TÜRK BİLİM ARAŞTIRMA VAKFI
TÜBAV
1995
TURKISH SCIENCE-RESEARCH FOUNDATION
TUBAV BILIM DERGISI

Yıl: 2011, Cilt:4, Sayı:1, Sayfa:52-63

ÖZGÜR DAĞITIMA KONU OLAN YAZILIMLAR: NEDİR, NE KADAR GÜVENLİDİR?

Hakan DEMİRTEL[1*], Şeref SAĞIROĞLU[2]

[1]Devlet Planlama Teşkilatı Müsteşarlığı, Yönetim Bilgi Merkezi Dairesi Başkanlığı, 06100, Ankara
[2]Gazi Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe, 06570, Ankara

Özet

Günümüzde yazılım sektöründeki telif hakları ve yazılım geliştirme konularına yeni bir anlayış ve yaklaşım getiren özgür dağıtım (copyleft) kavramı giderek daha yaygın olarak bilinmekte ve bu anlayış çerçevesinde üretilen yazılımlar çok çeşitli alanlarda kullanılmaktadır. Bu makalede telif hakları ve yazılım lisansları konularında genel kavramlar kısaca açıklanmış bazı açık kaynak kodlu yazılımlar tanıtılmış, açık kaynak kodlu yazılımların daha güvenli olup olmadığı konusundaki çalışmalar incelenmiş, özgür dağıtıma konu olan yazılımların testlerinde kullanılan yazılımlar kısaca özetlenmiş ve sonuçta özgür yazılımların güvenilirliği üzerine değerlendirmeler yapılmıştır.

Anahtar Kelimeler: Özgür dağıtım, özgür yazılım, açık kaynak kodlu yazılım, güvenli yazılım

COPYLEFTED SOFTWARE : WHAT AND HOW SECURE ARE THEY?

Abstract

Copyleft is a concept that brings new vision and understanding to copyright and software development in the software industry is better-known nowadays and software developed with this concept is widely used in many areas. In this paper; copyright and software license terms are briefly explained, literature and studies on open source software are reviewed if they are secure or not, and security test tools and results for open source software were also revised and finally some evaluations are given to conclude the study on the reliability inter mod vulnerabilities of open source software.

Keywords: Copyleft, Free software, Open source software, Secure software

* **E-posta:** demirtel@dpt.gov.tr, ss@gazi.edu.tr

1. Giriş

Bilişim teknolojilerindeki gelişmeler, insan, toplum ve dolayısıyla iş ve işleyiş yapısında değişimleri ve gelişmeleri zorunlu kılmıştır. Bu gelişim, bilişim teknolojilerinin kamu ve özel hizmetlerinin daha hızlı sunulması, yaygınlaştırılması, doğru ve yeterli bilgiye hızla ulaşma, güvenli kullanabilme, giderlerinin azaltılması, şeffaflaşma gibi beklentileri de beraberinde getirmiştir.

Elektronik ortamların hızla yaygınlaşması pek çok sorunu da beraberinde getirmektedir. Açık kaynak kodlu yazılımlar ve özellikle özgür yazılımlar son günlerde üzerinde çok konuşulan, tartışılan ve tartışılmaya da devam edecek konuların başında gelmektedir.

Çalışmada telif hakları ve korsan kullanım araştırmasına girilmemiştir. Çalışma kapsamında daha ziyade, yazılım geliştirme faaliyetlerine yeni bir boyut kazandıran ve bu alandaki birikimi tüm dünyanın birikimi kabul ederek konuya sosyal bir bakış açısı getiren özgür yazılım yaklaşımı ve bu yaklaşımın diğer yazılım geliştirme ve koruma yöntemlerinden farklılıkları açıklanmaya çalışılmıştır.

Çalışmada, açık kaynak kodlu yazılımlar ve özellikle özgür yazılımlar konusundaki tanım ve yaklaşımlar derlenmeye ve üzerinde sıkça tartışılan açık kaynak kodlu yazılımların güvenlik açıkları mercek altına alınmıştır. Özgür yazılım ve açık kaynak kodlu yazılımları temel yaklaşımlarındaki birlikteliği ya da benzerliği de dikkate alınarak güvenlik avantajları/zafiyetleri aynı kapsamda ele alınmış ve birlikte değerlendirilmiştir.

Bu makalede, konuyla ilgili olarak anlam kargaşasını ortadan kaldırmak amacıyla Bölüm 2'de Telif Hakkı (Copyright), Fikri Mülkiyet (Intellectual Property), Patent, Lisans, Gizlilik Sözleşmesi (Nondisclosure agreements), Kaynak Kod (Source Code), Özgür Dağıtım (Copyleft), açık kaynak kodlu (AKK) yazılımlar, özgür yazılım (Free Software) , GNU Lisansları, GNU GPL (General Public License), GNU LGPL (Lesser General Public License), GNU FDL (Free Documentation License), BSD (Berkeley Software Distribution) Lisansı gibi tanımlar ve yaklaşımlar kısaca özetlenmiştir. Bölüm 3'de ise Özgür Dağıtım örneklerinden bazıları PARDUS, OpenOffice.org ve MySQL açıklanmıştır. Bölüm 4'de ise özgür dağıtım yazılımlarının güvenliği konusu araştırılmış ve literatürdeki mevcut çalışmalar ışığında değerlendirmeler yapılmışır. Son bölümde çalışma genel olarak ele alınmış ve değerlendirmeler yapılmıştır.

2. Tanımlar ve Yaklaşımlar

Bu başlıklar altında önemli görülen hususlar alt başlıklar halinde aşağıda verilmiştir.

Telif Hakkı (Copyright)

Telif hakkı, fikri mülkiyet konusu olan fikir veya sanat eserinin, eseri oluşturan kişi ya da kuruma söz konusu eserin kopyalarını dağıtım hakkının belirli bir süre için verilmesidir ve bu hakkın koruyucusu devletlerdir. Başka bir deyişle telif hakkı, üretilen özgün ürünün kopyalarının dağıtımı konusunda ürün yaratıcısının hakimiyetinin sağlanmasıdır. 5.12.1951 tarihli ve 5846 sayılı Fikir ve Sanat Eserleri Kanununda fikir ve sanat eserleri; ilim ve edebiyat eserleri, musiki eserleri, güzel sanat eserleri ve sinema eserleri olmak üzere dört ana başlık altında değerlendirilmiştir:

Kanunun ilim ve edebiyat eserlerinin neler olduğunu sıralayan ikinci maddesinin 7 Haziran 1995 tarihli değişik birinci fıkrası ile yazılımlar kanun kapsamına dâhil edilmiştir. Söz konusu fıkra "Herhangi bir şekilde dil ve yazı ile ifade olunan eserler ve her biçim altında ifade edilen bilgisayar programları ve bir sonraki aşamada program sonucu doğurması koşuluyla bunların hazırlık tasarımları;" şekilde düzenlenmiştir:

Aynı maddenin son fıkrasında "Ara yüzüne temel oluşturan düşünce ve ilkeleri de içine almak üzere, bir bilgisayar programının herhangi bir öğesine temel oluşturan düşünce ve ilkeler eser sayılmazlar." hükmü getirilerek çözüm yoluna ilişkin düşünce ve ilkelerin telif hakkı içinde yer almadığı belirtilmiştir.

McGowan ve arkadaşları, telif hakkı konusundaki yaklaşımların patentli yazılım destekçileri, açık kaynak savunucuları, özgür yazılımcılar ve yazılım korsanları olarak dört farklı grup altında değerlendirilebileceğini ifade ederler [1].

Patentli yazılımı destekleyenler yazılımın fiziksel bir ürün gibi kontrol altında tutulması ve korunması gerektiğini düşünmektedirler [1]. Durell, yazılım firmalarının ürünlerini pazarlayarak hayatta kaldıklarını bu nedenle yazılımdaki yenilikçiliğin mutlaka korunması gerektiğini savunmaktadır [2]. Durell, bu korumanın monopollerin korunması şeklinde değil, yenilikçilik yapan firmaların ödüllendirilmesi ve bu alanda çalışmalarına devam etmelerinin garanti altına alınması şeklinde uygulanmasını önermektedir.

Ülkelerdeki mevzuat, patentli yazılım yaklaşımını desteklemek üzere tasarlanmıştır. Bu yaklaşımı destekleyenlerin genellikle Business Software Alliance (BSA) ve Software and Information Industry Association (SIIA) gibi organizasyonlarla birlikteliği söz konusudur [1].

Açık kaynak savunucuları, internet aracılığı ile birbirleri ile iletişimde bulunan ve beraber çalışan programcı "topluluk"ları mensuplarıdır. Open Source Initiative (OSI), Creative Commons gibi örnekler verilebilecek bu organizasyonlar yazılımların, programcıların kaynak kodları okuyup, geliştirip dağıtabilecekleri ortamlarda geliştirilmesine öncülük ederler. Özgür yazılımcılar ile açık kaynak savunucuları aslında aynı ideolojiden yola çıkmış olsalar da aralarında düşüncelerini yaymak ve geniş kabul görmek için uzlaşma arayışı açısından farklılıklar mevcuttur [1]. Açık kaynak savunucuları, telif haklarına daha ılımlı yaklaşan ve özgür yazılımcılara oranla daha az radikal girişimler içeren bir grup olarak değerlendirilebilir. Bununla beraber grup içinde telif hakkı sisteminin hızlı ve kaliteli yazılım geliştirmeye engel oluşturduğu görüşü kabul görmektedir.

Özgür yazılım yaklaşımı, kişi ya da firmaların patentli ve telif hakkı söz konusu olacak şekilde yazılım kodu üretmemesi prensibine dayalıdır. Bu grup mensupları, geliştirilen programlama tekniklerine patent verilmesinin yenilikçilik ve ilerlemeye engel olduğunu savunmaktadırlar [1].

Özgür yazılımcılar, özgür konuşma hakkı ile özgür kod geliştirme hakkını birbirine eş haklar olarak görürler. Telif hakkı yaklaşımı, geleneksel olarak, kendi kendine sansür üzerine inşaa edilirken özür yazılımcılar telif hakkının sahiplik unsurunun değil özgür konuşma hakkının koruyucusu olmasını istemektedirler. Bu anlayışa göre asıl olan toplumsal fayda yaratmaktır. Bu felsefenin temeli Richard Stallman'ın kurduğu Free Software Foundation (FSF) tarafından 1984 yılında atılmıştır [1]. Stallman özgür yazılımın, 1998 sonrasında gündeme gelen açık kaynak yaklaşımına olan üstünlüğünü "Açık kaynak hareketi için asıl konu işin uygulama tarafıdır, konuyu etik açıdan ele almaz. Söylendiği üzere açık kaynak bir yazılım geliştirme yöntemidir, özgür yazılım ise sosyal bir harekettir. Açık kaynak hareketi özgür olmayan yazılımı optimal olmayan bir çözüm olarak niteler. Özgür yazılım ise bunu sosyal bir problem olarak görür ve çözümün özgür yazılım olduğunu savunur" olarak açıklamıştır [3].

Bu kapsamda ülkemizde geliştirilen Pardus işletim sisteminin proje yöneticisi Erkan Tekman özgür yazılımların önemli bir unsurunun "topluluk" kavramı olduğunu Bilişim'08 etkinliği kapsamındaki sunumunda özellikle dile getirmiştir [4]. Aslında özgür yazılımlar bir kişi ya da kurumun değil bir "topluluğun" ürünü haline gelen yazılımlardır.

Yazılım korsanları ise bilgisayar korsanlarının genel yaklaşımını temsil eden grup olarak düşünülebilir. Bu grubun genel felsefesi yıkıcılıktır. Yazılımlar ve bilgi sistemleri üzerinde her türlü zarar verici faaliyeti haklı görürler. Teknoloji bağımlısı olan bu grup üretilen tüm teknolojinin incelenmek ve araştırılmak üzere ellerinin altında olmasını ister ve bu amaçla "saldırı (hacking)" ve "kırma (cracking)" faaliyetleri yürütürler. Bu yapısı ile grup üyeleri teknolojiye sahip olabilmek için etik olmayan ve yasa dışı sayılan pek çok yöntem kullanırlar [1].

Telif hakkı ile yakından ilgili diğer tanımlar

Fikri Mülkiyet (Intellectual Property)

Dünya Fikri Mülkiyet Örgütü (WIPO) kuruluş sözleşmesi fikri mülkiyeti aşağıdaki hakları içerecek şekilde tanımlanmıştır [5].

- Edebi, sanatsal ve bilimsel çalışmalar,
- İcracı sanatçıların eserleri, fonogram ve radyo yayınları,
- İnsan emeğinin tüm alanlarındaki buluşlar,
- Bilimsel buluşlar,
- Endüstriyel tasarımlar,

- Ticari markalar, hizmet markaları, ticari unvan ve isimler,
- Haksız rekabete karşı koruma,
- Sınai, bilimsel, edebi ya da sanatsal alanlarda fikri mülkiyet faaliyetlerinden kaynaklanan diğer tüm haklar.

Patent

Fikri mülkiyet haklarının buluşlarla ilişkili olanları patent adı altında ele alınır. Bu buluşlar bilgisayar yazılımlarından, sanatsal çalışmalara kadar çok geniş bir alanda olabilir. Patentte koruma altına alınan unsur fikirdir.

Stallman telif hakkı ile patent arasındaki beş farka değinmektedir [3]:

- Telif hakkı bir işe ilişkin detaylı açıklamalar içerirken, fikre ilişkin unsurları kapsamaz. Patentler sadece fikirleri ve bu fikirlerin kullanımını konu edinir.
- Telif hakkı otomatikman oluşurken patentler yetkili kurumlarca düzenlenir.
- Patent almanın maddi boyutu vardır ve zaman alır.
- Telif hakkı çok uzun sürelidir, bazı durumlarda 150 yıl bile sürebilir. Patentler yaklaşık 20 yıl sürer. Bu süre yazılım dünyası için çok uzun bir süredir.
- Telif hakkı sadece kopyalama hakkını verir. Telif hakkı olan bir kitapta kullandığınız bir terimin daha önce başkası tarafından kullanılmış olması sizi suçlu konumuna sokmaz, görmediğinizi söyleyip kendinizi savunabilirsiniz. Patent ise fikrin kullanımı üzerinde kesin bir tekel hakkı oluşturur. Her ne kadar siz fikri çalmadığınızı iddia etseniz de fikir patentli ise ihlal etmiş olursunuz.

Lisans

Telif hakkı olan bir ürün ya da hizmeti kullanma hakkı ya da ruhsatı olarak tariflenebilir. Yazılımlar için lisans, telif sahibinin haklarını korumak üzere geliştirilmiş belgedir.

Gizlilik Sözleşmesi (Nondisclosure agreements)

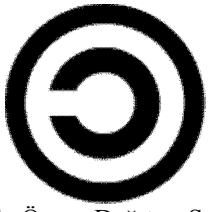
İki taraf arasında belirli bir amaç çerçevesinde paylaşılmaya konu olan gizli bilgi ve belgelerin başkaları ile paylaşılmaması amacı ile yapılan anlaşmalardır. Telif hakkına konu olan ya da olacak fikir veya ürünler için yapılan gizlilik anlaşmaları, telif haklarının korunmasına yönelik tek başına yeterli görülmemelidir. Ancak bu tür sözleşmelerin bir tedbir olarak kullanılmasının fikir ya da ürün sahibinin yararına olacağı da açıktır.

Kaynak Kod (Source Code)

Kaynak kod, bir bilgisayar dilinde yazılmış ve insanlar tarafından okunabilen komutlar kümesidir. Bu kodların bilgisayarlar tarafından uygulanabilmesi için derleyiciler ve diğer araçlar yardımıyla makine diline (executable code) çevrilmesi ya da yorumlayıcılar tarafından uygulanışı esnasında aynı işlevin yerine getirilmesi gerekir.

Özgür Dağıtım (Copyleft)

Hızla artan lisanssız yazılım kullanımı günümüzde yazılım alanında telif hakkının korunmasında büyük zafiyetlerin bulunduğunu göstermektedir. Hatta bu durum telif hakkı sahibinin haklarını koruyan lisanslı bir yazılımın yasal kullanımının hiçbir zaman tam anlamıyla uygulanabilir olmadığı düşüncesinin oluşmasına neden olmuştur. Temelinde yine telif hakları kanunu olan özgür dağıtım, bu alanda yeni ve çok farklı yaklaşımları içinde barındırmaktadır. Simgesi Şekil 1'de verilmiştir.



Şekil 1: Özgür Dağıtım Sembolü

"Copyleft", GNU GPL kısaca GPL lisansının kullandığı lisanslama metoduna verilen genel addır ve "özgür dağıtım" olarak ifade edilebilir. "Copyleft", telif hakkı terimi olan "copyright"ın aksine eser sahibinden ziyade bir "topluluk" tarafından ortaklaşa geliştirilen eserin tüm kullanıcılar tarafından özgürce kullanılmasının sağlanması esasına dayanır. "Özgür dağıtım" temel kanun olarak teklif hakkı kanunlarını kullanmakta ise de asıl amacı eserlerin özgürlüğünü sağlamaktır.

Açık kaynak kodlu (AKK) yazılımlar

Önceki bölümlerde de dile getirildiği üzere, geliştirilen yazılımların kaynak kodlarının açık olması ve bu kodların rahatça geliştirilebilir, değiştirilebilir olması açık kaynak kodunun temel unsurlarıdır.

Özgür yazılım (Free Software) [3]

Özgür yazılım kavramı, kullanıcıların, yazılımı çalıştırma, kopyalama, dağıtma, üzerinde çalışma, değiştirme ve geliştirme özgürlükleriyle ilgili bir kavramdır. Daha açık ifadeyle, "özgür yazılım" kavramı yazılım kullanıcıları için dört çeşit özgürlüğe dayanır:

- Herhangi bir amaç için yazılımı çalıştırma özgürlüğü
- Yazılımın nasıl çalıştığını öğrenme ve gereksiniminize göre uyarlayabilme özgürlüğü Yazılımın kaynak koduna ulaşmak, bu iş için önkoşuldur.
- Kopyaları dağıtma özgürlüğü. Böylece komşunuza yardım edebilirsiniz
- Tüm toplumun yarar sağlayabileceği şekilde programı geliştirme ve geliştirdiklerinizi yayınlama özgürlüğü kaynak koduna erişmek, bunun için bir önkoşuldur.

GNU Lisansları

Richard M. Stallman 1984 yılında GNU Projesine başladığında sadece popüler olmak değil aynı zamanda kullanıcılarına özgürlük vermek amacının da olduğunu belirtmektedir [3]. Stallman, GNU yazılımının patentli bir ürün olmasını engelleyecek şekilde yeni bir yazılım dağıtımı anlayışı getirmeyi başarmıştır. GNU Projesi üç tip kamu lisansı oluşturmuştur:

GNU GPL (General Public License)

Özgür dağıtımın asıl unsurunu GPL adı verilen genel kamu lisansı oluşturur. Bu lisans yazılım kaynak kodlarının özgürce dağıtımını garanti altına alan yapıdadır. GPL lisanslı ürün özgürce ve paralı ya da parasız olarak dağıtılabilir, üzerinde değişiklik yapılabilir ve garanti içermez. Ancak GPL lisanslı yazılımı para ile satan kişi ya da firmanın yazılıma garanti vermesi konusunda da bir sınırlama yoktur. Bu sistemde GPL ile lisanslanmış bir yazılımın değişikliklerden sonra yaşam döngüsüne yine GPL olarak devam etmesi istenmektedir. GPL'in ilk sürümü 1989 yılında kaleme alınmış, 1991'de ikinci sürümü hazırlanmıştır.

Özgür dağıtım güvencesinin sürdürülebilmesini teminen 2007 yılında yenilenen GPL'in üçüncü sürümü, yasal ve teknolojik gelişmelerden kaynaklanan değişikliklerin getirdiği belirsizlikleri ortadan kaldırmayı amaçlamaktadır. Bu sürüm ile kullanıcıların oluşan üç yeni tehditten korunması amaçlanmaktadır. Bu tehditler FSF tarafından sürüm 3'ün çıkış nedenleri olarak aşağıdaki şekilde açıklanmaktadır [6].

- Tivoization: Üzerinde GPL lisanslı yazılımlar çalıştıran çeşitli donanım üreticileri bu donanımlar üzerinde çeşitli düzenlemeler yaparak yazılım yenilemesinin ancak kendi geliştirdikleri yazılımlar kullanılarak yapılmasını zorunlu hale getirmektedir.

- Özgür yazılımı yasaklayan kanunlar: Digital Millenium Copyright Act ve EU Copyright Directive sayısal haklara zarar verecek yazılım geliştirme ve paylaşma faaliyetlerini yasaklamıştır. Özgür yazılım kullanıcılarının ve GPL lisansının bu konu ile alakasının olmadığı hususuna açıklık getirilmiştir.

- Ayrımcı patent anlaşmaları: Microsoft, yazılımın imtiyaz hakkı ödeyen sağlayıcılardan temin edilmesi halinde patent ihlallerine göz yumacağını söylemektedir. Sonuç olarak Microsoft özgür yazılım kullanımına ilişkin royaltileri toplamaya çalışmaktadır. Bu durum FSF tarafından kabul edilemez bir durum olarak değerlendirilmektedir.

Sürüm 3 konusunda özellikle Tivoization ile ilgili farklı düşünceler bulunmaktadır. Bu görüş farklılıklarının bir yansıması olarak Linux çekirdeği GPLv3 ile lisanslanmamış olup, GPLv2 lisanslı şekilde varlığını sürdürmektedir. Bahse ilk konu olan TiVo şirketinin donanımın doğru çalışmasını teminen geliştirdiği çözümde yanlış bir durumun olmadığı ve bu uygulamanın GPLv2'ye de uygun olduğu savunulmaktadır.

GNU LGPL (Lesser General Public License)

Özellikle kütüphane niteliğindeki yazılımların özgür olmayan yazılımlar tarafından kullanılmasına izin veren GPL lisansıdır. Bu lisans ile amaçlanan geniş kitlelerin söz konusu lisansa sahip kütüphaneleri kullanmalarının önünü açmak bu nitelikteki yazılımların defakto standart olmasının sağlanmasıdır. Adından da anlaşılacağı üzere LGPL her ne kadar daha az özgürlük demek olsa da yukarıda belirtildiği gibi bazı durumlarda özgür yazılım ve kullanıcı özgürlüğüne fayda sağladığı durumlar oluşabilmektedir.

Bu lisans sayesinde patentli yazılım kullanıcıları LGPL lisanslı kütüphanelerini ve bunların değiştirilmiş ve geliştirilmiş sürümlerini kullanma özgürlüğüne kavuşmuştur.

GNU FDL (Free Documentation License)

Özgür yazılıma ilişkin dokümantasyonun özgürce paylaşılmasını sağlamak üzere geliştirilmiştir. FDL ilgili dokümantasyonun özgürce kopyalanması, dağıtılması, değiştirilmesi, ticari ya da gayri ticari olarak kullanılabilmesine izin veren lisanstır.

BSD (Berkeley Software Distribution) Lisansı

BSD lisansı kaynak kodu açık olarak dağıtılan yazılımın sahipli yazılımlar tarafından da kullanılmasına izin veren yapıdadır. GPL nihai ürünün özgür olmasını zorunlu kılarken BSD lisansında böyle bir zorunluluk yoktur. Bu nedenle birçok ticari ürün BSD Lisanslı yazılımları kendi çözümlerini oluştururken kullanabilmişlerdir.

BSD lisansları özgür dağıtımdan daha çok her şeye açık lisanslar (permissive licenses) kapsamında değerlendirilmektedir.

3. ÖZGÜR DAĞITIM ÖRNEKLERİ

PARDUS [7]

2003 yılında fikri anlamda ilk adımları atılan ulusal işletim sistemi çalışmaları küçük bir yazılım ekibi ile 2004 yılında TÜBİTAK UEKAE bünyesinde başlatılmıştır. Linux temelli olması kararlaştırılan işletim sisteminin GPL lisansı kullanması benimsenmiş ve adı "Pardus" olarak belirlenmiştir.

Pardus'un kısa hikayesi şu şekilde verilmektedir: Pardus'un "bilişim okur-yazarlığına sahip bilgisayar kullanıcılarının temel masaüstü ihtiyaçlarını hedefleyen" bir işletim sistemi olmasına, "mevcut Linux dağıtımlarının üstün taraflarını kavram, mimari ya da kod olarak kullanmasına", ancak "otonom sisteme evrilebilecek bir yapılandırma çerçevesi ve araçları ile kurulum, yapılandırma ve kullanım kolaylığı sağlamasına" karar verilerek geliştirilmeye başlanmıştır [7].

Teknik hedefi ve yöntemi belirlenen bu proje hızla büyümüş ve 1 Şubat 2005 tarihinde ilk ürün olan Pardus Çalışan CD 1.0 çıkarılmıştır. Projenin amaçları ve teknik yaklaşımı hakkında Linux camiası ve kullanıcıları bilgilendirmeyi amaçlayan Çalışan CD beklenenin üzerinde ilgi görmüştür. Sonrasında geliştirme işlemleri daha çok özgün yenilik projelerine yoğunlaşmıştır. Nihayet 26 Aralık 2005'te Pardus'un ilk kurulabilir sürümü olan Pardus 1.0 Web üzerinden yayımlanmaya başlanmıştır." [7]

Pardus özgür yazılımın önemini, katkılarını ve oluşturduğu katma değeri görmek açısından önemli bir deneyimdir. 2 milyon satırdan fazla kod içeren bu yazılım 10 kişilik çekirdek bir ekiple başlayan bu süreç, Pardus topluluğu üyesi pek çok kişinin projeye destek vermesiyle topluluk halini almıştır.

OpenOffice.org [8]

OpenOffice.org açık kaynak kodu yaklaşımını benimsemiş bir projedir. Aynı zamanda bir ofis yazılımıdır. 13 Ekim 2000 tarihinde faaliyete başlamıştır. Projenin OpenOffice.org 1.0 sürümü 30 Nisan 2002 tarihinde çıkmıştır.

Proje ile tüm özgür yazılım projelerinde olduğu gibi topluluk temelinde kurulmuş olup uluslararası kabul görecek bir ofis yazılımı ortaya konulması hedeflenmiştir.

Proje Sun Microsystems'in sponsorluğunda yürütülmektedir. Sun Microsystems aynı zamanda kod geliştirmede de en önemli katkıyı vermektedir. Projeyi destekleyen diğer önemli firmalar Novell, RedHat, RedFlag CH2000, IBM ve Google'dır. Bunlara ilaveten tüm dünyadan 450.000 kişi projeye bağımsız olarak destek vermektedir. Asıl topluluk unsurunu da bu destekçiler oluşturmaktadır.

OpenOffice.Org projesi kapsamında XML tabanlı olarak geliştirilen açık doküman formatları 2005 yılında OASIS tarafından ODF standardı olarak kabul edilmiş ve daha sonra bu formatlar ISO/IEC 26300 numaralı ISO standardı haline gelmiştir.

OpenOffice.org LGPL lisanslıdır. Ayrıca ürüne ilişkin dokümanlar da FDL lisansı ile özgür dağıtım kapsamındadır.

MySQL

MySQL, Sun Microsystems tarafından geliştirilmiş GPL lisanslı bir veri tabanıdır [9]. Kurumsal çözümler açısından LAMP (Linux, Apache, MySQL, PHP/Perl/Python) adı verilen açık kaynak kodlu kurumsal çözümlerin önemli bir bileşenidir.

GPL lisanslı bir ürün olarak özgürce kullanılabilmektedir. Ancak, ürün geliştiricilerin kendi ticari ürünlerine entegre olarak MySQL kullanabilmeleri için Sun Microsystems'den ticari lisansını satın almaları gerekmektedir.

4. AKK TEST METOTLARI VE ARAÇLARI

Literatüre baktığımızda, AKK yazılımları test etme için pek çok yaklaşım olduğu gibi test etmede kullanılabilecekte pek çok yazılım vardır. Yaklaşımları incelediğimizde, performans, optimizasyon, web, kodlama, link testleri, kaynak kod analizleri, zafiyet analizleri, sisteme/ağa sızma denetimleri, genel ağ analizleri, kablosuz ağ analizleri, süreçlerin denetimi, şifre kırma, netBIOS/SNMP, ağ haritalama, SQL Enjeksiyon, XSS gibi pek çok yaklaşım ve bu gibi testlerin yapıldığı pek çok yazılım bulunmaktadır [16-25,18]. Güvenlik testlerinde kullanılan 13 farklı yazılım ve yapılabilecek testler [16] nolu kaynakta verilmiştir. AKK yazılımlaraın güvenliğine önem verildiğini göstermek amacıyla testlerde kullanılan yazılımlardan pek çoğu verilmiştir. Bunlar; TestLink, Selenium, Watir, Bugzilla, Mantis, JUnit, FIT/FitNesse, JMeter, OpenSTA, TestNG, RSpec, Ruby, WatiN, FireFox, Testopia, Wireshark, AutoIt, Canoo WebTest, Easyb, Emma, Findbug, jsUnit, Marathon, PHPUnit, RTH, WebInject, WebLOAD, WET, allpairs, Android, bugzillametrics, buildbot, checkstyle, Cobertura, Concordion, cppunit, cruise control, CubicTest, cunit, Dogtail, dotproject, EasyMock, Eclipse, Eventum, executequery, figleaf, FireBug, FireWatir, Grinder, HL7TestHarness, htmlunit, Hudson, hyperic, jenny, JRuby, jwebunit, Lou Wilson's VBA/Excel Web Test Automation Tool, nHapi, nose, NotePad++, NUnit, Paros, Perl, Perl WWW, Mechanize, perlclip, perl-Win32GuiTest, PHP, pmd, Pylot, Python, QaTraq, RubyWebBench, Sahi, SalomeTMF, SAMIE, SandCastle, Seagul, simpletest, SoapUI, STAF, STAF & STAX, SymbianOS Unit test, TestComplete, The Grinder, twill, Unitils, utPLSQL, Watij, xUnit, vb. olarak sıralanabilir [16, 18].

5. ÖZGÜR YAZILIMLARDA GÜVENLİK

Çalışma kapsamında, açık kaynak kodlu (AKK) yazılımlar ile özgür yazılımların nitelikleri açısından birbirlerine çok benzer özellikler arz ettiği düşüncesinden hareketle, bu yazılımların güvenlik açısından birlikte değerlendirilmesinde fayda görülmüştür. Bu nedenle bu bölümde geçen AKK yazılım ifadesi, AKK yazılımları ve özgür yazılımları kapsamaktadır.

AKK yazılımlar ile kodu kapalı yazılımlar güvenlik açısından karşılaştırıldığında hangisinin avantajlı olduğu konusunda iki ayrı görüş bulunmaktadır. Bir grup, açık kaynak kodunun güvenlik için avantajlı olduğunu savunurken diğer bir grup tersini söylemektedir.

Bu alanda yapılan çalışmalara ilişkin önemli birkaç örnek aşağıda sunulmuştur:

Fortify Araştırması [10]

Fortify tarafından on ayrı AKK yazılımı üzerinde Fortify SCA güvenlik analiz aracı kullanılarak bir araştırma yapılmıştır. Bu araştırmada incelenen yazılımlar Tablo 1'de verilmiştir.

Tablo 1: İncelenen AKK Yazılımlar [10]

AKK Yazılım	Açıklama
Derby	İlişkisel veritabanı
Geronimo	Uygulama sunucusu
Hibernate	CRM web uygulaması
JBoss	Uygulama sunucusu
JOnAS	Uygulama sunucusu
OFBiz	e-İş web uygulaması
OpenCMS	İçerik yönetim yazılımı
Resin	Uygulama sunucusu
Struts	Web uygulama platformu
Tomcat	Uygulama sunucusu

İncelenen yazılımlarda yüksek, orta ve düşük etkili çok sayıda güvenlik açıkları tespit edilmiştir. Bu araştırma sonucunda AKK yazılımlarında bulunan önemli güvenlik tehditlerinin çok zayıf olan ya da hiç olmayan güvenlik süreçlerinden kaynaklandığı ifade edilmiştir. AKK yazılımların işlevsel olarak büyük şirketlerin ihtiyaçlarına uygun olacak şekilde geliştirilmekle beraber güvenlik açısından endüstri standartlarının çok gerisinde olduğu görüşüne yer verilmiştir.

Buna karşın bazı AKK yazılım geliştirme takımlarının doğru adımlar attığı, örneğin Mozilla'nın, ürünün güvenliğini artırmak amacıyla güvenlik danışmanı Rich Mogul ile çalışmaya başladığı aynı araştırmada belirtilmekte, bu tür yaklaşımların diğer AKK yazılım projeleri tarafından da benimsenmesinin gerektiği vurgulanmaktadır. Her ne kadar iyi örnekler olsa da AKK yazılımların çoğunlukla güvenlik açısından üç temel noktada büyük eksiklikleri olduğu çalışma tarafından dile getirilmektedir. Bu eksiklikler; eksik uzman, açıklıkların giderilmesinde teknolojiden faydalanmama ve güvenli süreç geliştirme yaklaşımlarını uygulamama olarak sıralanmaktadır.

Araştırmada bu eksikliklere ilişkin temel bulgular aşağıdaki verilen üç başlık altında özetlenmiştir.

1. Uzman eksikliği

Güvenlik konusunda uzman sayısının yeterli olmaması her iki tarafında sorunudur. Bu konuda üç temel ihtiyaç söz konusudur. Bunlar:

- Dokümantasyon: Söz konusu yazılımın güvenliğine ve güvenli kurulumuna ilişkin bilgilendirme dokümanlarına erişilebilmelidir.
- e-Posta: Kullanıcıların güvenlik açıklarını bildirebilecekleri bir e-posta adresi bulunmalıdır.
- Güvenlik uzmanlarına erişim: Kullanıcıların güvenlik sorunlarını tartışabilecekleri uzmanlara kolay erişimi sağlanmalıdır.

Bu konuda incelenen yazılımların durumları ise Tablo 2'de verilmiştir.

Tablo 2: Güvenlik Uzmanlarına Erişim [10]

AKK Yazılım	Güvenliğe özel e-posta adresi	Güvenlik hakkında bilgi	Güvenlik uzmanlarına kolay erişim
Derby	Yok	Yok	Yok
Geronimo	Yok	Yok	Yok
Hibernate	Yok	Yok	Yok
JBoss	Yok	Var	Var
JOnAS	Yok	Yok	Yok
OFBiz	Yok	Yok	Yok
OpenCMS	Yok	Yok	Yok
Resin	Yok	Yok	Var
Struts	Yok	Yok	Yok
Tomcat	Var	Var	Var

2. Güvenli yazılım geliştirme süreçlerinin uygulanmaması

Yazılımların sürümler itibarı ile incelendiğinde güvenlik açığı sayısının azalmadığı aksine bazılarında artışlar olduğu gözlendiği belirtilmektedir. Bu durum AKK yazılım geliştirmede güvenli kod geliştirmeye yönelik süreçlerde zafiyet olduğu şeklinde yorumlanmaktadır. İncelenen bazı yazılımlarda tespit edilen açıkların sayısı Tablo 3'de verilmiştir.

Tablo 3: İncelenen Yazılımların Satır ve Tespit Edilen Açık Sayıları [10]

AKK Yazılım	Açık sayısı	Satır sayısı
Cayenne 1.14	0	35.373
Cayenne 1.24	3	43.741
Cayenne 2.04	3	43.751
Derby 10.2.2.0	398	224.712
Derby 10.3.1.4	312	251.659
Derby 10.3.2.1	313	252.347
Geronimo 2.0	41	87.375
Geronimo 2.01	103	85.001
Geronimo 2.02	106	85.483
Hibernate 3.12	18	62.143
Hibernate 3.13	18	62.865
Hibernate 3.25	23	74.834
Hipergate 2.1.20	10.734	108.276
Hipergate 3.0.26	14.425	80.941
Hipergate 3.0.30	14.423	83.875
JOaAS 4.84	193	132.013
JOnAS 4.85	198	132.396
JOnAs 4.86	196	133.145

3. Açıkların azaltılmasında teknolojiden faydalanmama

İncelenen AKK yazılımların yaygın olarak bilinen XSS (cross-site scripting) ve SQL enjeksiyonu açıklarının yüksek sayılara ulaştığı belirtilmektedir. İncelenen yazılımların son sürümleri için tespit edilen XSS ve SQL injection güvenlik açığı sayıları Tablo 4'de verilmiştir.

Tablo 4: XSS ve SQL Injection Açık Sayıları [10]

Güvenlik Açığı	Tespit Edilen Açık Sayısı
XSS	22.828
SQL Enjeksiyonu	15.612

AKK yazılımlarda rastlanan çok sayıdaki güvenlik açığı, incelen yazılımlar bağlamında AKK yazılım geliştiricilerinin çok temel güvenlik açığı tarayıcılarını kullanmadığı düşüncesini akla getirdiği söylenmektedir. İncelenen yazılımların son sürümlerinde Fortify SCA kullanılarak tespit edilen etkisi yüksek açık sayısı ve ortalamaları Tablo 5'de verilmektedir.

Tablo 5: Açık sayıları ve ortalamalar [10]

AKK Yazılım	Açık Sayısı	Satır Sayısı (.000)	1000 satırda ortalama açık sayısı
Derby 10.2.2.0	398	224,7	1,8
Geronimo 2.02	106	85,5	1,2
Hibernate 3.25	23	74,8	0,3
Hipergate 3.0.26	14.425	80,9	178,2
JBoss 4.22GA	72	264,0	0,27
JOnAS 4.85	198	132,4	1,5
Ofbiz 4.00	56	110,6	0,51
OpenCMS 7.01	241	130,3	1,9
Resin 3.025	204	92,6	2,2
Struts1.27	41	34,3	1,2
Tomcat 6.014	469	80,2	5,8

Ross Anderson Çalışması [11]

Ross Anderson "Açık ve Kapalı (kodlu) Sistemler Eşittir (ideal dünyada)" isimli makalesinde özet olarak kaynak kodun açık ya da kapalı olmasının güvenlik açısından çok önemli bir fark oluşturmayacağı teorisini ortaya atmış ve bu teorisini hata ayıklama süreçlerindeki mevcut uygulamaları ve Paul Klemperer'in "açık artırma teorisi"ni [12] kullanarak açıklamıştır.

Anderson, kendi teorisini açıklarken açık artırma teorisinin "İdeal ve adil bir açık artırmada birbirinden farklı açık artırma yöntemlerinin aynı maddi sonuca ulaşacağı" tezinden hareketle, kod geliştirme yöntemi ne olursa olsun ideal durumda aynı oranda hata sayısı ve güvenlik açığına ulaşılacağı benzetmesini yapmaktadır.

Bu benzetmeyi destekler mahiyette, hata ayıklama sürecinde alfa testlerinin kaynak koda sahip yazılımcılar tarafından yapıldığını, devamında ise eğer yazılım kapalı kodlu ise beta testlerinin kodu görmeyen kullanıcılar tarafından yapıldığını anlatmaktadır. Fenton ve Neil'in çalışmasına referans veren Anderson, kaynak koda erişimi olmayan test edicilerin kaynak kodu görenlere göre gelişmiş sistemler için 3 ila 5 katı zaman harcadığını söylemektedir [13]. Buradan hareketle, ekonomik açıdan yapılan bir analiz ile maliyetli kod geliştiriciler yerine belirli bir hata oranı giderildikten sonra kalan testlerin beta test olarak dış dünyaya havale edilmesinin daha etkin olduğu ifade edilmektedir. Bu durum aslında kodu görmeden de (her ne kadar tersine mühendislik ile kod dönüştürülebilse de) aynı etkililikte hata ve güvenlik açığı ayıklamasının yapılabileceğini göstermektedir.

Anderson'a göre ideal dünyada açık ve kapalı kod eşit olduğu kadar aynı zamanda birbirlerine rakiptirler. Bu durum da dengenin oluşmasında önemli bir unsur olarak belirtilmektedir. Her iki taraf da birbirinin açığını bulmak için faaliyet göstermektedir. Açık kaynak kodu, kod geliştiriciler ve kullanıcıların hata ve açık bulmalarını kolaylaştırırken, karşı tarafın sızmaları için de aynı kolaylığı sağlamaktadır. Kapalı kod için tam tersi olan durum da yine aynı sonucu oluşturmaktadır. Anderson, çalışmasında testin etkililiğinden de bahsetmektedir. Windows kullanıcı sayısının 10-20 katı ve dolayısıyla test süresinin de 10-20 katı olduğunu ve bunun sonucunda da Windows'un aynı oranda güvenilir olması gerektiğini söylemektedir. Devamında ise böyle bir durumun olmadığını, sebebinin ise test etkililiğinin GNU/Linux lehine aradaki farkı kapattığını anlatmaktadır.Anderson, dengenin oluşmasına katkı sağlayan unsurlar arasında şirket seviyesindeki bazı politik insentifler olduğunu da söylemektedir. Şöyle ki: Amerikan hükümetinin bazı ürünler için hataların önce yetkili makamlara bildirilmesi tercihinden bahsederek, bildirilen açık kullanılarak saldırılar yapılmaya başlanana kadar ürün sahipleri tarafından hazırlanan yamaların yayımlanmamasının sağlandığı ve bu açıklardan yasaların uygulanması ve istihbarat amaçlı olarak yararlanılması yönünde kullanıldığını ifade etmektedir. Buna ilaveten, üreticilerin bir tercihi ve serbest rekabetin bir sonucu olarak ekonomik verimi sağlamak üzere, belirlenen açık sayısı belirli bir sayıya ve etkiye ulaşmadan yamaların dağıtımı yapılmamaktadır. Sahipli ve kapalı kodlar için geçerli olan bu tip uygulamalar da iki taraf arasında güvenlik açısından denge oluşturan faktörler arasında sayılmaktadır.

Tüm bu tartışma ve örneklerden sonra Anderson, yine de dünyanın ideal bir dünya olmadığını ve bu nedenle anlattıklarının kesin olarak kanıtlanmasının zor olduğunu sonuç olarak ifade etmektedir.

6. SONUÇ VE YORUMLAR

Mayıs-Haziran 2008 aylarında Gartner tarafından yapılan ve Asya Pasifik, Avrupa ve Kuzey Amerika'da faaliyet gösteren 274 organizasyonu kapsayan araştırma sonuçlarına göre bu şirketlerden %85'inde AKK yazılımlar kullanıldığı ve kalan %15'inde de bir yıl içinde AKK yazılımların kullanılmaya başlanacağı sonucuna ulaşılmıştır [14]. Bu bilgiler, aslında AKK yazılımların hayatımıza ne kadar hızlı ve yoğun biçimde girdiğini göstermektedir.

Bu konuda yapılan çalışmalar aslında kasıtlı olarak güvenlik açığı oluşturulmadığı varsayımı ile hareket edildiğinde açık ya da kapalı kod yazılımların aynı güvenlik seviyesine yakınsayacağı sonucuna bizi götürmektedir.

Ancak, AKK yazılımlarda kasıtlı olarak oluşturulan açıkların çok kısa sürede keşfedilerek düzeltildiği, kapalı kodlu yazılımlarda ise bu açıkların fark edilip düzeltilmesinin uzun süreler alabildiği örneklerle ortaya koyulmuş [15] olduğu unutulmamalıdır.

Bu değerlendirmeler ışığında, AKK yazılımlarında kapalı kodlu yazılımlar gibi güvenlik riskleri içerebileceğini göstermektedir. Burada asıl olan ve AKK yazılımları güvenlik açısından öne çıkaran unsur bu yazılımların fonksiyonlarının tam anlamı ile izlenebilir olması ve güvenlik testlerinden çok daha kolay geçirilebilmesi imkânıdır. Önemli olan bu imkânlardan uygun iş modelleri ve geliştirme yöntemi ile azami oranda yararlanma ortamının hazırlanmasıdır. Bu amaçla özellikle özgür yazılım kapsamında geliştirilen projeler için;

- Yazılım güvenliğine ilişkin bilgilendirme dokümanları hazırlanmalı ve kullanıcıların erişimine sunulmalı,
- Güvenlik açıklarının raporlanacağı eposta adresleri verilmeli ve bu kanaldan gelen bildirimler değerlendirilmeli,
- Güvenlik danışmanlığı alınmalı ve
- Kod geliştiricilerin güvenli yazılım geliştirme süreçlerini öğrenmesi sağlanmalıdır.

Yukarıda sayılan ön şartların yerine getirilmesi hem güvenlik açıklarının daha çok kişi tarafından incelenmesi ve bu incelemelerden azami oranda yararlanmayı sağlayacak, daha güvenli ürünlerin geliştirilmesinin önünü açacaktır.

Ülkemizde gerek AKK yazılımları gerekse lisanslı veya paralı yazılımların dengeli yaygınlaşmasının ülkemizin kaynaklarının daha efektif kullanılmasına uygun rekabetçi bir ortamın oluşmasına, markalara bağımlılığı ortadan kaldırılmasına, konuya ilgi duyan uzman sayısının arttırılmasına ve en önemlisi daha güvenli yazılımlar geliştirilmesine büyük katkılar sağlayacağı değerlendirilmektedir.

Sonuç olarak; yapılan çalışma genel olarak değerlendirldiğinde AKK yazılımların çok güvensiz olduğu veya lisanslı yazılımların çok güvenli olduğu gibi bir değerlendirme yapmak mümkün olmasa da lisanslı ve ücretli yazılım kullananların güvenliğin sağlanmasında daha önde olduklarını belirtmekte fayda vardır.

Unutmamalı ki yazılımların talep görmesinde en önemli faktör kullanıcı ihtiyaçlarının ne kadar karşılandığıdır. Elektronik ortamdaki güvenlik ihtiyacı konusunda kullanıcıların yeterince farkında ve duyarlı olmadıkları düşünülse de aslında her kullanıcının temel ihtiyaçlarının en başında gelmektedir.

Kaynaklar

[1] Matthew K. McGowan, Paul Stephens, Dexter Gruber, "An Exploration of the Ideologies of Software Intellectual Property: The Impact on Ethical Decision Making. Journal of Business Ethics", Vol. 73, 409-424 (2007).

[2] Karen Lynne Durell, "Intellectual Property Protection for Computer Software: How Much and What Form is Effective?", *International Journal of Law and Information Technology*, ISSN 0967-0769, Vol. 8 No.3, Oxford University Press (2000).

[3] M.Richard Stallman, "Free Software, Free Society", *Free Software Foundation*, ISBN: 1-882114-98-1 (2002).

[4] http://www.ozgurlukicin.com/haber/bir-yol-hikyesi-bilisim-08

[5] A.Soyak, "Küreselleşme Sürecinde Ulusal Teknoloji Politikası ve Türkiye: Sınai Mülkiyet Hakları ve Ar-Ge Teşvikleri Açısından Bir Çözümleme", *Bilim ve Teknik Yayınevi* (2000).

[6] B.Smith, "A Quick Guide to GPLv3", *Free Software Foundation Inc.* (2007).

[7] Pardus Projesi Sitesi, www.pardus.org.tr.

[8] OpenOffice.Org Projesi Sitesi, www.openoffice.org.

[9] MySQL Sitesi, www.mysql.com.

[10] Fortify Software Inc, "Open Source Security Study: How Are Open Source Development Communities Embracing Security Best Practices?" (2008).

[11] R. Anderson, "Open and Closed Systems are Equivalent (that is, in an ideal world)", Perspectives on Free and Open Source Software, ISBN 978-0-262-06246-6, 127-142. *The MIT Press.* (2005).

[12] P.Klemperer, Auction Theory: A Guide to the Literature. Journal of Economic Surveys Vol 13, No 3 (1999).

[13] N.E.Fenton, M.Neil, A Critique of Software Defect Prediction Models, *IEEE Transactions on Software Engineering* Vol 25, No 5, 675–689 (1999).

[14] Computerworld UK, Open Source in Every Business within 12 Months, *Gartner Report* (2008).

[15] C.Payne, On the Security of Open Source Software, *Information Systems Journal*, Vol 12, No 1, 61-78 (2002).

[16] Open Source Testing – Security http://www.opensourcetesting.org/security.php

[17] F.Özavcı, Bilgi Güvenliği Denetim Sürecinde Özgür Yazılımlar, www.gamasec.net/files/bgdsoy.pdf

[18] http://www.opensourcetesting.org/survey.php

[19] http://www.isecom.org/

[20] OWASP http://www.owasp.org

[21] http://www.sqaforums.com/postlist.php?Cat=0&Board=UBB19

[22] http://www.cigital.com/papers/download/bsi4-testing.pdf

[23] Sectools 2007 http://sectools.org

[24] GamaLAB, http://www.gamasec.net/gamalab.html

[25] Enderunix, http://www.enderunix.org/?lng=tr&page=papers