Intermedia International e-Journal, Fall - December - 2018 5(9)

*DOI NO: 10.21645/intermedia.2018.53 *Submit Date: 20.06.2018 *Acceptance Date: .17.12.2018 *ISSN: 2149-3669

YENİ MEDYADA TİCARİLEŞME VE KİŞİSELLEŞTİRME STRATEJİSİ OLARAK KURGU PROGRAMLARININ DENETİMİ "DONGLE PAZARLAMASI"

Supervision of Fiction Programs as a Commercialization and Personalization Strategy in New Media and "Dongle Marketing"

Arş.Gör. Mustafa AYDEMİR,
Ege Üniversitesi İletişim Fakültesi,
Radyo Televizyon ve Sinema Bölümü,
İZMİR [1]
Orcid Numarası: 0000-0001-9414-4053

ÖZ

Medya içeriklerinin, kullanıcı eksenli hazırlanması ve sunulması için birçok video, grafik ve tasarım programı kullanılmaktadır. Yeni medyanın işleyiş yapısı gereği içerik sahipliği kadar denetlenebilir yazılım mülkiyeti önemli hale gelmektedir. Web 3.0 altyapısının etkisiyle tüm yazılımların ve eklentilerinin güncellenmesi pazar stratejisi gereği zorunlu hale gelmektedir. Program yazılımcılarının ticarileşme algısı ile kullanıcıların ücretsiz erişim talebinin yarattığı karşıtlık nedeniyle gerçek talep ve kullanıcı sayısı belirlenememektedir. Tekil kullanıcı ile ticari tabanlı son kullanıcıların bu yazılımları satın almasına ek olarak illegal kullanımların da sınırlandırılması anlayışıyla dongle adı verilen bir usb ile söz konusu programların çalışmasını ortaya çıkarmaktadır. Ticarileşen ve son kullanıcıların eylem alanını şekillendiren bu sistemin uygulanması ile yapım sonrası içerik üretim hizmetlerinin değişimi ve denetimi sağlanmaktadır. Bu çalışma program kimliğinin kişiselleştirilmesinin temel ilkeleri ile sürecin nasıl işlediğini incelemektedir.

Anahtar Kelimeler: *Yapım, Edius, Dongle, Kullanıcı Kimliği, Yazılım Korsanlığı, Kod Karıştırma.*

Extended Abstract: Many video, graphic and design programs are used for user-oriented preparation and presentation of media contents. Due to the functioning of the new media, as far as content ownership is concerned, ownership of auditable software is becoming more important. As a result of the Web 3.0 infrastructure, updating all software and add-ons is becoming mandatory due to market strategy. The real demand and number of users cannot be determined on the grounds of the contrast created by the commercialization perception of software programs and the free access demand of the users. İn addition to the purchase of these software by end-users based on commercial users and individual users, else dongle is also known as a USB to restrict the illegal use of these programs to reveal the work. With the implementation of the system, developments that require change and control of post-production content production services were experienced.

Innovations within the software bring freedom of choice to the user with different types of programs. The fact that these programs are powerful graphics and easy to use software have enabled the development of the computer technologies market. On the other hand, due to R & D activities and the increase in prices of companies due to market policies, usage costs increased as well. Nowadays, the USA and Far East companies, which direct the software sector, sell their products by using the installed computer infrastructure with domestic applications.

[1] mustafa.aydemir@ege.edu.tr

In the field of informatics, while the software developers are producing and marketing services, users try to obtain and use this service free of charge. Legal regulations on software are needed with sectoral globalization of information and communication technologies.

Companies are also developing new techniques to secure their products through copy protection. It is a popular method to experience software with time-restricted demo versions for other users. However, it offers limited access to output from projects. This causes an illegal approach to programs to purchase the paid version. Today, it is defined as an indirect service to break the passwords of software and try to market it to other users at a lower price or for free. They use protective apparatus called dongle to protect the rights of software companies. Dongle is created to identify the user and to prevent or restrict the illegal conduct of the user. Because companies are trying to make this fee permanent as a license renewal fee instead of making money from software once. The use of software for advanced computer products is exploited by both commercial firms and end users.

A hardware lock that is usually attached to a USB or serial connection is a hardware-based software protection technique designed specifically for authorized users to use licensed software applications. Normally, dongles are used with expensive applications. The individual, who uses the program for himself and is selfish in the digital environment, is increasingly moving away from being part of the collective illegal practice. Therefore, using the software's website, it creates an ID and shows a tendency to avoid sharing with others. The user's digital step can be easily calculated and other user profiles can be prevented from entering the program in a multiple model. The concepts of media, information and telecommunication are integrated disciplines. There is a functioning structure that includes the principle of reciprocity according to Producer-Consumer, Seller-Buyer and Acceptance-Rejections. This operation covers activities within the framework of commercial rights such as Guarantee, Distance Selling agreement, Multiple User Rights and Free Update Support.

The manufacturer (Software Owner / Firm) is looking for solutions with “encryption, coding, confusing, hiding, watermarking, reverse engineering and algorithmic actions to protect its property on the product at the information level. Software piracy is the person or people who uses reverse engineering methods for trying to infiltrate the software coding system in this way such as trying to convert the process in their favor. In the process from the production of information to the stealing of the software, problems and solutions are examined with the dongle sample. Besides, the software firms are using IP based licensing ways. Via this way the attacker can crack the software but they can’t get update packages. Software and media protection methods are different.

Media protection used to use watermarking and server side streaming protection techniques but software protection can be only dongle, ip restriction, etc… Finally, there is no just one technique about these protections. There are several methods are used by the software providers. This study examines the basic principles of personalizing the program identity and how the process works.

***Keywords**: Production, Edius, Dongle, User İdentity, Software Piracy, Obfuscation.*

Giriş

Bilgi ve iletişim teknolojilerinin 1980 sonrası liberal politikalar sonucunda etkisini daha fazla gösteren küreselleşme eğilimleri sayesinde yakaladığı pazarlama ivmesi, teknik araç ve yazılım gibi bilgi çağının gereksinimlerini karşılama ortamı oluşturmuştur. Bilginin dolaşım hızına ek olarak bilgisayar ve telekomünikasyon hizmetlerinde yaşanan yenilikler, arz politikalarını da şekillendirmeye başlamıştır. Bu ürünlerin birbirleriyle bağlantılı ve ardışık düzlemde tüm kullanıcı grupları için kitle iletişim araçları üzerinden “modernleşme aracı ve uyumlu olabilme” gibi duygusal kodlamalar yoluyla pazarlandığı bir dönem yaşanmıştır. Bu süreç, sonraki yıllarda üretim bandında sağlanan çeşitlilik ve rekabete açık küresel markalar tarafından özellikle internet ortamı kullanılarak sosyal paylaşım ağlarını da kapsayan bir algoritma ile şekillendirilmeye çalışılmıştır.

Günümüzde teknolojinin “en yeni” olana duyulan merak ve deneyimleme arzusuna göre kültür kazanan bireylerin hızlı tüketim davranışı göstermesiyle sektörün her yıl hızla daha pahalı ve az erişilebilir olması gibi bir

riskin ortaya çıkmasına zemin hazırlamıştır. Bireysel kullanıcıların sistem güncellemesine ek olarak ilgili ürünün yeni modeliyle güncellenmesi isteği, pazarın işleyiş dinamiklerinin, ilkelerinin ve pazarlama tekniklerini derinden etkilemektedir. Yeni yönelim bireylerin "tüketerek mutlu olabilme" yönüne doğru geçiş yapmasını sağlamaktadır. Bilgisayar teknolojileri konusunda kuramsal ölçekte firmaların yeni hizmet modelleri geliştirerek pazar hakimiyetlerini ve mevcut pozisyonlarını koruma düşünceleri, pazarın tüm kullanıcı gruplarına uygun hukuki düzenlemeler yapması gereksinimini ortaya çıkarmıştır.

Medya içinde yaşanan teknolojik gelişmeler ve değişen yayıncılık modelleri, interaktif kullanıcılar üzerine şekillenen yeni bir sistemin ortaya çıkmasını sağlamaktadır. Yeni medya olarak tanımlanan, yakınsama ve kullanıcı odaklı tasarlanan yenilik hareketi, internetle bağlantılı sosyal paylaşım ağları ile hızlı bir pazar ağı elde edebilmektedir. Tüm kullanıcı gruplarının (birey, kamu ve özel) gündelik eylemlerinde önemli bir zamanı bilgisayar ve mobil cihazlar ile şekillendirmesi, bilişim sektörünün işleyiş yapısını derinden etkilemektedir. Bilgisayar ve bağlantılı ürünlerin yazılım olarak ilgili medya ortamına göre tasarlanmasıyla ağ politikaları "kalite ve kullanım" yönüyle de güncellenmeye başlanmaktadır. Reklam içeren ücretsiz yazılımlar ve paralı yazılımlar dışında çevrimiçi olarak sunulan ürünler de ortaya çıkmaktadır. Bu ürünler, özellikle kopyalama ve yasadışı yollarla program kurulumlarının önüne geçmeyi hedefleyen bir yapıda oluşturulmaktadır. Zira kullanıcılar içinde ekonomik, ruhsal, kişisel zaafiyet, kişisel tatmin, diğer kullanıcılarla paylaşma gibi farklı gerekçeler ile bu alanı tahrip eden bilgisayar korsanlığının asgari düzeye çekilmesi adına yazılımları korucuyu bazı modeller ve ürünler hazırlanmaya çalışılmaktadır.

Medya alanında içeriği izleyen ve tüketen kullanıcı figürü yerine içeriği oluşturan ve denetleyen etkin kullanıcı davranışları oluşturulması adına hukuki yapıyı özendiren uygulamalar piyasaya sürülmektedir. Dongle adı verilen usb, akıllı kart, cd ve anahtar (İng. Key) aygıtları üzerinen yazılımları bir kilit üzerinden kullanıcı kimliğiyle çalıştıran hukuki yönü güçlü bir sistem oluşturulmaktadır. Bu sistem, kullanıcıların genel profilini belirlemek ve kullanım düzeylerini saptamak gibi stratejilerin dışında pazarın üretim bandının yüzdelik değişimine göre hareket etmeyi hedefleyen bir pazarlama modeli üzerine inşa edilmektedir.

1. Yazılım Koruma Sisteminin Yapısı

Yazılım kavramı bilgisayar teknolojilerinin önemli bir içerikleşme konusunu yönlendiren sistem araçlarını ifade etmektedir. Yazılım üretim yapısı olarak farklı programlama dilleri üzerinden oluşturulan kodlama ve algoritmik hesaplamalara dayalı, çeşitli güvenlik işlemlerini yaparak, lisans alarak piyasaya sunulan ürünleri içermektedir. Programlama dilleri olarak Pascal, Basic, C, C#, C++, Java, JavaScript, Cobol, Perl, PHP, Python, Ada, Fortran, Delphi gibi eski tip kullanımlara yeni dönem içinde katılan iki dil ise Google GO ve Apple Swift'tir.

Programlama dili, en temel haliyle yazılımcının bir algoritmayı ifade etmesi adına bir bilgisayara ne (ler) yapmak istediğini anlatmasıdır. Programlama dilleri, uygulama yapısı olarak bir yazılımcının bilgisayara hangi veri üzerinde işlem yapacağını, verinin nasıl ve hangi koşullarda depolanıp iletileceği mantığını içermektedir. Her bir yazılım için yeni bilişim hizmetleri kapsamında bulut (İng.Cloud) tabanlı olarak hazırlanan lisanslama çözümlemeleri ile söz konusu ürünün herhangi bir şekilde lisanssız kullanımı ve üçüncü kişilere dağıtımını engelemek adına kurumsal ya da özel yazılım ürünleri oluşturulmaktadır.

Bu çözüm, yazılım üzerinde kullanım verilerinin izlenmesi gibi yazılım takibi de dahil olmak üzere zamanlamaya dayalı ve içerik olarak aşamalara göre bölümlendirilmiş işlemleri içermektedir.

1.1. Önceki Çalışmalar

Bir görüş olarak "Yazılım dünyayı yiyor – Software is eating the World." diyen Marc Andreessen gibi birçok kişi için sektör, gelişime oldukça açık olan programlama dilleri üzerinden yepyeni bir düzen içeren bir eğilim içindedir. Yazılım süreçlerinin nitelikli olarak yürütülebilmesi için, "Planlama, Tasarım, Analiz, Soyutlama ve Uygulama" gibi konuların nitelikli olarak yapılması gerekmektedir. Bu nedenle sürecin bir döngü biçiminde tanımlanmasıyla

yazılımın oluşturulmasından geliştirilmesine kadar derinlikli bir süreç önemli hale getirilmektedir. Yazılımın geliştirilmesi süreçleri kendi içinde (CMM, CMMI, ISO 9000, ISO 9001 ve SPICE)[2] gibi temel örnekleri içermektedir.

Yazılım konusu üretim kademelerinin içerik ve uygulama özellikleri açısından teknik olduğu kadar ticari bir ilgi konusudur. Sadece bilgisayarla kullanılarak üretilen programların çıktı değerinin her yıl milyar dolarlık bir bilişim sektörünün parçası haline dönüşmesi bu alana yapılan yatırımların ve akademik çalışmaların düzeyini de derinden etkilemektedir. Yazılım konusu üretim ve tüketim nesnesi olmakla birlikte piyasaya sunulduktan sonra kopyalama ve telif gibi hukuki konuları içeren bir modele dönüşmektedir. Lee ve Kim, korsanlığın önlenmesinin "telif hakkı koruması" ya da "kopya koruması" olarak sınıflandırılabileceğini öne sürmektedir. Burada, bir itilaf durumunda mülkiyetin ispat edilmesi için bir önlem geçerliyken ikincisi korsanlığın kendisi için bir engeldir (1999). Yazılım korsanlığı temelde yazılımın izinsiz olarak elde edilmesi (indirme, şifresini kırma, paylaşma, kiralama ve çoğaltma) gibi farklı eylemleri içermektedir. Chen (2001), Bahaa-Eldin vd., (2014), Chang ve Atallah, (2002) yazılım konusunu güvenlik boşluğundan faydalanma ve kazanç elde etme (Karpavelli ve Arundevi, 2017) amaçlarıyla gerçekleştiğini belirtmektedir. Yazılım korsanlığına karşılık ortaya atılan bazı öneriler ve koruma teknikleri arasında Filigran (İng. Watermarking) tekniği üzerine Collberg ve Thomborson, (1999), Collberg vd. (1998) yazılım filigranı için detaylı tanımlar sunmuştur.

Obfuscation tekniği temelde, bir programın, uygulanan bir transformasyona bağlı olarak biçimsel değişiklini ifade eden kavramdır. Bu konuda yapılan çalışmalar Collberg vd., (1998) tarafından temellendirilmekle birlikte Kuzurin vd., (2007) şaşırtma eyleminin belirgin hale getirildiğini vurgulamış; Jakubowski vd., (2009) ise türsel olarak sınıflandırma yoluna gitmişlerdir.

Kişisel ya da ticari nitelikli verilere üçüncü kişilerin ulaşmasını önlemek adına yapılan uygulamalardan biri olan "cryptography" tekniği ise kamuya açık anahtar (İng. Public Key) üzerinden şifreli ve saklanmış verilerin kullanıcının inisiyatifine göre dosya ve paylaşmak istediği veriler üzerindeki hakimiyetini ifade etmektedir. Bu konuda, veri gizliliğini dijital imza ve şifreleme üzerinden ele alan bazı çalışmalarda (Rivest, Shamir & Adleman, 1978) şifreli mesaj ya da e-posta gönderimi gibi kapalı erişim içeren, şifreli iletilerin işlevlerini koruyucu bir hamle olarak tanımlayan (Boneh, Sahai & Waters, 2012) bazı görüşler gibi kullanıcı reflekslerinin güçlü olduğunu vurgulayan yapıya göndermede bulunmaktadır. Pennsylvania Üniversitesi dil kayıtlarına göre, dongle kelimesinin en erken atıfları 1982'de ortaya çıktığını belirten Smith (2013) dışında dongle konusunu genellikle yüksek kaliteli düşük miktarlı yazılımı korumak için kullanıldığını ve korunması amaçlanan ilgili yazılımla birlikte müşterilere gönderilen yazılımın, gönderildiği ortamda belirli dongle'ı algılamadıkça yüklenememesi ya da çevrimiçi olarak kullanılamamasını ifade eden (Piazzalunga, Salvaneschi, Balducci, Jacomuzzi ve Moroncelli, 2007) görüşlerle birlikte yetkilendirme ve yetkisizleştirme niteliği olduğunu (York ve Muratore 2004) belirten dongle tipleri ile uygulamanın getirdiği psikolojik yapıyı inceleyen (Varian vd., 2003) sistemin işleyişini değerlendiren çalışmalar da yer almaktadır. Çalışmanın bir başka inceleme alanını oluşturan ve negatif bir değerlendirme yapılan alan yazılım korsanlığı'dır. Yazılım sektörü içinde yazılım korsanlığı'nın, (İng.Software Piracy) en büyük tehdit olduğunu (Campidoglio M, & Frattolillo F & Landolfi F., 2009) savunan bazı görüşler kullanıcı ile üretici reflekslerinin gerekçelerinin ne olduğu sorunsalına odaklanmaktadır.

1.2. Yazılım Korsanlığı

Yazılım sektörü; bilgiye erişebilme, uyumlu olma, küresel sistemin gerektirdiği enformatik yapıyı kişisel niteliklere indirgeyerek gelişim sağlayabilmek adına ürettiği her bir program için kamusal fayda sağlamanın dışında ticari gerekçelere de dayandığı için kazanım konusu dikkate değer bir yapıya dönüşmektedir.

Üretilen her bir yazılım en yalın haliyle bir eserdir. Bu eser bir kitabın yeni baskısı gibi kendi içinde güncelleme yaparak tüketicilerin kullanım deneyimleri ve sistemde ortaya çıkan açıkları önleyici bilişimsel

---

[2] CMM (Yetenek Olgunluk Modeli), CMMI (Bütünleşik Yetenek Olgunluk Modeli), ISO 9000 (Üretim ve Yönetim Süreçleri için Avrupa Standardı) ISO 9001 (Üretim ve Yönetim Süreçleri İçin Genel Avrupa Standardı) SPICE (Yazılım Süreci İyileştirme ve Yeterlilik Belirleme) şeklinde ifade edilen tüm modeler yazılımsal sürecin kullanılan teknoloji ve kalite yönetimi olarak zaman, maliyet ve verimlilik durumlarının analizini içermektedir.

eylemleri içermektedir. Yazılım korsanlığı konusu Karpakavallı ve Arunadeviye göre "yazılımın yasal olmayan bir şekilde kopyalanması, dağıtılması veya kullanılmasıdır. Bir dizi ülkede organize suç gruplarının dikkatini çekmiş olması çok kârlı bir "iş" tir. Yazılım korsanlığı, yayıncılar için kayda değer bir gelir kaybına neden olur ve bu da tüketiciye daha yüksek fiyatlar getirir (2017:603). Yazılımsal üretim lisanlarının aygıtsal numaralar ile ya da ürünü kullanan kullanıcıları içerdiğini belirten Alawneh ve Abbadi, "yazılım lisanlarının içinde birbirinden farklı denetleyici kuralların birçok güvenlik akışı ile sınırlandırıcı nitelikler gösterdiğini" (2008:509-523) açıklamaktadır. Yazılım korsanlığı konusunda sosyo-psikolojik yapıyı öne çıkaran bazı görüşler ise (Harran vd., 2015) "insanların yanlış olduğuna inandığı ve sosyal olarak kabul edilebilir olan gerçek davranış arasında yaşamın birçok alanında bir altta yatan dikotomi[3]" olarak tanımlanmaktadır. Yazılım Korsanlığı, fikri mülkiyet yasaları uyarınca korunan mülkün yasadışı kullanımı veya dağıtımıdır. Yazılım korsanlığı aşağıdaki kategorilere ayrılabilir:

- Son Kullanıcı Korsanlığı
- İstemci Sunucu Aşırı Kullanımı
- İnternet Korsanlığı
- Sabit Disk Yükleme
- Yazılım Sahteciliği (Chen, 2001: 2)

Yazılım korsanlığının bunlara ek olarak soflifting (izinsiz aktarım) ve renting (kiralama) olarak kulanıcılardan diğer kullanıcılara aktarılması konusu daha önce sözünü ettiğimiz 1980 sonrasında geniş kapsamda ele alınarak değerlendirilmeye başlanmıştır. Hunt ve Vitell'in (1986) izinsiz aktarım konusundaki görüşleri, motivasyon faktörü olarak kullanımı (Simpson vd., 1994), kontrol edilebilme yolları (Gopal ve Sandars, 1997) dışında Thong ve Yap (1998) gibi bazı araştırmacılar tarafından bu eylemlerin temel gerekçeleri, (Conner ve Armitage, 1998) ile hesap yönetimlerinin denetimi (Moseley ve whitis, 1995) ve planlanmış davranış olarak nasıl gerçekleştirildiği (Conner ve Armitage, 1998) bağlamlarında incelenmiştir.

Yazılım korsanlığının yükselişi, bu alanda nitelikli araştırma ve geliştirmeler yapılarak farklı yazılım kopyalama ve koruma tekniklerinin oluşturulmasına yol açmaktadır. Yazılım kopyalamasını katmanlar olarak değerlendiren Bahaa-Eldin ve Sobh (2014), bu katmanları; *"kod koruması, lisanslama ve sayısal haklar yönetimi katmanı"* şeklinde sınıflandırmaktadır.

Yazılım kırılması, (İng. Cracking) yazılım sektöründe birçok kişi için ciddi bir tehdittir. Saldırı yapmak istediği yazılımın bir kopyasını elde eden bir korsanın içine girmiş olan korumayı kırmayı başarması sorun teşkil etmektedir. Chang ve Atallah, yazılımların koruyucu kodlaması üzerine yaptıkları çalışmada, bu noktada yapılan işlemin genel yapısını ele almaktadır. Chang ve Atallah'a göre "Tipik olarak, kraker, yazılımın değiştirilmiş sürümlerini veya kopya koruması veya kullanım kontrol mekanizmaları devre dışı bırakılmış olan kırılmalar oluşturur. Kırık yazılım daha sonra kamuya yasadışı olarak yeniden dağıtılabileceğini ve yazılım korsanlığı sorununu şiddetlendirebileceğini" (2002: 160) dolayısıyla da olası eylemler karşısındaki pasif tavrın sonuçlarının neler olabileceğini saptamaya çalışmaktadırlar. Yazılım korsanlığının önlenmesi konusunda yapılabileceklerin neler olacağı sorusunu dört ana özellikle ele alan Chang ve Atallah, güvenilmeyen ortamlarda çalışan yazılımların etkili bir şekilde korunmasını sağlayan koruma mekanizmaları aşağıdaki özelliklere sahip olması gerektiği konusunda önerilerini Esneklik, Kendini Savunma, Yapılandırılabilirlik ve Beyaz Kutu Güvenliği" şeklinde sıralamaktadırlar. Buna göre:

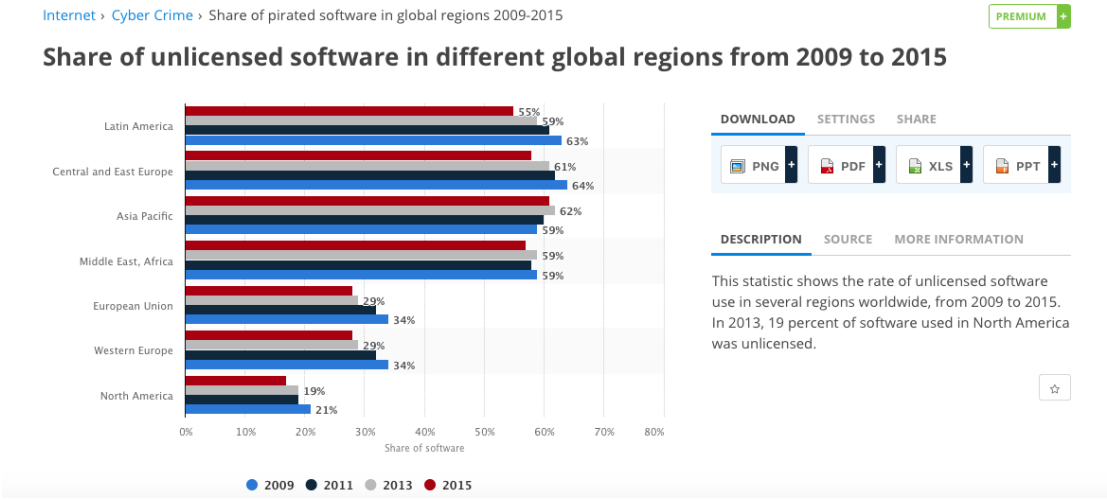
a. **Esneklik:** Korumanın tek bir arıza noktası yoktur ve devre dışı bırakmak zordur.
b. **Kendini Savunma:** Kurcalamaya karşı eylemleri algılayabilir ve alabilir (ör. Kod değiştirme).
c. **Yapılandırılabilirlik:** Koruma özelleştirilebilir ve ihtiyaç duyulduğu kadar güçlü yapılabilir.
d. **Beyaz Kutu Güvenliği:** Koruma için herhangi bir planın zaman içinde herkes tarafından

[3] Dikotomi kavramı dilimize "ikileşim" olarak gemiştir. Kelime köken olarak Yunanca; "dichia": ikili; "temnein": kesmek, iki eşit parçaya ayrılmak üzere büyüme noktasından ikiye bölünerek dallanma; İngilizce "dichotomy"; Fransızca, "dichotomie" sözlüğünden üretilmiş olup, sosyal bilimler alanında bir sistemin içinde iki farklı işleyiş olarak kullanılmaktadır.

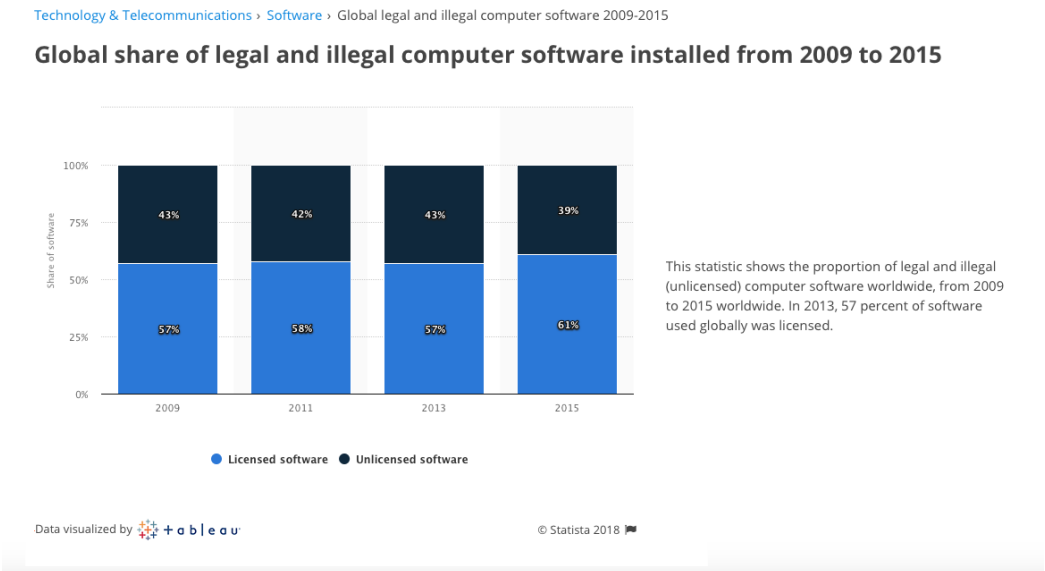
bilinmesi muhtemel olduğundan, gücünün gizliliğine değil, koruma yükleme süresinde kullanılan gizli bir anahtar bilgisine dayanması gerekir (ancak korunan program herhangi bir yerde saklanmaz) (2002: 161).

e.

Chang ve Atallah, gibi birçok araştırmacının önerileri karşısında sektörün geldiği durumun olumsuz yönlerinden birini oluşturan lisanssız yazılım paylaşımı konusunda Statista şirketi tarafından yapılan bir araştırmaya göre 2009 ile 2015 yılları arasında dünyanın farklı bölgelerinde durum incelenmiştir. Araştırma kapsamında seçilen yedi bölge "Latin Amerika, Merkez ve Doğu Avrupa, Asya Pasifik, Ortadoğu-Afrika, Avrupa Birliği, Batı Avrupa, Kuzey Amerika" şeklinde sıralanmaktadır. Aşağıda yer alan **Grafik 1** değerlendirildiğinde, Avrupa Birliği, Batı Avrupa ve Kuzey Amerika "gelişmiş ekonomileri ve üretim ile satış yönetimine sahip şirketleriyle" lisanssız kulanıma 2 yıllık ölçütlerle belirlenmiş süreçte maksimum %34 oranında gerçekleşirken, diğer bölgelerde ise "ekonomik gelişmemişlik, teknolojik eksiklik vd gerekçelerle" kullanım oranının % 64 seviyelerine çıktığı görülmektedir. Buna göre birinci grup içinde yer alan kullanıcılar arasında her üç kullanıcının bir tanesinin lisanssız yazılım kullandığı, ikinci grupta yer alan kullanıcılar arasında ise her üç tanesinden ikisinin lisanssız yazılım kullanma eğiliminde olduğu belirtilmektedir.



Grafik 1: Dünyanın Farklı bölgelerinde 2009-2015 Dönemi Lisanssız Yazılım Paylaşımı (www.statista.com, 2018)



Grafik 2: 2009-2015 Arası Yasal ve Yasadışı Bilgisayar Yazılım Kullanımlarının Küresel Payı (www.statista.com, 2018)

Bilişim sektörüne yön veren şirketlerin bu tür sorunlar karşısında kopya koruma teknikleri üreterek korsanlık konusunu en azından sınırlandırabilme amaçları bulunmaktadır. Bilgiye sahip olma yerine hakim olma ve bilgiyi sızdırma, virüs yoluyla hesapları kırarak bilginin elde edilmesi, üreticilerin ve kullanıcıların hesapları ile yazılımları üzerindeki hakimiyetleri konusunda yaşanan karmaşa karşısında çözüm yolları gösterilerek olası kayıpların önüne geçilmesi hedeflenmektedir. Kopya koruma teknikleri yazılımlar, patentler, algoritmalar ve internet tabanlı eylemler içermektedir. Bunları sırasıyla aşağıda yer aldığı gibi yapısal ve türsel olarak sınıflandırarak değerlendirebiliriz.

1.3. Kopya Koruma Teknikleri

1.3.1. Watermarking (Filigran/Damgalama)

Yazılım koruma teknikleri arasında en bilinen uygulamalardan birisi filigran/damgalama olarak tanımlanan (ing. Watermarking) bu sistem Oxford sözlüğünde watermark olarak "Üretim sırasında bir kağıt yaprağına etkilenen ayırt edici bir işaret ya da aygıt, genellikle kağıt güçlü bir ışığa karşı tutulduğu zaman neredeyse farkedilemez" bir yapı olarak tanımlanmaktadır (Simpson & Weiner, 2000:176). Bu tekniğin yazılım alanıyla etkileşime geçmesi (Cox vd., 1996) ilk olarak 1954 yılında gerçeklemiş olup, telif alanında yapılan patent çalışmaları (Davidson ve Myhryold., 1996) ile desteklenmiş ve 1990 yılında ise (Tamada vd., 2004) dijital olarak bu tekniğin kullanımı popülar hale gelmiştir. İlk kez, (Collberg ve ark. 1998 & 1999) yazılım filigranı için detaylı tanımlar sunulmuş; yazılımların algoritması konusunda (Wenkatesen vd., 2001) bazı öneriler de ileri sürülmüştür. Kopya koruma uygulamaları, filigranın, telif hakkı korsanları tarafından bile herkes tarafından okunabilmesini gerektirmektedir.

Filigran gereksinimleri uygulamalar arasında farklılık gösterir. Cox ve Linnartz'e göre bu noktada "önemli bir ayırt edici özellik, filigran okuma kabiliyetine getirilen kısıtlama seviyesidir. Örneğin, birçok durumda, bu bilginin birçok alıcı tarafından okunabileceği şekilde ses, görüntü veya video içeriğine bilgi gömülmesi arzu edilmektedir" (1998: 587). Watermaking konusunda temel bir örnek olarak Springboard (6 Mile Creek Systems, 6sys.com/Springboard) örneğini veren Bardosh, programın genel içeriği ve uygulama modüllerini şöyle açıklamaktadır:

> *"Springboard, Windows bilgisayarları için bir yazılımdır. Çeşitli projeler için ücretsiz ve yararlıdır. Ücretsiz sürüm, resimlerde filigranlar yerleştirir, bazı baskı ve ihracat sınırlamalarına sahiptir ve yayınlanma tarihinden dört ay sonra sona ermektedir. Ancak, süresi dolduğunda yeni bir ücretsiz sürümü indirmekten engellenmezsiniz. Hoşlanıp karar verirseniz, bunu 35 $ için kaydedebilir ve filigrandan kurtulabilirsiniz" (2007: 111).*

Yazılım sahibi veya telif hakkı sahibi, daha sonra korsanlık ya da yazılımın yetkisiz kullanımıyla ilgili bir kanıt elde etmek için gizli mesaj oluşturarak yazılımın içinde gizleme yolunu seçebilmektedir. Yazılım filigranı, yazılımı korsanlıktan korumak ve ticari yazılımlar için telif hakkı korumasını sağlamak için reaktif bir yaklaşım (Zhang vd., 2008), (Sasirekha ve Hemalatha, 2002) olarak kabul eden bazı görüşlerle birlikte filigranlama tekniğini, "video, görüntü ve ses dosyaları gibi belirli dijital ortam içerikleri için tasarlandığını ancak ilişkisel veritabanları ve doğal dil metin / belge dosyası" (Atallah, 2005) gibi medya içeriklerinde filigran uygulamasına ilgiyi artırdığını ileri süren çalışmalar da bulunmaktadır. Filigran tekniğinin çalışma düzeni (Miller vd., 1999), aşağıda yer alan Şekil 1'de, orjinal enformasyon ve medya sinyalleri işlenerek medya ortamına daha sonra tarayıcı kodlanarak algılanan filigran bilgisine ve insan algılama sistemi üzerinden ise algılanan medya yapısına dönüştürülmektedir.

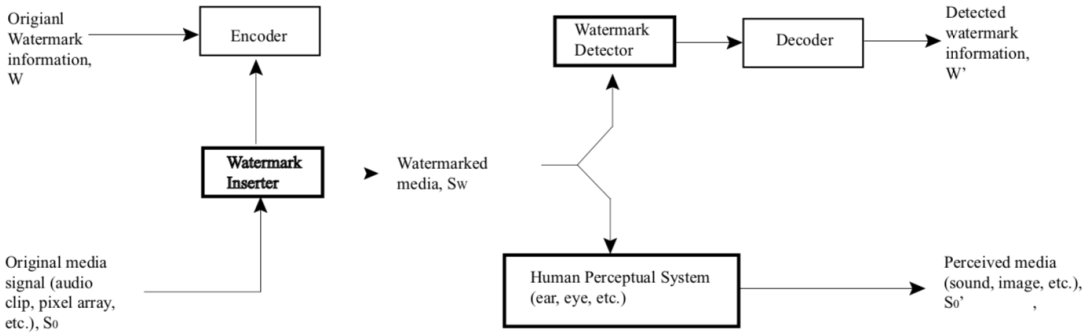


Figure 18.1: Watermarking framework.

Şekil 1: Watermarking (Filigran) Çalışma Düzeni, (MILLER vd., 1999:4)

Watermarking (Filigran) tekniğini gizli ve kamuya açık olmak üzere iki'ye ayıran Arnold'a göre;

a. Gizli filigranlar, çeşitli şekillerde kimlik doğrulama ve içerik bütünlüğü mekanizmaları olarak kullanılabilir. Bu, filigranın yalnızca sır hakkında bilgi sahibi olan yetkili kişiler tarafından okunabilen güvenli bir bağlantı olduğunu gösterir.
b. Kamusal filigranlar, herkes tarafından okunabilir, bir bilgi taşıyıcısı olarak hareket eder. Bu kamu filigranları üçüncü bir tarafça tespit edilemez veya çıkarılmamalıdır. Bu filigranlar bilgi bağlantıları olarak hareket ederse, bu gereksinim azaltılabilir (2000:1).

Teknik olarak, yazılım filigran tekniklerinin çoğu iki kategoriye ayrılır: statik filigran (Davidson ve Myhrvold, 1996; Moskowitz ve Cooperman, 1996) ve dinamik filigranlama (Collberg ve diğerleri, 1990; Collberg ve Thomborson, 2004; Collberg ve Thomborson, 1999). Kamela ve Albluwid'e göre; "Statik filigranda filigran, ya veri bölümünde ya da kod bölümünde, kaynak kodunda saklanmaktadır (2009: 395-396). Zhu vd., (2005) ise "dinamik bir filigranın bir yazılım nesnesinin yürütme durumuna ekleneceğini ve filigranı çıkarmak için programı yürütmek gerektiğini" vurgulamaktadır. Collberg ve Thomborson, bu noktada üç temel fonksiyonun önemine vurgu yapmaktadır. Bunlar;

(a) program dönüşümlerine tabi tutulduktan sonra numara güvenilir bir şekilde alınabilir,
(b) yerleştirme bir rakip tarafından algılanamaz
(c) yerleştirme, programın performansını düşürmez (1999: 311).

Sonuç olarak yazılım filigranı bir programa çok sayıda yerleştirme işlemidir. Bu işlemin yapılma süreçleri statik ve dinamik tüm modeler ile fonksiyonlar programın veri yönetimi ve kaynağın açık ya da kapalı tutulmasına bağlı olarak şekillenecektir.

1.3.2. Obfuscation (Kod Karıştırma)

Yazılım alanında uygulanan bir başka teknik olarak kod karıştırma tekniği mevcut kodların üzerinde yapılan küçük hileler ve değişiklikler ile yasadışı kullanımların önüne geçilmesi işlemidir. Collberg vd., bu işlemde, herhangi bir programın, uygulanan bir transformasyona bağlı olarak, bir insan rakibi veya otomatik bir veya her ikisi için anlaşılması daha zor olan bir forma dönüştürülmesi tekniğini içerdiğini vurgularken; (1998) Kuzurin vd., ise "bir programı şaşırtmak, bir programın metninden algoritmalar, veri yapıları, sekrete anahtarları vb. ile ilgili bazı değerli bilgilerin çıkarılmasını mümkün olduğunca engelleyen bir forma sokmak anlamına geleceğini" (2007:281) belirtmektedir. Kod Karıştırmayı ve Filigran tekniklerini birbirine bağlı yazılım koruma teknikleri olarak kabul eden (Zhu, 2007), benzer ilkeler çerçevesinde sistematize edildiğini ön plana çıkarmaktadır. Collberg ve Thomborson ise bu uygulama ile "işlevsel olarak orijinal programa eşit olacak şekilde dönüştüğünü" belirtmektedir. Kod

karıştırma konusu, kuşkusuz "gizleme, çeşitlilik ve kurcalamaya karşı korumayı birleştiren" bir yazılım koruma tekniğidir. Zeng vd, bu noktada ince bir ayrıntı olduğunu tekniğin özellikle "bir programın kendisini gizlemek yerine, programın yorumlama sürecini şaşırtarak" yaptığını söylemektedir (2011:329-340). Ancak burada dikkat edilmesi gereken bir başka husus ise kaynak kodunun kullanımıdır. Leite ve Cappeli'ye göre "kaynak kodu, farklı bilgisayar programlama dillerinde yazılabildiğinden, o belirli programlama dilindeki okuma yazma sorunu, gizlenmeye katkıda bulunan bir sorun olabilmektedir" (2010:127).

Bu tür teknikler çeşitli bağlamlarda iyi hizmet vermesine rağmen az sayıda korunan bir programın pratikte ne kadar uzun kalmayacağını tahmin etmek için pratik bir güvenlik analizi önermiştir. Çözümler, bazı mühendislik varsayımları altında mevcutken (Dedic ve diğerleri, 2007), Jakubowski vd, bu konuda mevcut açık problemin gerçekçi bir güvenlik değerlendirmesini destekleyen pratik koruma teknikleri geliştirdiğini" ileri sürmektedir (2009:1). Ceccato vd., kod karıştırma konusunu transformasyonlar açısından üç gruba ayırmaktadır. Bunlar: "düzenleri gizleme, davranışlarını değiştirmeden koddaki ilgili bilgileri kaldırma; veri gizleme, uygulama verilerini ve veri yapılarını dönüştürme (ör. veri kodlama, veri bölme); ve kontrol akışı kod değiştirmenin uygulamanın orijinal akışını değiştirmesidir" (2009:1) Yazılım gizleme, tüm işlevsellikleri korurken anlaşılması güç hale gelen kod üzerindeki dönüşümleri ifade eder. Hui vd., "gizli veri ve algoritmaların tersine mühendislik veya virüs modifikasyonundan korunmasında önemli bir rol oynadığını belirtirken, (2011: 168) CERT[4] tarafından 2014 yılında hazırlanan raporda ise bu işlem "Kodu gizleme, güvenliği geliştiren, fikri mülkiyet haklarını koruyan dizi meşru nedene sahiptir. Ayrıca, kötü amaçlı yazılım yazarları tarafından kodlarının hayatta kalması ve algılamayı önleme yeteneklerini arttırmak için kullanılmıştır" (2014:1) şeklinde açıklanmaktadır.

Kod karıştırma konusunda Kieseberg ve Weippl'e göre iki temel kavramın özellikle ümit verici olduğu kanıtlanmıştır. Bunlar: " (i) gizleme / çeşitlendirme ve (ii) filigran kavramı"dır. Zira, "gizlemenin ardındaki temel fikir, bilginin herhangi bir saldırgan için anlaşılmaz hale getirilmesi, dolayısıyla değerinin kaldırılmasıdır" (2018:12). Genellikle kodu korumak için kullanılırken, diğer bir deyişle programın kolayca geri çevrilemeyen bir versiyonunu üreterek verileri gizlemek için üretilen stratejiler de bulunmaktadır.

1.3.3. Cryptography

Yazılım alanında iletilen bilginin, istenmeyen şahıslar tarafından anlaşılmaması adına yapılan bir tekniktir. Burada, temel mantık; veri gizliliğini bütün olarak, kimlik denetimi yaparak, inkar edilmeksizin ve erişim kontrolü sağlanarak yapılmasıdır. Eski Mısır döneminden itibaren bilinen ilk uygulamalara bakıldığında erişimi sınırlamak ya da şifreleyerek erişime kapatma düşüncesiyle yapılan işlemler bulunmaktadır. Günümüzde bilgisayar ortamında açık ve kapalı anahtarlama şeklinde kamu ve özel niteliğe ilişkin bir sistem bulunmaktadır. Bu teknik, iki farklı yöntem geliştirilerek uygulanmaya başlanmıştır.

Naor ve Shamir, 1995 yılında gizli mesajların korunması için görsel kriptografi (Visual Cryptography) olarak adlandırılan devrimsel bir kriptografik yapı önermiştir (1995:1-12). Bu yeni kriptografik yapının iki önemli özelliği vardır:

(1.) Gizli mesajları korumak için mükemmel bir güvenli yol sağlamaktadır.
(2.) İnsan görsel sistemi (HVS) şifrelenmiş mesajları alırken herhangi bir hesaplama yapmadan gizli mesajları doğrudan belirleyebilmektedir (2007: 1).

Yukarıda bu iki filigran sınıfını kamu ve özel filigran olarak belirrtik. Ancak bu, kriptografide "kamusal" teriminin iyi bilinen anlamı göz önüne alındığında yanıltıcı olabilmektedir. (Miller vd., 1999) bu konuda "bir kamu anahtar şifreleme algoritması iki sır içerir; Bir mesajı şifrelemek bir sırrı bilmeyi gerektirir ve bir mesajın şifresini çözmek ikinciyi de bilmeyi gerektirir" şeklinde açıklama yapmaktadır.

[4] CERT (Computer Emergency Response Team/ Bilgisayar Acil Müdahale Ekibi) CERT'in işlevi, özellikle olayların önlenmesi, ele alınması ve raporlanmasına odaklanarak, bilgisayar güvenliği konularında en iyi uygulama tavsiyesini ve en iyi uygulamayı sağlamaktır.

Yeni teknolojilerin ortaya çıkmasında kriptografik teknikler, kimlik doğrulama araçları ve güvenlik duvarları önem kazanmıştır. Sürekli olarak artan denetim ve düzenlemeler ile bu tekniğin etkinliği de artmaktadır.

1.3.4. Dongle (Yazılım Kilidi)

Bilgisayar programları satın alan ve ilgi alanına göre bunun üzerinde çalışan figüre yaşadığımız yüzyılın en temel gereksinimi olan bilgiyi üretebilme konusunun doğal bir sonucudur. Çalışmamızın ana konusunu oluşturan koruma tekniği olarak "Dongle" kavramının ne olduğu ve yayıncılık konusundaki rolünün nasıl şekillendiğini sistemin yönelimlerini anlamaya başlayarak çözümlemek mümkündür. Yeni medya sistemi ile özellikle sosyal ağlar üzerinde görsel ve işitsel temalı paylaşımlar yapmak için en etkileyici görsel etki araçlarını mobil ortamda (cep telefonları) bile olsa yapmaya çalışmak paylaşılan şeyin önemine atıfta bulunmaktadır. Zira günümüzün ön plana çıkabilme nesnesi olarak medya alanına yönelmek ve bu yolla tanınırlık kazanmak için (Youtuber, Viber) gibi paylaşım ağıyla eşleşen kullanıcı kimlikleri inşa edilmektedir.

Dongle kavramı yazılıma bağlı olarak bir usb üzerinden ilgili programı kullanmak üzere bilgisayara takılan bir aygıtı ifade etmektedir. Bu kavramın donanım kilidi olarak ya da donanım aygıtı olarak tanımlanmasında usb'nin temelde donanım olarak kabul edilmesinden kaynaklanmaktadır. Buna karşın aygıtın asıl kaynak noktası yazılım olduğu ve yazılıma bağlı olarak çalışma özelliği göstermesi nedeniyle yazılım kilidi olarak açıklanması daha doğru bir görüş olarak düşünülmektedir.

Dongle'lar (Djekic ve Loebbecke, 2007; Piazzalunga, Salvaneschi, Balducci, Jacomuzzi, & Moroncelli, 2007) ayrıca donanım tuşları olarak da adlandırılır ve bu, USB portu gibi uygun bir bağlantı noktasına takılabilen küçük bir çubuğu kullanan donanım tabanlı bir önlemdir. Örneğin bir kullanıcı yazılımı çalıştırmak istiyorsa, belirtecin bilgisayara bağlı olması gerektiğinden bu tür korumanın kırılması için kaynak kodunun değiştilmesi gerekmektedir.

Tek bir seri numarası, satıcı tarafından her dongle müşterisine atanan benzersiz bir bağımsız yazılım satıcısı kimlik numarası dahil dongle çeşitli mantıksal kaynaklar, dongle'ın işlevselliğini, yazılımın yazabileceği kalıcı bir belleği ve simetrik anahtarlar için simetrik bir şifreleme içermektedir. Dongle, depolamayı açmak için bir erişim şifresini içermektedir (Piazzalunga ve diğ., 2007). Bununla birlikte yukarıda bahsedilen tüm mantıksal kaynakların her zaman bir dongle'da mevcut olmadığını belirtmek de önemlidir.

Dongle, "normal olarak ayrı bir harici aygıta bağlantı için kullanılan bir bilgisayardaki bir bağlantı noktasına takılan bağımsız bir aygıt" (Smith, 2013) olmasının dışında York ve Muratore ise "donanım veya yazılımın yetkisiz kullanımını engelleyen bir cihaz" ifadesinde bulunmaktadır. Bir dongle genellikle bir cihaza bağlı küçük bir kablodan veya donanımı sabitleyen bir anahtardan oluşmaktadır. Terim, ayrıca çevre birimleri için genel bir bağdaştırıcıyı belirtmek için kullanılmaktadır (2004: 404). Schneider'e göre Dongle, kopya korumada teknolojinin mevcut durumunu yansıtmakta olup, genellikle paralel bağlantı noktasına bilgisayara takılan bir donanım parçasını" ifade etmektedir (2004:147). Stevenson, dongle kullanımının kurgu programları ile olan ilişkisini örneklendirirken "Avid gibi pahalı video düzenleme yazılım sistemleriyle yazılımı kullanarak bilgisayara bağlanması gereken bir USB dongle'ına sahip olmak gerekirken, daha az pahalı olan yazılım sistemleri yazılımınızı internet üzerinden etkinleştirmeniz gerektiğine" vurgu yapmaktadır (2006: 286). Dongle kullanımı konusunu ironik bir biçimde değerlendiren Bardosh ise şunları söylemektedir:

> *"Dingle, dangle, dongle! Hangi dongle nedir? Bir dongle, bir USB atlama sürücüsü gibi görünen küçük bir alettir. Bir yazılım uygulaması dongle güvenliği tarafından korunduğunda, yazılımı başlatmadan önce cihazınızdaki bir USB bağlantı noktasına dongle takılmalıdır. Dongle olmadan başlatmaya çalışırsanız, dongle'ı takmanızı söyleyen bir hata mesajı alırsınız. Bir dongle gerektiren bir yazılım uygulamanız varsa, bilgisayarınıza takılı değilken güvenli bir yerde sakladığınızdan emin olun. Sloganı hatırla, "Dongle'ını kaybetme!" (2007: 231).*

Aşağıda yer alan eski tip dongle örneklerine bakıldığında yeni teknolojilerin boyut, içerik, nitelik ve

kapasite gibi değişimler dışında kullanım amacıyla da değişkenlik geçirdiği anlaşılmaktadır.

Fotoğraf 1: Dongle Çeşitleri, (https://medium.com)

Son yıllarda tanıtılan evrensel seri veri yolu (USB) dongle, kısa menzilli kablosuz iletişim için bilgisayarla ilişkili cihazlarda çok popüler hale gelmektedir. USB dongle uygulamaları için tek bantlı antenler üzerinde önemli miktarda çalışma yapılmıştır. Birden fazla hizmet için artan talep, çok dilli anten ile entegre bir USB dongle sahip olmak çekici olmaktadır (Sun vd., 2013: 307). Dongle uzun süredir yazılım koruması için endüstri tarafından kullanılmaktadır. Bunlar, bilgisayara takılı donanım anahtarlarıdır; bunlar, dongle'larla gelen programların yürütülemez. en büyük dezavantajı, her dongle özellikli yazılımın genellikle farklı bir dongle gerektirmesidir. Dahası, koruma genellikle dongle'lar ve programları arasındaki iletişim trafiği engellenip değiştirilebildiği için baypas edilebilmektedir (Chang ve Atallah, 2002: 163). Bir dongle kullanmak, bilgisayar cihazlarıyla sağlam ve güvenli bir şekilde entegre edilmediğinden tanımlanan problemi çözmemektedir. Alawneh ve Abbadi bu konuda kullanım yapısına odaklanmaktadır. Temel düşünce "Dongle ürünlerinin pratik değil ve pahalı olduğu eleştirisidir. Bunun nedeni, bir yazılım ürününe özgü bir yazılım olmasından kaynaklanmasıdır. Zira aygıt, tipik olarak, her biri bir yazılımın çalıştığı her zaman bir aygıt bağlantı noktasına bağlanacak belirli bir dongle gerektiren birden fazla yazılım ürününe sahiptir". (2008:512). Aslında birden farklı donanım için birden fazla dongle kullanılması halinde ortam ve kullanım işlevinin yitirilebileceği kaygısı, birden fazla programın aynı anda kullanılması halinde performans olarak nasıl sonuçlar vereceğini de kapsamaktadır. Varian vd ise dongle kullanımının kullanıcı üzerindeki ruhsal etkisini dile getirirken şunları söylemektedir:

> *"Dongle denen bir şey var, insanların PC'yi bilgisayarın arkasına iliştirdiği ve yazılımı çalıştırmak için küçük bir donanım parçası var. İnsanlar bundan nefret ediyordu. Artık kimse bir dongle kullanmıyor. Evet, pazar tarafından nefret edilen ve pazara yanıt olarak piyasadan çıkarılmış olan DRM'ler var" (2003: 750).*

Genellikle USB veya seri bağlantı noktasına takılan bir donanım kilidi, yalnızca yetkili kullanıcıların lisanslı yazılım uygulamalarını kullanabilmelerini sağlamak için özel olarak tasarlanmış donanım tabanlı bir yazılım koruma tekniğidir. Normalde, dongle'lar pahalı uygulamalarla kullanılır. Khan vd., bu tür uygulamaları, "kullanılmaya başlatıldıklarında bağlantı noktalarında dongle olup olmadığını kontrol edilen ve şu anda, ticari yazılımı korsanlıktan korumak için en güvenilir tekniklerden biri" (2015: 288), olduğunu belirterek aslında tüm kullanım dezavantajlarına karşın güvenlik ve çalışma yapısı itibariyle bu eksikliği kapatabileceği tezini işlemektedirler.

Yazılım koruması, bilgisayar bilimi ve bilgi teknolojisi alanlarında önemli bir rol oynamaktadır. İçeriği işleyen ve yazılımı koruyan temel içerik üzerindeki gereksinimlerini korumaktadır. Sasirekha ve Hemalatha, "Kopya korumasının tersine mühendislik ve yazılım kurcalamaya karşı aynı korumalara ihtiyaç duyduğu seviyeye kadar bir başka yazılım koruma biçimi" olduğunu vurgulamaktadır (2012: 53). Bu nedenle dongle gibi önemli bir modelin her zaman tercih konusu olacağı ve gelişim göstereceği anlaşılmaktadır. Burada yazılım kilidinin mimari yapısı ile sektörel örneklerinin giderek artan ihtiyaçlar karşısındaki çözümleyici rollerinin niteliği aslında içeriğin korunmasını sağlamaktadır. Kullanıcı, bu bağlamda değerlendirildiğinde tüm eksikliklere karşın ihtiyaçlarını karşılayan sistemleri kullanmaya ve sadakat göstermeye eğilimlidir.

2. Temel Mimari Yapı Ve Örnekler

Yazılımın yasal olmayan kullanımıyla mücadelede birçok yöntem vardır ve bunlar iki ana gruba ayrılabilir: yazılım tabanlı koruma ve donanım tabanlı koruma. Popüler donanım tabanlı yazılım kopya koruma yöntemlerinden biri, "dongle" adı verilen özel bir donanıma dayanmaktadır. Dongle, uygulamaların yasadışı olarak orijinal kopyadan çoğaltılmasını engelleyen USB flash kalem gibi genellikle küçük bir USB, RS232 veya LPT arabirim aygıtıdır (http://www.bastioninfotech.com). Dongle tipleri temel olarak yazılım koruması çözümü olmak dışında veri yönetimi, gerçek zaman saati ve kimlik doğrulama gibi farklı amaçlara hizmet etmek üzere tasarlanmaktadır (Bastion, 2017:4). Dongle sistemi daha önce incelemesini yaptığımız koruma tekniklerinde görüldüğü gibi yazılımın saklanması ve bilgilere ulaşılamaması için uygulanan algoritma tabanlı kod sistemleri dışında farklı özellikleri mimari açıdan içinde barındırmaktadır. Bunlar özellikle usb dongle modelinde görüldüğü üzere sisteme kurulmuş bir programın tümleşik bir ortam haline getirilerek çalışması esasına dayanmasıdır.

Yazılım konusu bu yönüyle paranın yazı ve turası gibi birbirinden ayrılmayan, ayrı olarak düşünüldüğünde ise tek başına anlam ifade etmeyen biçime dönüşmektedir. Burada yazılım bilgisayar ortamında program dosyası olarak bulunurken donanımsal açıdan dongle aygıtı ile aktif hale gelmektedir. Dolayısıyla kullanıcı gerek programı çalıştırabilme ve kullanabilme becerisine gerekse söz konusu yazılımın uzantısı olarak dongle kullanma becerisine sahip olmakla yükümlüdür. Kullanıcının her iki ortama hakim olması ve bunu disipline etmesi yazılım korsanlığının reddedilerek kişisel bir eşyaya dönüşmesidir. Aslında burada amaçlanan, sadece satın aldığı ürünü kullanan ve çıktı üzerinden maksimum faydayı kendine ilke edinen kullanıcı profili oluşturabilmektir. Bu modeldeki bireyin herhangi bir risk alması ya da yazılım firmasına zarar verecek bir çalışma içerisine girmesine gerek kalmamaktadır. Bu mantığın genele uyarlanmış hali diğer tüm kullanıcıların da dahil olduğu birbirinden bağımsız eylem ağlarının sağlanabilmesidir.

Programı kendi için kullanan ve dijital ortamda bencilleştirilen birey kollektif bir yasadışı uygulamasının parçası olmaktan giderek uzaklaşmaktadır. Bu programların ilgili yazılımın web sitesini kullanarak ID oluşturması ve diğer kişilerle paylaşmaktan kaçınan bir yönelim göstermesi, kullanıcının dijital adımının kolaylıkla hesap edilebilmesini ve diğer kullanıcı profillerinin çoklu bir modelde programa sızmalarını engelleyebilecektir.

Aşağıda yer alan koruma tekniklerinin eleştirel analizini içeren tablo 2 dikkatle incelendiğinde tüm koruma yöntemlerinin kullanıcının tavrı ve ahlaki yapısıyla ilişkili olduğunu göstermektedir. Her bir koruma tekniğinin farklı uygulama biçimleri bulunsa da özü itibariyle her zaman küçük bir risk oluşturabilecek açığa sahip olabilceği gerçediğidir. Zira "hiçbir yazılım kusursuz değildir".

Table 1. Critical analysis of software protection techniques.

Ref #	Technique/ Methodology	Strengths	Scope/Limitation
Sasirekha and Hemalatha [26]	Cryptography based software protection technique.	Thwarts static analysis and static tempering of the program.	Is not effective against dynamic analysis when the original code becomes available in the memory.
Guoyuan *et al.* [12]	Shareware protection schemes to protect software from anti-debugging by thwarting the possibilities of reverse engineering.	-	The scope is limited as it is based on melting the protection solution into the development lifecycle and it only addresses the anti-revensing methodology for sharewares.
Jamkhedkar and Heileman [13]	Open layered framework that incorporates various interoperating technologies, for developing DRM systems.	It ensures a strong mechanism for security of the digital contents.	It is a complex solution as for each middleware service of framework it requires implementing different types of security controls and business logic.
Zhang [29]	Suggests employing effective usage control technologies in DRM systems to facilitate user to access, download, transfer and share protected or copyrighted contents.	It employs usage control mechanisms and models that allows RELs, authentication and authorization management security models and secure utilization of end-user digital devices.	-
Mafia and Pimentel [19]	Software protection scheme based on tamperproof processor by exploiting smart card technology. It is based on asymmetric cryptosystem in which private key is embedded on the smart card.	It is a robust technique against attacks as it can bypass code substitution and threats to license management protocols.	The limitation of the scheme is that asymmetric cryptosystem is computationally expensive which results in performance degradation.
Zhang [28]	Watermarking technique which employs hash function that contains watermark signature. Hash function extracts the embedded watermark at the run-time.	The strength of this technique is that watermark is calculated dynamically through hash function.	-
Ghosh *et al.* [11]	Employs obfuscation in the forms of encryption and checksum guards through process-level virtualization.	It is an effective technique for protecting the software from unauthorized use with enhanced security.	Scope of the proposed technique is limited as it requires periodic discarding of the code from the memory which results in decrypting the original code again and again for a single execution resulting in performance degradation. The specialized VM executable is also limitation for each software. It also doesn't suggest mechanism to protect the VM software itself.
Kimball [17]	Emulation-based software protection techniques to protect software from reverse engineering by page-granularity code signing and encrypted code execution within emulators (sandbox).	It minimizes the chances of reverse engineering as it requires sandbox or an emulator to execute the software.	Is not efficient because the encrypted code needs to be decrypted before execution which will cause performance degradation.
Erlingsson *et al.* [8]	A software guards model named XFI is proposed to protect user-mode and kernel-mode address spaces.	XFI supports low-level architectural features (e.g., language-based protection) which facilitates safe execution of the code.	The proposed technique has overhead of watching both the user-mode and kernel-mode address spaces with administrative privileges.
Zhu *et al.* [31]	Software Watermarking techniques supplemented with watermarking attack models, its taxonomy and algorithms.	Four types of watermark models ensure security, ownership, user authenticity and unauthorized uses of software.	-
Lin *et al.* [18]	Suggests Hardware virtualization for self-protection against anti-debugging/reverse engineering.	-	-
Dedic *et al.* [7]	Program-transformation algorithm by simulating hacker's steps to reversing a program in the form of a flow graph.	This proposed technique is useful for DRM systems.	The program-transformation algorithm may suffer from exponential or polynomial time complexity which limits scope of the proposed technique.
Birrer *et al.* [2]	Suggests adding a metamorphic layer of protection in the form of program fragmentation on top of the traditional obfuscation techniques.	The proposed technique adds further complexity to the already obfuscated code making reverse engineering more difficult.	-
Min *et .al.* [22]	Methodology uses encryption of the MAC address and generates a unique registration code for each installation of the software.	-	The proposed methodology can only work within an enterprise network where machines are interconnected, and will fail if the same software is replicated on other isolated networks or standalone machines.
Cappaert *et al.* [4]	The technique employs various chunks of the program codes to encrypt and decrypt the other segments of the code.	The proposed technique is an effective safeguard against both static and dynamic analyses of software code as it employs code dependencies.	-

Tablo 1: Yazılım Koruma Tekniklerinin Eleştirel Analizi (KHAN vd., 2015:295)

Keylok Firması Dongle sisteminin nitelikli çözüm ortaklarından birisi olarak ürettiği sistemin yazılım mimarisi olarak genel içeriği başlıklar halinde şu şekilde sıralamaktadır:

- Tescilli Şifreleme
- Anti-Debugger (Kod Ayıklamasını Engelleme)
- Code Vault (Kritik Kod Saklanması)
- Şifreli Algoritmalar (2018b, https://www.keylok.com).

Keylok Firması Dongle sistemi üzerinde yazılımsal olarak getirdiği çözüm yollarını ise şu şekilde maddeleştirmektedir:

- Yazılım Lisans Koruması
- Yazılım Para Kazanma
- Gömülü IoT[5]
- Markalı Dongle Kullanımı
- CodeVault
- Sanal Sunucu Ortamları
- Otomatik Yazılım Koruması
- PDF ve PPT Dosya Koruması
- Özelleştirilmiş Çözümler (2018, https://www.keylok.com).
-

Dongle konusunda çözümler üreten yerli bir yazılımı kullanıcılara sunan O-Key firması ise yazılım mimarisiyle kullanıcılara getireceği katkıları şöyle sıralamaktadır:

[5] **IoT** Internet of Things/Nennelerin İnterneti: (çeşitli haberleşme protokolleri sayesinde birbirleri ile haberleşen ve birbirine bağlanarak, bilgi paylaşarak akıllı bir ağ oluşturmuş cihazlar sistemini ifade etmektedir.

- Mikroişlemci Tabanlı Mimari
- Gömülü Şifreleme Algoritması
- Hat Güvenliği Algoritması
- Şifreli Gömülü Hafıza
- Okuma/Yazma Modu
- PIN Kontrolü
- Her Alana Farklı Şifre
- Çiftli Kabuklama
- Sürücüsüz USB Desteği
- Güvenli Uzak Güncelleme
- Kullanım Adedi Sınırlaması
- Kredi Sınırlaması
- Ağ Üzerinden Koruma
- Geniş Programlama Dili Desteği
- İşletim Sistemi Desteği
- İzinsiz Haberleşme Tespitinde Kendini Kilitleyebilme
- Hızlı Uygulama ve Teknik Destek
- Firmaya Özel Anahtar ve Tekrarsız Seri Numarası
- İmzalama ve Doğrulama Desteği
- Özetleme Algoritmaları Desteği
- Kendi Şifreleme Algoritmanızı İşletebilme Desteği
- Ortalama İşlem Süresi = 30ms!
- Kimliğinizi Yansıtma
- Aynı Kod İle İster Lokal İster Ağ Üzerinden Çalışma (O-Key, 2018, http://www.okeydongle.com).

3. Sistem Özellikleri Ve Uygulama Alanları

Dongle koruma tekniği durumunda, kullanıcı Dongle'da saklanan bir özel sertifika ile tanımlanır. Dongle tipik olarak bir Smart Token olduğu için sertifika deposu ve yönetimi bakımından oldukça kolay bir yapıya sahiptir.

Dongle'ların kullanımlarını kısıtlayan bazı dezavantajları vardır. Kullanıcılar genellikle çeşitli nedenlerle dongle sevmemektedir. Chen'e göre dongle kullanımı konusundaki avantaj ve dezavantajlar şunlardır:

Avantajları

1. Değerli kaynak kodlarını, hassas metin bilgilerini ve veri dosyalarını korur;
2. Diğer donglelarla etkileşimi azaltmaya yardımcı olan tamamen şeffaftır.

Dezavantajları

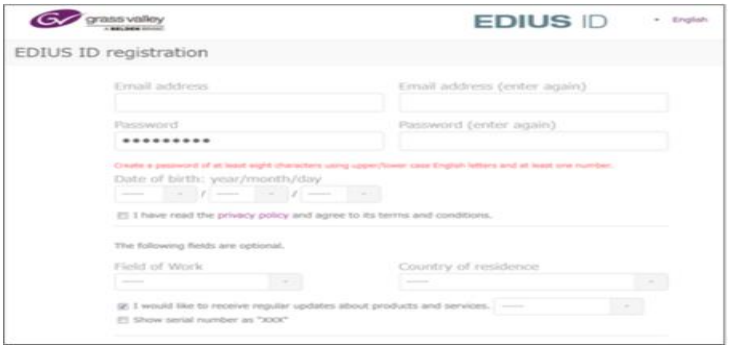
1. Özel donanım sürücüsü gerektirdiğinden kurulumu ve kullanımı zahmetli olabilir;
2. Ek üretim gideri nedeniyle birçok yazılım şirketi için bir seçenek değildir;
3. Her bir müşteriye gönderilmesi gerektiğinden, yazılımların internet tabanlı dağıtımını kolaylaştırmaz;
4. Çoğu yazılım uygulaması için gerçekleşebilen bir kopya koruma mekanizması değildir (Chen, 2001: 5).

Yeni medya sisteminin bireye içeriğe katılma ve etkinlik üretme katkısı sağlamasıyla başta video ve ses olmak üzere birçok kitle iletişim aracının ve sosyal paylaşım ağlarının yoğun bir şekilde kullanıldığı gözlemlenmektedir. Yazılım değeri açısından zengin markaların bulunduğu yapım sektörü içinde özellikle kurgu programları gibi görsel ve işitsel anlatıya odaklanan programların kullanım oranı da paralel olarak artmaktadır. EDIUS, Steinberg, AVID ve ADOBE gibi şirketlerin liderliğindeki programların nitelikli üretim yollarına ulaşmak ve diğer markalar karşısında daha fazla tercih edilen bir konumda olabilmeleri için her sürüm içinde farklı menü

özellikleri ve buna bağlı paneller oluşturulduğu görülmektedir.

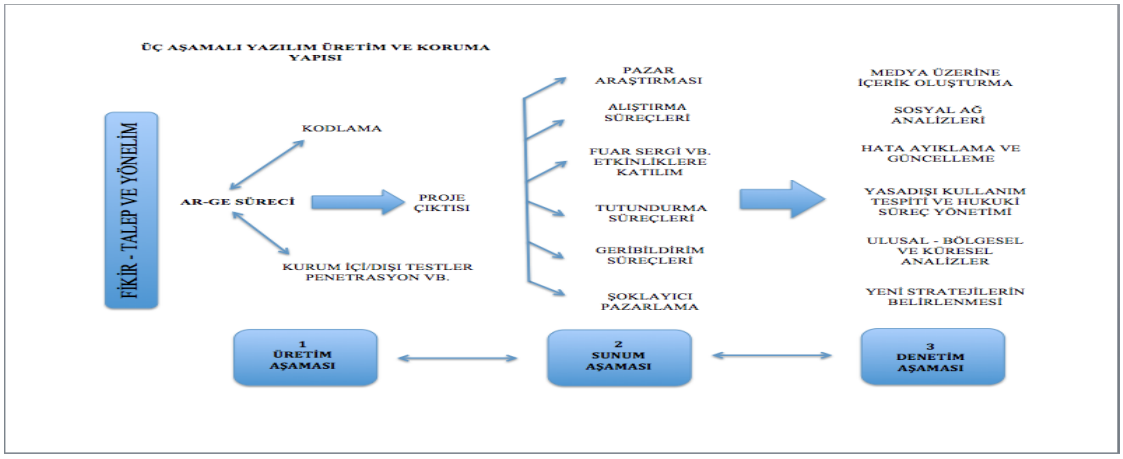
Bu markaların nitelikli hizmet verebilmesi için bünyelerinde çalışan birçok yazılım mühendisi gerek yazılımın kalitesini ve geniş çevrelere satılmasını gerekse olası sistem açıklarının neler olduğunu saptayarak yasadışı kullanımının önlenmesine yönelik çalışmalar ile ilgilenmektedir. Örneğin EDIUS firması piyasaya tutunma sürecini tamamladıktan sonra Dongle kullanımına geçmiş ve ilerleyen dönemlerde ise 6.5 sürümünden itibaren Flexera adlı (Belden Brand, 2015) çevrimiçi kimlik sistemine geçerek işlemlerini yönetmeye başlamıştır. Şirket aynı zamanda dongle aygıtının kaybolması halinde kullanılmak üzere bazı ön çalışmalar da yapabilmiştir. Bu, yazılım 2007 yılında hazırlanan tanıtım rehberinde dongle aletinin kaybedilmesi halinde neler olabileceği şeklindeki kurmaca soruya şu yanıtı vermektedir.

> "Lisansınızı kaybetme eşdeğeri olacaktır, ama güvenlik kilidini dongle'dan PC'nize aktarmak mümkündür, böylece cihaz kilidi güvenli bir yerde saklanabilir. Daha sonra bir disk çökmesine sahipseniz ve EDIUS'u yeniden yüklemeniz gerekiyorsa, dongle, hiçbir ek ücret ödemeden yeniden serileştirme için Grass Valley'e döndürülebilir (2007:16).



Grafik 2: EDIUS ID Kayıtlanma Sistemi, (Grass Walley a Belden Brand, 2015:2)

Medya, Bilişim ve Telekomünikasyon kavramları birbiriyle uyumlu olarak büyüme davranışı gösteren disiplinlerdir. Üreten-Tüketen, Satan-Satın alan ve Kabul-Red durumlarına göre karşılıklılık ilkesini içinde barındıran bir işleyiş yapısı bulunmaktadır. Bu işleyiş, "Garanti, Mesafeli Satış Sözleşmesi, Birden Fazla Kullanıcı Hakkı ve Ücretsiz Güncelleme Desteği" gibi güven esasına dayanan ticari haklar çerçevesinde düzenlenen faaliyetleri kapsamaktadır. Bu ilişkinin taraflarından herhangi birinin hukuku bozması halinde ilgili süreçler baz alınarak dengenin korunması hedeflenmektedir. Bununla birlikte bilginin, ar-ge faaliyetlerinin ve üretim hizmetlerinin ekonomik nitelikler arz etmesiyle konu; fikri ve sınai haklar başta olmak üzere, marka ve patent hakkı, telif hakkı ve sayısal haklar yönetimine göre yeniden biçimlenmektedir. Yazılım Üretim ve Koruma Yapısı tarafımızca üç aşamalı olarak gerçekleşen bir model olarak görülmektedir. Bu aşamalar ve temel ögeleri şunlardır:



Şekil 2: Üç Aşamalı Yazılım Üretim ve Koruma Yapısı

Bilgiye kolay erişmenin mevcut şartları içerisinde aidiyetlik ve telif içeren tüm ürünlere "bedel ödemeksizin" kullanma arzusu, "fikir hırsızlığından, yasadışı kullanıma, izinsiz kopyalamadan programların kırılmasına" kadar geniş bir yelpazede yeni sorunlar ortaya çıkarabilmektedir. Yazılım alanının ana ögesi olan kullanıcının tipoloji ve alana olan yönelim düzeyi ana işleyişin nasıl şekillendiğini de gösterebilmektedir.

YAZILIM KULLANICI TİPOLOJİSİ VE YÖNSEL ANALİZ					
SEVİYE	KULLANICI TİPİ	YÖNSEL YAPI	KULLANIM DÜZEYİ	EĞİLİM YAPISI	KOD DURUMU
1	BİRİNCİL	ÇİFT YÖNLÜ	YÜKSEK	SATIN ALMA VE KULLANMA	POZİTİF
2	İKİNCİL	TEK YÖNLÜ	NORMAL	YASADIŞI EDİNİM VE KULLANMA	POZİTİF
3	DOLAYLI	YÖNSÜZ	DÜŞÜK	YASAL VE YASADIŞI KULLANIM	DAHA ÇOK NÖTR KISMEN NEGATİF

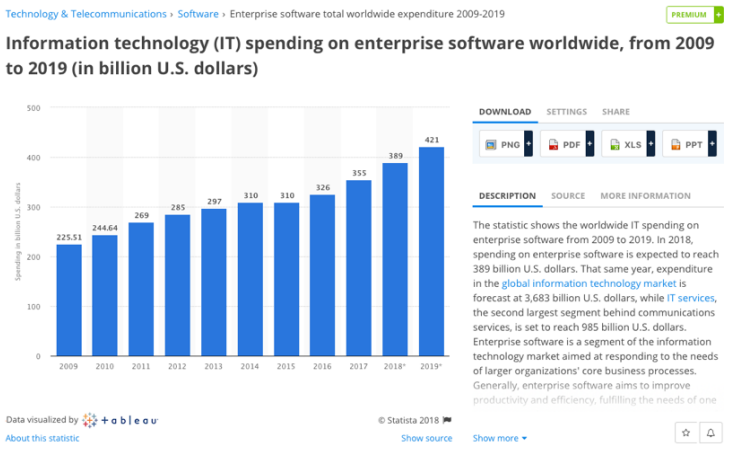
Tablo 2: Yazılım Kullanıcı Tipolojisi ve Yönsel Analiz

Yukarıdaki bilgiler ışığında kullanıcının yazılım konusunda yasal ya da yasadışı davranışlarını belirleyen sosyo-ekonomik ve ruhsal etkilerin dışındaki genel seviyesinin "tip, yapı, düzey, eğilim ve kod" açısından değişkenliklerinde kişisel yapısının da etkili olduğu ve bu alanlarının düzeylerinde ise birden fazla faktörün olduğu saptanmaktadır.

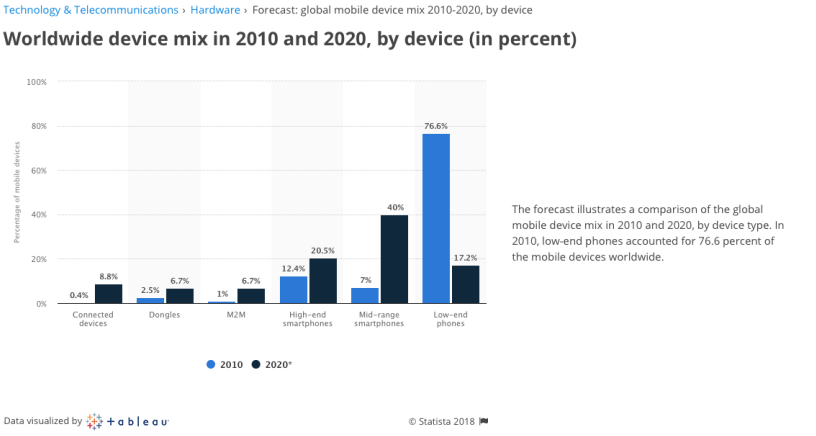
4. Küresel Görünüm ile Dongle Pazar Analizi

Yazılım alanı, bilginin paraya nasıl dönüştüğünü ve bu sistemin sağlıklı işleyebilmesi için hangi önlemler alınması gerektiğini gösteren bazı ipuçları içermektedir. Yeni medya ortamında tüketici karakteristiğine sahip daha fazla kullanıcının etkin katılım, bireysel odaklı içerik, içeriğe katılma ve diğer kişilerle hukuki platformlarda uygun biçimde paylaşımda bulunması gerçeği mevcuttur. Bu sistemin kullanıcıdan bir başka beklentisi ise üretilen yazılımlara ilgi duyması, bunları satın alması ve test ederek geribildirimde bulunabilmesi şeklindedir.

Yazılım firmalarının her yıl yaptığı ar-ge faaliyetleri ve sundukları ürün hizmetine karşılık yüksek gelirler elde edebilmesi için kullanıcıların telif konusuna hassasiyet göstermesi ve üretici haklarına saygı göstermesi talebinde bulunmaktadır. Dünya genelinde, bilgi teknolojilerinin yazılım harcamalarının 2009-2019 dönemi içinde geçirdiği değişimi inceleyen "STATISTA" 2009 yılında 25 milyar dolar seviyesindeki yazılım geliri oranının 2019 döneminde yaklaşık iki katına ulaşarak 421 milyar dolar seviyesine uluşabileceğini öngörmektedir.



Grafik 3: Dünya Genelinde Bilgi Teknolojileri Alanında Yazılım Harcaması 2009- 2019. https://www.statista.com/statistics/203428/total-enterprise-software-revenue-forecast/ (05.01.2018 Tarihinde Erişildi).



Grafik 4: Dünya Genelinde Mobil Aygıt Ürünlerinin 2009-2020 Dönemi Analizi https://www.statista.com/statistics/219008/global-mobile-device-mix-since-2010-by-device/ (10.01.2018 Tarihinde Erişildi).

Yukarıda yer alan grafik 4 dünya genelinde aygıt kullanımındaki yıllara göre değişimi incelemektedir. Bu grafik içinde "dongle" aygıtlarının değişim parametrelerine bakıldığında 2009-2019 arası on yıllık dönemde yaklaşık % 300 oranında bir gelişme göstererek bu alandaki ticari gelişim ortamı ile diğer aygıtlardaki gelişime paralel bir seyirde gerçekleşeceğini öngörebilmek mümkün olmaktadır.

Sonuç

Tüm sistemlerin kendi içinde eksik ve kusurlu yapı içermesi nedeniyle yüksek bütçeli yazılım hizmetlerinin satın alınması yerine yazılım korsanlığı (İng. Software Piracy) yoluyla elde edilmesi çabası, kopya koruma tekniklerinin ortaya çıkmasına zemin hazırlamaktadır. Üretici (Yazılım Sahibi/Firma), ürün üzerindeki mülkiyetini bilişim düzeyinde de korumak üzere "şifreleme, kodlama, şaşırtma, gizleme, filigran, ters mühendislik ve algoritmik eylemler" ile çözümler aramaktadır. Yazılım korsanlığı yapan kişi ya da kişiler ise tersine mühendislik yöntemleri ile yazılım kodlama sistemine sızmaya çalışarak süreci kendi lehlerine çevirmeye çalışmaktadır. Bilginin üretilmesinden çalınmasına kadar ilgili süreçte yaşananları ve çözüm yollarını "dongle" örneği üzerinden ele aldığımız yazılım konusu, her iki tarafın yeni hamleleri ile değişkenlik göstermektedir.

Yazılım firmalarının, korsanlık ve yasadışı kullanım konusundaki eylemler üzerindeki temel ilkelerini iki ana başlıkta toplayabiliriz.

a. *Firmaya Bağlı Etkenler:* Yeni medya, sektörlerin birbirleriyle iç içe geçtiği pazarlama kültürünün doğal bir gereksinimi olan "alıştırma" eyleminin gerçekleşmesi adına piyasaya sürdüğü bir ürünün lansmanını yapmaktadır. Bu lansman, pazarda yer tutabilmek adına ürünün ücretsiz sürümünü piyasaya sürmesi ve yasadışı yollarla kırılması karşısında pasif bir yol izlenmesine kadar geniş bir ticari yapıyı içermektedir. Yazılım alanına giren ürünün, diğer firmaların pazardaki mevcut ürünlerine karşı duyulan ilgi ve pazar paylarını azaltabilmek adına yaptığı dolaylı bir pazarlamayı içermektedir. Kullanıcı yeni bir ürüne olan ilgisini mevcut menü ve uygulamaların getirdiği nitelikli çözümler ile canlı tutabilmektedir. Dolayısıyla her yeni ve ücretsiz olanın kabul edilmesinden ziyade çözümleyici bir niteliğe sahip olması koşulu beklenmektedir. Yazılım firmasının pazarın dengesini bozmaya yönelik hamlesi, makyevelist bir kapital düzenin en belirgin özelliğidir. Piyasaya giren ürünün yeni sürümleri ve güncellemeleri ile "tutundurma" işlemlerinin başarılı olmasıyla söz konusu program(lar)a alışan işleyiş sayesinde "ihtiyaç yaratma" planı başarıya ulaşmış ve tercih edilen bir konuma da erişebilmiştir. Koruma yöntemlerinin, büyük oranda ticari kayıpları önleyecek kadar zorunlu bir tercih biçiminde kullanıcılara sunulması dongle sisteminin genel yönetim mantığını açıklamaktadır.

b. *Firmadan Bağımsız Etkenler:* Bilgisayar ve bilişim sektörlerinde internet ortamı üzerinden "verinin elde edilmesi, kullanılması ve paylaşılması" sıklıkla tercih edilmektedir. Ekonomik değeri yüksek olan bazı programların yasadışı kurulumu, bu programların tanıtım ve eğitim videoları ile içerik desteği sağlanması, birey eksenli medya kültürünün en bilinen uygulama örnekleridir. Yazılım şirketlerinin içeriksel ve türsel açıdan yarattığı zenginliğe karşın, kullanıcı grubunun ekonomik çıktı değerinin aynı düzeyde olmaması (burada psikolojik gerekçelerle yapılan yasadışı eylemler de bulunmaktadır) tüm üretim biçiminin illegal kullanımına ya da diğer kullanıcılara paylaşılmasına neden olabilmektedir. Bilgiye erişmek, eşzamanlı kullanıma sahip olmak, diğer kullanıcılarla bütünleşmek ve ortak verinin parçası olmak adına programların kırılması (İng. Crack) hamleleri yapılmaktadır. Sistem bu bağlamda gerçek kullanıcılar ile dolaylı kullanıcılar şeklinde işlemektedir.

Yukarıda belirtilen her iki sistemin ortak özelliği, koruma önlemlerinin yetersiz olmasına karşılık büyüme özelliklerinin olabilmesidir. Yasal kullanıcılar ile yazılım satışı sektörünün gelişimi devam ederken, yasadışı kullanıcıların yazılım üretmek yerine mevcut sistemin ürünlerini kullanma davranışları gösterdiği görülmektedir. Her iki kullanıcı tipinin birbirinden beslendikleri hatta gelişimlerine katkıda bulundukları anlaşılmaktadır.

Sonuç olarak, yasadışı kullanımın tamamen ortadan kaldırılamayacağı sadece sınırlandırılabileceği ve kısmi engellemeler oluşturulacağı düşünülmektedir. Yakın gelecekte bilişim sektörünün hızlı gelişim sağlamasıyla, özellikle "kodlama ve yapay zeka" gibi alanların ciddi bir talep görerek küçük yaştaki yasal ve yasadışı kullanıcı yaş gruplarının sisteme dahil olacağı da öngörülmektedir. Bu durumda yasal olan ve olmayan her türlü kullanım için serbest piyasa kurallarının yeniden düzenlenmesine bağlı olarak hukuki düzlemde de karşılık bulması beklenmektedir.

Kaynakça

Alawneh, M. and ABBADI, I.M., (2008), Software Licence Protection and Management for Organisations, 2008, in IFIP International Federation for Information Processing, Volume 278; Proceedings of the IFIP TC 11 23rd International Information Security Conference; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), 509–523pp.

Arnold, Michael (2000) AUDIO WATERMARKING: FEATURES, APPLICATIONS AND ALGORITHMS, Multimedia and Expo, 2000. ICME 2000. IEEE International Conference on, Volume: 2, 1-3pp. https://www.researchgate.net/publication/3864166_Audio_watermarking_features_applications_and_algorithms (13.02.2018 Tarihinde Erişildi.)

Atallah, Mikhail Jibrayil (2005) "A Survey of Watermarking Techniques for Non-media Digital Objects," in Proceedings of the 3rd Australasian Information Security Workshop, Australia, 73pp.

Bahaa-Eldin, Ayman Mohammad & SOBH, Mohamed A.A. (2014) A comprehensive Software Copy Protection and Digital Rights Management platform, Electrical Engineering, Shams Engineering Journal (2014) 5, 703-720pp.

Bardosh, Karl (2007) The Complete Idiot's Guide to Digital Video, New York, Published by the Penguin Group.

Bastion Infotech Pvt. Ltd. Report (2017) "Hardware Dongle For License Keys, AuthGuru Hardware Keys For Software and Data Protection", http://www.bastioninfotech.com/india/products/hardware-keys/authguru-hardware-donglekeys.jsp (04.11.2017 Tarihinde Erişildi.)

Boneh, D., & Sahai, A. & Waters, B. (2012) Functional Encryption: A New Vision for Public-Key Cryptography. Communications of the ACM, 55(11), 56–64, doi: 10.1021/ac60289a702.

Campidoglio M, & Frattolillo F. & Landolfi F. (2009) The Copyright Protection Problem, Challenges and Suggestions, In: Internet and web applications and services conference, ICIW'09; 2009.

Ceccato, Mariano & Di Penta, & Massimiliano NAGRA, & FALCARIN, Jasvir Paolo & Ricca, Filippo & Torchiano, Marco & Tonella, Paolo (2009) "The Effectiveness of Source Code Obfuscation: an Experimental Assessment, Program Comprehension", 2009. ICPC'09. IEEE 17th International Conference on, 17/19 May 2009, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5090041 (21.12.2017 Tarihinde Erişildi.)

Cert Report (2014) "Code Obfuscation," A Cert-UK Publication Copyright. https://www.ncsc.gov.uk/content/files/protected_files/guidance_files/Code-obfuscation.pdf. (16.02.2018 Tarihinde Erişildi.)

Chang Hoi & Atallah, Mikhail J. (2002) Protecting Software Code by Guards, T. Sander (Ed.): DRM 2001, LNCS 2320,160–175pp, Springer-Verlag Berlin Heidelberg 2002.

Chen, Min (2001) Software Product Protection, HUT TML 2001 T-110.501 Seminar on Network Security, 1-12pp.

Collberg, Christian & THOMBORSON, Clark (2002) Watermarking, tamper-proofing, and obfuscation – Tools for software protection, IEEE Trans. on Software Engineering, Vol. 28, No. 8, pp. 735-746pp.

Collberg, Christian. & Thomborson, Clark. (1999) "Software watermarking: Models and dynamic embeddings," in Proceedings of Symposium on Principles of Programming Languages, POPL'99, 1999, 311–324 pp.

Collberg, Christian & THOMBORSON, Clark & LOW, Douglas (1998) "On the limits of software watermarking" in Technical Report #164, Department of Computer Science, The University of Auckland.

Conner, M. & ARMITAGE C. (1998) 'Extending the Theory of Planned Behavior: A Review and Avenues for Further Research', fournal of Applied Social Psychology 28(15), 1429-1464pp.

Cox, Ingemar J. & Linnartz, Jean-Paul M. G. (1998) "Some General Methods for Tampering with Watermarks", IEEE Journal On Selected Areas in Communications, Vol. 16, No. 4, May 1998, 587-593pp.

Cox, Ingemar J. & Kilian, Joe & Leighton, F. Thomson & Shamoon, Talal (1996) "A secure, robust watermark for multimedia," in LNCS 1174, 1996, 317–333pp.

Davidson Robert L. & Myhrvold, Nathan (1996) "Method and system for generating and auditing a signature for a computer program," US Patent, vol. 5,559,884, 1996.

Djekic, Petar & Loebbecke, Claudia (2007). Preventing application software piracy: An empirical investigation of technical copy protections. The Journal of Strategic Information Systems, 16(2), 173–186pp. http://doi.org/10.1016/j.jsis.2007.05.005.

Edius, Grass Walley a Belden Brand (2015) Overview of the EDIUS ID Activation System http://www.en.ediusworld.com/docs/Application_Notes/professional/edius/GVB-1-0525A-EN-AN_EDIUS_Activation_WW.pdf (05.01.2018 Tarihinde Erişildi.)

Edius Nle (2007) Software Reviewer's Guide, July 2007, http://www.ediushd.com.ar/EDIUS_4.5_Reviewers_Guide[1].pdf (05.01.2018 Tarihinde Erişildi.)

Fang, Hui & Wu, Yongdong & Wang, Shuhong & Huang, Yin (2011) Multi-stage Binary Code Obfuscation Using Improved Virtual Machine, X. Lai, J. Zhou, and H. Li (Eds.): ISC 2011, LNCS 7001, 168–181pp, Springer-Verlag Berlin, Heidelberg.

Glass, R. & Wood W. (1996) 'Situational Determinants of Software Piracy: An Equity Theory Perspective', Journal of Business Ethics 15, 1189-1198pp.

Gopal, R. & Sanders L. (1997) 'Preventive and Deterrent Controls for Software Piracy', Journal of Management Information Systems 13(4), 29-47pp.

Harran, Martin & Mckelvey, Nigel & Curran, Kevin & Subaginy, Nadarajah (2015) Software Piracy, Category: Cyber and Network Security, IGI Global, 726-731pp.

Hunt, S. & Vitell S. (1986) 'A General Theory of Marketing Ethics', Journal of Micromarketing 6, 5-16pp.

Jakubowski, Mariusz & Chit H. & Saw, W. (Nick) & Venkatesan, Ramarathnam (2009) Iterated Transformations And Quantitative Metrics For Software Protection, International Conference on Security and Cryptography (SECRYPT 2009), Inproceedings, https://www.microsoft.com/enus/research/publication/iterated-transformations-and-quantitative-metrics-for-software-protection/ (13.01.2018 Tarihinde Erişildi)

Kamela, Ibrahim & Albluwib, Qutaiba (2009) A robust software watermarking for copyright protection, doi:10.1016/j.cose.2009.01.007, Computers & Security 28 (2009) 395–409pp. Elsevier Ltd.

Karpakavalli,B. & Arunadevi R. (2017) Software Piracy Protection System International Journal of Advance Research, Ideas and Innovations in Technology, Volume 3, Issue 1, 603-605pp.

Keylok Report (2018) "Solutions", https://www.keylok.com/solutions (04.01.2018 Tarihinde Erişildi.)

Keylok Report (2018b) Secure Software Licensing, The Best And Only Line Of Defense Insights, https://www.keylok.com/resources/insights/secure-software-licensing-best-and-only-line-defense (04.01.2018 Tarihinde Erişildi.)

Khan, Muhammad & Akram, Muhammad & Riaz, Naveed (2015) A Comparative Analysis of Software Protection Schemes, The International Arab Journal of Information Technology, Vol. 12, No. 3, May 2015, 286-295pp.

Kieseberg, Peter & Weippl, Edgar (2018) Security Challenges in Cyber-Phyiscal Production Systems, 3-16pp, (in) Software Quality: Methods and Tools for Better Software and Systems, 10th International Conference, SWQD 2018, Vienna, Austria, January 16–19, 2018, Proceedings.

Kuzurin, Nikolay & Shokurov, Alexander & Varnovsky, Nikolay & Zakharov, Vladimir (2007) On the Concept of Software Obfuscation in Computer Security, J. Garay et al. (Eds.): ISC 2007, LNCS 4779, pp. 281–298, 2007. Springer-Verlag Berlin Heidelberg 2007.

Lee, Byoungcheon & Kim, Kwangjo (1999) Copyright Protection of Software using Public Key Infrastructure. Proc. of SCIS99. Jan.26-29, Symposium on Cryptography and Information Security, The Institute of Electronics, Information and Communication Engineers.

Leite, Julio Cesar Sampaiodo Prado & Cappelli, Claudia (2010) Software Transparency, Business & Information Systems Engineering 3|2010, 127 -139 pp.

Liutkevicius, Agnius & Vrubliauskas, Arunas & Kazanavicius, Egidijus (2011) "Assessment of Dongle-based Software Copy Protection Combined with Additional Protection Methods", Electronics and Electrical Engineering, 2011. No. 6(112), 111-116pp.

Miller, Matt L. & Cox, Ingemar J. & Linnartz Jean-Paul M.G. & Kalker, Ton (1999) A review of watermarking principles and practices, 1-32pp, "Digital Signal Processing in Multimedia Systems, Ed. K. K. Parhi and T. Nishitani, Marcell Dekker Inc., 461-485pp.

Ming, Shi Wang & Wei, Che Chen (2007) "Digital Image Copyright Protecton Scheme Based On Visual Cryptography and Singular Value Decomposition", Optical Engineering 46-6, 067006 June 2007, 1-8 pp.

Moseley, O. & Whitis, R. (1995) 'Preventing Software Piracy', Management Accounting (December), 42-47pp.

Naor, Moni & Shamir, Adi (1995) "Visual cryptography," in Proc. Advances Cryptol. EUROCRYPT94, LNCS 950, pp. 1–12, Springer-Verlag, Berlin 1995.

O-Key Raporu (2018) "O-Key Dongle Genel Bakış" http://www.okeydongle.com/urunler/129/genel-bakis (04.01.2018 Tarihinde Erişildi.)

Piazzalunga, Ugo & Salvaneschi, Paolo & Balducci, Francesco & Jacomuzzi, Pablo & Moroncelli, Cristiano (2007) "Security Strength Measurement for Dongle-Protected Software", IEEE Security & Privacy Magazine, 5(6), 32–40pp. http://doi.org/10.1109/MSP.2007.176.

Rivest, Ronald L. & Shamir, Adi & Adleman, Len (1978) A method for obtaining digital signatures and public-key cryptosystems. Communications of the Acm, 21(2), 120-126pp. http://portal.acm.org/citation.cfm?doid=359340.359342 doi: 10.1145/359340.359342 (09.01.2018 Tarihinde Erişildi.)

Sasirekha N. And Hemalatha M., (2012) "A Survey on Software Protection Techniques Against Various Attacks," Global Journal of Computer Science and Technology, vol. 12, no. 1, 53-58pp.

Schneier, Bruce (2004) "Secrets and Lies, Digital Security in a Networked World", John Wiley & Sons.SIMPSON, J.A. & Weiner, E.S.C. (eds) (2000) Oxford English dictionary, Second Edition edn, Oxford University Press.

Simpson, P. & Banerjee, D. (1994) 'Softiifting: A Model of Motivating Factors', Journal of Business Ethics 13, 431-438pp.

Smith, Kevin (2013) What The Heck is a Dongle? http://www.businessinsider.com/what-is-a-dongle-2013-3 (09.11.2017 Tarihinde erişildi.)

Stevenson, David (2006) The Downloader's Handbook, Your complete guide to using broadband for downloading, ripping and converting music and film, Hampshire, Harriman House.

Sun, Xiaolei & Cheung, Sw William & Yuk, Tung Ip Tony (2013) A dual-band antenna for wireless USB dongle applications, The 2013 International Workshop on Antenna Technology (iWAT 2013), Karlsruhe, Germany, 4-6 March 2013, In Conference Proceedings, 2013, 307-310pp.

Tamada, Haruaki & Nakamura, Masahide & Monden, Akito & Matsumoto, Kenichi (2004) "Design and evaluation of birthmarks for detecting theft of java programs," in Proc. IASTED International Conference on Software Engineering (IASTED SE2004), Feb 2004, 569–575pp.

Thong, J. And Yap, C. (1998) 'Testing an Ethical Decision-Making Theory: The Case of Softlifting', journal of Management Information Systems 15(1), 213 237pp.

Varian, Hal & Farber, David & Manferdelli, John & Green, Lucky & Alben, Alex (2003) Impacts Of Drm On Innovation, Competition And Security, Berkeley Technology Law Journal, Vol. 18, No. 2 (Spring 2003), 697-771pp, Published by: University of California, Berkeley, School of Law Stable, http://www.jstor.org/stable/24116756 (09.01.2018 Tarihinde Erişildi.)

Venkatesan, Ramarathnam & Vazirani, Vijay V. And Sinha, Saurabh (2001) "A graph theoretic approach to software watermarking," in 4th International Information Hiding Workshop, Pittsburgh, PA.

York, Matt & Muratore, Stephen (Ed) (2004) The Videomaker Guide to Digital Video and DVD Production Second Edition, Oxford, UK, Elsevier.

Zhang, Xuesong & He, Fengling & Zuo, Wanli (2008) "Hash Function Based Software Watermarking," in Proceedings of the Advanced Software Engineering and its Applications, Hainan Island, 95-98pp.

Zeng, Ying & Liu, Fenlin & Luo, Xiangyang & Yang, Chunfang (2011) Software Watermarking Through Obfuscated Interpretation: Implementation and Analysis 329, Journal of Multimedia, ISSN 1796-2048 Volume 6, Number 4, August 2011, 329-340pp.

Zhu, William Feng (2007) Concepts and Techniques in Software Watermarking and Obfuscation, A thesis submitted in partial fulfillment of the requirements of Doctor of Philosophy in Computer Science, The Department of Computer Sciences The University of Auckland New Zealand, August 2007.

Zhu, William Feng & Thomborson, Clark & Wang, Fei-Yue (2005) "A Survey of Software Watermarking", In Springer-Verlag Berlin ISI 2005, LNCE 3495, 454-458pp.

Şekil- Tablo-Fotoğraf Ve Grafik Bilgileri

Şekil 1: Watermarking (Filigran) Çalışma Düzeni, (MILLER, Matt L. & COX, Ingemar J. & LINNARTZ Jean-Paul M.G. & KALKER, Ton (1999) A review of watermarking principles and practices, 1-32pp, "Digital Signal Processing in Multimedia Systems, Ed. K. K. Parhi and T. Nishitani, Marcell Dekker Inc., 461-485pp.

Şekil 2: Üç Aşamalı Yazılım Üretim ve Koruma Yapısı

Fotoğraf 1: Dongle Çeşitleri https://medium.com/@hartsock/unit-tests-withdonglesbd942db2e37e (20.01.2018 Tarihinde Erişildi).

Tablo 1: Yazılım Koruma Tekniklerinin Eleştirel Analizi (KHAN, Muhammad & AKRAM, Muhammad & RIAZ, Naveed (2015) A Comparative Analysis of Software Protection Schemes, The International Arab Journal of Information Technology, Vol. 12, No. 3, May 2015, 286-295pp).

Tablo 2: Yazılım Kullanıcı Tipolojisi ve Yönsel Analiz

Grafik 1: Dünyanın Farklı bölgelerinde 2009-2015 Dönemi Lisanssız Yazılım Paylaşımı (https://www.statista.com/statistics/263161/percentage-of-pirated-software-in-several-global-regions/ (05.01.2018 Tarihinde Erişildi).

Grafik 2: 2009-2015 Arası Yasal ve Yasadışı Bilgisayar Yazılım Kullanımlarının Küresel Payı https://www.statista.com/statistics/263187/global-distribution-of-legal-and-illegal-computer-software/ (05.01.2018 Tarihinde Erişildi).

Grafik 3: EDIUS ID Kayıtlanma Sistemi, (Grass Walley a Belden Brand, 2015:2) http://www.en.ediusworld.com/docs/Application_Notes/professional/edius/G_VB-1-0525A-EN-AN_EDIUS_Activation_WW.pdf (05.01.2018 Tarihinde Erişildi).

Grafik 4: Dünya Genelinde Bilgi Teknolojileri Alanında Yazılım Harcaması 2009- 2019. https://www.statista.com/statistics/203428/total-enterprise-software-revenue-forecast/ (10.01.2018 Tarihinde Erişildi).

Grafik 5: Dünya Genelinde Mobil Aygıt Ürünlerinin 2009-2020 Dönemi Analizi https://www.statista.com/statistics/219008/global-mobile-device-mix-since-2010-by-device/ (10.01.2018 Tarihinde Erişildi).