzu. In the 5th century book War Art, unfortunately there is no recommendation about what to do in the cyber war, which is the modern day war. Even if cyber warfare is different from traditional warfare, an analogy can be made and we can show malware as bullet of cyber warfare. These reasons are the fight against malware and the accumulation of knowledge is an important advantage and defense strategy. It was aimed to contribute to this thesis, prepared in the field of combat with modern malware, as part of the national cyber security strategy. In this context, modern malware trying to overcome (evasive) the analysis process is examined and probing solution is introduced through deep learning.

Keywords: Malware, mobile malware, sandbox, malware analyses, evasive malware, anti-analyses, zero-day malware, deep learning, android

BÖLÜM 1

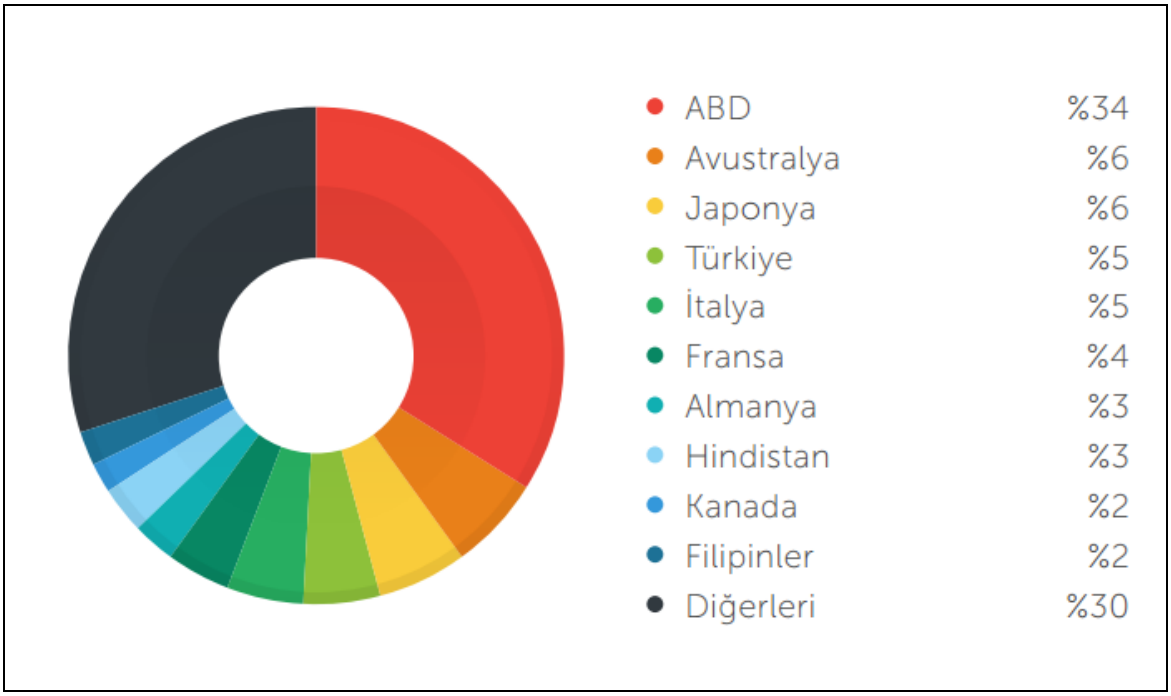
GİRİŞ

1.1 Literatür Özeti

Bilişim, insanoğlunun teknik, ekonomik ve toplumsal alanlardaki iletişiminde kullandığı ve bilimin dayanağı olan bilginin özellikle elektronik cihazlar aracılığıyla düzenli ve akla uygun bir biçimde işlenmesi/aktarılması bilimi olarak tanımlanmaktadır. Suçun tanımı ise, toplumsal düzen içerisinde oluşturulmuş kanunların yasakladığı ve yapıldığında cezai bir müeyyidesinin olduğu her türden davranış olarak tarif edilebilir. Günümüz dünyasında ise özellikle kişisel bilgisayarlar ve mobil iletişim cihazlarının gelişmesi ve geniş bant internetin yaygınlaşması ile orantılı olarak genişleyen ve karmaşıklaşan bir iletişim ağı (siber uzay) oluşmuştur. İnsan hayatını kolaylaştıran yönlerine paralel olarak, bu iletişim ağı üzerinde artan bir suç potansiyelini de beraberinde getirmektedir. Yukarıda da anlattığımız üzere bilişim ve suç tanımları dikkate alındığında bilişim suçu kavramı daha iyi anlaşılabilir olmaktadır. Bu noktada bilişim suçlarının merkezine zararlı yazılımları koymak ve zararlı yazılımların odağında siber suç/bilişim suçlarını değerlendirmek yanlış olmayacaktır. Bizim de çalışmamızda odaklandığımız konu olan zararlı yazılımları bilişim suçları ekseninde incelemek gerekir.

İlk zararlı yazılımın ortaya çıkış tarihi tam olarak bilinmemesine rağmen tahmini olarak 1960-70'li yıllar olduğu kabul edilmektedir. İlk zamanlar zararlı yazılımlar için hedef sistemler sadece bilgisayarlardı ve yazılan kodlar deneysel çalışmalar içermekteydi. 1987 yılında Fred Cohen Bilgisayar Virüsleri Teori ve Deneyler (Computer Viruses – Theory and Experiments) konulu çalışmasıyla bilgisayar virüsü terimi ilk kez akademik

olarak kullanılmış oldu. 1987 yılında “Christmas Tree” adındaki solucan ilk geniş çaplı zararlı yazılımlardan biriydi ve tam bu yıllarda ilk anti-virus Bernd Fix tarafından geliştirilmiştir. Almanya’da ortaya çıkan CASCADE adlı virüs ise kendini şifreleyerek saklamaya çalışan ilk virüs olmuştur. 1988 yılında Robert Tappan Morris’in yazdığı solucan (Morris Worms) bilinen ilk geniş çaplı salgındı. Morris solucanı dahil, o günlerde hazırlanan zararlı yazılımların çoğu zarar vermek amacıyla geliştirilmemişlerdi.



Şekil 1.1 Fidye yazılımların ülkelere göre dağılımı [30]

Zararlı yazılımlar yeterince test edilmedikleri için kolayca kontrolden çıkabiliyor ve istenmeden de olsa ciddi zararlara yol açıyorlardı. Morris solucanı da yayılmak için o günlerde UNIX sistemlerde olan bir açıklığı kullanmıştı. Hedef sistemi yeniden enfekte etmemek için bir kontrol mekanizması koyulmadığı için bilgisayarların servis dışı kalana kadar yeniden ve yeniden (recursive) enfekte olmalarına yol açmış ve öngörülemeyen şekilde tüm Interneti etkilemiştir. Fakat başta bu şekilde ilerleyen zararlı yazılımlar belirli bir aşamadan sonra deneysel olmaktan çıkmış ve finansal bir gelir kaynağı ve suç unsuru haline gelmiş. Daha sonra ise devletler açısından stratejik bir noktaya evrilmiştir. Bu noktadan sonra stratejik bir silah olarak ciddi bir problem olarak kullanıcılarının, şirketlerin ve devletlerin karşısına çıkmaya başlamıştır. Şekil 1. 1 ‘de görebileceğimiz üzere Türkiye 2015 yılına ait verilere göre fidye türü zararlı yazılımların

en çok görüldüğü ülkeler arasındandır. Eğlence ve kendini kanıtlama amaçlı zararlı yazılımlar yerini

- reklam,
- istenmeyen eposta (*spam*) gönderimi,
- dosyalarınızı şifreleyip karşılığında fidye isteme
- kişisel bilgilerin çalınması (*kredi kartı, bankacılık şifreleri vs.*),
- büyük ölçekli servis dışı bırakma saldırıları (*DDoS*)
- devletler arası istihbari ve stratejik bilgiler elde etme

gibi çok çeşitli hedefleri gerçekleştirebilmek için geliştirilen zararlı yazılımlara bırakmıştır. Bu faaliyetlerde elde edilen kazançlar, suçlular arasındaki rekabet, yakalanmama ve daha çok kazanma motivasyonu zararlı yazılımların teknik gelişimini ciddi biçimde arttırmış ve karşımıza profesyonel zararlı yazılımlar çıkmaya başlamıştır.

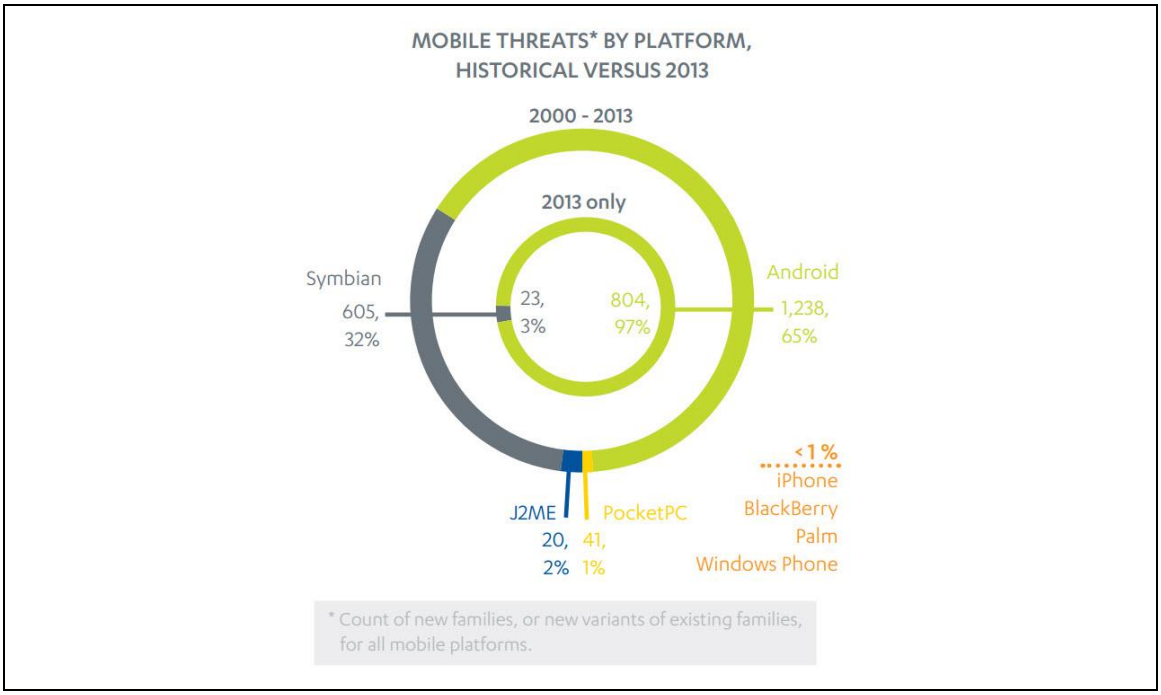
Geçmişte bilişim sistemleri arasında veri alış verişi sınırlı yöntemlerle (*taşınabilir disketler/kısıtlı internet bağlantıları*) sağlanabiliyordu ve sistemler arasındaki etkileşim çok yaygın olmamasından dolayı bir zararlı yazılımın yayılabilmesi için çok geniş bir zaman dilimi gerektiriyordu [2]. Bu da zararlı yazılımın çok yayılmadan ve zararlı etkileri büyümeden keşfedilebilmesini ve gerekli önlemlerin zamanında alınmasına olanak sağlıyordu. Fakat bugün Internet'in çok hızlı ve yaygın olması, neredeyse tüm cihazların internet bağlantısını olduğu/olacağı bir siber dünyada zararlı yazılımların çok hızlı bir şekilde yayılabilmesine ortam sağlamıştır. Geçmişe göre kullanılan uygulamalardaki çeşitlilik ve işlevsellik beraberinde bir çok güvenlik açıklığının ortaya çıkmasına ve zararlı yazılımlara bu açıklıkları (*tarayıcı vb.*) istismar (*exploit*) edebilmesinin yollarını açmıştır. Ayrıca işletim sistemlerinin yapısal olarak gelişmesi (*daha büyük ve daha karmaşık bir eve dönüşmesi gibi*) ile zararlı yazılımlara bulaşabilecekleri, sonrasında gizlenebilecekleri ve daha sonra varlıklarını uzun süre devam ettirebilecekleri bir ekosistem sağlamıştır.

Zararlı yazılımların gelişmesinde rol oynayan diğer faktör ise hukuki faktörlerdir. Zararlı yazılım geliştiricilerin/grupların yakalanması ve cezalandırılması, caydırıcılığa neden olduğu için saldırganların bir kısmı (*amatörler*) bu işten çıkmasına geride kalanların ise daha yetenekli ve tespit edilmesi zor zararlı yazılımlar geliştirmelerine sebep olmuştur [1]. Zararlı yazılımların etkili olmasında bir diğer faktör de bilişim sistemlerinin kullanımının yaygınlaşmasına rağmen kullanıcıların güvenlik bilincinin çok düşük

olmasıdır. Hala siber dünya da en zayıf halka insan ve bu durum uzun süre değişmeyecek gibi durmaktadır. Tehlike sadece siber suçlular tarafıdan ele geçirilen bilgiler , bozulan sistemler veya giden paralarınız değil aynı zamanda kişisel bilgilerinizi toplamaya çalışan devlet(*ler*) mekanizması da bireylerin zararlı yazılımlara ve genel olarak siber güvenlik kavramına gereken önemi vermesini zorunlu kılmaktadır. Bu nokta da bireylerde hakim olarak bulunan "benim saklayacak bir şeyim yok. Bilgilerim başkalarının eline geçse ne fark eder ki?" gibi algısal problemlerinin olmasıdır. Burada şu unutulmamalıdır ki kişisel bilgilerinizin siz izin versenizde vermesenizde elde edilmesi yani sizden onay alınmaması ve "*Herkes, özel hayatına ve aile hayatına saygı gösterilmesini isteme hakkına sahiptir. Özel hayatın ve aile hayatının gizliliğine dokunulamaz.*" ilkesine ters olarak haklarınızın ihlal edilmesidir. Bu nedenle devletlerden veya siber korsanlardan saklıyacak hiç bir şeyiniz olmasa bile hakkınızın gasp edilmesine/özgürlüğünüzün elinizden alınmasına izin vermemek bilinçli bir birey ve sağlıklı bir toplum için önem arzetmektedir.

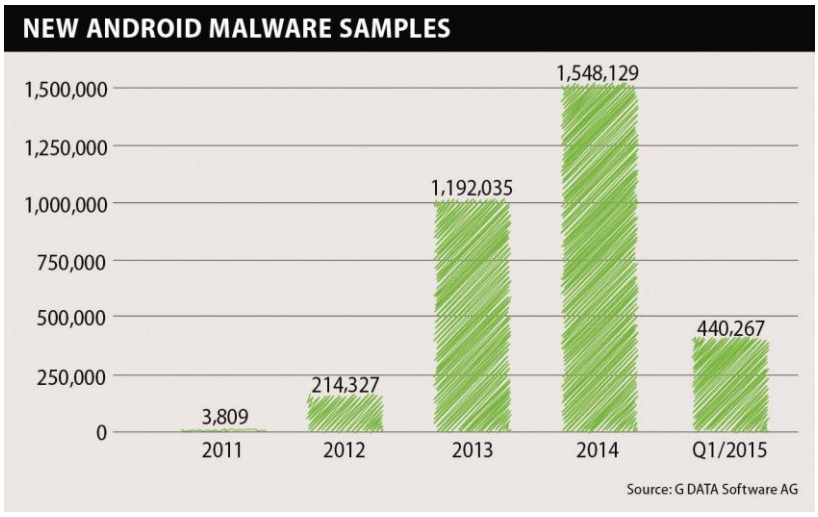
Zararlı yazılımlar değişen ortamla birlikte başlangıçta virüs adında ilerlese de bu tür yazılımların gösterdiği gelişim ile beraber virüs kelimesi günümüzdeki zararlı yazılımları ifade etmek için yetersiz kalmıştır. Daha genel ve kapsayıcı olarak ingilizcede **mal**icious soft**ware** kelimelerinden türetilen **malware** (*Türkçe'de bunun karşılığı olan bir terim üretilmese de*) bir üst çatı olarak kullanılmaya başlanmıştır ve **virüs** zararlı yazılımın türlerinden birini ifade eden bir terim haline gelmiştir. Zararlı yazılımların analiz ve tespitinde başlangıçta insan merkezli ve statik olarak gerçekleştirilmekteydi. Ardından zararlı yazılım geliştiricilerin statik analizi zorlaştıracak teknikler bulmasından dolayı ve sanallaştırma teknolojisindeki gelişmelerin sağladığı imkanlarla hedef yazılım çalıştırılarak bir sonuç elde edilmeye çalışılmıştır. Fakat daha sonra üretilen zararlı yazılım sayısı o kadar gelişme göstermiştir ki kaçınılmaz olarak analiz ve tespit süreci otomatize (*yapay sinir ağları ve makine öğrenmesi yöntemleri*) edilmeye çalışılmıştır [3] [4] [5]. Fakat zararlı yazılım geliştiricileri de buna karşı otomatize analizi atlatmak için farklı yöntemler geliştirmişlerdir [1] [6]. Bu noktada tezin ikinci amacı olarak derin öğrenme algoritmalarıyla bu durumu çözmeye yönelik bir yaklaşım getirilmiş ve kayda değer bir başarı sağlanmıştır. Mobil zararlı yazılım düyasında en önemli hedef platform Android işletim sistemidir. Aşağıda Şekil 1. 2'de görülebileceği üzere 2015 yılına ait

tespit edilen zararlı yazılım verilerinde mobil cihazlarda zararlı yazılım istatistikleri gözükmektedir.



Şekil 1. 2 Mobil zararlı yazılım hedef işletim sistemi oranları [29]

Bu noktada hedef platform olarak Android'in seçilmesinin nedeni tehlikenin daha çok mobil dünyada bu platformuna yönelik olmasıdır. Şekil 1. 3'de 2011 yılından 2015 yılına kadar olan Android platformu için tespit edilen zararlı yazılım örnekleri sayısına bakılırsa üretilmesi gereken çözümün otomatize olmasının kaçınılmaz zorunluluğunu göstermektedir.



Şekil 1. 3 Yeni android tabanlı zararlı yazılım üretilme miktarları [31]

1.2 Tezin Amacı

Günümüzde en önemli siber güvenlik sorunlarından birisi yeni/daha önceden tespit edilememiş (*zero-day malware*) zararlı yazılımların hızlı bir şekilde tespit edilmesidir. Mevcut AV (*anti-malware*) sistemleri, daha önceden tespit edilen zararlı yazılımların imzalarını bulundurdukları veritabanları üzerinden bilinen zararlı yazılımlara karşı oldukça başarılı olurken, daha önceden karşılaşılmamış ya da tespit edilmemiş yeni zararlı yazılımlara karşı başarısız olmaktadır [1]. Ayrıca günümüzün akıllı ve Tinba[1] örneğinde olduğu gibi hedef odaklı ve karmaşık yetenekleri (*Polymorphic & Metamorphic Malware*) olan zararlı yazılımlarla mücade etmek için hızlı çözümler üretmek gerekmektedir. Bu nedenle hızlı çözümlerde kaçınılmaz yöntem **analiz sürecinin otomatize** edilmesidir. Fakat burada ise karşımıza otomatize analiz sürecinin atlatılması (*evasive*) için yeteneklerini geliştiren (*özellikle son yıllarda çok fazla görmeye başladığımız*) zararlı yazılımlar çıkmaktadır [6]. Yapılacak basit kontrollerle analiz edildiğini anlayabilmek ve kendisini masum bir yazılım olarak sisteme göstermek özellikle mobil sistemler için çok kolay hale gelmiştir. Bu sayede analiz sürecini atlatmak (*evasive malware*) başvurulan yaygın bir yöntem haline gelmiştir.

1.3 Orjinal Katkı

Sonuç olarak tez sonucunda yapılan orjinal katkıyı iki başlıkta değerlendirmek mümkündür. Tezden üretilen birinci katkı zararlı yazılım analizi açısından kendini gizleme, analiz edilmesini ve tespit edilmesini zorlaştıran yöntemleri belirtmek ve yalıtılmış bir ortamda çalıştığını anlamak için nelere başvurabileceğini ortaya çıkarmak ve bu noktada farkındalık oluşturmaktır. Üretilen ikinci katkı ise günümüzde GPU sistemlerinin işlem gücünün gelişmesiyle beraber daha çok alanda görmeye başladığımız derin öğrenme (*deep learning*) yöntemini [7] kullanarak bu tür zararlı yazılımlar için etkin bir tespit etme mekanizması ortaya koymaktır.

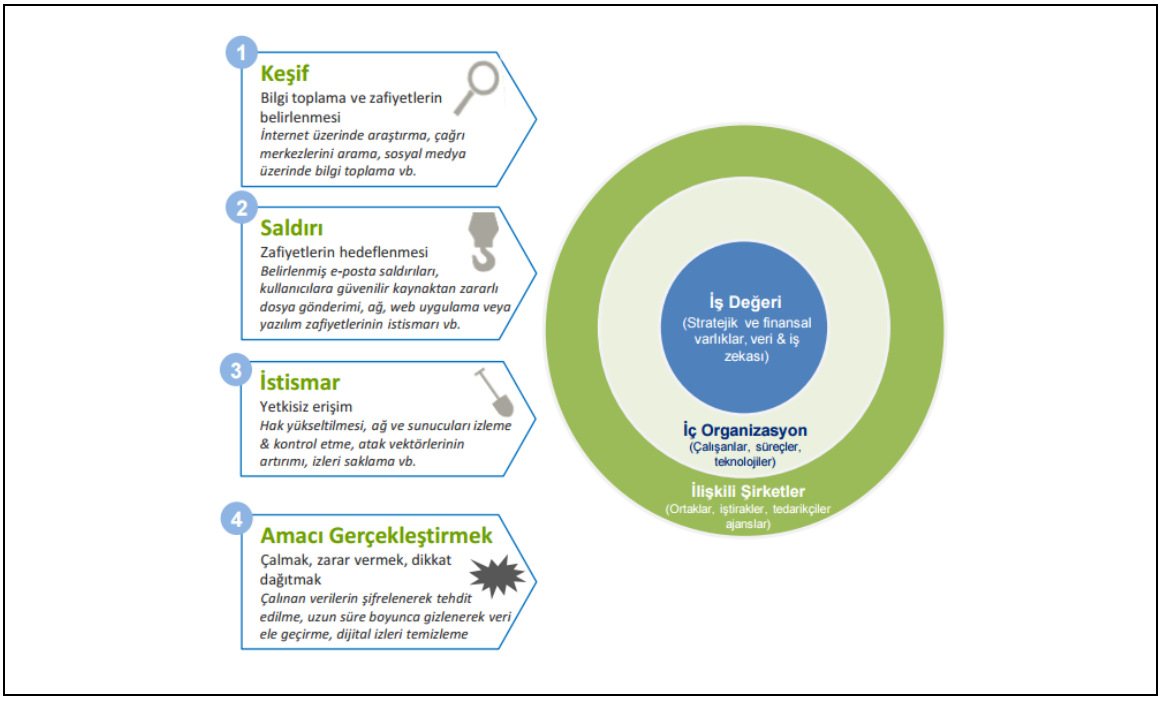
---

[1] Tinba zararlı yazılımı tiny (küçük) ve bank (banka) kelimelerinin birleştirilmesinden oluşmaktadır. Tinba'yı asıl önemli kılan ise özellikle Kuzey Kore, Pakistan, Filistin, Türkiye gibi askeri ve stratejik açıdan önemli ülkelerde için özelleşmiş olması ve Türkiye üzerinde de aktif olmasıdır.

BÖLÜM 2

SİBER SALDIRI ANATOMİSİ

Siber saldırı, hedef seçilen şahıs, şirket, kurum veya devletin bilişim sistemlerinin işleyişinin engellenmesi, bozulması, değiştirilmesi yada stratejik bilgilerin elde edilip yayınlanması yoluyla iş, ticaret, kamu düzeninin ve hiyerarjisinin bozulması için yapılan siber girişimlerin toplamıdır. "*Bir siber saldırı nasıl gerçekleşmekte ve zararlı yazılım bir siber saldırının hangi aşamasında devreye girmektedir?*" sorusunu açıklığa kavuşturmak zararlı yazılımların siber dünyadaki yerini anlamak için büyük önem arzetmektedir. Bu noktada bir siber saldırını anatomisini anlamak için 4 genel gruplandırma yapmak mümkündür.



Şekil 2. 1 Bir siber saldırının anatomisi [32]

2.1 Keşif (*Bilgi Toplama*)

Saldırı yapılacak hedef sistem ya da sistemlerin zayıf noktalarını belirleyebilemek ve saldırı noktalarını tespit edebilmek için bilgi toplama süreci çok önemli bir yer tutmaktadır. Bu noktada iyi bir çalışma yapılması saldırının başarıya ulaşması için en önemli adımı temsil etmektedir. Keşif süreci aslında klasik savaş stratejisinde düşmanın zayıf noktasını bulma aşamasına karşılık gelmektedir. Bu aşama ne kadar iyi yapılırsa bundan sonra atılacak adımların o kadar başarılı olmasına etki edecektir.

2.2 Saldırı

Keşif işlemi gerçekleştirildikten sonraki adım bulunan güvenlik açıkları sayesinde sisteme sızma sürecidir. Bu süreç sistemin güvenlik parametrelerine veya hedef sistemin siber güvenlik konseptine yakınlığı ile test orantılı zorlukta gerçekleşir. Saldırının anatomisine göre zararlı yazılım bu noktada (*mail ile zararlı yazılımın indirme linkinin gönderilmesi, dışarıdan usb ile sisteme bulaştırma vb.*) devreye girmeye başlar.

2.3 İstismar (*Yetkisiz Erişim*)

Saldırı süreci sonunda başarılı bir şekilde sisteme sızıldı ise sonraki adım sisteme erişimdir. Bu noktadan sonra zararlı yazılım siber saldırganlara hizmet etmeye başlar. Saldırgan bilgiye yada sistem kaynaklarına (*yazılım, donanım veya veri*) yetkisi olmadığı halde erişebilmesi bu adımın başarılı olduğu anlamına gelir. Aynı bilgiye yetkili kullanıcılar da olağan şekilde erişebilirler, yani bilginin kendisinde bir bozulma olmayabilir. Bu da saldırının farkedilmesini ilk etapta zorlaştırmaktadır. Bununla birlikte o bilgiye erişmesi istenmeyen kişilerin erişmesi de, istismar olarak nitelendirilir.

2.4 Amacı gerçekleştirme

Yetkisiz erişim gerçekleştiktan sonra sisteme sızıldı ise bu noktadan sonra hedeflenen aksiyonların gerçekleştirilme süreci başlamaktadır. Bu noktada yapılanlar aşağıdaki gibi gruplanabilir:

- **Erişimi Engelleme veya Zarar Verme**: Bilgiye erişim engellenir. Bilgi kaybolmuştur, silinmiştir veya kullanılamaz durumdadır. Veriler ulaşılabilir

durumda (DdoS vb.) değildir veya yetkili kullanıcılar tarafından kullanılamaz (şifreleme vb.) durumdadır.

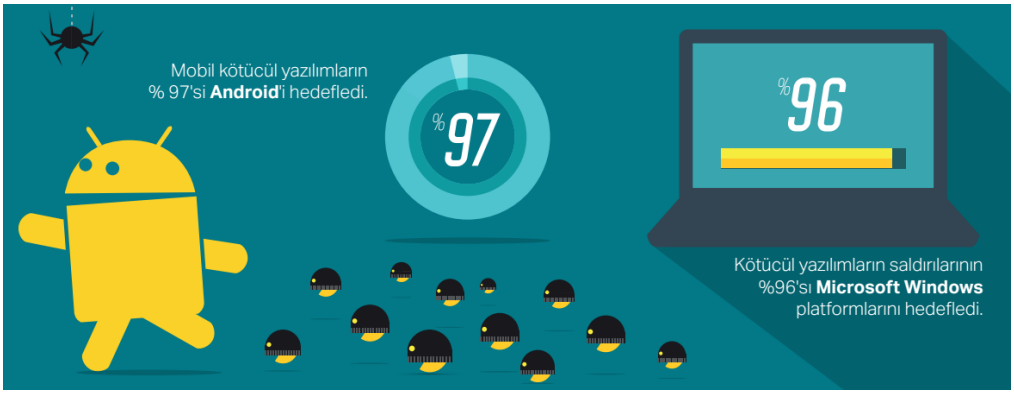
- **Değişiklik Yapma**: Bilginin yetkili kullanıcıya ulaşmadan önce saldırganın amaçları doğrultusunda bilgide değişiklik yapması anlamına gelir. Üretilen veri, daha önceki gerçek bir verinin taklidi olabileceği gibi, gerçeğe uygun tamamen yeni bir veri şeklinde olabilir.
- **Kopyalama**: Gerçekte kişisel veya özel olması gereken verinin çoğaltılması ve yetkisi olmayan kişilerin erişmesi anlamına gelir.
- **Sistem kaynaklarını sömürme**: Doğrudan bulaştığı sisteme bir zararı olmasa da DDoS benzeri amaçlar için sistem kaynakları kullanılabilir.
- **Kalıcı olmaya çalışma**: Sistemde uzun süre kalınmak isteniyorsa bunun için gerekli stratejik hamlelerin yapılmasıdır.

Sonuç olarak zararlı yazılımlar saldırı yapılacak sistem(ler) ne kadar önemli ya da korunaklı ise geliştirilen zararlı yazılımlar da o kadar konveksiyonel olmaktadır. Ayrıca uzun süreli amaçlar hedeflenmiş ise kendini gizlemek ve tespit edilme olasılığını aza indirmek için o kadar gelişmiş yöntemler kullanılmaktadır. Dolayısıyla şu tespiti yapmak yanlış olmayacaktır: “siber saldırı ne kadar konveksiyonel ise geliştirilen yazılım da o kadar teknik bilgi gerektirmektedir.”

BÖLÜM 3

SİBER GÜVENLİKTE TÜRKİYE'NİN DURUMU

"2016-2019 Ulusal Siber Güvenlik Stratejisi ve Eylem Planının ana amacı; siber güvenliğin ulusal güvenliğin ayrılmaz bir parçası olduğu anlayışının tüm kesimlerde yerleşmesi, ulusal siber uzayda bulunan sistem ve paydaşların tamamının güvenliğini sağlamak üzere idari ve teknolojik önlemlerin alınmasını sağlayacak yetkinliğin eksiksiz bir şekilde kazanılması" hedeflenmiştir [35]. Bu amaca yönelik olarak siber uzayın güvenliğinde bir alt başlık olan zararlı yazılımlarla mücadele önemli bir yer tutmaktadır. Türkiye'de merkezi bir otorite tarafından zararlı yazılımlarla ve daha genelinde siber güvenlik ile ilgili yeterince istatistik toplanarak paylaşılmamasına rağmen güvenlik konusunda çalışan uluslararası firmaların yayımlanan raporları ve Türkiye ile ilgili kısımları incelendiğinde tüm dünyada olduğu gibi Türkiye'de de zararlı yazılımların artarak büyüyen bir tehdit oldukları görülmektedir [8].



Şekil 3. 1 Türkiye zararlı yazılım ortam istatistiği [33]

Çizelge 3. 1 'de Kaspersky tarafından dünya üzerinde rapor edilen şifreleme zararlı yazılımların ülke bazlı yüzde ağırlıklarını bulabilirsiniz [36]. Burada Türkiye 8. sırada yer

alması dikkate değerdir. Türkiyeyi ilgilendiren bir diğer raporda ise Labris firmasının verilerine göre ülkemizde gerçekleştirilen saldırılarda Windows ve Android (*Şekil 3. 1*) platformlarına yönelik ciddi bir oran göze çarpmaktadır.

Çizelge 3. 1 Ülkelere göre şifreleme zararlı yazılımların oranı

	Ülke	Şifreleme zararlıların kullanıcılara saldırı oranı (%)
1	Hollanda	1,06
2	Belçika	1,00
3	Rusya	0,65
4	Brezilya	0,44
5	Kazakistan	0,42
6	İtalya	0,36
7	Litvanya	0,34
8	Türkiye	0,31
9	Ukrayna	0,31
10	Avusturya	0,30

Microsoft'un 2013 yılına ait raporda Türkiye, maalesef 10 ülke arasında bir yıl içinde en fazla zararlı yazılım artışı gözlenen ülke olarak öne çıkmaktadır [34]. Türkiye'nin bu oranda zararlı yazılıma maruz kalmasının sebebi olarak, Türkiye'deki son kullanıcıların siber tehditler konusunda farkındalık eksiklikleri ve bilgisayarlarında ya da mobil cihazlarında gerçek zamanlı güvenlik yazılımlarının eksiklikleri olduğunu söyleyebiliriz.

Şekil 3. 2 Dünya genelinde botnet komuta kontrol bilgisayarları ile en çok trafik oluşturan ülkeler [8]

Şekil 3. 2 ve Şekil 3. 3 botnet ağlarına ilişkin STM firmasının yayınladığı 2016 raporunda dramatize bir Türkiye verisine ulaşmak mümkündür. Bunun nedenlerinden biri de ülkemizdeki son yıllarda geniş bant internet kullanımına ilaveten mobil abone sayısı ve mobil internet kullanım oranlarının artmasının da etkili olduğu görülebilir.

Şekil 3. 3 Asya ülkeleri arasında botnet komuta kontrol bilgisayarları ile en çok trafik oluşturan ülkeler [8]

Ülkemizde siber güvenlikte saç ayağının diğer kısımları ise hukuki altyapı ve siber suçlarla mücadele/müdahele teşkilatlanmalarının sağlanmasıdır. Bu amaçlarla Türk ceza kanununun bilişim suçları ile ilgili bölümü 10. bölümdür. Bölüm 5 alt başlık altında değerlendirilmiştir. Bu bölümler;

- Bilişim sistemine girme
- Sistemi engelleme, bozma, verileri yok etme veya değiştirme
- Banka veya kredi kartlarının kötüye kullanılması

- Yasak cihaz veya programlar
- Tüzel kişiler hakkında güvenlik tedbiri uygulanması şeklindedir.

Yaklaşık 1,5 sayfalık metnin tamamını tezin Ek - A kısmında bulabilirsiniz. Bilişim teknolojileri kullanılarak işlenen suçların soruşturulması ve dijital delillerin incelenmesi için destek veren görevli daire başkanlıklarının ve taşra teşkilatındaki birimlerin dağınık yapısının tek bir çatı altında toplanması, mükerrer yatırımların önüne geçilmesi, siber suçlarla mücadelenin etkin ve verimli olarak yürütülmesini sağlamak amacıyla 2011/2025 sayılı Bakanlar Kurulu Kararı ile **Emniyet Genel Müdürlüğü** bünyesinde Bilişim Suçlarıyla Mücadele Daire Başkanlığı kurulmuştur. Daha sonra bu daire 28/02/2013 tarih ve B.05.1.EGM.0.65.35539/31772 sayılı Bakanlık Oluruna istinaden *Bilişim Suçlarıyla Mücadele Daire Başkanlığı*nın ismi **Siber Suçlarla Mücadele Daire Başkanlığı** olarak değiştirilmiştir. Teknik olarak da siber güvenlik kavramı en üst çatı olarak kullanılmaya başlanmıştır. Türkiye de yasama & yürütme konusunda durum yukarıda olduğu gibi buna ek olarak Avrupa komisyonunun 2011 yılında siber suçları kategorize eden bildirisi aşağıdaki gibidir:

- Elektronik Ağlar Aracılığıyla İşlenen Klasik Suçlar
    - Dolandırıcılık
    - Sahtecilik
    - Siber Taciz ve Şantaj
- Elektronik Medya Üzerinde Yayınlanan Yasadışı İçerik
- Elektronik Ağlara Özgü Suçlar
    - Bilgisayar Korsanlığı
    - Hizmeti Engelleme
    - Zararlı Yazılımlar
    - Sosyal Mühendislik Saldırıları

Bu ayrım zararlı yazılımların ayrı bir kategori olarak yer alması dikkate değer bir noktadır. Bu noktada ülkemizde siber güvenliğin durumunu inceledikten sonra zararlı yazılımlar konusuna geçebiliriz.

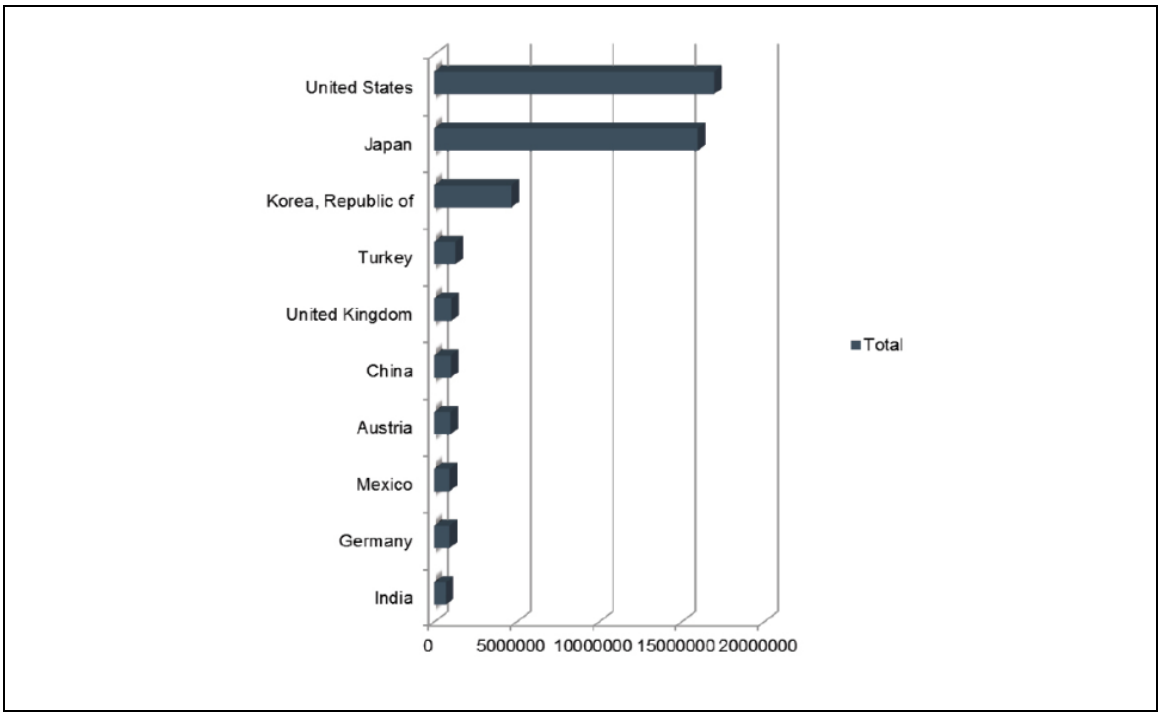
BÖLÜM 4

ZARARLI YAZILIMLAR

Zararlı yazılımlar, genel olarak siber dünyada hedeflenen eylemleri gerçekleştirmek için siber saldırganlar tarafından geliştirilen yazılımlardır. Kendisi doğrudan bir hedef olacağı gibi, bir amaç doğrultusunda yardımcı araç olarak da geliştirilebilir. Zararlı (*malware*) yazılımlar;

- sistemin çalışmasını bozma,
- sistem güvenliğini riske atma,
- sistemlere yetkisiz erişim hakkı kazanma,
- reklam,
- istenmeyen eposta (*spam*),
- dosyalarınızı şifreleyip karşılığında para isteme,
- hassas bilgileri (*kredi kartı, bankacılık şifreleri vs.*) elde etme,
- büyük ölçekli servis dışı bırakma saldırıları (*DDoS*) yapma,
- devletler arası istihbari ve stratejik bilgiler elde etme vb. gibi

kötü amaçlı işlevleri gerçekleştiren yazılımlarıdır [8] [9] [34]. Bu noktada ülkemiz dünyada zararlı yazılım konusunda ciddi tehlike arz eden bir konumda olduğunu Şekil 4. 1 'de görebilirsiniz. Zararlı yazılımlar başlangıçta sadece virüs şeklinde nitelendirilse de daha sonra süreç içinde *Virüs, Truva Atı, Rookit, Solucan, Arka Kapı*, vb. gibi değişik türlerde sınıflandırılmaya başlanılmıştır [10] [11] [34]. Şimdi sırasıyla çeşitlerini inceleyelim.



Şekil 4. 1 En fazla zararlı yazılım görülen ülkeler (*Temmuz-Eylül 2016*) [8]

4.1 Zararlı yazılım çeşitleri

Zararlı yazılımlar, yaygın olarak hedef sistemlerdeki davranış biçimlerine göre sınıflandırılmaktadır. Günümüzde zararlı yazılımlar birden fazla tehlikeli aktiviteyi beraber gerçekleştirdikleri için pratikte çok yalın kategorizasyon yapmak zorlaşmaya başlamıştır. Genel olarak bir sınıflandırma (*taxonomy*) oluşturmak istenirse literatürte olan sınıflandırma aşağıdaki gibidir.

4.1.1 Klavye Dinleme Sistemi (*Keylogger*)

Klavye dinleme sistemleri, klavye girişi bilgilerinizi (*klavye, sanal klavye, vb*) dinleyerek *şifrelerinizi*, *online bankacılık bilgileriniz*, *e-posta* ve Whatsapp, Telegram gibi *yazışma verilerinizi* almaya çalışan zararlı yazılım türüdür. Bu türün önemli özelliği bu bilgileri klavye üzerinden elde etmeye çalışmasıdır.

4.1.2 Truva atı (*Trojan*)

Truva atı zararlı yazılımları ön planda kullanıcının işine yarayacak bir yazılımın/uygulamanın içine gömülmüş zararlı yazılımlardır. Bu sayede kullanıcıyı

şüphelendirmeden arka planda dışarıdan erişim sağlama gibi işleri yerine getirirler. İsmi mitolojide yer alan truva atından gelir.

4.1.3 Solucan (*Worm*)

Bağımsız bir şekilde kendi başına çalışabilen ve kendisini ağdaki başka cihazlara kopyalabilen zararlı yazılımlardır. Yayılmak veya çalışmak için kullanıcı faktörüne ihtiyaç duymamaları en belirgin özellikleridir.

4.1.4 Casus Yazılım (*Spyware*)

Casus yazılımlar, hedef sistemdeki kullanıcı (*hesap bilgileri, şifreler, kişisel veri*) ya da sistem bilgilerini toplayan ve belirli merkezlere aktaran yazılımlardır.

4.1.5 Reklam Yazılımı (*Adware*)

Reklam Yazılımları, sisteminizde reklamlar görüntülemek veya özelleştirilmiş reklamların görüntülenmesi gibi amaçlar için ziyaret ettiğiniz web sitelerini, ilgi alanları olabilecek her tür veriyi toplamak amacıyla tasarlanmış zararlı yazılım türleridir. Bu yazılımların tek gelir odağı reklamlar üzerinedir. Zararlı yazılımlar arasında en az zararlı olan yazılım türüdür.

4.1.6 Fidye Yazılımı (*Ransomware*)

Fidye yazılımı genel olarak korkutma (*klasik yöntem*) veya verilerinizi şifreleyerek fidye isteyen zararlı yazılımlardır. Son yıllarda fidye yazılımları en çok görülen zararlı yazılım türleridir. Ülkemizde de aktif olan CryptoLocker adında çok tehlikeli bir sürümü bir çok insanı etkilemiştir. Fakat bu yazılımın genel kullanıcı kitlesine zararlı yazılımların ne kadar tehlikeli olduğuna ilişkin bir farkındalık etkisi de olmuştur.

4.1.7 Kök Kullanıcı Takımı (*Rootkit*)

Rootkit'ler, en tehlikeli zararlı yazılımlar kategorisinde yer alır. Bulaşılan sistemde kendisini işletim sistemi gibi merkezi noktalara gizlediğinden dolayı hem yaptığı tüm işlemleri (*çalışan süreçler, dosyalar, kayıtlar, internet bağlantıları*) gizleyebilmekte hem de tespit edilmesini zorlaştırabilmektedir. Genelde çekirdek seviyesinde

çalışmalarından dolayı analiz ve tespit süreçleri oldukça zordur. Diğer zararlı yazılım türlerinden farklı olarak ileri seviye işletim sistemi veya donanım bilgisi gerektirir. APT türündeki zararlı yazılımlarla beraber en gelişmiş ve tehlikeli kategorisindedir.

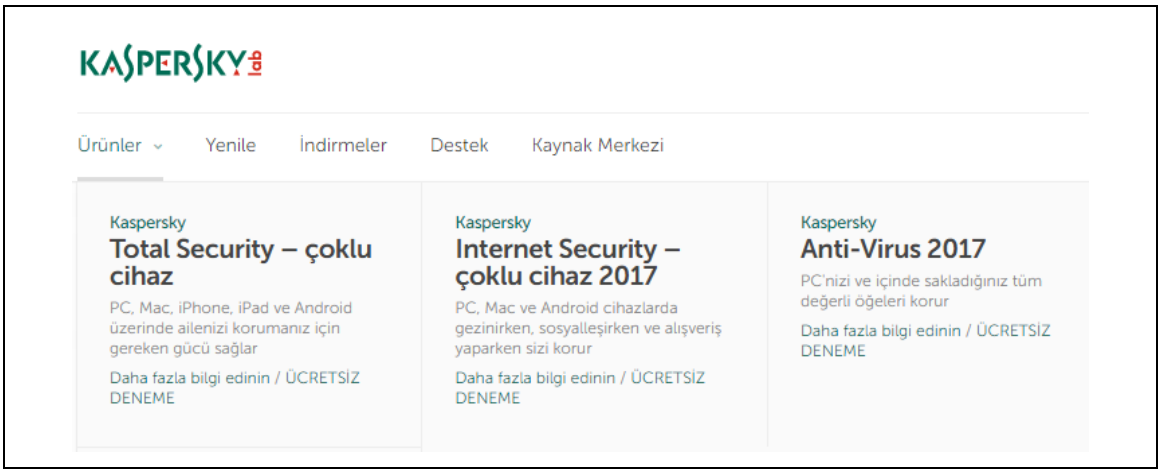
4.1.8 APT (*Advanced Persistent Threat*)

Advanced Persistent Threat (*APT*); ülkemizde "*hedef odaklı saldırı*" veya "*gelişmiş sürekli tehdit*" olarak iki farklı şekilde ifade edilen saldırı aynı zamanda bir zararlı yazılım türünü de ifade etmektedir. Zararlı yazılımın bilgi birikim ve teknoloji olarak geldiği noktayı anlayabilmek için APT kavramını biraz irdelememiz gerekmektedir. Günümüzde kamu kurumlarını, kritik altyapıları ve büyük şirketleri hedef alan çok ciddi bir tehdittir. APT'yi oluşturan terimlerin ne anlama geldiğini kısaca açıklamak gerekirse;

- **Advanced**: Saldırıyı yapan kişi sistemlere sızmak için elindeki tüm olanakları ve *gelişmiş* araçları kullanır. Bu aşamada, tek bir yöntem yerine, birden fazla sızma yöntemini birleştirerek saldırma işlemini gerçekleştirir.
- **Persistent**: Saldırgan bilindik saldırı tekniklerinden; sisteme en hızlı şekilde girmek yerine yavaş ve farkedilmeden uzun süre kalabilecek stratejiler izler.
- **Threat**: Saldırgan bilinen ve otomatize edilmiş saldırı araçları kullanmak yerine insan faktörünü de devreye sokar. Tüm süreç zararlı yazılımlar tarafından yapılmamakta yeri geldiğinde insan faktörünün de işin içinde olduğu bir saldırı türüdür.

4.2 Zararlı yazılımlarla mücadele etme yöntemleri

Zararlı yazılımlarda baştan itibaren en yaygın savunma güvenlik firmalarının ürettiği çözümler ile yapılmaktadır. Başta virüs olarak isimlendirildiği için AV olarak literatüre giren yazılımlar virüs kelimesini bir alt başlık, malware kelimesini bir üst başlık olarak kullanılmaya başlamasına rağmen yerleşik isimlendirmeden dolayı hala AV olarak kullanılmaya devam etmektedir.



Şekil 4. 2 Kaspersky firmasının güvenlik çözümleri

Yukarıdaki Kaspersky firmasının Şekil 4. 2 'de görüleceği üzere güncel güvenlik çözümlerinde AV hala isim olarak kullanılmaya devam etmekte fakat bir üst çatı olarak Total Security benzeri isimlerle daha genel bir kavram konulmaya çalışılmış ve diğer firmaların da benzer şekillerde davrandığını görebilirsiniz.

4.2.1 AV sistemleri nasıl çalışır?

AV yazılımlarının genel çalışma prensibi daha **önceden tespit edilen** zararlı yazılımların imzalarını (*MD5 vb.*) ve karakteristik bilgilerini bir veritabanında tutmak ve şüpheli dosyaları bu veritabanındaki verilerle karşılaştırmak üzerinedir. Burada AV programlarının ya da genelde tüm güvenlik firmalarının sunduğu çözümlerde en büyük problem daha önceden karşılaşılmamış bir dosyanın güvenilir mi yoksa zararlı mı olduğuna hızlı bir şekilde karar vermesi sorunsalıdır. Problemi şu şekilde örneklendirirsek daha anlaşılır olacaktır: "*Bilgisayarınıza bir USB benzeri bir depolama cihazı bağladınız ve dosyaları açmadan önce AV yazılımına taratmak istiyorsunuz. AV yazılımı dosyaları tararken daha önce görmediği bir dosyaya denk geldiğinde hızlı bir şekilde karar vermesi gerekir. Burada sizi uzun süre bekletme şansı çok azdır. O an için güvenilir deyip detaylı bir analiz sonucunda zararlı yazılım/dosya çıkması bir problem, bilmediği ve şüpheli olan her şeye hemen zararlı yazılım/dosya demesi ayrı bir probleme yol açacaktır.*" İmza tabanlı yöntem hızlı ama günümüzde çok başarılı sonuçlar vermeyen bir sistemdir. Neden artık çok başarılı olmadığını ilerleyen bölümlerde açıklık getireceğiz.

Zararlı yazılımlarla mücadelede kullanılan imza karşılaştırma yöntemi çok devrimsel bir değişiklik olmadan benzer şekillerde çalışmaya devam etmiştir. İlerleyen yıllarda imza karşılaştırma yöntemlerinin etkisizliği fark edilmiş ve alternatif oluşturabilecek yöntemler geliştirilmeye çalışılmıştır. Bu yöntemlerin en genel olanı sezgisel (*heuristic*) adı verilen yöntemdir. Tasarlandığı ve uygulandığı ilk yıllarda oldukça başarı göstermiş olan sezgisel yöntemler zaman içerisinde zararlı yazılım geliştiricilerinin geliştirdiği yeni tekniklerle etkinliği giderek azalmıştır. Şimdi AV yazılımlarının kullandığı bu iki yöntemi daha detaylı bakalım.

4.2.1.1 İmza karşılaştırma yöntemi

AV'ler imzası bilinmeyen bir zararlı yazılımı doğrudan tespit edemezler. Çünkü AV'ler o yazılımın ne yapısına ne de sistemde neler yaptığına bakmazlar. Çünkü bu işlem, teknik olarak zahmetli, çalıştığı istemi zorlayan ve zaman gerektiren bir iştir. Bu sebepten AV'ler tarafından kullanılamazlar. Bu yöntem, eğer zararlı kendi imzasını değiştirme (*polymorphic*) ya da kodlarını düzenleyebilme (*metamorphic*) yetisinden birine sahip değilse etkilidir. AV'lerin çok etkin olamamasının nedenlerinden en başta birçok zararlı yazılım, sadece imzasını değil aynı zamanda çalışan kodların kendisini de değiştiren yapıda olmasındandır.

4.2.1.2 Sezgisel (*Heuristic*) Yöntem

Sezgisel yöntem, imzası olmayan zararlı yazılımları tespit edebilmek için imza tabanlı analizin yetersiz kaldığı noktada devreye girmiştir. Mantık olarak sezgisel yöntemler kullanan AV'ler, bir programı önce kendi üzerinde inceler (*reverse engineering*), zararlı aktivite içeren kodlarlarla karşılaştığında ya da şüphelendiğinde belirli ölçütlerden geçirerek bir çıkarımda bulunmaya çalışır. İlk başlarda başarılı sonuçlar veren bu yöntem zararlı yazılım geliştiricilerinin karşı atlatma teknikleri (polymorphic ve metamorphic) ile atlatılmaya çalışılmıştır.

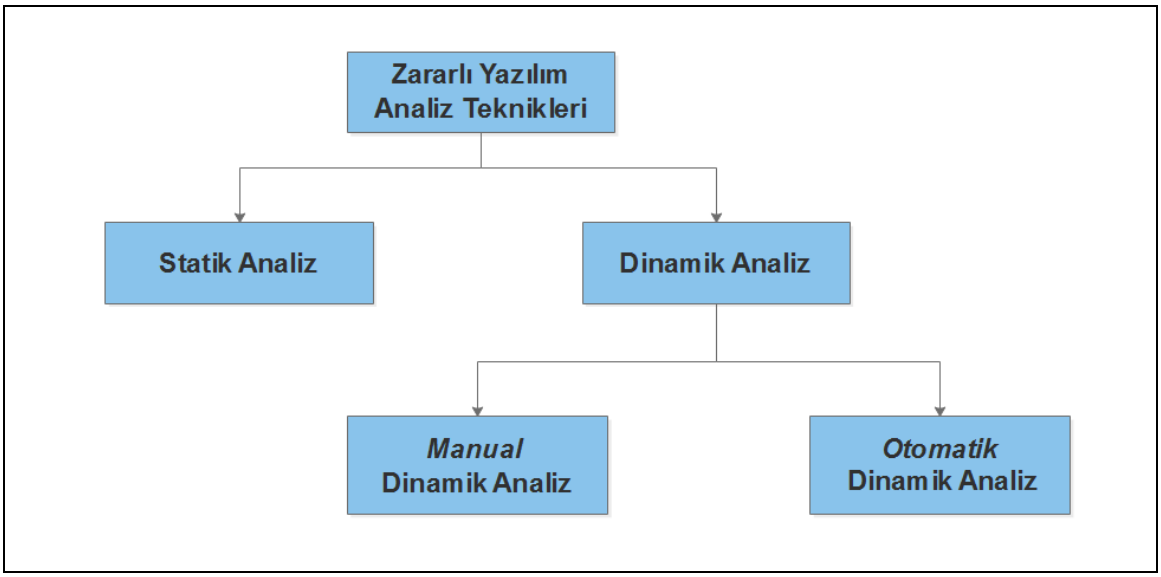
BÖLÜM 5

ZARARLI YAZILIM TESPİT ETME YÖNTEMLERİ

Günümüzdeki bir yazılımı analiz işlemi temelde iki ana kısma ayrılmaktadır [1] [12]:

- Statik Analiz
- Dinamik Analiz

Statik analiz hedef yazılım çalıştırılmadan kaynak dosya üzerinden çıkartılan bilgiler üzerinden yapılan analiz çeşididir. Dinamik analiz ise analiz edilmek istenen yazılımın çalıştırılması ile elde edilen bilgiler üzerinden yapılan analiz işlemidir.



Şekil 5. 1 Zararlı Yazılım Analiz teknikleri

5.1 Statik Analiz

Eğer analiz edilmek istenen yazılımın kaynak kodları elimizdeyse tüm statik analiz teknikleri kolayca uygulanabilir. Fakat çoğunlukla analiz edilmek istenen yazılımın

kaynak kodları elimizde olmamaktadır. Elimizde sadece çalıştırılabilir (*binary*) bir dosya vardır. Yardımcı programlar ile export/import tablo bilgisi, url, ip adresi vb bilgiler elde edilerek bir çıkarımda bulunulmaya çalışılır. Buradan elde edilen verilerle çok gerçekçi ve doğru sonuçlar üretilmesi mümkün değildir. Bu nedenle dinamik analiz yöntemi olmadan yapılan bir statik analiz **artık** çok doğru karar verilmesini sağlamayacaktır [1].

5.2 Dinamik Analiz

Dinamik analiz genellikle yalıtılmış (*sanallaştırılmış*) bir ortamda hedef yazılımı çalıştırmak ve ortama ait tüm durumları (*registry, file system, process status vb*) takip ederek analiz etme tekniğidir.

- **Fonksiyon çağrı analizi**, yazılımın çağırdığı sistem fonksiyonlarına bakarak
- **Fonksiyon parametre analizi**, yazılımın çağırdığı fonksiyonara parametre olarak neleri gönderdiğine bakarak
- **Bilgi Akışını İzleme,** yazılım içerisindeki veriyi işleyiş akışına bakarak
- **Sistemin başlangıç değerlerine ekleme,** işletim sistemi açılması sırasında otomatik olarak yüklenmesine bakarak
- **Ağ** ve **Dosya sistemi**nde yaptığı davranışlar vb. bilgiler üzerinden yapılan analiz işlemidir.

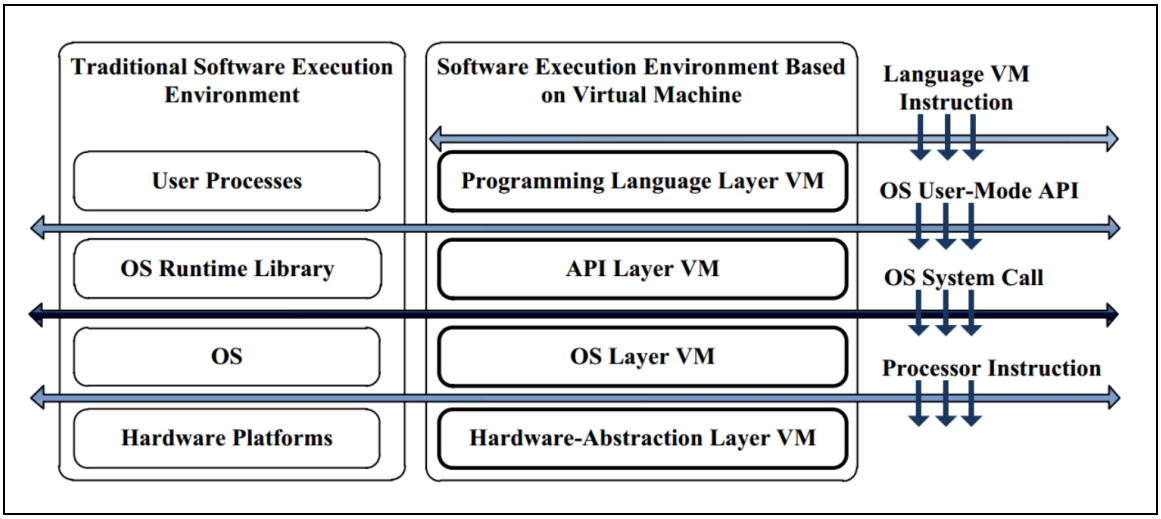
5.2.1 Otomatize Dinamik Zararlı Yazılım Analizi

Dinamik analiz ve genelde zararlı yazılım analizi başlangıçta insan odaklı ilerleyen bir süreç olsada zararlı yazılım teknikleri o kadar ilerleme göstermiştir ki zararlı yazılım üretim sayısındaki hızlı artıştan dolayı analizin elle yapılması **artık** mümkün olmaktan çıkmıştır. Bu nedenle süreç sanallaştırma teknolojisinin gelişmesiyle beraber, yapay sinir ağları konusunda bilgi birikiminin artmasıyla birlikte otomatize bi zararlı yazılım sürecine evrilmiştir.

5.2.1.1 Sanallaştırma Nedir?

Sanallaştırma (*Virtualization*), IBM'in 1960'lı yıllarda geliştirmeye başladığı bir teknolojidir. Bu teknolojinin temelinde fiziksel sistemlerin sanal kopyalarının oluşturulması temelindedir. Sanallaştırma sayesinde işlemci (*CPU*), bellek (*RAM*), sabit disk ve ağ adaptörleri gibi mevcut fiziksel kaynaklarınızı mantıksal bölümlere ayırabilir ve her mantıksal bölümün farklı bir sistem gibi hareket etmesi sağlanabilmektedir [13]. Sanallaştırma teknolojisinin yaygın olarak kullanılmakta olan çeşitli türleri vardır:

- Donanım seviyesinde sanallaştırma
- İşletim sistemi seviyesinde sanallaştırma
- API seviyesinde sanallaştırma
- Programlama dili seviyesinde sanallaştırma



Şekil 5. 2 Sanallaştırma Katmanları [13]

5.2.1.2 Zararlı yazılım analizinde sandbox ortamı nedir?

Sandbox ortamını kapalı bir sistem gibi düşünürsek bu sistemin içinde tıpkı bir otomasyona süreci gibi giriş ve çıkış süreci yer almaktadır. Dışarıdan bize görünen kısmı ise sadece zararlı yazılım örneğini bu sisteme vermek ve sonuç raporunu almaktır. Öncelikle bu sistemin içindeki yapıya bir göz atalım. Sandbox ortamları temelde üç parçadan oluşur. Bunlar;

- Analizin sürecinin yönetildiği parça

- Analiz edilmek istenen yazılımın çalıştırıldığı sanallaştırma (*virtualbox*, *vmware*, *quemu v.s.*) ortamı parçası
- Elde edilen veriler üzerinden çıkarımda bulunulduğu parça'lardır.

Analizin yapıldığı kısımda sanal ortam ile konuşmayı sağlayan uygulama ya da betikler (*scriptler*) ve raporlamayı sağlayan araçlar dışında başka bir işlem yapılmamaktadır. Çalıştırma ortamında ise hedef yazılımın davranışlarını analiz etmek için gerekli network araçları (*wireshark, windump, tcpdump, etheral v.s.),* dosya sistemi izleme uygulamaları, registry kayıtlarını izleme uygulamaları (*process bat, captureBAT, regshot*), Process izleme uygulamaları (*sysinternals, process hacker, process explorer*), Servisleri izleme uygulamaları ve bellek inceleme uygulamaları (*Volatility vb*) bulunmaktadır. Tüm bu araçlardan elde edilen verileri üzerinden yapay sinir ağı gibi yöntemlerle karar verilmeye çalışılmaktadır. Online olarak veya farklı şekillerde zararlı yazılım analiz hizmet veren sandbox sistemleri[1] bulunmaktadır. Sanallaştırma teknolojisinin zararlı yazılım analizinde en büyük avantajı sistemi istediği zaman ayağa kaldırabilmesi ve istediği zaman ilk haline dönebilme kolaylığıdır.

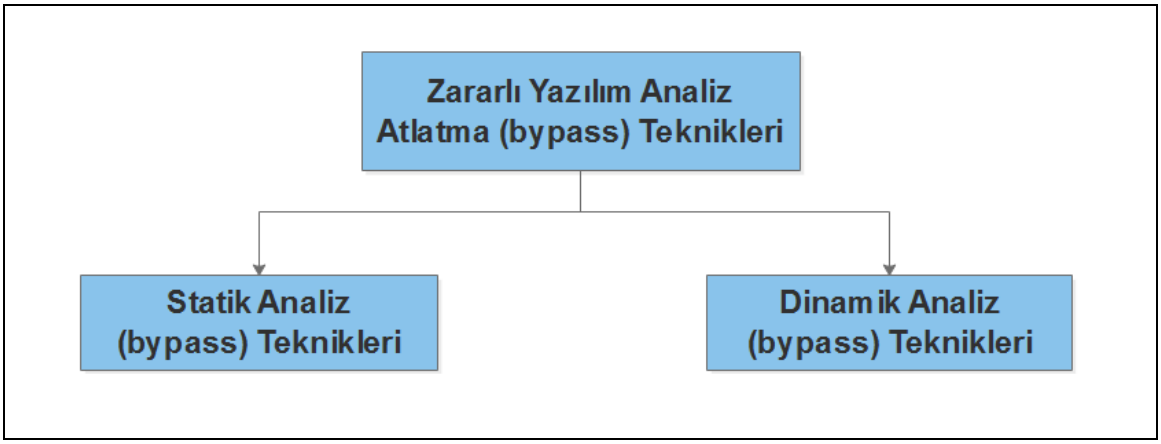
[1] Sistemlerin bir kısmı:

- Comodo Valkyrie
- Joe Sandbox
- Cuckoo Sandbox
- Minibis BitBlaze

BÖLÜM 6

ZARARLI YAZILIM ANTİ-ANALİZ YÖNTEMLERİ

Zararlı yazılımlar güvenlik firmalarının kendileriyle mücadele için buldukları çözümlere sürekli bir karşı atlatma yöntemi bulmaya çalışmaktadırlar ve bu karşı atlatma yöntemleri genelde iki ayrı kategoride değerlendirilebilir. Bu kategoriler statik analizi ve dinamik analizi atlatma (bypass) üzerinedir. Şimdi sırasıyla bu yöntemleri inceleyelim.



Şekil 6. 1 Analiz Atlatma Teknikleri

6.1 Statik Analiz Atlatma (*bypass*) yöntemleri

Statik analiz edilme sürecini engellemek için zararlı yazılım geliştiricileri yaygın olarak iki yöntem kullanmaktadırlar [14].

6.1.1 Gizleme (*Obfuscation*)

Türkçe karşılığı "*Şaşırtma*" veya "*Gizleme*" olan obfuscation, zararlı yazılımın analiz edilmesini zorlaştırmak/atlatmak için başvurulan yöntemlerden biridir [14]. Tersine mühendislik (*reverse engineering*) yöntemleri ile çalıştırılabilir (*binary*) kodların kaynak koduna çevrilmesi ve analiz edilmesini zorlaştırılması ya da engellenmesi amaçlanmaktadır.

6.1.2 Paketleme (*Packet*)

Zararlı yazılım geliştiricileri, zararlı yazılım programına tersine mühendislik ile analiz edilmesini önlemek için çalıştırılabilir kodların bir kısmını şifreli olarak saklar ve bu şifreleme mekanizmasını çözecek kodu çalıştırılabilir kodun içine ekler. Kaynak kodun bir kısmının şifrelemesiyle birlikte kötü amaçlı programın genelde kötü işleri yapacak kod kısmı sıkıştırılmış/paketlenmiş hale gelir. AV programları paketlenmiş öğelerin içeriğini programı çalıştırmadan bilemeyecekleri için statik analizi kolayca atlatmış olacaklardır. Statik analiz tek başına kullanılmayıp daha çok dinamik analiz sürecine katkı sağlayacak şekilde olduğundan genelde zararlı yazılım geliştiricileri de daha çok dinamik analiz sürecini atlatmaya çalışmaktadırlar.

6.2 Dinamik Analiz Atlatma (*bypass*) yöntemleri

Dinamik analiz süreci zararlı yazılım analiz sürecinin en etkin olduğu yöntem aynı zamanda zararlı yazılım üreticilerinin atlatmaya çalıştığı ve etkili yöntemler geliştirdiği ve farkındalığın arttığı alan olmaktadır. Özetle iki taraf arasında zorlu savaşlar burada gerçekleşmektedir. Burada zararlı yazılım geliştiricilerin kullandığı yöntemler kısaca aşağıdaki gibidir [14].

6.2.1 Anti-Debugging

Debugging işleminin tespit edilip zorlaştırılması/engellenmesi sağlanır.

- API'ler aracığılıyla (*IsDebuggerPresent() gibi*) ve Process Environment Block ile API'siz debugger tespit işlemi

- Debugger'ın yönetemiyeceği (*handle*) hatalar üretilerek debugging işlemini başarısız hale getirme
- Programın çalışma aralığına gibi yöntemlere bakılarak debugger tespit edilmeye çalışılır.

Kısaca debugging üzerinden yapılan zorlaştırma ya da engelleme üzerinden analizden kaçmaya çalışılır.

6.2.2 Anti-Disassembly

Disassembler programlarını yanıltarak, kodun yorumlanması, analiz edilmesini zorlaştırmak hedeflenmektedir.

6.2.3 Anti-Antivirus

Sistemde yüklü AV'ün varlığı veya yazılımında bulunan açıklıkları (*vulnerability*) kullanarak kendini gizleme şeklinde olan yöntemlerdir.

6.2.4 Anti-Sandbox

Otomatize analiz süreci kaçınılmaz olarak bir sandbox ortamında gerçekleşmekte ve gerçek bir ortamla sanal bir ortamın ayrımının kolay yapılabiliyor olması beraberinde ciddi bir aşılması gereken problem olarak karşımızda durmaktadır. Bir sonraki bölümde son zamanlarda daha da fazla karşımıza çıkan bu tehlikenin detaylarından ve ardından **derin öğrenme ile bu problemi çözmeye yönelik** bir çalışma yapacağız.

BÖLÜM 7

SANDBOX TESPIT ETME YÖNTEMLERİ

Günümüzde gelişmiş zararlı yazılımlar, geliştiricileri tarafından analiz edildiğini anlamak ve bu süreci atlatarak kendini uzun süre gizlemek üzerine gelişmiş yetenekleri olacak şekilde tasarlanmaya başlanmıştır. Bu süreçte kendini uzun süre gizleyebilmek için en çok başvurulan yöntem kum havuzu (*sandbox*) denilen yalıtılmış bir ortamda çalışıp çalışmadığını anlamak üzerine kurmaktadır. Bu hem çok basit parametrelere bakarak kolayca anlayabilmek üzerine kurulmuş olabileceği gibi daha gelişmiş zararlı yazılımlarda kullandıkları yöntemler daha da farklılaşabilmektedir. Tabi ki burada bilgisayar sistemleri ve mobil sistemlede kullandıkları parametreler benzer ve farklılıklar içermektedir. Bu noktada benzerliklere ve ayrımlara göz gezdirelim:

7.1 Donanımsal parametreler

Bilgisayar sistemlerinde ve mobil sistemlerde zararlı yazılımların yalıtılmış ortamda olup olmadığı anlamak için çeşitli donanım bilgilerinden yola çıkarak anlama sürecidir. Bu özelliklerin bir kısmı aşağıdaki gibidir.

- Disk/SD hafıza kartı boyutuna bakma,
- MAC adresine bakma, IMEI numarası
- Telefon model bilgisi
- Telefonun pil durumu
- Ram/CPU vb. bilgisine bakarak çıkarımlarda bulunmaya çalışır.

7.2 Yazılımsal parametreler

İşletim sisteminde kurulu programlara bakarak çıkarımda bulunulmaya çalışılır. Wireshark benzeri özel amaçlı programların kurulu olması bir şüpheli duruma karşılık geleceği gibi, Whatsapp benzeri anlık mesajlaşma uygulamalarından herhangi birinin mobil bir sistemde kurulu olmaması da ters yönden şüphe uyandırıcı bir durum oluşturur. *Vmware* benzeri sanallaştırma sisteminde sanallaştırma sistemine ait bir process'in çalıştığının görülmesi de zararlı yazılım üreticileri için kritik bir bilgi olmaktadır.

7.3 Kullanıcı yaşam parametreleri

Zararlı yazılım üreticileri için en geniş veri elde etme alanı burasıdır. Ve güvenlik firmalarının karşısında problem oluşturan alan burasıdır. Sistemde gerçek bir kullanıcı mı var yoksa simüle mi ediliyor bu ayrım şu anda zararlı yazılım geliştiricilerinin lehine durumdadır. Örneğin e-posta ya da kısa mesaj trafiğinin olması aktif bir kullanım görülmesi bir analiz ortamında çalışmadığı hakkında bilgi vermektedir. Aktif bir internet bağlantısının olamaması şüpheli bir durumun olduğunu gösterir. Mobil sistemlerde sensör bilgileri sistemlerin yeteneklerini arttırmak ve kullanıcının yaşam parametlerini alıp hayatlarını kolaylaştırmak adına çok önemli bir yer tutmaktadır. Fakat bu sensör

- jiroskop,
- dijital pusula
- İvmeölçer,
- yakınlık sensörü,
- ortam ışığı sensörü,
- sıcaklık sensörü,
- mikrofon,
- GPS,
- WiFi ve
- Bluetooth

bilgileri alınarak zararlı yazılım tarafından kullanılabilmektedir. Birçok zararlı yazılım analiz ortamı bu sensör bilgilerini simüle eden bir mekanizma sunmamaktadır. Sensörün olup olmadığı bakmanın bile tek başına yeterli olması tehlikeyi açıklamak açısından yeterli olacaktır. Kalvye girdilerinin takip edilmesi ve hızı anlamsal bir çıkarım yapılabilmesine olanak sağlamaktadır. Mobil cihaz sözkonusu olduğunda multimedya

(*fotoğraf*, *video* vb) dosyalarının olmasıveya olmaması aktif bir kullanıcı kararının verilip verilmemesine olanak sağlayacaktır.

Çizelge 7. 1 Mobil & Bilgisayar sistemleri için Statik & Dinamik Kontrol Parametreleri (*S: Statik, D: Dinamik*)

Taxonomy	Methods	Computer Systems	Mobile Systems	Type
Human Interaction Patterns	Keyboard activities	✓	✓	D
	Mouse activities	✓	×	D
	Applications using	✓	✓	D
	Login/logout times	✓	×	D
	Window/GUI Interactions	✓	✓	D
	Keystrokes/touchpad speed	✓	✓	D
	Call history	×	✓	D
	Message history	×	✓	D
	Calendar history	✓	✓	D
	Address Book	✓	✓	D
	Multimedia files	✓	✓	D
Internet Usage Pattern	Web browsing history	✓	✓	D
	File uploads / downloads	✓	✓	D
	Sent / received emails	✓	✓	D
	Chat / messenger conversations	✓	✓	D
	Search engine queries	✓	✓	D
	Installed applications	✓	✓	D

Çizelge 7. 1 Mobil & Bilgisayar sistemleri için Statik & Dinamik Kontrol Parametreleri (*S: Statik, D: Dinamik*) (Devam)

Taxonomy	Methods	Computer Systems	Mobile Systems	Type
System Sensor Data Pattern	Accelerometer	×	✓	D
	Geomagnetic Field	×	✓	D
	Orientation	×	✓	D
	Gyroscope	×	✓	D
	Light	×	✓	D
	Pressure	×	✓	D
	Temperature	×	✓	D
	Proximity	×	✓	D
	Gravity	×	✓	D
	Linear Acceleration	×	✓	D
	Rotation Vector	×	✓	D
	Relative Humidity	×	✓	D
	Ambient Temperature	×	✓	D
	Mag Field	×	✓	D
	Game Rot Vector	×	✓	D
	Gyroscope Uncal	×	✓	D
	Significant Motion	×	✓	D
	Step Detector	×	✓	D
	Step Counter	×	✓	D
	Geomag Rot Vector	×	✓	D
	Tilt Detector	×	✓	D
	Pick Up Gesture	×	✓	D
	Hall Effect	×	✓	D
	Gesture Grid Change	×	✓	D

Çizelge 7. 1 Mobil & Bilgisayar sistemleri için Statik & Dinamik Kontrol Parametreleri (*S: Statik, D: Dinamik*) (Devam)

Taxonomy	**Methods**	**Computer Systems**	**Mobile Systems**	**Type**
Hardware Information Pattern	IMEI number	×	✓	S
	System disk/memory size	✓	✓	S
	System disk/memory usage	✓	✓	D
	GSM operator	×	✓	S
	CPU	✓	✓	S
	RAM	✓	✓	S
	MAC address	✓	✓	S
	Device ID	×	✓	S
	Battery status	×	✓	D
	Camera	✓	✓	D
	Microphone	×	✓	D
	Bluetooth	×	✓	D
	3G	×	✓	D
	Wi-Fi	✓	✓	D
	Printer	✓	×	S
	Phone model	×	✓	S

Çizelge 7. 2 Zararlı yazılım analiz ortamlarında sensör simülasyon bilgileri

	Malware Analysis Services							
Sensors	**Android Sandbox**	**Andrubis**	**APKScan**	**CopperDroid**	**Mobile Sandbox**	**SandDroid**	**Trace Droid**	**Visual Threat**
Accelerometer	✓		✓	✓	✓	✓		✓
Geomagnetic Field	✓		✓		✓	✓		✓
Orientation	✓		✓		✓	✓		✓
Gyroscope								
Light								
Pressure								
Temperature	✓		✓		✓	✓		✓
Proximity	✓		✓		✓	✓		✓
Gravity								

Çizelge 7. 1'de detaylı olarak çıkarmaya çalıştığımız çizelgeye bakılarak nelerin zararlı yazılım üreticileri açısından analiz edilmekten kaçmak için başvurabilecekleri parametreler olabileceğini ortaya koymaya çalıştık. Ardınan Çizelge 7. 2'de yaygın sanbox'ların hangi sensör bilgilerinin sanal ortamda gerçeklendiğine bakarsak tehlikenin boyutları daha net görülebilir.

BÖLÜM 8

DENEYSEL ÇALIŞMA

Makine öğrenmesi gibi yöntemler üzerinden başarılı sonuçlar elde edebilmek için gerekli olan belirleyici özniteliklerin çıkartılması amacı ile zararlı yazılımların çalıştırılması ve sitemdeki davranışları gözlemlenmesi gerekmektedir. Fakat bu gerekliliğe karşı sınıflandırma sonuçlarını olumsuz etkileyen yukarıdaki bölümlerde detaylı bir şekilde bahsettiğimiz anti-analiz (*bypass*) teknikleri geliştirilmiştir. Burada iki yol ayrımı bulunmaktadır. Ya fark edilemeyecek şekilde insan ve sistem davranışlarını simüle edebilecek sandbox (*sanallaştırma*) sistemleri oluşturulacak ya da daha efektif bir çözüm olan yazılımları **çalıştırmadan** yüksek başarı oranları elde edebilecek tespit metotlar geliştirilecektir. Derin öğrenme yöntemleri kullanılarak, analiz edilmek istenen yazılımların çalıştırılmasına gerek kalmadan tespit edilmesi hedeflenmiştir. Bu noktada Mobil yazılımların kullandıkları izinler üzerinden öznitelikler çıkartılmış ve bu özniteliklerin ağırlıkları otomatik kodlayıcı ile optimize edilmiş ve sonrasında en optimum sonuç üreten beş katmanlı yapay sinir ağı ile sınıflandırma yapılmıştır. Sınıflandırma sonucunda zararlı yazılımlar %93,67 doğruluk oranı ile tespit edilmiştir.

8.1 Benzer Çalışmalar

Makine öğrenmesi yöntemleri, zararlı yazılımların tespiti konusunda daha başarılı bir sistem geliştirmek adına kullanılmaya başlanmıştır [15] , [16], [17]. Zararlı yazılım tespiti konusunda yapılan bir çalışmada makine öğrenmesi yöntemlerinden destek vektör makineleri (*SVM*) ve J48 Karar Ağacı algoritmaları ve torbalama (*bagging*) yöntemi ile izinler ve API (*Application Programming Interface*) çağrılarından oluşan bir veri seti üzerinde kullanıldığında %92,36 ile %96,88 arasında değişen doğruluk oranları

elde edilmiştir [16]. Bir diğer çalışmada, sadece uygulama izinleri kullanılarak oluşturulmuş bir veri seti üzerinde Bayesian, CART (*Classification and Regression Tree*), J48 Karar Ağacı, Rastgele Orman (*Random Forest*) ve SMO (*Sequential Minimal Optimization*) yöntemleri kullanılarak zararlı yazılım tespiti yapıldığında %72,78 ile %94,90 arasında değişen tespit oranları elde edilmiştir [18]. Bu çalışmaların yanısıra, sadece Android çekirdek sistem çağrıları kullanılarak oluşturulmuş bir veri seti üzerinde otomatik kodlayıcı tabanlı derin YSA yöntemi, SVM, YSA, Naive Bayes (*NB*) ve Karar Ağaçları zararlı yazılım tespiti amacıyla uygulandığında sırası ile %93,68, %88,24, %87,88, %77,94 ve %87,42 tespit oranları elde edilmiştir. Bu çalışmada derin YSA yönteminin, zararlı yazılımların tespiti konusunda diğer makine öğrenmesi yöntemlerine kıyasla daha başarılı performans gösterdiği saptanmıştır [19]. Yapılan bir çalışmada ise Multi-Layer Perceptron (*MLP*), J48 Karar Ağacı, k-NN (*k-Nearest Neighbour*), Rastgele Orman, NB algoritmalarını kullanarak sistem çağrılarının üzerinden oluşturulan bir veri seti üzerinden farklı öznitelik setleri ile en yüksek başarı oranı %83 olarak elde edilmiştir [20].

Derin öğrenme yöntemini kullanan bir çalışmada ise uygulamaların kurulum anında istediği ve çalışma anında kullandığı izinlerden iki ayrı veri seti oluşturulmuş ve oluşturulan bu iki veri seti üzerinde otomatik kodlayıcı kullanılarak yapılan zararlı yazılım tespiti sonucunda sırasıyla %87,1 ve %80,9 başarı oranları elde edilmiştir [12]. Bu çalışmaya göre kurulum anında istediği izinler üzerinden yapılan analiz daha doğru sonuçlar elde edilmesini sağlamaktadır. [21] çalışmasında ise API çağrıları ve uygulamanın istediği izinlerden tehlikeli olarak etiketlenenler dikkate alınmıştır. Sadece tehlikeli olarak etiketlenen API çağrılarına göre SVM tabanlı algoritmayla %81 ve hem tehlikeli API çağrıları hem de riskli olarak etiketlenen izinler üzerinden SVM algoritmasıyla %86'lık bir başarı elde edilmiştir.

8.2 Veri Setinin Hazırlanması

Mobil cihazlar için geliştirilmiş olan uygulamalar, işletim sistemi versiyonuna ve gerçeklenme şekline bağlı olarak ya ilk yüklenme sırasında ya da kameraya, rehber bilgisine erişilmesi vb. şekilde çalışma esnasında ihtiyaç duydukları izinleri kullanıcıya onaylatmaktadırlar. Yapılan çalışmalar, kullanıcıların uygulama izinlerine çok dikkat

etmediği veya uygulama açısından bu izinlere ihtiyaç olup olmadığı hakkında yeterli bilgiye sahip olmadığını ve istenen izinlerin genel olarak doğrudan onaylandığını ortaya koymuştur [22].

Çizelge 8. 1 – Veri Sınıfı

Veri Sınıfı	**Alt Veri Seti**		
	Eğitim	**Test**	**Doğrulama**
Zararsız Uygulamalar (*Benign*)	2261	484	484
Zararlı Uygulamalar (*Malware*)	1166	251	251

Bu çalışma kapsamında derin öğrenme teknikleri ile zararlı yazılım tespiti için önerilen sistem, Android işletim sistemi için geliştirilen mobil uygulamaların ihtiyaç duydukları izinleri öznitelik olarak kullanmaktadır. Bahsi geçen bu izinler uygulama içinde yer alan AndroidManifest.xml isimli manifesto dosyasında yer almaktadır. Herhangi bir uygulama tarafından Android işletim sisteminden talep edilen izinler, bu uygulamanın sistemde yapabileceği tüm aktiviteleri belirleyen ve uygulama için çalıştırılmadan elde edebilecek en nitelikli özniteliklerdir [16], [18], [23].

Android uygulamaları için toplam 138 adet izin tipi[1] tanımlanmıştır [24]. İhtiyaç duyulan bu izinler, manifest dosyasında xml formatında ifade edilmektedir. Geliştirilen bir yazılım ile zararlı ve zararsız olmak üzere iki gruba ayrılmış uygulamalardan oluşan veri seti içerisindeki tüm uygulamalar için apk dosyalarından belirtilen 138 adet izin tipine ait değerler elde edilmiştir. Tüm uygulamalar için “0” veya “1” olan izin tipleri gereksiz korelasyonların önüne geçmek ve veri setinin kirletilmesini engellemek için çıkartılmıştır. Bunu sonucunda 138 adet izin 128 adet izin tipine indirgenmiştir. Elde edilen bu 128 adet izin tipi öznitelik olarak kullanılmıştır.

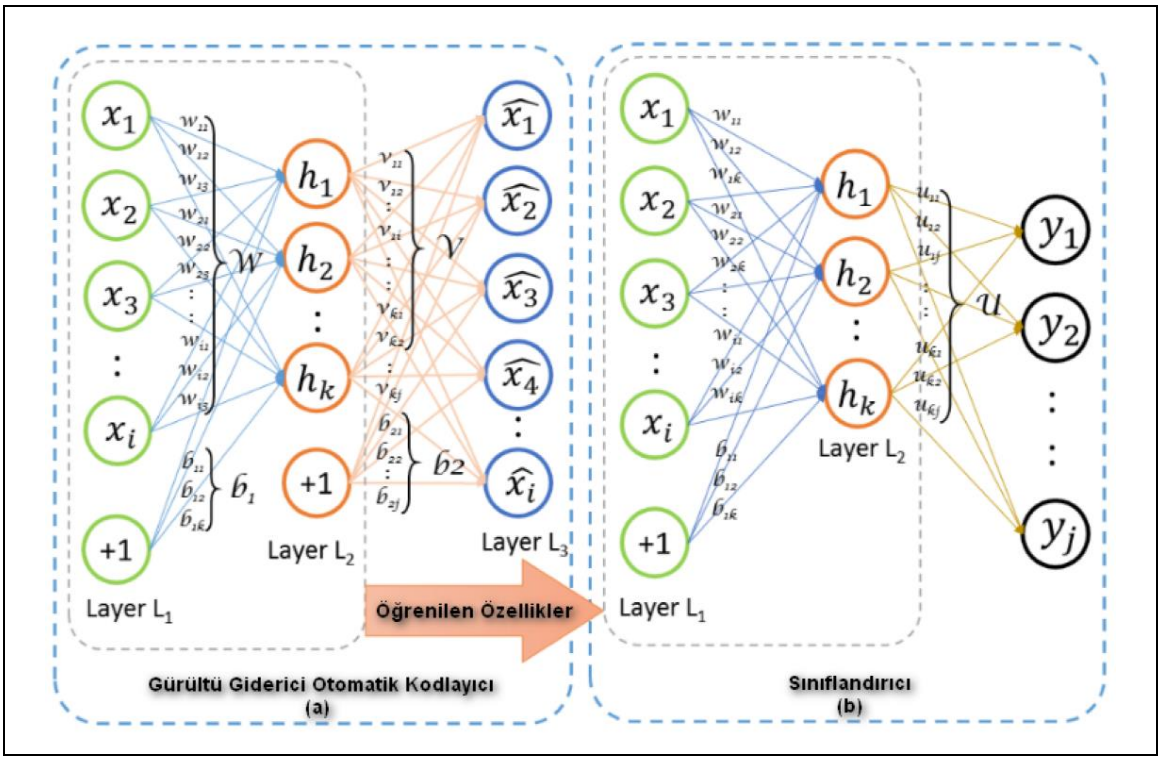
Çalışma kapsamında 3229 zararsız, 1668 zararlı mobil uygulama kullanılmıştır. Zararsız sınıfındaki uygulamalar, Android uygulama marketinde en yüksek indirme oranına

[1] Kullanılan izinler Ek – B’de görülebilir.

sahip, herhangi bir şekilde zararlı uygulama olarak belirtilmemiş ve uzun zamandır markette bulunan uygulamalardan seçilmiştir. Zararlı sınıfındaki uygulamalar ise Comodo güvenlik firması tarafından zararlı uygulama olarak tespit edilmiş uygulamalardan oluşturulmuştur [25]. Sistemin eğitim, test ve doğrulama aşamalarında kullanılmak üzere veri seti, %70'i eğitim, %15'i test, %15'i de doğrulama olacak şekilde Çizelge 8. 1'de belirtildiği gibi alt veri setine ayrılmıştır. Veri setindeki tüm öznitelikler ikili ("0" veya "1") olduklarından herhangi bir normalizasyon yapılmasına gerek kalmamıştır.

8.3 Derin öğrenme Tabanlı Zararlı Yazılım Tespit Modeli

Önerilen yöntemde gürültü giderici otomatik kodlayıcı (*denoising autoencoder*) (*GGOK*) ve çok katmanlı YSA (*multilayer perceptron*) Şekil 8. 1'de gösterildiği gibi bir arada zararlı yazılım tespiti için kullanılmıştır. GGOK'lar arka arkaya eklenerek çok katmanlı bir YSA oluşturmak amacıyla kullanılabilmektedir. GGOK tabanlı derin YSA ön eğitim (*pre-training*) ve ince ayar (*fine tuning*) aşamalarından oluşur. İlk olarak etiketsiz veri kullanılarak gözetimsiz bir şekilde önemli özniteliklerin keşfedilmesi amaçlanır, sonrasında da keşfedilen bu önemli özniteliklerin YSA sınıflandırcısının gözetimli eğitimini iyileştirmek için kullanılır.



Şekil 8. 1 Derin yapay sinir ağının eğitim süreci a) Gözetimsiz öznitelik öğrenme [26]

Şekil 8. 1 (a) görüleceği gibi ön eğitim aşamasında GGOK etiketsiz olarak eğitim verisiyle eğitilir. Bu aşamada derin sinir ağlarının her gizli katmanı ayrı birer otomatik kodlayıcı (OK) olarak kabul edilmiştir. Buna ek olarak bu OK'lar giriş katmanından en derindeki katmana doğru sırasıyla ve ayrı ayrı eğitilir. Bu şekilde yapılan eğitim greedy-wise eğitim olarak adlandırılmaktadır. Burada amaçlanan YSA'nın gözetimli olarak eğitilmesinden önce ağırlıkların optimum seviyeye çekilmesini sağlamaktır. Rastgele ağırlıklarla ilklendirilmiş YSA'nın hata fonksiyonu, meyilli azalım (*gradient descent*) ile eğitilirken yerel minimum değerlerine takılabilmekte ve sistemin başarısı azalabilmektedir. GGOK sayesinde YSA'nın ağırlıkları güncellenerek gözetimli eğitime uygun hale getirilir. Ön eğitim aşamasından sonra ince ayar aşamasında Şekil 8. 1 (b)'de görüleceği gibi ağırlıkları düzeltilmiş YSA etiketli veri ile gözetimli olarak sınıflandırma hatasını düzeltmek üzere eğitilir ve tüm modelin ağırlıkları ideal değerlere ulaşıncaya kadar eğitime devam edilir [27], [28]. Bu çalışmada GGOK'lar içerisindeki nöron aktivasyonu için sigmoid fonksiyonu seçilmiştir ve sınıflandırıcının çıkış katmanında softmax aktivasyon fonksiyonu kullanılmıştır. Ayrıca, modelin aşırı öğrenmesini engellemek amacıyla erken durma (*early-stopping*) ve L1, L2 regularization (*weight decay*) yöntemleri kullanılmıştır.

8.4 Deneysel Sonuçlar

Bu çalışmada zararlı yazılım tespiti amacıyla kullanılan derin öğrenme modeli; giriş katmanı, üç gizli katman ve bir çıkış katmanın sahip olacak şekilde oluşturulmuştur. Veri setinde 128 adet öznitelik bulunduğu için giriş katmanında 128 nöron kullanılmıştır. Sonrasında, gizli katmanlardaki nöron sayısı, GGOK'nın gözetimsiz eğitimde en ideal başarımı elde etmesi amacıyla giriş katmanından fazla olacak şekilde yapılan testler sonucunda en iyi tespit oranlarını veren 153 değeri belirlenmiştir. Son olarak, iki sınıflı (*zararlı ve zararsız yazılım*) bir sınıflandırma işlemi yapıldığından çıkış katmanında iki adet nöron kullanılmıştır.

Çizelge 8. 2 Terminoloji ve Açıklamalar Tablosu

Terminoloji	Kısaltmalar	Tanımlar
True Positive	TP	Zararsız yazılımlardan zararsız olarak seçilenler
True Negative	TN	Zararlı yazılımlardan zararlı olarak seçilenler
False Negative	FN	Zararlı yazılımlardan zararsız olarak seçilenler
False Positive	FP	Zararsız yazılımlardan zararlı olarak seçilenler
Precision	Precision	TP/ (TP + FP)
Recall	Recall	TP/ (TP + FN)

Elde edilen sonuçlar Çizelge 8. 3'de verilmiştir ve sonuçları yorumlarken kullanılan tanımlamalar ve kısaltmalar Çizelge 8. 2'de açıklanmıştır. Çizelge 8. 3'de görüldüğü gibi model, zararsız ve zararlı veriler için sırasıyla %91,9 ve %93,7 hassasiyet (*precision*) değeri ve %97,1 ve %83,6 geri çağırma (*recall*) değeri üretmiştir. Bu sonuçlara bakıldığında model, zararsız verilerin sınıflandırılması konusunda zararlı verilere kıyasla daha yüksek bir başarım göstermektedir. Ayrıca zararlı ve zararsız verilerin tamamı için ağırlıklı ortalama yöntemiyle bulunmuş olan %92,3 F1 değeri sistemin zararlı yazılım konusundaki genel başarımını tariflemektedir. Ek olarak uyguladığımız sistemin, test verisinin tamamı üzerindeki doğruluk oranı %93,67 olarak tespit edilmiştir.

Çizelge 8. 3 Deneysel Sonuç Tablosu

Veri Sınıfı	**Sonuçlar**		
	Precision	**Recall**	**F1 Score**
Zararsız Uygulamalar (*Benign*)	0,919	0,971	0,944
Zararlı Uygulamalar (*Malware*)	0,937	0,836	0,883
Ağırlıklı Ortalamalar	0,925	0,925	0,923

Çizelge 8. 4'de ise veri setimiz diğer algoritmalarla başarı oranları kıyaslanmıştır. Uygulanan algoritmalar Weka ile denenmiş ve verilen değerler ağırlıklı ortalama olarak hesaplanmıştır. Derin öğrenme için hazırlanan veri seti Weka üzerinde uygun hale getirilmiş ve algoritmalar çalıştırılmıştır. Alınan sonuçlar üzerinde derin öğrenmeye en yakın sonuç %93.19 başarı oranı ile MLP algoritmasıyla elde edilmiştir. MLP'nin başarı oranın derin yapay sinir ağına yakın olmasının sebebi veri kümesindeki örnek sayısının kısıtlı olmasıdır. Bu yüzden MLP yöntemi local minimum problemine takılmadan derin öğrenmeye yöntemine yakın sonuçlar vermiştir.

Çizelge 8. 4 Weka Deneysel Sonuç Tablosu

Algoritma	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Naive Bayes	0.858	0.859	0,858	0,859
Logistic Regression	0,909	0.91	0.909	0.91
MLP	0.932	0.932	0.931	**0.9319**
SVM	0.909	0.91	0.909	0.91
Bagging	0.919	0.919	0.919	0.918
J48	0.919	0.92	0.92	0.919

BÖLÜM 9

SONUÇ VE ÖNERİLER

Zararlı yazılımlar bilişim sistemleri var olduğu sürece mücadele etmemiz gereken bir olgu olarak karşımızda durmaktadır. Biz de ülkemizin siber güvenlik hedeflerine katkı sağlayacak şekilde son yılların gelişmiş zararlı yazılımlarının kullandıkları atlatma yöntemlerini ve kullanabilecekleri yöntemleri belirlemeye çalıştık. Yaptığımız çalışmayla buna çözüm getirmeye çalıştık. Çalışmada 3229 zararsız, 1668 zararlı Android uygulamasına ait işletim sisteminden talep ettikleri izinler GGOK ile ön işlemden geçirilmiş ve çok katmanlı YSA ile sınıflandırılmıştır. Sınıflandırma sonucunda önerilen sistem önceden tanımadığı zararlı yazılımları %93,67 doğruluk ile tespit etmiştir. Elde edilen başarı dinamik analiz ile edilen öznitelikler kullanılarak makine öğrenmesi yöntemleri ile edilen başarı oranları ile eşdeğerdir ve özellikle anti-analiz tekniklerinden etkilenmemesi açısından üstünlük göstermiştir.

Son olarak güvenlik algısı ülkemizde halen çok sınırlı bir düzlemde (*yaygın olarak pentest odaklı ya da eğitim/danışmanlık hizmeti vermek*) ilerlemektedir. Bu durum da siber güvenlik konusunda nitelikli alanlarda yetişmiş insan kaynağı eksikliğine yol açmaktadır. Bu noktada zararlı yazılım analisti (*malware reseacher*), tersine mühendislik (*reverse engineering*) uzmanı, açıklık analisti (*vulnerability researcher*), zafiyet geliştirici (*exploit developer*) gibi alanlarda çalışmak isteyen kişilerin kendi fedekarlıklarının ötesinde devletin doğrudan desteklemesi, cazibe konuları haline getirmesi gerekmektedir. Dolayısıyla zararlı yazılım ve ilişkili alanlara insanları yönlendirmek için ciddi çalışılması gerekmektedir. Yoksa hiçbir aklı başında mühendisin/çalışanın bir programlama dilini ve çalıştığı/çalışacağı firmanın

ekosistemini bilmenin yeterli olduğu ortamı bırakıp assembly dilini bilmek, işletim sistemini bilmek, tersine mühendislik bilmek, sistemler arası haberleşme mimarisini bilmek, yeterince donanım bilmek vb. gibi uzun bir listeye sahip bir alana yönlenmesini beklemek gerçekçi olmayacaktır. Hele ki yeterli iş olanaklarının neredeyse olmadığı ülkemizde bu daha da zor hale gelmektedir. Kişisel çıkarlar, kendi refah düzeyimiz için başka devletlere veya o ülkenin vatandaşlarına/kaynaklarına saldırmak gibi anlayışa sahip olmasamda ülkem adına bir gün Stuxnet'i yazabilecek bilgi birikimine sahip insanların ve ekosistemin oluşması dileğiyle.

KAYNAKLAR

[1] A. Moser, C. Kruegel ve E. Kirda, (2007). “Limits of Static Analysis for Malware Detection”, Twenty-Third Annual Computer Security Applications Conference, 421-430.

[2] E. H. Spafford, (1989). “The Internet Worm Program: An Analysis”, SIGCOMM Comput. Commun. Rev., 19(1):17-57.

[3] R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen ve A. Czech, (2015). “Analysis of Malware behavior: Type classification using machine learning”, International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, 1-7.

[4] J. Sahs ve L. Khan, (2012). “A Machine Learning Approach to Android Malware Detection”, European Intelligence and Security Informatics Conference, Odense, 141-147.

[5] C. Willems, T. Holz ve F. Freiling, (2007). “Toward Automated Dynamic Malware Analysis Using CWSandbox”, IEEE Security & Privacy, 5(2):32-39.

[6] Xu Chen, J. Andersen, Z. M. Mao, M. Bailey ve J. Nazario, (2008). “Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware”, IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN), Anchorage, AK, 177-186.

[7] S. Hou, A. Saas, L. Chen ve Y. Ye, (2016). “Deep4MalDroid: A Deep Learning Framework for Android Malware Detection Based on Linux Kernel System Call Graphs”, IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), Omaha, NE, USA,104-111.

[8] STM, 2016 Türkiye siber tehdit durum raporları, https://www.stm.com.tr/tr/yayinlar/makalelerraporlar, 07 Mayıs 2017.

[9] Tübitak, Bilgi Güvenliği, http://www.bilgiguvenligi.gov.tr, 17 Mayıs 2017.

[10] Symantec, 2016 Internet Security Threat Report, https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf, 17 Mayıs 2017.

[11] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez ve A. Ribagorda, (2014). “Evolution, Detection and Analysis of Malware for Smart Devices”, IEEE Communications Surveys & Tutorials, 16(2):961-987.

[12] Xu, L., Zhang, D., Jayasena, N.,Cavazos, J., (2016). “Hadm: Hybrid analysis for detection of malware”, SAI Intelligent Systems Conference, London, UK.

[13] Chiueh, Susanta Nanda Tzi-cker, ve Stony Brook, (2005). A survey on virtualization technologies, RPE Report : 1-42.

[14] Y. Gao, Z. Lu ve Y. Luo, (2014). “Survey on malware anti-analysis”, Fifth International Conference on Intelligent Control and Information Processing, Dalian, 270-275.

[15] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee ve K. P. Wu, (2012). “DroidMat: Android Malware Detection through Manifest and API Calls Tracing”, Seventh Asia Joint Conference on Information Security, Tokyo, 62-69.

[16] N. Peiravian ve X. Zhu, (2013). “Machine Learning for Android Malware Detection Using Permission and API Calls”, IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, 300-305.

[17] J. Sahs ve L. Khan, (2012). “A Machine Learning Approach to Android Malware Detection”, 2012 European Intelligence and Security Informatics Conference, Odense, 141-147.

[18] U. Pehlivan, N. Baltaci, C. Acartürk ve N. Baykal, (2014). “The analysis of feature selection methods and classification algorithms in permission based Android malware detection”, 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Orlando, FL, 1-8.

[19] S. Hou, A. Saas, L. Chen ve Y. Ye, (2016). “Deep4MalDroid: A Deep Learning Framework for Android Malware Detection Based on Linux Kernel System Call Graphs”, 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), Omaha, NE, USA, 104-111.

[20] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat ve R. Yusof, (2014). “Analysis of Features Selection and Machine Learning Classifier in Android Malware Detection”, 2014 International Conference on Information Science & Applications (ICISA), Seoul, 1-5.

[21] W. Li, J. Ge ve G. Dai, (2015). “Detecting Malware for Android Platform: An SVM-Based Approach”, 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, New York, NY, 464-469.

[22] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev and C. Glezer, (2010). “Google Android: A Comprehensive Security Assessment”, IEEE Security & Privacy, 8(2):35-44.

[23] W. Shin, S. Kiyomoto, K. Fukushima ve T. Tanaka, (2009). “Towards Formal Analysis of the Permission-Based Security Model for Android”, Fifth International Conference on Wireless and Mobile Communications, Cannes, La Bocca, 87-92.

[24] Google Android Developers, Manifest.permission, https://developer.android.com/reference/android/Manifest.permission.html, 07 Şubat 2017.

[25] Comodo, Security Solutions from Comodo, http://www.comodo.com, 07 Şubat 2017.

[26] Javaid, A., Niyaz, Q., Sun, W., Alam, M., (2015). "A Deep Learning Approach for Network Intrusion Detection System", 9th EAI International Conference on Bio-inspired Information ve Communications Technologies, New York, USA, 21–26.

[27] Y. Bengio, P. Lamblin, D. Popovici ve H. Larochelle, (2007). "Greedy Layer-Wise Training of Deep Networks", Advances in Neural Information Processing Systems 19 (NIPS'06), 153-160.

[28] P.Vincent, H. Larochelle Y. Bengio ve P.A. Manzagol, (2008). "Extracting and Composing Robust Features with Denoising Autoencoders", Twenty-fifth International Conference on Machine Learning, 1096 - 1103.

[29] F-Secure, Mobile Threat Report 2013, https://business.f-secure.com/category/resources, 1 Mayıs 2017.

[30] TrendLabs, TrendLabs 1Ç 2015 Güvenlik Özeti, http://www.trendmicro.com.tr/media/misc/trendlabs-security-roundup-q1-2015-report-tr.pdf, 1 Mayıs 2017.

[31] G Data, Mobile Malware Threat Report (Q1/2015), https://public.gdatasoftware.com/Presse/Publikationen/Malware_Reports/G_DATA_MobileMWR_Q1_2015_US.pdf, 1 Mayıs 2017.

[32] Deloitte, Siber Güvenlik Sunumu, https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/risk/siberguvenlik-sunum.pdf, 1 Mayıs 2017.

[33] Labris Networks, 2014 Güvenlik Raporu, http://labrisnetworks.com, 1 Mayıs 2017.

[34] Microsoft Security Intelligence Report, SIR v20, v21, https://www.microsoft.com/security/sir/default.aspx, 1 Mayıs 2017.

[35] Ulaştırma Denizcilik ve Haberleşme Bakanlığı, Ulusal Siber Güvenlik Strateji Belgesi 2016-2019, http://www.udhb.gov.tr/doc/siberg/2016-2019guvenlik.pdf, 1 Mayıs 2017.

[36] Kaspersky, Security Bulletin 2015, https://securelist.com/files/2015/12/Kaspersky-Security-Bulletin-2015_FINAL_EN.pdf, 1 Mayıs 2017.

EK-A

BİLİŞİM ALANINDA SUÇLAR

ONUNCU BÖLÜM - *Bilişim Alanında Suçlar*

Bilişim sistemine girme

Madde 243- (1) Bir bilişim sisteminin bütününe veya bir kısmına, hukuka aykırı olarak giren veya orada kalmaya devam eden kimseye bir yıla kadar hapis veya adlî para cezası verilir. (1)

(2) Yukarıdaki fıkrada tanımlanan fiillerin bedeli karşılığı yararlanılabilen sistemler hakkında işlenmesi halinde, verilecek ceza yarı oranına kadar indirilir.

(3) Bu fiil nedeniyle sistemin içerdiği veriler yok olur veya değişirse, altı aydan iki yıla kadar hapis cezasına hükmolunur.

(4) **(Ek: 24/3/2016-6698/30 md.)** Bir bilişim sisteminin kendi içinde veya bilişim sistemleri arasında gerçekleşen veri nakillerini, sisteme girmeksizin teknik araçlarla hukuka aykırı olarak izleyen kişi, bir yıldan üç yıla kadar hapis cezası ile cezalandırılır.

Sistemi engelleme, bozma, verileri yok etme veya değiştirme

Madde 244- (1) Bir bilişim sisteminin işleyişini engelleyen veya bozan kişi, bir yıldan beş yıla kadar hapis cezası ile cezalandırılır.

(2) Bir bilişim sistemindeki verileri bozan, yok eden, değiştiren veya erişilmez kılan, sisteme veri yerleştiren, var olan verileri başka bir yere gönderen kişi, altı aydan üç yıla kadar hapis cezası ile cezalandırılır.

(3) Bu fiillerin bir banka veya kredi kurumuna ya da bir kamu kurum veya kuruluşuna ait bilişim sistemi üzerinde işlenmesi halinde, verilecek ceza yarı oranında artırılır.

(4) Yukarıdaki fıkralarda tanımlanan fiillerin işlenmesi suretiyle kişinin kendisinin veya başkasının yararına haksız bir çıkar sağlamasının başka bir suç oluşturmaması halinde, iki yıldan altı yıla kadar hapis ve beşbin güne kadar adlî para cezasına hükmolunur.

Banka veya kredi kartlarının kötüye kullanılması

Madde 245 – (Değişik: 29/6/2005 – 5377/27 md.)

(1) Başkasına ait bir banka veya kredi kartını, her ne suretle olursa olsun ele geçiren veya elinde bulunduran kimse, kart sahibinin veya kartın kendisine verilmesi gereken kişinin rızası olmaksızın bunu kullanarak veya kullandırtarak kendisine veya başkasına yarar sağlarsa, üç yıldan altı yıla kadar hapis ve beşbin güne kadar adlî para cezası ile cezalandırılır.

---

(1) 24/3/2016 tarihli ve 6698 sayılı Kanunun 30 uncu maddesiyle, bu fıkrada yer alan "ve" ibaresi "veya" şeklinde değiştirilmiştir.

(2) Başkalarına ait banka hesaplarıyla ilişkilendirilerek sahte banka veya kredi kartı üreten, satan, devreden, satın alan veya kabul eden kişi üç yıldan yedi yıla kadar hapis ve onbin güne kadar adlî para cezası ile cezalandırılır.

(3) Sahte oluşturulan veya üzerinde sahtecilik yapılan bir banka veya kredi kartını kullanmak suretiyle kendisine veya başkasına yarar sağlayan kişi, fiil daha ağır cezayı gerektiren başka bir suç oluşturmadığı takdirde, dört yıldan sekiz yıla kadar hapis ve beşbin güne kadar adlî para cezası ile cezalandırılır.

(4) Birinci fıkrada yer alan suçun;

a) Haklarında ayrılık kararı verilmemiş eşlerden birinin,

b) Üstsoy veya altsoyunun veya bu derecede kayın hısımlarından birinin veya evlat edinen veya evlâtlığın,

c) Aynı konutta beraber yaşayan kardeşlerden birinin,

Zararına olarak işlenmesi hâlinde, ilgili akraba hakkında cezaya hükmolunmaz.

(5) **(Ek: 6/12/2006 – 5560/11 md.)** Birinci fıkra kapsamına giren fiillerle ilgili olarak bu Kanunun malvarlığına karşı suçlara ilişkin etkin pişmanlık hükümleri uygulanır.

Yasak cihaz veya programlar

Madde 245/A- (Ek: 24/3/2016-6698/30 md.)

(1) Bir cihazın, bilgisayar programının, şifrenin veya sair güvenlik kodunun; münhasıran bu Bölümde yer alan suçlar ile bilişim sistemlerinin araç olarak kullanılması suretiyle işlenebilen diğer suçların işlenmesi için yapılması veya oluşturulması durumunda, bunları imal eden, ithal eden, sevk eden, nakleden, depolayan, kabul eden, satan, satışa arz eden, satın alan, başkalarına veren veya bulunduran kişi, bir yıldan üç yıla kadar hapis ve beşbin güne kadar adli para cezası ile cezalandırılır.

Tüzel kişiler hakkında güvenlik tedbiri uygulanması

Madde 246- (1) Bu bölümde yer alan suçların işlenmesi suretiyle yararına haksız menfaat sağlanan tüzel kişiler hakkında bunlara özgü güvenlik tedbirlerine hükmolunur.

EK-B

KULLANILAN İZİNLER

"android.permission.ACCESS_CHECKIN_PROPERTIES"

"android.permission.ACCESS_COARSE_LOCATION"

"android.permission.ACCESS_FINE_LOCATION"

"android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"

"android.permission.ACCESS_MOCK_LOCATION"

"android.permission.ACCESS_NETWORK_STATE"

"android.permission.ACCESS_SURFACE_FLINGER"

"android.permission.ACCESS_WIFI_STATE"

"android.permission.ACCOUNT_MANAGER"

"android.permission.AUTHENTICATE_ACCOUNTS"

"android.permission.BATTERY_STATS"

"android.permission.BIND_APPWIDGET"

"android.permission.BIND_DEVICE_ADMIN"

"android.permission.BIND_INPUT_METHOD"

"android.permission.BIND_REMOTEVIEWS"

"android.permission.BIND_WALLPAPER"

"android.permission.BLUETOOTH"

"android.permission.BLUETOOTH_ADMIN"

"android.permission.BRICK"

"android.permission.BROADCAST_PACKAGE_REMOVED"

"android.permission.BROADCAST_SMS"

"android.permission.BROADCAST_STICKY"

"android.permission.BROADCAST_WAP_PUSH"

"android.permission.CALL_PHONE"

"android.permission.CALL_PRIVILEGED"

"android.permission.CAMERA"

"android.permission.CHANGE_COMPONENT_ENABLED_STATE"

"android.permission.CHANGE_CONFIGURATION"

"android.permission.CHANGE_NETWORK_STATE"

"android.permission.CHANGE_WIFI_MULTICAST_STATE"

"android.permission.CHANGE_WIFI_STATE"

"android.permission.CLEAR_APP_CACHE"

"android.permission.CLEAR_APP_USER_DATA"

"android.permission.CONTROL_LOCATION_UPDATES"

"android.permission.DELETE_CACHE_FILES"

"android.permission.DELETE_PACKAGES"

"android.permission.DEVICE_POWER"

"android.permission.DIAGNOSTIC"

"android.permission.DISABLE_KEYGUARD"

"android.permission.DUMP"

"android.permission.EXPAND_STATUS_BAR"

"android.permission.FACTORY_TEST"

"android.permission.FLASHLIGHT"

"android.permission.FORCE_BACK"

"android.permission.GET_ACCOUNTS"

"android.permission.GET_PACKAGE_SIZE"

"android.permission.GET_TASKS"

"android.permission.GLOBAL_SEARCH"

"android.permission.HARDWARE_TEST"

"android.permission.INJECT_EVENTS"

"android.permission.INSTALL_LOCATION_PROVIDER"

"android.permission.INSTALL_PACKAGES"

"android.permission.INTERNAL_SYSTEM_WINDOW"

"android.permission.INTERNET"

"android.permission.KILL_BACKGROUND_PROCESSES"

"android.permission.MANAGE_ACCOUNTS"

"android.permission.MANAGE_APP_TOKENS"

"android.permission.MASTER_CLEAR"

"android.permission.MODIFY_AUDIO_SETTINGS"

"android.permission.MODIFY_PHONE_STATE"

"android.permission.MOUNT_FORMAT_FILESYSTEMS"

"android.permission.MOUNT_UNMOUNT_FILESYSTEMS"

"android.permission.NFC"

"android.permission.PROCESS_OUTGOING_CALLS"

"android.permission.READ_CALENDAR"

"android.permission.READ_CONTACTS"

"android.permission.READ_EXTERNAL_STORAGE"

"android.permission.READ_FRAME_BUFFER"

"android.permission.READ_HISTORY_BOOKMARKS"

"android.permission.READ_INPUT_STATE"

"android.permission.READ_LOGS"

"android.permission.READ_PHONE_STATE"

"android.permission.READ_SMS"

"android.permission.READ_SYNC_SETTINGS"

"android.permission.READ_SYNC_STATS"

"android.permission.REBOOT"

"android.permission.RECEIVE_BOOT_COMPLETED"

"android.permission.RECEIVE_MMS"

"android.permission.RECEIVE_SMS"

"android.permission.RECEIVE_WAP_PUSH"

"android.permission.RECORD_AUDIO"

"android.permission.REORDER_TASKS"

"android.permission.RESTART_PACKAGES"

"android.permission.SEND_SMS"

"android.permission.SET_ACTIVITY_WATCHER"

"android.permission.SET_ALARM"

"android.permission.SET_ALWAYS_FINISH"

"android.permission.SET_ANIMATION_SCALE"

"android.permission.SET_DEBUG_APP"

"android.permission.SET_ORIENTATION"

"android.permission.SET_POINTER_SPEED"

"android.permission.SET_PROCESS_LIMIT"

"android.permission.SET_TIME"

"android.permission.SET_TIME_ZONE"

"android.permission.SET_WALLPAPER"

"android.permission.SET_WALLPAPER_HINTS"

"android.permission.SIGNAL_PERSISTENT_PROCESSES"

"android.permission.STATUS_BAR"

"android.permission.SUBSCRIBED_FEEDS_READ"

"android.permission.SUBSCRIBED_FEEDS_WRITE"

"android.permission.SYSTEM_ALERT_WINDOW"

"android.permission.UPDATE_DEVICE_STATS"

"android.permission.USE_CREDENTIALS"

"android.permission.USE_SIP"

"android.permission.VIBRATE"

"android.permission.WAKE_LOCK"

"android.permission.WRITE_APN_SETTINGS"

"android.permission.WRITE_CALENDAR"

"android.permission.WRITE_CONTACTS"

"android.permission.WRITE_EXTERNAL_STORAGE"

"android.permission.WRITE_GSERVICES"

"android.permission.WRITE_HISTORY_BOOKMARKS"

"android.permission.WRITE_SECURE_SETTINGS"

"android.permission.WRITE_SETTINGS"

"android.permission.WRITE_SMS"

"android.permission.WRITE_SYNC_SETTINGS"

"com.android.launcher.permission.INSTALL_SHORTCUT"

"com.android.browser.permission.READ_HISTORY"

"com.android.browser.permission.READ_HISTORY_BOOKMARKS"

"com.android.launcher.permission.UNINSTALL_SHORTCUT"

"com.android.email.permission.ACCESS_PROVIDER"

"android.permission.WRITE_CALL_LOG"

"android.permission.READ_CALL_LOG"

"vdsoft.spying.sjin.permission.C2D_MESSAGE"

"com.google.android.c2dm.permission.RECEIVE"

"android.permission.ACCESS_DOWNLOAD_MANAGER"

"android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"

"android.permission.USES_POLICY_FORCE_LOCK"

"android.permission.ACCESS_PROVIDER"

"android.permission.READ_ATTACHMENT"

"android.hardware.camera.autofocus"

"android.permission.RECORD_VIDEO"

"android.permission.WRITE_OWNER_DATA"

"android.permission.SYSTEM_OVERLAY_WINDOW"

"android.permission.RECEIVE_USER_PRESENT"

"com.android.launcher.permission.READ_SETTINGS"

"android.webkit.permission.PLUGIN"

"com.android.browser.permission.WRITE_HISTORY_BOOKMARKS"

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : İrfan BULUT

Doğum Tarihi ve Yeri : 13.01.1985 - İstanbul

Yabancı Dili : İngilizce

E-posta : msc.irfanbulut@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Mühendislik	Dokuz Eylül Üniversitesi	2010
Lise	Fen bilimleri	Adnan Menderes Anadolu Lisesi	2003