
GAZİ ÜNİVERSİTESİ
Gazi
1926

WEB UYGULAMA SALDIRILARINA YÖNELİK OTOMATİK TESPİT VE DÜZELTME SİSTEMİ

Durmuş AYDOĞDU

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2015

Durmuş AYDOĞDU tarafından hazırlanan "WEB UYGULAMA SALDIRILARINA YÖNELİK OTOMATİK TESPİT VE DÜZELTME SİSTEMİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

DanışmanProf. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Başkan : Prof. Dr. Önder Haluk TEKBAŞ

Savunma Yönetimi Anabilim Dalı, Kara Harp Okulu

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Üye : Yrd. Doç. Dr. Mehmet DEMİRCİ

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Tez Savunma Tarihi: 10/07/2015

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

…….…….

Prof. Dr. Şeref SAĞIROĞLU

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Durmuş AYDOĞDU
20/07/2015

WEB UYGULAMA SALDIRILARINA YÖNELİK OTOMATİK TESPİT VE DÜZELTME SİSTEMİ

(Yüksek Lisans Tezi)

Durmuş AYDOĞDU

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2015

ÖZET

İnternetin yaygınlaşmasıyla da beraber, web uygulamaları kişisel ve kurumsal ihtiyaçlar doğrultusunda çeşitli alanlarda kullanılmaya başlanmış ve özellikle de üzerinde hassas bilgi bulunan, web uygulamalarının güvenliği önemli hale gelmiştir. Bu tez çalışmasında, web uygulamalarının güvenliğinin arttırılması amacıyla, web uygulamalarının güvenlik durumu ve siber saldırıların kurumlara maliyeti incelenmiştir. Ayrıca, web uygulamalarında bulunan açıklıklar araştırılmış ve istismar edildiğinde web uygulamasının bütünlüğünü bozabilecek açıklıklara değinilmiştir. Bu açıklıkların istismar edilmesine karşı önlem olarak geliştirilen güvenlik çözümleri araştırılmış, ve elde edilen sonuçlar doğrultusunda, uygulamanın bütünlüğünü kontrol eden ve sağlayan özgün bir sistem önerilmiş ve gerçekleştirilmiştir. Önerilen sistem, herhangi bir saldırı ya da yetkisiz işlem sonucunda bütünlüğü bozulan web uygulamasının, bütünlüğünün bozulduğunun tespiti, bütünlüğün otomatik olarak tekrardan sağlanması ve tespit ettiği saldırı hakkında uyarı üretecek şekilde gerçekleştirilmiştir. Gerçekleştiren sistem WUBKU olarak adlandırılmış ve GPL sürüm 3 ile lisanslanarak herkesin kullanımına sunulmuştur. WUBKU'nun web uygulama güvenliği çözümlerine getirdiği yeni yaklaşımla bundan sonraki güvenlik çözümlerine yeni ufuklar açması ve web uygulama güvenliğine katkı sağlaması beklenmektedir.

Bilim Kodu : 902.1.014

Anahtar Kelimeler : Web uygulama, güvenlik duvarı, tasarım, güvenlik, açıklıklar, bütünlük kontrol, uygulama, WUBKU

Sayfa Adedi : 104

Danışman : Prof. Dr. Şeref SAĞIROĞLU

AUTOMATIC DETECTION AND CORRECTION SYSTEM AGAINST WEB APPLICATION ATTACKS

(M. Sc. Thesis)

Durmuş AYDOĞDU

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2015

ABSTRACT

Web applications have begun to be used for the personal and institutional needs in various fields with the spread of the Internet and especially the security of web applications, which provide an interface to sensitive information, increased in importance. In this thesis, the security of web applications and the cost of cyber attacks to the institutions are analyzed in an effort to increase the security of web applications. Also, vulnerabilities, found in web applications, are researched and vulnerabilities which can impair the integrity of web applications when exploited, are touched on in this thesis. Security solutions, improved as a precaution against the exploitation of these vulnerabilities are investigated. According to the results a novel system, checking and providing the integrity of the application, is suggested and implemented. The suggested system identifies the application whose integrity is exploited as a result of any attacks or unauthorized process, provides the integrity again automatically and delivers a warning about the identified attack. The system is called WUBKU and licensed with GPL version 3. Thus, WUBKU is presented to use for everyone. WUBKU brings a new approach to the web application security solutions. This approach is expected to open new horizons and make contributions to the future web application security solutions.

Science Code : 902.1.014

Key Words : Web application, firewall, design, security, vulnerabilities, integrity control, application, WUBKU

Page Number : 104

Supervisor : Prof. Dr. Şeref SAĞIROĞLU

TEŞEKKÜR

Çalışmalarım boyunca bana yol gösteren, destek ve yardımlarını benden esirgemeyen danışman hocam Sayın Prof. Dr. Şeref SAĞIROĞLU ile maddi ve manevi her türlü destekleriyle beni hiçbir zaman yalnız bırakmayan abim Ali AYDOĞDU ve yengem Ayşenur AYDOĞDU ile çok değerli aileme teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
DDR	Çift Veri Hızı (Double Data Rate)
D-WAV	Web Uygulama Güvenlik Açıklıkları Detektörü
FP	Hatalı Pozitif (False Positives)
GB	Gigabayt (Ölçü Birimi)
GHZ	Gigahertz (Ölçü Birimi)
GPL	Genel Kamu Lisansı
HTML	Hiper Metin İşaret Dili
HTTP	Hiper Metin Aktarım Protokolü
IDE	Bütünleşik Geliştirme Ortamı
IPAAS	Girdi Parametresi Analiz Sistemi
ISDFS	Uluslararası Adli Bilişim ve Güvenlik Sempozyumu
KB	Kilobayt (Ölçü Birimi)
LAMP	Linux Apache MySQL PHP
LDAP	Basit Dizin Erişim Protokolü
LTS	Uzun Dönem Destek
OWASP	Açık Kaynak Web Uygulama Güvenliği Projesi
MB	Megabayt (Ölçü Birimi)
MD	Mesaj Özeti (Message-Digest)
MEDULA	Medikal Ulak
MHZ	Megahertz (Ölçü Birimi)
MITM	Ortadaki Adam
PHP	Ön İşlemci Betik Dili
RIPEMD	Yarış Bütünlük İlkeleri Değerlendirme Özet Değeri
SHA	Güvenli Sağlama Protokolü
SMS	Kısa Mesaj Servisi
SSL	Güvenli Soket Katmanı

Kısaltmalar	Açıklamalar
SQL	Yapılandırılmış Sorgu Dili
TSL	Güvenli İletim Katmanı
ULAKBİM	Ulusal Akademik Ağ ve Bilgi Merkezi
UYAP	Ulusal Yargı Ağı Bilişim Sistemi
W3AF	Web Uygulama Saldırı ve İnceleme Çatısı
WASC	Web Uygulamaları Güvenlik Konsorsiyumu
WASP	Web Uygulama SQL Enjeksiyon Koruyucu
WUBKU	Web Uygulamaları Bütünlük Kontrol Uygulaması
WWW	Dünya Geniş Ağı
XSS	Çapraz Site Kod Çalıştırma

1. GİRİŞ

Bilgi teknolojisindeki hızlı gelişmeler sonucu internet, özellikle de WWW (Dünya Geniş Ağı) ile dünya üzerinde en çok kullanılan iletişim ortamlarından biri haline gelmiştir [1]. Web uygulamaları ise sunduğu imkanlar ile günlük yaşamın önemli bir parçası haline gelmiş ve bu şekilde web uygulamalarına olan bağımlılık gittikçe artmıştır [2,3]. Ayrıca, kurumsal web uygulamalarının sayısı da giderek artmakta ve çoğu kurum, işlemleri için yeni uygulama kullanmaya devam etmektedir [4]. İletişim Kullanımı Yönelim Araştırmasına (Communications Usage Trend Survey) göre, iş yaşamı web uygulamalarına giderek daha bağımlı hale gelmektedir ve web siteleri üzerindeki sosyal etkileşimin büyümeyi sürdürmesi beklenmektedir [5].

Web uygulamaları kişisel sayfalar, bilgi edinme, iletişim, sosyal ağlar, e-ticaret gibi çeşitli kişisel ihtiyaçlardan kullanılabildiği gibi UYAP (Ulusal Yargı Ağı Bilişim Sistemi), e-Devlet ve MEDULA (Medikal Ulak) gibi kurumsal ihtiyaçlardan da kullanılabilmektedir [1,3,6]. Kullanım alanı çok çeşitlilik gösteren web uygulamaları hemen hemen her türlü bilgiye erişimde, bu bilgilerin saklanması, işlenmesinde kullanılabilmektedir ve bu da web uygulamalarını kritik hale getirmektedir [2,3].

Web uygulamalarındaki hızlı büyüme, karmaşıklık ve uyumsuzluğun yanında güvenlik problemlerini de beraberinde getirmektedir [4]. Ayrıca, web uygulamaları yaygın şekilde kullanılmakta ve saldırıların hedefi haline gelmektedir [3,7,8]. Bu durum ise hem yaygın şekilde kullanılan hem de saldırıların hedefi olan web uygulamalarının güvenliğini önemli hale getirmektedir [9]. Web uygulamaları çeşitli sebeplerden dolayı saldırıların hedefi haline gelebilmektedir [2,6]. Web uygulamalarının saldırının hedefi haline getiren etkenler aşağıda maddeler halinde verilmiştir [3,7,10,11,12].

- Dünyanın herhangi bir yerinden erişilebilir olması,
- Uygulamalar üzerinde hassas veri bulunması ve veriler için arayüz sağlaması,
- Kullanım oranının yüksek olması,

Yukarıda da maddeler halinde belirtildiği gibi web uygulamalarının hedef olmasında birçok etken rol oynamaktadır. Saldırıların hedefi olan web uygulamalarına çeşitli kaynaklardan saldırı yapılabilmektedir [4]. Web uygulamalarına karşı yapılan saldırıların kaynakları aşağıda maddeler halinde verilmiştir.

- Dahili saldırganlar, sistem üzerinde yetkili kişilerdir.
- Yetkili kullanıcılar, sisteme kendini tanıtmış kişilerdir.
- Yetkisiz kullanıcılar, sisteme kendini tanıtmadan, uygulama üzerindeki yetkilendirme mekanizması ya da web sunucudaki açıklığı kullanan kişilerdir.

Web uygulamalarının hedef olmasında birçok etken bulunurken, web uygulamalarına karşı yapılan saldırı yöntemleri de çeşitlilik göstermektedir [13]. Bu saldırı yöntemleri ile web uygulamalarında bulunan açıklıklar kullanılarak uygulamalar istismar edilebilmektedir. Bir web uygulaması, web uygulamasının üzerinde çalıştığı web sunucusu, web uygulaması ile istemci arasındaki iletim protokolü ya da doğrudan web uygulaması üzerinde bulunan açıklık kullanılarak istismar edilebilmektedir [6]. Bu yüzden web uygulama güvenliğinin tam olarak sağlanabilmesi için, web uygulamasının üzerinde çalıştığı web sunucusunun, web uygulaması ile istemci arasındaki iletim protokolünün ve web uygulamasının güvenliği göz önünde bulundurulmalıdır.

Web uygulamaları üzerinde açıklık bulunmasının çeşitli sebepleri bulunmaktadır [1,3,9,11,14]. Bu sebepler aşağıda maddeler halinde verilmiştir.

- Yazılım geliştirme yaşam döngüsünün uygulamasındaki eksiklikler,
- Web uygulama güvenliğinde, sunucu ve iletim protokolü güvenliğinin göz ardı edilmesi,
- Uygulama güvenliği konusunda birçok çalışma yapılıyor olmasına rağmen, tüm bu çalışmaların takip edilmesinin zor olması,
- Güvenlik farkındalığındaki eksiklik,
- Tasarım kusurları,
- Uygulamadaki hatalar,
- Web uygulama pazarının hızlı gelişmesi,

- Güvenlik konusunda yeterli bilgi ve tecrübesi olmayan web geliştiricisi, yöneticisi ve kullanıcısının çok sayıda bulunması,
- Uygulama geliştiricilerinin güvenli kod geliştirmedeki yetersizliği,
- Uygulama geliştirmede güvenliğin ihmal edilmesi,

Yukarıda da belirtildiği gibi web uygulamaları çeşitli alanlarda yaygın şekilde kullanılmaktadır ve saldırıların hedefi durumundadır. Yaygın şekilde kullanılan ve saldırıların hedefi halinde gelen web uygulamalarını güvenlik açısından inceleyen ve güvenlik durumunu ortaya koyan çeşitli raporlar bulunmaktadır. Raporlarda yer alan verilere göre, web uygulamaları üzerinde birçok açıklık bulunmaktadır [15-18]. Bu açıklıkların istismar edilmesi ise kişi ya da kurumlar için maddi ve manevi sonuçlar doğurabilmektedir [18].

Bu çalışmada, web uygulamalarında bulunan güvenlik açıklıklarından ve mevcut güvenlik çözümlerinden yola çıkılarak, web uygulama güvenliğine katkı sağlamaya çalışılmıştır. Bu kapsamda, web uygulamalarının yetkisiz şekilde değişip değişmediğinin tespiti için bir sistem önerilmiştir. Önerilen sistem temel alınarak, herhangi bir saldırı ya da yetkisiz iyi veya kötü niyetli bir işlem sonucunda, bütünlüğü bozulan web uygulamasının bütünlüğünün bozulduğunun tespitini yapan bir uygulama geliştirilmeye çalışılmıştır. Ayrıca, uygulamaya, bütünlüğü bozulan uygulamanın bütünlüğünü otomatik olarak tekrardan sağlama ve bütünlüğün bozulduğu ve düzeltildiğiyle ilgili bir uyarı üretme gibi ilave özellikler kazandırılmaya çalışılmıştır. Önerilen sistemin ve gerçekleştirilen uygulamanın web uygulama güvenliğine katkı sağlayacağı değerlendirilmektedir.

Bu tezin ikinci bölümünde, web uygulamaları üzerindeki açıklıklar verilmiş ve bu açıklıkların istismar edilmesi sonucu web uygulamasının bütünlüğünün bozulmasına sebep olabilecek tehditlere yer verilmiştir. Ayrıca, web uygulamalarının güvenlik durumunu ve siber saldırıların kurumlara maliyetini oraya koyan raporlar özetlenmiştir. Üçüncü bölümde ise web uygulama güvenliği hakkında detaylı bir literatür taraması yapılmış ve elde edilen bulgular özetlenmiştir. Dördüncü bölümde, tez kapsamında geliştirilen sistem detaylı olarak sunulmuştur. Tezin beşinci bölümünde ise gerçekleştirilen sistemin test çalışmalarına yer verilmiştir.

Son bölümde ise bu tez kapsamında yapılan inceleme ve araştırmalar sonucunda elde edilen bilgilerden ve bulgulardan yola çıkılarak yapılan çalışmanın genel bir değerlendirilmesi yapılmış ve geliştirilen uygulamanın web uygulama güvenliğine sağlayacağı katkılar üzerinde durulmuştur.

2. WEB UYGULAMA GÜVENLİĞİ AÇIKLIKLARI

Web uygulamaları kullanımı günlük yaşama birçok kolaylık getirirken, kurumlar için her zamankinden daha maliyetli ve riskli hale gelmektedir [16]. Bu uygulamaların güvenliğinin arttırılması amacıyla, WASC (Web Uygulama Güvenliği Birliği (Web Application Security Consortium)) ve OWASP (Açık Kaynak Uygulama Güvenliği (Open Web Application Security Project)) kurumları çalışmalar yapmaktadır [19,20].

WASC tarafından 2010 yılında yayınlanan "WASC Tehdit Sınıflandırması Sürüm 2.0'da (WASC Threat Classification v2.0)" web site güvenliği için tehditler ve açıklıklar ele alınmaktadır. Bu belgede yer alan açıklıklar Çizelge 2.1'de verilmiştir.

Çizelge 2.1. WASC web site açıklıkları [13]

Sıra Nu.	Açıklıklar
1.	Uygulama Yanlış Yapılandırma (Application Misconfiguration)
2.	Dizin İndexleme (Directory Indexing)
3.	Uygun Olmayan Dosya Sistemi İzinleri (Improper Filesystem Permissions)
4.	Uygun Olmayan Girdi İşleme (Improper Input Handling)
5.	Uygun Olmayan Çıktı İşleme (Improper Output Handling)
6.	Bilgi Sızıntısı (Information Leakage)
7.	Güvenli Olmayan İndexleme (Insecure Indexing)
8.	Uygun Olmayan Anti-otomasyon (Insufficient Anti-automation)
9.	Yetersiz Kimlik Denetimi (Insufficient Authentication)
10.	Yetersiz Yetkilendirme (Insufficient Authorization)
11.	Yetersiz Şifre Kurtarma (Insufficient Password Recovery)
12.	Yetersiz Süreç Doğrulama (Insufficient Process Validation)
13.	Yetersiz Oturum Bitiş Süresi (Insufficient Session Expiration)
14.	Yetersiz İletim Katmanı Koruması (Insufficient Transport Layer Protection)
15.	Sunucu Yanlış Yapılandırma (Server Misconfiguration)

OWASP tarafından, web uygulama güvenliğinde farkındalığın arttırılmasına katkı sağlamak amacıyla, web uygulama güvenliğinde en kritik açıklıklar listesi yayınlamaktadır

[19]. OWASP en bilindik açıklıklar 2013 listesinde yer alan açıklıklar Çizelge 2.2'de verilmiştir.

Çizelge 2.2. OWASP 2013 Açıklıklar Listesi [19]

Sıra Nu.	Açıklık
1	Enjeksiyon (Injection)
2	Kırık Kimlik Doğrulama ve Oturum Yönetimi (Broken Authentication and Session Management)
3	Çapraz Site Kod Çalıştırma (Cross-Site Scripting (XSS))
4	Güvensiz Doğrudan Nesne Başvurusu (Insecure Direct Object References)
5	Güvenlik Yanlış Yapılandırma (Security Misconfiguration)
6	Hassas Veriyi Açıkta Bırakma (Sensitive Data Exposure)
7	İşlev Seviyesi Erişim Kontrol Eksikliği (Missing Function Level Access Control)
8	Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery (CSRF))
9	Bilinen Açıklık Bileşenlerini Kullanma (Using Known Vulnerable Components)
10	Doğrulanmayan Yönlendirme ve İletme (Unvalidated Redirects and Forwards)

Tez kapsamında önerilen sistem, web uygulamalarının bütünlük kontrolü üzerinden çalışan bir güvenlik çözümü sunmaktadır. Önerilen sistemde, bütünlük kontrolü özet değeri üzerinden yapılmaktadır. Korunan web uygulamasının dosya ve dizin yapısının özet değeri özet algoritması kullanılarak hesaplanmakta ve bu özet değerin değişip değişmediğinin kontrolü üzerinden, web uygulamasının dosya ve dizin yapısının bütünlüğü kontrol edilmektedir. Bu sebeple, WASC ve OWASP tarafından yayınlanan açıklıklardan istismar edildiğinde, uygulamanın bütünlüğünün bozulmasına sebep olabilecek açıklıklar bu bölümde ele alınmıştır. Ayrıca, açıklığın nasıl istismar edilebileceğine değinilmiştir.

2.1. Uygulama Yanlış Yapılandırma (Application Misconfiguration)

Web uygulamaları üzerinde bulunan yapılandırma açıklığıdır [13]. Web uygulamalarının gereksiz özellikler ve güvenli olmayan özelliklerinin bir araya gelmesinin bir sonucu olarak, bu özellikler üzerinden kötü niyetli kişiler, kimlik denetim yöntemlerini atlatma ve bir ihtimal yükseltilmiş ayrıcalıklarla hassas bilgilere erişim imkânı sağlayabilmektedir [13]. Bu açıklık istismar edildiğinde, erişilen alanın sistem üzerindeki yetkileri doğrultusunda sisteme zarar verilebilir. Bu açıklığa, kurulumla gelen varsayılan kullanıcı adı ve şifreler, kodun içinde bulunan arka kapı (backdoor) hesapları, özel erişim mekanizmaları ve web sunucu üzerinden erişilebilen dosya izinlerinin yanlış olması durumları örnek olarak verilebilir [13].

2.2. Uygun Olmayan Dosya Sistemi İzinleri (Improper Filesystem Permissions)

İşletim sistemi üzerindeki, dosya, dizin ve sembolik bağlantı izinlerinin doğru olmaması sonucu ortaya çıkan açıklıktır [13]. Bu açıklığın istismar edilmesi, uygulamaların gizliliği, bütünlüğü ve erişilebilirliğini tehdit edebilmektedir [13]. İşletim sistemi üzerindeki izinlerin düzgün ayarlanmaması sonucu, yetkisiz kişiler tarafından, erişilmemesi gereken dosya ve dizinler üzerinde okuma, silme ve değiştirme işlemlerinin yapılması mümkün olabilmektedir [13]. Bir sunucu üzerinde farklı yetkilerin kullanıcılara işe göre verilmemesi durumunda, yetkilerin kötüye kullanılması ile başka kullanıcılara ait dosyaların değiştirilmesi bu açıklığın istismar edilmesine örnek olarak verilebilir.

2.3. Uygun Olmayan Girdi İşleme (Improper Input Handling)

Günümüzde uygulamaların karşılaştığı en bilindik açıklıklardan biridir [13,19]. Sistemde ve uygulamalar üzerindeki kritik açıklıkların arkasında girdi kontrolünün zayıf olması bulunmaktadır [13]. Uygulamalar çeşitli kaynaklardan, çeşitli biçimlerde girdi almaktadır. Alınan bu girdilerin kaynağına ve biçimine bakılmaksızın bütün girdiler güvensiz ve potansiyel olarak kötücül kabul edilmelidir [13]. Güvenilir olmayan girdinin kullanımı sonucu SQL (Yapılandırılmış Sorgu Dili) komutları, LDAP (Basit Dizin Erişim Protokolü) komutları ve işletim sistemi komutları uygulama üzerinden çalıştırılması bu açıklığın istismar edilmesine örnek olarak verilebilir [13,19,21].

2.4. Sunucu Yanlış Yapılandırma (Server Misconfiguration)

Web sunucu ya da uygulama sunucu üzerindeki yanlış yapılandırmadan dolayı ortaya çıkan açıklıktır [13]. Birçok sunucu yapılandırma dosyaları, betikler (script) ve web sayfalarını içeren gereksiz varsayılan ve örnek sayfalar ile gelmektedir [13]. Uygulama güvenliğinin sağlanabilmesi için web sunucu, uygulama sunucu, veritabanı sunucusu gibi ortamlar üzerindeki bir açıklığın istismar edilesi sonucu uygulamanın da istismar edilmesine neden olabilecek ortamlar için güvenlik ihtiyaçları tanımlanmalı ve uygulanmış olmalıdır [19]. Bu ortamlar ve üzerinde çalışan yazılımların güncel tutulması gerekmektedir [19]. Apache Tomcat web sunucu uygulaması kurulduğunda varsayılan içerik yönetimi ile gelmektedir. Bu içerik yönetim alanın varsayılan kullanıcı adı ve şifrenin değiştirilmemesi sonucu, yetkisiz kişiler tarafında sunucu üzerine uygulama yüklenebilir. Bu şekilde kötücül uygulama yüklenerek sistemin istismar edilmesi bu açıklığın istismar edilmesine örnek olarak verilebilir.

2.5. Bilinen Açıklık Bileşenlerini Kullanma (Using Known Vulnerable Components)

Web uygulaması tarafından kullanılan, modüller, çatılar (frameworks) ve kütüphaneler gibi bileşenlerdeki açıklıklar üzerinden, uygulamanın istismar edilmesine sebep olan açıklıklardır [19]. Bu bileşenlerin üzerindeki açıklığın istismar edilmesi, bileşenin sistem üzerindeki yetkileri doğrultusunda, sistem üzerinde veri kaybı ya da sunucunun ele geçirilmesine sebep olabilir. Uygulamalarda bilindik açıklıkları olan bileşenlerin kullanılması uygulamanın güvenlik seviyesini düşürebilir ve mümkün saldırı alanları ve etkilerine olanak sağlamaktadır [19]. Uygulamada kullanılan üçüncü parti bir yazılım üzerindeki açıklık üzerinden, uygulamanın istismar edilmesi durumu bu açıklığın istismar edilmesine örnek olarak verilebilir.

2.6. Yetersiz Kimlik Denetimi (Insufficient Authentication)

Yetersiz kimlik denetimi açıklığı, web uygulaması üzerinde düzgün kimlik denetiminin yapılmaması sonucu, kötü niyetli kişilerin hassas içeriğe erişmesine sebep olan bir açıklıktır [13]. Kimlik denetiminin düzgün yapılmaması sonucu, web tabanlı yönetim alanlarına erişim bu açıklığın istismar edilmesi durumuna örnek olarak verilebilir ve erişilen alanın yetkileri doğrultusunda sisteme zarar verilebilir [19].

2.7. Yetersiz Yetkilendirme (Insufficient Authentication)

Yetersiz yetkilendirme açıklığı, kullanıcıların yaptığı işlemler ve eriştiği verilerde yetkili olup olmadığının kontrolünün yapılmaması sonucu ortaya çıkmaktadır [13]. Bir işlem yapılırken, işlemi yapmak isteyen kaynağın, işlemi yapmaya yetkili olup olmadığının kontrol edilmemesi, yapılan işleme göre sisteme zarar vermektedir. Yetkilendirme işlemlerinde, kullanıcı, servis ya da uygulamanın işlemi yapıp yapmaya yetkili olup olmadığının kontrolü yapılmalıdır [13]. Yapılmadığı durumlarda, bir kullanıcının başka bir kullanıcı dosyalarına erişimi ve bu dosyalar üzerinde işlem yapması durumu bu açıklığın istismar edilmesine örnek olarak verilebilir.

2.8. Yetersiz İletim Katmanı Koruması (Insufficient Transport Layer Protection)

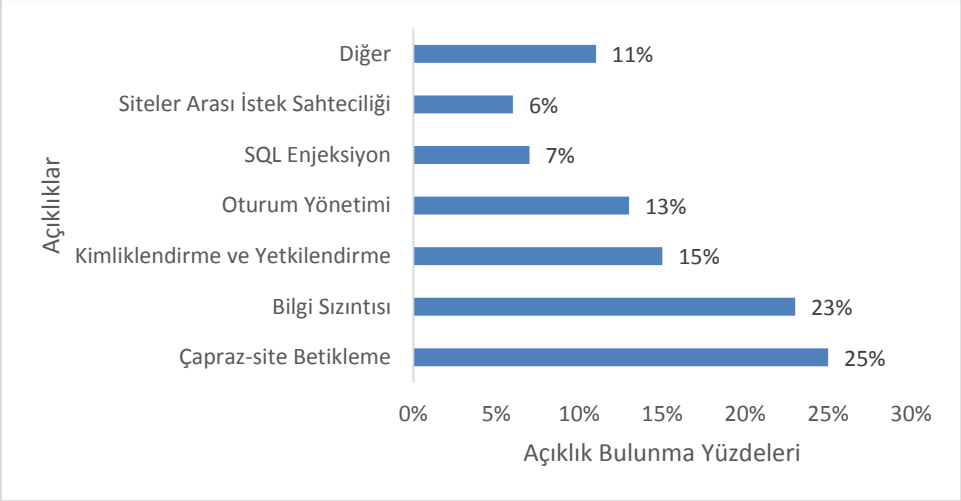
Yetersiz iletim katmanı koruması açıklığı, sunucu ile istemci arasındaki iletişimin açığa çıkmasına sebep olan bir açıklıktır [13]. Açıkta olan iletişim bilgileri, kötü niyetli kişiler tarafından dinlenerek, kullanıcı adı ve şifre bilgilerinin alınması ya da veri iletişimi esnasında araya girilerek verinin değiştirilmesi gibi saldırılarda kullanılabilmektedir [13]. Bu katmanda koruma sağlanabilmesi için SSL/TSL kullanılarak şifreleme sağlanmalıdır. MITM (Ortadaki Adam) saldırısıyla, sunucu ile istemci arasındaki trafik bilgisinin ele edilmesi sonucu, elde edilen bilginin önemi ve kullanım biçimi doğrultusunda sisteme zarar verilebilmektedir [13]. MITM saldırıları bu açıklığın istismar edilmesine örnek olarak verilebilir.

2.9. İşlev Seviyesi Erişim Kontrol Eksikliği (Missing Function Level Access Control)

Bu açıklık, uygulamalar üzerindeki fonksiyonlara erişim sağlanırken, sunucu tarafından sağlanan erişimlerin kontrolünün düzgün yapılmaması sonucu ortaya çıkmaktadır [19]. Sunucu üzerindeki fonksiyonlara erişimde sunucu tarafından kontrollerinin yapılması gerekmektedir. Kontrol yapılarak, yapılmak istenen işlemin saldırgandan mı yoksa yetkili kişiden mi geldiği doğrulanmazsa; saldırganların, yetkisi olmadığı işlemleri, kimlik denetimine girmeden sahte isteklerle yapması mümkün olabilmektedir [19]. Şifre değiştirme işleminde, kullanıcı tanımlayıcı bilgisi değiştirilerek başka bir kullanıcıya ait şifre bilgisinin değiştirilmesi bu açıklığın istismar edilmesine örnek olarak verilebilir.

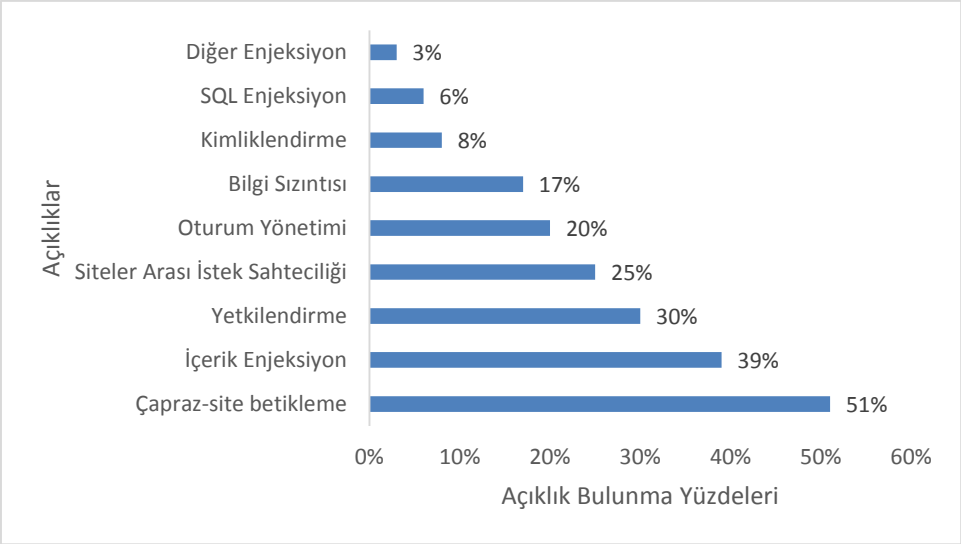
Web uygulamalarını güvenlik açısından inceleyen ve güvenlik durumunu ortaya koyan çeşitli raporlar da bulunmaktadır. İleriki bölümlerde bu raporlardan elde edilen istatistiki bilgilere yer verilmiştir.

Mobil, bulut ve web uygulama güvenliği üzerine çalışan bir şirket olan CENZIC'in yayınladığı "Uygulama Güvenliği Açıklık Eğilimleri Raporu 2014 (Application Vulnerability Trends Report 2014)"e göre; 2013 yılında test edilen uygulamaların %96'sında bir ya da birden fazla kritik açıklık bulunmaktadır [15]. CENZIC değerlendirdiği uygulamalar üzerinde çeşitli açıklıklar tespit etmiştir [15]. Tespit edilen bu açıklıklar ve oranları Şekil 2.1'de gösterilmektedir.

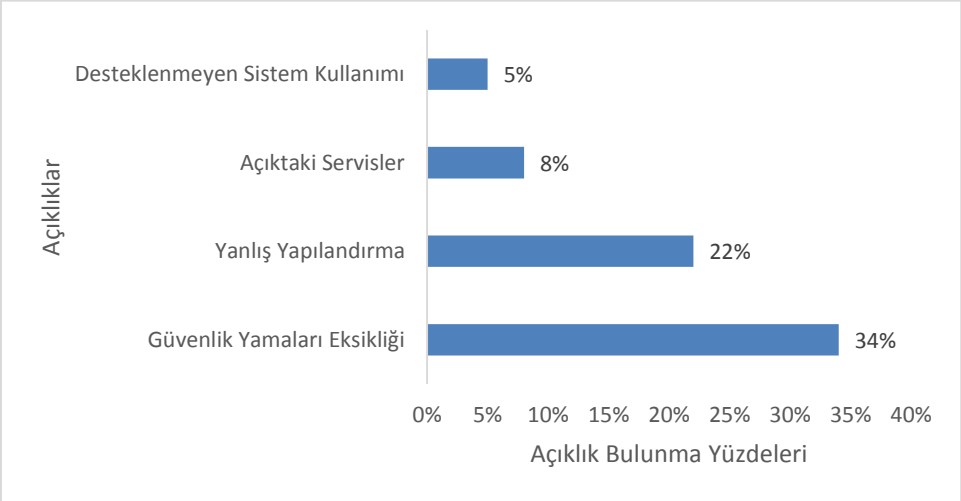


Şekil 2.1. CENZIC tarafından web uygulamalarında tespit edilen açıklıklar ve bulunma oranları [15]

Edgescan, bulut tabanlı açıklık yönetim servisi olup, hem uygulamalar hem de uygulamaların üzerinde çalıştığı sunucu üzerindeki açıklıkları tespit etmeye yardımcı olmaktadır [16]. "Açıklık İstatistikleri Raporu 2014 (2014 Vulnerability Statistics Report)"te edgescan tarafından geçmiş 12 ayda bulunan açıklık türleri ele alınmıştır [16]. Raporda açıklıklar değerlendirilirken, web uygulamasının üzerinde çalıştığı sunucu ile sunucu ve istemci arasındaki iletişimin güvenlik durumu da değerlendirilmektedir [16]. Edgescan tarafından yayınlanan raporda uygulama katmanı üzerinde bulunan açıklıklar Şekil 2.2'de, sunucu üzerinde bulunan açıklıklar ise Şekil 2.3'te gösterilmektedir.



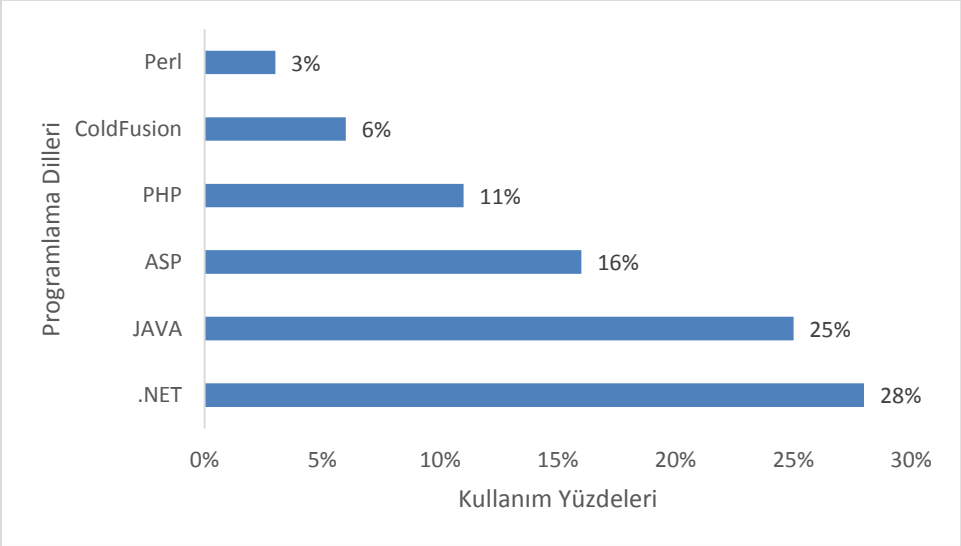
Şekil 2.2. Edgescan tarafından uygulama katmanında bulunan açıklıklar ve oranları [16]



Şekil 2.3. Edgescan tarafından sunucu katmanında bulunan açıklıklar ve oranları [16]

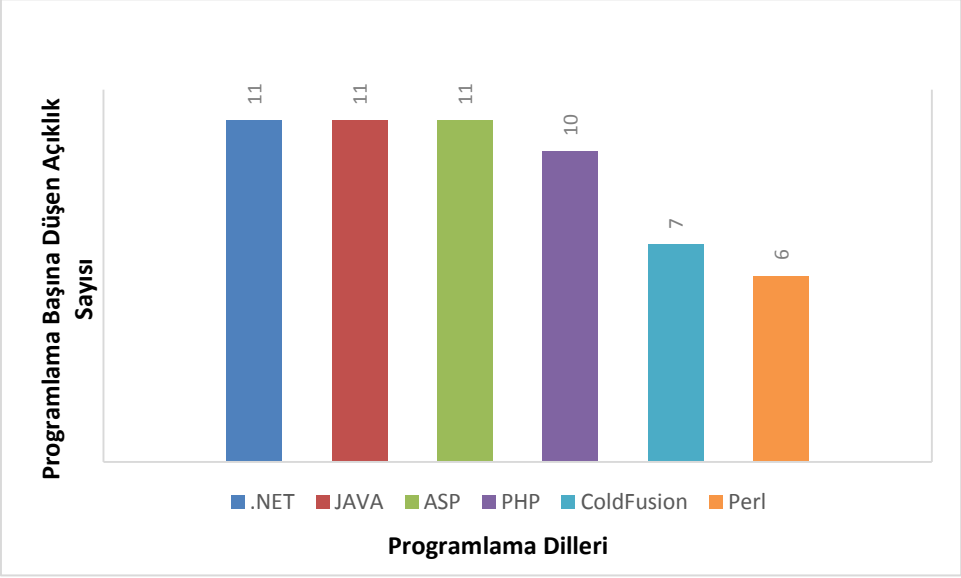
Ayrıca raporda, sunucu üzerindeki en yaygın açıklığın SSL (Güvenli Soket Katmanı) ya da TSL (Güvenli İletim Katmanı) şifreleme ile verinin iletim güvenliğiyle ilgili olduğu da ortaya konulmaktadır. Değerlendirilen her iki sunucudan birinde, SSL/TSL şifreleme ile ilgili en az bir tane yüksek ya da orta seviye risk bulunduğu görülmektedir[16]. Bu da sunucuların %50'sinin istemci tarayıcısı ile sunucu arasındaki iletişimin saldırganların müdahalesine açık olduğu anlamına gelmektedir [16].

Web uygulama güvenliği konusunda çözüm sağlayan WhiteHat Security şirketinin "2014 Web Site Güvenliği İstatistikleri Raporu (2014 Website Security Statistics Report)" na göre, web uygulamalarının geliştirildiği programlama dillerinin kullanım oranları Şekil 2.4'te gösterilmektedir.



Şekil 2.4. Web uygulamalarında programlama dillerinin kullanım oranları [17]

Ayrıca aynı raporda yer alan, her bir programlama dilinde geliştirilen uygulamaların üzerindeki açıklık dağılımı Şekil 2.5'te gösterilmektedir.



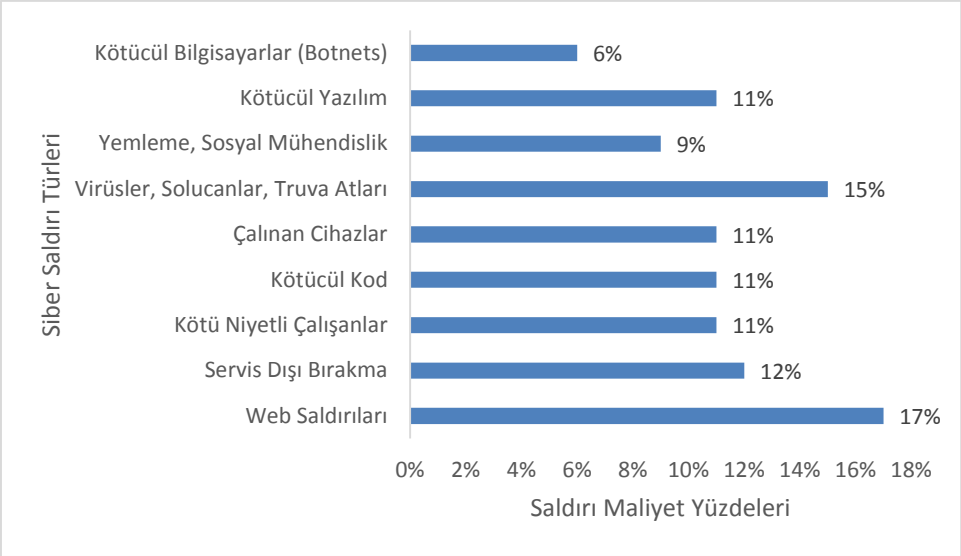
Şekil 2.5. Programlama dili başına düşen açıklık sayıları [17]

Web uygulamalarının güvenlik durumunu ortaya koyan raporlardan da anlaşıldığı üzere, web uygulamaları üzerinde yüksek oranda çeşitli açıklıklar bulunmaktadır. Bu açıklıklar istismar edildiğinde finansal kayıplar, bilgi kaybı ya da itibar kaybı gibi maddi ve manevi zararlara yol açmaktadır[16,18]. Bu zararların neler olabileceği ve nelere yol açabileceği hakkında fikir sahibi olabilmek için, siber suçların maliyeti hakkında bilgi içeren, "Siber Suçların Küresel Maliyeti 2014 Raporu (2014 Global Report on the Cost of Cyber Crime)" verilerinden yararlanılmıştır. Raporda yer alan veriler oluşturulurken 7 farklı ülkede yer alan çeşitli 257 kurum, Birleşik Devletler (United States) kurum çalışanlarıyla yapılan 2081 röportaj ve toplam maliyetin hesaplanması için 1717 siber saldırı verilerinden yararlanıldığı belirtilmektedir [18].

Rapora göre, genel olarak siber suçların maliyetine bakıldığında, 257 kurumun yıllık maliyeti 7,6 milyon dolardır ve bu maliyet yıllık şirket başına 0,5 milyon dolarla 61 milyon dolar arasında değişmektedir. Maliyetin 2013 yılında 7,2 milyon dolar olduğu göz önünde bulundurulursa, maliyetin %10,4 arttığı görülmektedir [18].

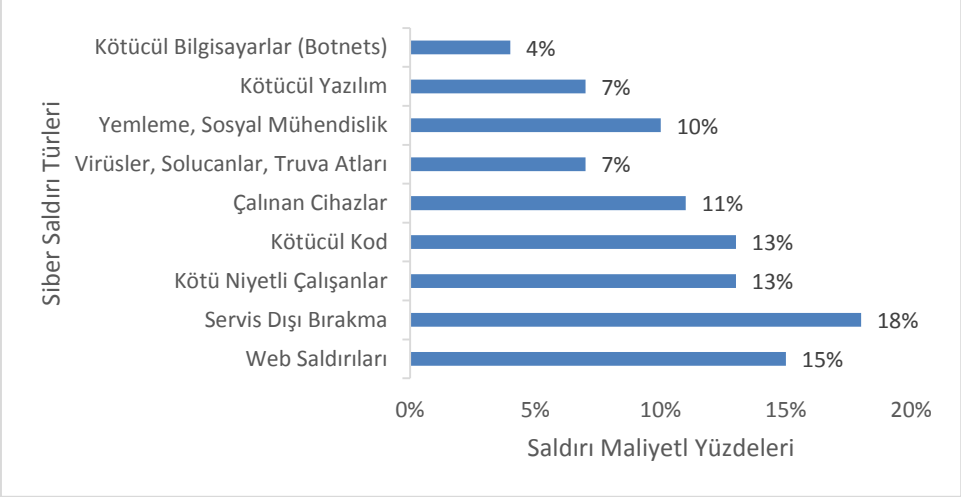
Siber saldırıların en maliyetlilerine bakılacak olursa, kötü niyetli kurum çalışanları, servis dışı bırakma ve web tabanlı saldırılar gelmektedir. Bu suçlar yıllık olarak kurum başına tüm siber suç maliyetinin %55'inden daha fazlasını oluşturmaktadır [18].

Saldırıların maliyeti şirketlerin büyüklüklerine göre farklılık göstermektedir [18]. Ayrıca, aynı saldırı farklı büyüklükteki şirketler için farklı maliyet oluşturabilmektedir. Farklı saldırıların küçük şirketler için maliyeti Şekil 2.6'da gösterilmektedir. En fazla maliyeti sırasıyla web tabanlı saldırılar, virüsler, solucanlar ve truva atları ile servis dışı bırakma saldırıları oluşturmaktadır [18].



Şekil 2.6. Belirli saldırıların küçük şirketler için maliyet dağılımı [18]

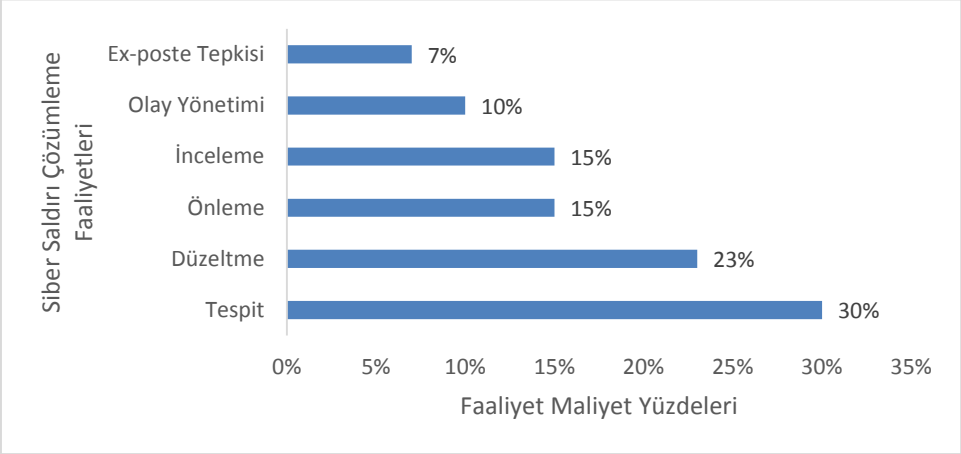
Yine aynı saldırı çeşitlerinin büyük şirketler için maliyet oranı Şekil 2.7'de gösterilmektedir. Şekilde de görüldüğü üzere, en fazla maliyeti sırasıyla servis dışı bırakma saldırıları, web tabanlı saldırılar, kötü niyetli çalışanlar ve kötücül kod saldırıları oluşturmaktadır [18].



Şekil 2.7. Belirli saldırıların büyük şirketler için maliyet dağılımı [18]

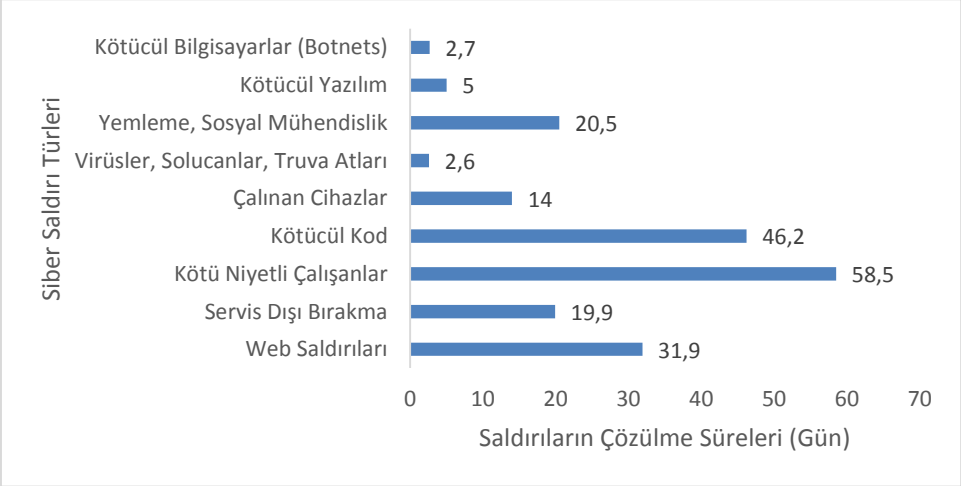
Saldırılardan korunmak için çeşitli güvenlik önlemlerinin alınması maliyetli olduğu gibi, bir saldırı sonrası saldırının verdiği zararın düzeltilmesi, saldırının incelenmesi gibi çeşitli maliyetlerde bulunmaktadır. Bir saldırının çözülmesindeki faaliyetlerin maliyet dağılımı Şekil 2.8'de gösterilmektedir. Şekilde de görüldüğü üzere, en fazla maliyeti tespit ve

düzeltme oluşturmaktadır. Tespit ve düzeltme maliyeti toplam faaliyet maliyetinin %53'üne tekabül etmektedir [18].



Şekil 2.8. Saldırıların çözülmesindeki faaliyetlerin maliyet oranları [18]

Siber saldırılar hızlı bir şekilde çözülmezse daha maliyetli olabilmektedir [18]. Siber saldırıların ortalama çözülme zamanları gün bazında Şekil 2.9'da gösterilmektedir. Şekilde görüleceği üzere, kötü niyetli çalışanlar, kötücül kod ve web saldırılarının sırasıyla çözülmesi için geçen süre en fazladır.



Şekil 2.9. Saldırıların çözülmesi için geçen süre bilgisi [18]

3. LİTERATÜR TARAMASI

Bu tez çalışması kapsamında, web uygulama güvenliği için çözüm sağlayan akademik çalışmalar taranmış ve incelenmiştir. Yapılan çalışmalarda web uygulamalarının güvenliğinin sağlanması için; uygulama kodunun güvenliği, yazılım yaşam döngüsü ile güvenliğin bütünleşik hale getirilmesi, güvenlik testleri, uygulama katmanı güvenlik duvarı gibi birçok güvenlik çözümü sunulmaktadır [8,11,12,22-32]. Bu güvenlik çözümlerinden bazıları, web uygulama güvenliği konusunda sadece bir soruna çözüm getirirken, birden fazla güvenlik sorununa çözüm getiren güvenlik çözümleri de bulunmaktadır. İleriki bölümlerde bu güvenlik çözümlerine değinilmiştir.

Web uygulamalarında uygulama kodunun güvenli geliştirilmesine yönelik birçok güvenlik çözümüyle karşılaşılmıştır. Bu güvenlik çözümleri arasında, OWASP'ın en bilindik açıklıklar listesinin birinci sırasında yer alan enjeksiyon açıklıklarından biri olan SQL enjeksiyon ve üçüncü sırada yer alan XSS (Çapraz Site Kod Çalıştırma) açıklıklarının istismar edilmesini engellemeye yönelik çözümlerin daha çok öne çıktığı görülmektedir.

SQL enjeksiyon açıklığının istismar edilmesine karşı önlem olarak, giriş parametre veri türünün otomatik tespitine dayanan, IPAAS (Girdi Parametresi Analiz Sistemi) olarak adlandırılan özgün bir teknik sunmuştur. Bu teknik sayesinde, web uygulama parametreleri test esnasında öğrenilmiş ve sistemin çalışmasına etki etmeden, çalışma esnasında parametreleri kontrol etmiştir [11].

Başka bir çözüm ise WASP (Web Uygulama SQL Enjeksiyon Koruyucu (Web Application SQL Injection Preventer)) aracını, var olan mevcut tekniklerin kavramsal ve pratik avantajları üzerine kurulu yüksek otomasyonlu yeni bir yaklaşımla sunmuştur. Bu yaklaşım üzerinden geliştirilen WASP aracı, SQL enjeksiyon açıklığını istismar eden saldırıları karşı FP (Hatalı Pozitif) uyarı üretmeden etkin şekilde koruma sağlamıştır [22].

Diğer bir çözüm ise SQL enjeksiyon ve XSS açıklıklarının istismar edilmesine karşı önlem olarak, girdi temizleme ve doğrulama kod biçimlerinin tanımlandığı statik kod öznitelikleri (attributes) üzerinden, açıklık öngörücülerin oluşturulduğu bir model önermiştir. Önerilen

öznitelikler üzerinden oluşturulmuş açıklık öngörücüler, yapılan testlerde düşük FP uyarı oranı ile açıklıkların %80'inden daha fazlasını tespit etmiştir [23].

Başka bir çözüm ise geliştiricinin farkındalığının arttırılması ve güvenli programlama tekniğine teşviki için IDE (Bütünleşik Geliştirme Ortamı) içerisinde programcıya güvenli programlama tekniklerini etkileşimli olarak hatırlatan bir uygulama önermiştir [8].

Web uygulamalarının geliştirilmesi esnasında güvenli geliştirilmesine yönelik çözümler olduğu gibi, geliştirilen bir web uygulamasının güvenli olup olmadığının ortaya konulması için, web uygulamalarının güvenlik testleri üzerine yapılmış çalışmalar ve ortaya konulmuş çözüm önerileri de bulunmaktadır. Bu çözümlerden biri, kırık yetkilendirme ve oturum yönetimi açıklığının var olup olmadığını otomatik tespit eden bir teknik önermiştir. Önerilen teknik, web uygulaması hakkında detaylı bilgiye ya da herhangi bir kurulum gerektirmeyen, uygulama hakkında birkaç temel bilginin test operatörü tarafından girilmesiyle, gerçek saldırıların benzeri yapılarak oturum yönetimi açıklıklarının tespitini otomatik olarak yapmıştır [24].

Bir başka yöntem ise web uygulamalarındaki enjeksiyon açıklıkların tespiti için dinamik ve statik yaklaşımların güçlü yanlarını birleştiren hibrid analiz çatısı yaklaşımını sunmuştur. Sunulan yaklaşım uygulanarak Phan olarak adlandırılan bir araç haline getirilmiştir. Araç, statik olarak PHP (Ön İşlemci Betik Dili) bytecode'larını analiz ederek tehlikeli kod ifadelerini aramaktadır. Bu şekilde dinamik analiz aşamasın da ise bu ifadeler izlenmektedir [12].

Başka bir çözüm ise giriş değerlerinin alındığı web formları üzerinden açıklıkların tespiti ile özgün bir yaklaşım sunmuştur. Öncelikle verilen bir web formunun karakteristiğini analiz etmiş ve formdaki alanlar için test verileri oluşturmuştur. Bu test verilerini form üzerinde çalıştırmış ve gelen HTTP (Hiper Metin Aktarım Protokolü) yanıt kodları ve HTML (Hiper Metin İşaret Dili) cevaplarına dayalı olarak sonuçları analiz etmiştir. Yaklaşım D-WAV (Web Uygulama Güvenlik Açıklıkları Detektörü) adlı bir araç olarak uygulamış ve sonuçlar yaklaşımın SQL enjeksiyon ve XSS açıklıkları gibi web uygulama açıklıklarını hızlı, otomatik ve etkin bir şekilde tespit ettiğini göstermiştir [25].

Bir başka çözüm ise web uygulamalarındaki dizi (string) ile ilgili muhtemel açıklıkların tespiti, görüntülenmesi ve kapatılmasına çevirim içi olarak imkân sağlayan yeni bir web servis ortamı sunmuştur [26].

Diğer bir çözüm önerisi ise W3AF (Web Uygulama Saldırı ve İnceleme Çatısı (Web Application Attack and Audit Framework)) isimli web uygulama güvenlik açıklık tarayıcısı olan bir aracın tarama modüllerini ve betiklerini özelleştirmiştir. Bu şekilde web uygulama güvenlik açıklık tarayıcısı servisi tasarlamış ve web siteleri için güvenlik tarayıcısı ve değerlendirme servisi sağlamıştır [27].

Web uygulama güvenliğinde sorunlardan biri de, uygulama ve uygulamanın üzerinde çalıştığı sunucu üzerindeki yapılandırma hatalarıdır [13,28]. Yapılandırma hatalarından kaynaklanan açıklıkların istismar edilmesine çözüm olarak, potansiyel yapılandırma açıklıklarının nerede olduğunun tespiti, standart ölçütlere dayanan önem derecesini ölçme ve açıklığın düzeltilmesini kolaylaştırma için aşamalı yapılandırma taraması ve web uygulamalarının ilk kaynak kod analizini etkin bir şekilde birleştiren bir yaklaşım sunulmaktadır. Confeagle olarak adlandırılan bu yaklaşım, bilgi sızıntısı, servis dışı bırakma ve oturum çalma (session hijacking) saldırılarına sebep olan potansiyel yapılandırma açıklıklarını tespit etmiştir [28].

Web uygulama güvenliğinin sağlanması için birden fazla katmana sahip hibrid çözüm önerisi de bulunmaktadır. Sunulan çok katmanlı yaklaşım, saldırıları düşük FP uyarı ile yüksek oranda tespit etmiştir. Önerilen sistem uygulama seviyesinde bilinen ve bilinmeyen, özellikle SQL enjeksiyon ve XSS, saldırılarını tespit etmiştir. Önerilen çok katmanlı yaklaşım sırasıyla filtre, tespit modülü ve analiz&doğrulama modülü katmanlarından oluşan bir sistem sunmuştur [29].

Web uygulama güvenliğinde güvenliğin arttırılması amacıyla yazılım yaşam döngüsünün web uygulamaları geliştirilirken uygulanması ve bu evrelerde güvenliğin nasıl olması gerektiğini öneren çalışmalarda bulunmaktadır. Bir çözüm önerisi, web uygulama açıklıklarının çoğunun doğrudan yazılım geliştirme yaşam döngüsünün uygulanmasının yetersiz olmasından kaynaklandığı varsayımıyla, yazılım geliştirme yaşam döngüsünde, güvenlik tekniklerinin (practices) bütünleşmesi üzerine odaklanmış ve güvenlik faaliyetlerin yazılım geliştirme yaşam döngüsünün aşamalarına göre tanımlamıştır [30].

Diğer bir çözüm önerisi ise web uygulama güvenliğinde çeşitli ticari ve açık kaynak kodlu çözümlerin olduğu ve uygun güvenlik çözümü seçiminin önemli olduğu varsayımıyla, web uygulama katmanı güvenlik duvarı çözümlerinin, uygulama katmanı güvenliği için ihtiyaç duyulan, önemli özellikleri karşılaştırmıştır. Bu şekilde kullanıcıların kendi ortamları için en uygun çözümü tercih etmelerine yardımcı olunmuştur. Karşılaştırma da kullanılan kriterler aşağıda verilmiştir [31].

- Güvenlik politikası kontrolü
  - Zaman verimli
  - İyi organize
  - Etkin
- İzleme
- Engelleme
- Cevapları filtreleme
- Saldırı önleme
- Web sitesi gizleme sistemi
- Kimlik denetimi ve web tek oturum açma
- Derin inceleme
- Oturum koruma
- Genel güvenlik performansı

Başka bir çözüm önerisi, web uygulamalarının güvenliği için geliştirilen çözümleri sınıflandırmıştır. Sınıflandırma alt başlıkları aşağıda maddeler halinde verilmiştir [32].

- Yeni web uygulamalarının güvenli yapılması,
- Var olan uygulamaların güvenlik analizleri/testleri,
- Var olan uygulamaların çalışma zamanı koruması,

Literatür çalışmaları değerlendirildiğinde, önerilen güvenlik çözümlerinin uygulamaların güvenli geliştirilmesine yoğunlaştığı görülmektedir. Sunulan güvenlik çözümlerinin, uygulama üzerinde güvenlik açığı olmaması çabasında olduğu ve varsa da güvenlik açığı tarayıcıları ile açıklıkların test aşamasında tespit edilip kapatılması yönünde olduğu gözlemlenmiştir. Ayrıca, yapılan çalışmalarda mevcut güvenlik çözümlerinin

karşılaştırılması yapılarak, daha uygun bir güvenlik çözümünün tercih edilmesi yönünde çalışmalar da bulunmaktadır.

Yapılan tüm bu çalışmalar incelendiğinde, çalışmaların açıklıkların tespiti ve açıklıkların istismar edilmesinin önlenmesine yönelik olduğu görülmektedir. Kısacası çözümler siber saldırılarının tespiti ve önlenmesi faaliyetlerine yöneliktir. Bununla beraber, bir açıklığın istismar edilmesinin sonrasında, yapılan saldırının verdiği zararın düzeltilmesine yönelik bir çalışma olmadığı görülmektedir. İlerleyen bölümlerde, bir saldırı sonucu verilen zararın tespiti ve düzeltilmesine yönelik geliştirilen uygulama ayrıntılı bir şekilde ele alınacaktır.

4. ÖNERİLEN SİSTEM (WUBKU) ve GERÇEKLEŞTİRİLMESİ

Bu tez çalışmasında, geçmiş bölümlerde değinilen bilgiler, araştırmalar, incelemeler ve analizler sonucunda web uygulama güvenliğinin arttırılmasına yönelik özgün bir sistem önerisi ve sistemin gerçekleştirilmesi hedeflenmiştir. Bu bölümde, önerilen sistem geçmiş bölümlerdeki bilgilerle ilişki kurularak anlatılacaktır.

Bilindiği üzere kişisel ve kurumsal ihtiyaçlar doğrultusunda web uygulamaları hayatımızın ayrılmaz bir parçası olmuştur. Bu web uygulamaları çeşitli sebeplerden dolayı kötü niyetli kişilerin hedefi haline gelmesinin bir sonucu olarak, web uygulamalarının güvenliğinin sağlanması hayati önem arz etmektedir.

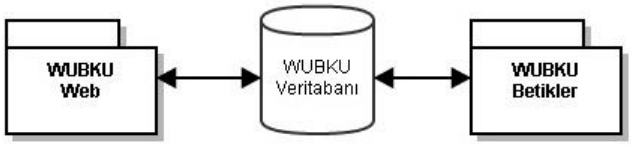
Web uygulamalarını güvenlik bakış açısıyla irdeleyen raporlar incelendiğinde, web uygulamalarının büyük bir bölümünün güvensiz olduğu açıkça görülmektedir. Web uygulamalarının geliştirildiği programlama dilleri kullanımı ve açıklıkların programlama dillerine göre dağılımı göz önünde bulundurulduğunda, açıklıkların programa diliyle ilgisi olmadığı sonucuna varılmaktadır [17]. Web uygulama güvenliği için geliştirilen çözüm önerileri incelendiğinde ise çözümlerin daha çok saldırıların tespiti ve önlenmesine yönelik olduğu görülmektedir. Saldırı sonucunda oluşan zararın düzeltilmesine yönelik çalışmaların olmaması dikkat çekmektedir.

Siber suçların şirketlere olan maliyetini ortaya koyan bilgiler incelendiğinde, farklı saldırı türlerinin farklı büyüklükteki kurumlar için farklı maliyetler oluşturduğu görülmektedir. Saldırılardan korunmak için yapılan faaliyetlere bakıldığında ise en fazla maliyetin saldırının tespiti ve düzeltilmesi faaliyetlerinde olduğu görülmektedir. Tespit ve düzeltme faaliyetleri tüm korunma maliyetinin %53'lük kısmını oluşturmaktadır. Siber saldırıların çözülme süreleri göz önünde bulundurulduğunda, kötü niyetli çalışanların yaptığı saldırılar ile kötücül kod kullanılarak yapılan saldırılarının çözülme sürelerinin en fazla olduğu görülmektedir.

Bu tez çalışmasında, yukarıda ki belirtilen problemlere çözüm getirmek amacıyla, bütünlük kontrolü üzerinden çalışan bir sistem önerilmiştir. Önerilen sistem, web uygulamasının bütünlüğünün bozulmasına sebep olabilecek herhangi bir yetkisiz işlem ya

da saldırı sonucunda, bütünlüğün bozulduğunun tespiti, bozulan bütünlüğün otomatik olarak tekrardan sağlanması ve bütünlüğün bozulduğu ve düzeltildiğiyle ilgili bir uyarı üretecek şekilde gerçekleştirilmeye çalışılmıştır. Bu şekilde, web uygulama güvenliğine katkı sağlanması amaçlanmaktadır.

Önerilen sistem, WUBKU (Web Uygulamaları Bütünlük Kontrol Uygulaması) olarak adlandırılmıştır WUBKU ile web uygulamalarının bütünlüğünün kontrol edilmesi ve bütünlük kontrolü üzerinden web uygulama güvenliğinin arttırılmasına katkı sağlayacağı değerlendirilmektedir. WUBKU bileşenleri Şekil 4.1'de gösterilmektedir.



Şekil 4.1. WUBKU bileşenleri

WUBKU, web uygulamalarının bulunduğu dizin içerisinde bütünlük kontrolü yaparak herhangi bir saldırı ya da yetkisiz bir işlem sonucunda bütünlüğü bozulan uygulamanın, bütünlüğünün bozulup bozulmadığını tespit edebilmektedir. WUBKU bütünlüğün bozulması durumunu saldırı olarak kabul etmektedir ve bütünlüğü bozulan uygulamanın bütünlüğünü otomatik olarak tekrardan sağlayabilmektedir. Bütünlüğün otomatik olarak tekrardan sağlanması için, WUBKU veritabanı üzerinde tutulan bütünlüğü doğrulanmış dosya ve dizin yapısı kullanılmaktadır. Bir uygulamanın bütünlüğünün bozulması, yetkisiz şekilde uygulama üzerinde aşağıdaki işlemler yapıldığında gerçekleşmektedir.

- Uygulama içerisine dizin eklenmesi,
- Uygulama içerisinde var olan dizininin silinmesi ya da değiştirilmesi,
- Bütün uygulamaların bulunduğu kök dizin üzerine dosya ya da dizin eklenmesi,
- Uygulama içerisinde dosya oluşturulması,
- Uygulama içerisinde var olan dosyanın silinmesi ya da değiştirilmesi,
- Uygulama dizinin silinmesi ya da değiştirilmesi,

Gerçekleştirilen uygulamada bütünlüğün bozulduğunun nasıl tespit edileceği konusunda imza tabanlı mı yoksa anormallik tabanlı bir sistem kullanılmasının mı daha iyi olacağı konusunda çalışılmıştır. Çalışma sonucu, imza tabanlı çalışan sistemlerde yanlış uyarı üretilme durumunun daha az görülmesi ve bu şekilde daha doğru sonuçlar üretilmesinden imza tabanlı bir sistem olarak gerçekleştirilmesine karar verilmiştir. Gerçekleştiren uygulamada bir dosyanın ya da dizinin yapısının değişip değişmediğinin kesin çözümü için özet algoritması kullanılarak hesaplanan özet değeri kullanılmıştır. Bu özet değeri imza olarak kabul edilmiş ve imza tabanlı çalışan bir uygulama geliştirilmiştir.

Özet fonksiyonu, herhangi bir uzunluktaki ya da boş girdi karakter dizgilerinin genellikle daha kısa, tekil (unique) ve sabit uzunlukta bir hale dönüştürülmesi olarak tanımlanmaktadır [33,34]. Özet fonksiyonları bütünlük kontrolü, şifre koruma, rastgele numara üretme, mesaj kimlik doğrulama ve dijital imza gibi çeşitli alanlarda kullanılabilmektedir [33,34]. Özet fonksiyonlarının kullandığı çeşitli algoritmalar bulunmaktadır. Bu algoritmalar ve özellikleri Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. Özet algoritmaları [35]

Sıra Nu.	Özet Algoritması	Çıktı Değeri Uzunluğu
1.	MD4	128 bit
2.	MD5	128 bit
3.	MD2	128 bit
4.	RIPE-MD	128 bit
5.	Haval	128, 160, 192, 224 bit
6.	SHA-0	160 bit
7.	SHA-1	160 bit
8.	SHA-256	256 bit
9.	SHA-384	384 bit
10.	SHA-512	512 bit

Bu özet algoritmalarının çoğu güvenilirliğini kaybetmiştir. MD4 (Mesaj Özeti 4), MD5, RIPEMD (Yarış Bütünlük İlkeleri Değerlendirme Özet Değeri), Haval-128, SHA (Güvenli Sağlama Protokolü) 0, SHA-1 algoritmalarında farklı girdi değerleri için aynı çıktı değeri üretilerek güvensizlikleri ispatlanmıştır [36-40].

Mevcut durumda SHA-256, SHA-384, SHA-512 algoritmaları güvenli gözükmektedir. Geliştirilen uygulamada ise özet algoritması olarak SHA-512 algoritması kullanılmıştır. Bu algoritmanın kullanılmasının nedenleri ise SHA-512 özet algoritmasının 64 bit makineler üzerinde SHA-256 özet algoritmasından daha hızlı olduğunun kanıtlanmasıdır [41]. SHA-512 özet algoritmasının çıktı değerinin uzun olmasından donanım ortamları üzerinde kullanımı maliyetli olabilmektedir [41]. Geliştirilen uygulama, işletim sistemi üzerinde çalıştığından böyle bir kısıt bulunmadığının varsayılmasıdır. Gerçekleştirilen sistemin üzerinde çalıştığı işletim sistemi üzerinde SHA-512 algoritmasını kullanarak özet değeri üreten hazır komut bulunması ve SHA-512 özet algoritmasının güvensizliğinin ispatlanmamış olmasıdır.

Önerilen sistemde bütünlük kontrolü işlemleri için, Linux işletim sistemi üzerinde hazır bulunan sha512sum komutu kullanılmaktadır. Bu komut girdi olarak verilen bir dosyanın özet değerini SHA-512 algoritmasını kullanarak hesaplayabilmektedir. Önerilen sistemin gerçekleştirilmesinde, hem web arayüzünden yapılan işlemler hem de komut satırı üzerinden gerçekleşen işlemler bu komut kullanılarak gerçekleştirilmektedir. Önerilen sistemde bu komut, dosya veya dizinlerin özet değerlerinin oluşturulmasında kullanıldığı gibi, uygulamaya ait mevcut durumdaki dosya veya dizinlere ait özet değerlerinin tutarlı olup olmadığının kontrolü içinde kullanılmaktadır.

Bütünlük kontrolü için SHA-512 özet algoritmasını kullanan sha512sum komutunun kullanılmasına karar verildikten sonra, imzaların nasıl oluşturulacağı üzerinde çalışılmıştır. Yapılan çalışma sonucu uygulamaların yönetimini sağlayan ve aynı zamanda uygulamalar için imza oluşturulmasını sağlayan bir arayüz geliştirilmiştir. Geliştirilen arayüz kullanılarak uygulama üzerindeki dosya ve dizin silme/güncelleme ve değiştirme işlemleri betikler aracılığıyla yapılmaktadır. Arayüz üzerinden yapılan işlemler sırasında sha512sum komutu kullanılarak uygulamaların özet değeri üzerinden imza değeri hesaplanmaktadır. Bu şekilde, arayüz kullanılmadan yapılan her türlü işlem yetkisiz olarak kabul edilmekte ve saldırı olarak algılanmaktadır.

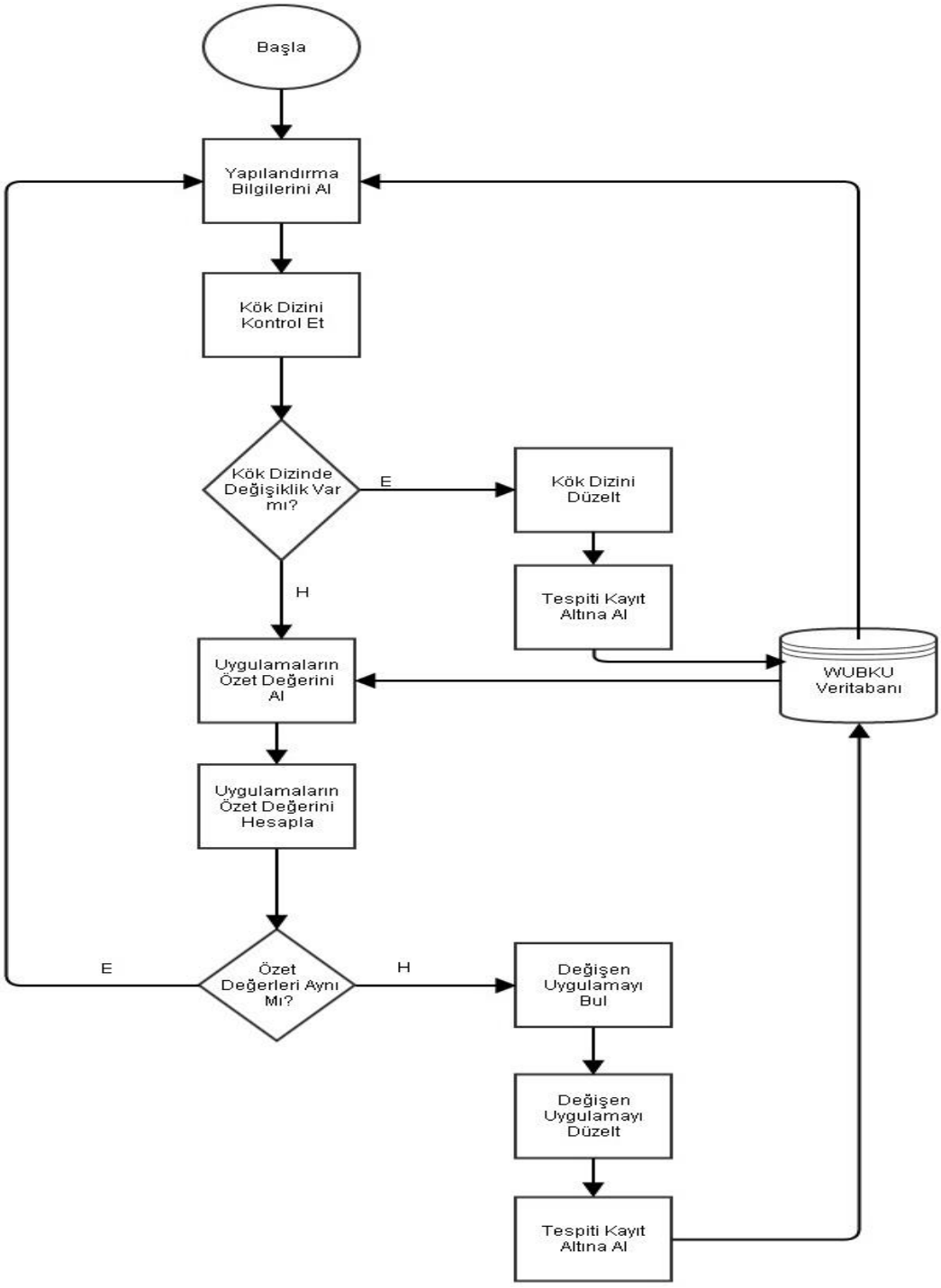
WUBKU gerçekleştirilirken hangi dosyaların güvenli kabul edileceğinin yanında, bu güvenli kabul edilen dosyaların nerede saklanacağı sorunuyla karşılaşılmıştır. Bu soruna çözüm olarak ilk önce tüm dosyaların web uygulamasının bulunduğu dizinden başka bir dizinde tutulması planlanmıştır. Ancak, herhangi bir ihlal durumunda, uygulamanın

bütünlüğünü bozan saldırı ile farklı dizinde bulunan güvenilir dosyaların bütünlüğü de bozulabileceğinden bu fikirden vazgeçilmiştir. Bu yüzden güvenli dosyaların veritabanı üzerinde tutulmasına karar verilmiştir. Böylelikle veritabanına erişiminde kullanılan kullanıcı adı ve şifre ile güvenlik bir kat daha arttırılmıştır.

Bir uygulamanın bütünlüğünün bozulup bozulmadığının kontrolü için Bash uygulamasıyla yazılan betik kullanılmıştır. Betik bütünlüğün kontrolü için ilk olarak web arayüzü üzerinden tanımlanan yapılandırma bilgilerini veritabanından almaktadır. Bu yapılandırma bilgileri, bütünlük kontrolünün yapılması ve bütünlüğün bozulması durumunda bütünlüğün otomatik olarak tekrardan sağlanması için temel bilgileri içermektedir. Temel bilgiler alındıktan sonra ilk olarak bütün uygulamaların bulunduğu kök dizinin bütünlüğü kontrol edilmektedir. Kontrol sonucu kök dizin üzerinde oluşturulmuş herhangi bir dosya ya da dizin varsa silinerek, yapılan işlem hakkında bilgi içeren bir uyarı mesajı üretilmektedir. Kök dizinin kontrol edilmesinden sonra veritabanı üzerinden kök dizine ait özet değeri bilgisi alınmaktadır. Bu işlemden sonra kök dizine ait özet değeri hesaplanarak sonuç veritabanından alınan değer ile karşılaştırılmaktadır. Bu şekilde WUBKU üzerinde birden fazla uygulama bulunsa bile iki özet değerinin karşılaştırılmasıyla uygulamaların bütünlüğünün bozulup bozulmadığının kontrolü yapılabilmektedir. Bu işlem sürekli tekrarlanarak uygulamaların değişip değişmediğinin kontrolü anlık olarak yapılmaktadır. Veritabanından alınan değer ile hesaplanan değerin aynı olmaması sonucu hangi uygulamanın bütünlüğünün bozulduğunun tespiti işlemine başlanmaktadır. Hangi uygulamanın bütünlüğünün bozulduğunun tespiti için, veritabanı üzerinden her bir uygulamaya ait dosyaların ve dizinlerin toplam özet değeri bilgileri alınmaktadır. Bu değerler ile her bir uygulamaya ait dosyaların ve dizinlerin hesaplanan özet değerleri karşılaştırılmaktadır. Bu karşılaştırma üzerinden hangi uygulamanın bütünlüğünün bozulduğunun tespiti yapılmaktadır. Uygulamaya ait veritabanından gelen dosya ya da dizin özet değerlerinin hesaplanan değerlerle aynı olmaması durumunda, dizinler üzerinde bir değişiklik varsa dizinlerin bütünlüğünün düzeltilmesi, dosyalar üzerinde bir değişiklik varsa dosyaların bütünlüğünün düzeltilmesi işlemi yapılmaktadır. Bu işlemlerde dizinin silinmesi durumuna karşı ilk önce dizinlerin bütünlüğü kontrol edilmektedir. Dizinlerin bütünlüğünün sağlanmasının ardından, ilgili dizinlerde bütünlüğü bozulan dosyalar tekrardan oluşturulmaktadır.

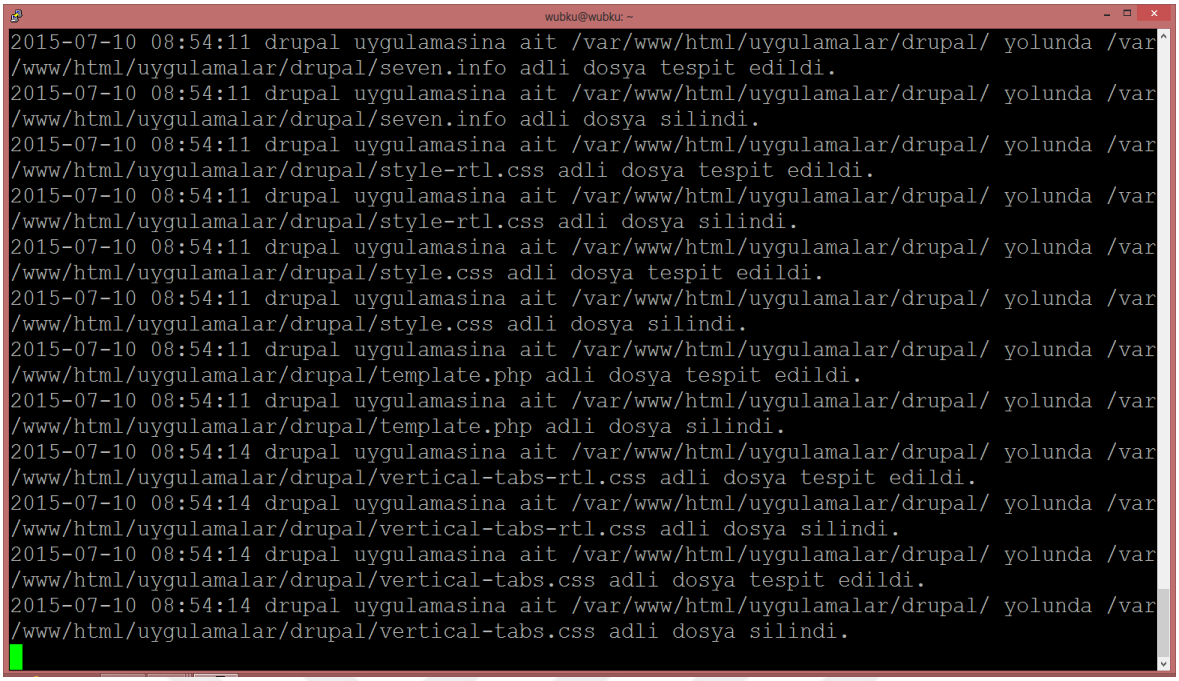
Uygulamaya ait dizin üzerinde değişiklik tespit edilmesi durumunda ilk olarak uygulama üzerindeki mevcut dizin bilgileri alınmaktadır. Bu işlemden sonra olması gereken dizin yapısı veritabanı üzerinden alınmaktadır. Alınan bu değerlerin karşılaştırılması sonucu, uygulama üzerine yeni eklenen dizinler tespit edilip silindiği gibi, veritabanından gelen bilgilere göre olması gerektiği halde var olmayan diziler de tekrardan oluşturulmaktadır. Sonra, yapılan bu işlemler ile ilgili bilgi içeren bir uyarı mesajı üretilmektedir.

Uygulamaya ait dosya üzerinde değişiklik tespit edilmesi durumunda, ilk olarak uygulama üzerine yeni bir dosya eklenip eklenmediğinin kontrolü yapılmaktadır. Bu işlem için uygulama üzerindeki mevcut dosya bilgileri alınmaktadır. Bu işlemden sonra uygulama üzerinde olması gereken dosya bilgileri veritabanı üzerinden alınmaktadır. Alınan bu bilgilerin karşılaştırılması sonucu, uygulama üzerine yeni eklenen dosya olup olmadığının kontrolü yapılmaktadır. Veritabanından gelen bilgilere göre olmaması gereken dosyalar otomatik olarak silinmektedir. Bu işlem ile ilgili bilgi içeren bir uyarı mesajı üretilmektedir. Uygulama üzerine yeni eklenen dosyalar silindikten sonra, mevcut dosyaların bütünlüğü kontrol edilmektedir. Mevcut dosyaların bütünlüğünün kontrolü için, uygulama ait dosya, dosyanın özet değeri ve dosyaların bulunduğu dizin bilgileri veritabanından alınmaktadır. Bu bilgiler kullanılarak ilk önce bilgileri alınan dosyanın ilgili dizinde olup olmadığının kontrolü yapılmaktadır. Dosyanın bulunmaması durumunda dosya, veritabanında bulunan bilgiler kullanılarak tekrardan oluşturulmaktadır. Dosyanın bulunması durumunda ise dosyanın özet değeri hesaplanmaktadır. Hesaplanan özet değeri ile veritabanından alınan özet değeri karşılaştırılmaktadır. Özet değerlerinin aynı olmaması durumunda, bütünlüğü bozulan dosya silinerek yerine, veritabanı üzerinde bulunan bilgiler kullanılarak dosya tekrardan oluşturulmaktadır. Bu işlem ile ilgili bilgi içeren bir uyarı mesajı üretilmektedir. Kullanılan betiğin akış diyagramı Şekil 4.2’de gösterilmektedir.



Şekil 4.2. WUBKU saldırı tespit ve uyarı sistemi akış diyagramı

WUBKU tespit ettiği bütünlüğü bozucu durumları ve bütünlüğün tekrardan sağlanması için yapılan işlemleri veritabanının yanında, "/var/www/WUBKU.log" dosyasına da kaydetmektedir. Ayrıca, WUBKU'nun çalışması esnasında karşılaşılan çalışma zamanı hataları da bu dosyaya kaydedilmektedir. WUBKU.log dosyasının içeriği Şekil 4.3'te gösterilmektedir.



Şekil 4.3. WUBKU.log dosyası içeriği

WUBKU mevcut sürümü Linux işletim sistemi üzerinde çalışacak şekilde gerçekleştirilmiştir. Bunun sebebi bütünlüğü bozulan uygulamaların tespiti ve bütünlüğü bozulan dosyaların otomatik olarak bütünlüğünün tekrardan sağlanması için Bash uygulaması ile Linux komut satırının hazır fonksiyonlarının kullanılmasıdır. WUBKU açık kaynak kodlu olan PHP programlama dili, Bash uygulaması ve MySQL veritabanı kullanılarak geliştirilmiş ve açık kaynak kodlu uygulamaların lisanslanmasında kullanılan GPL (Genel Kamu Lisansı) sürüm 3 lisanslı bir uygulamadır.

WUBKU ilk olarak geliştirildiğinde sadece belirlenen bir uygulama için bütünlüğü kontrol edici bir betik ve dosyaların özet değerlerinin hesaplanması için kullanılan bir sayfadan oluşmaktaydı. WUBKU uzman görüşleri sonrasında ilk olarak birden fazla uygulama için koruma sağlaması ve tespit ettiği işlemler ile ilgili uyarı üretecek şekilde geliştirilmesine devam edilmiştir. WUBKU'nun, http://www.wubku.com bağlantısı üzerinden indirilebilir sürümünün, özellikleri aşağıda verilmiştir.

- Bir web uygulaması için dosya ve dizin bazında bütünlük kontrolü,
- Web uygulamalarının bulunduğu kök dizin silinse bile uygulamaları ve kök dizini otomatik düzeltebilme,

- Bütünlüğü bozucu bir saldırı sonucu, bütünlüğü bozulan dosyaların otomatik tespiti ve düzeltilmesi,
- Birden fazla web uygulaması için koruma sağlayabilme,
- Tespit edilen saldırıların web ara yüzünden takip edilmesini sağlama,
- Tespit edilen saldırıların eposta ile bildirilmesi,
- Web uygulamalarının içeriğinin yönetilebilmesi için arayüz sağlama,
- Yönetici, yazılımcı ve izleyici şeklinde görev paylaşımı,
- Var olan yazılımların WUBKU göç sayesinde WUBKU'ya aktarılması,
- Kayıt takibi için kullanıcı yetkilendirmesi,
- Uygulamalar üzerinde yapılan işlemlerin kayıt altına alınması,

Web uygulama güvenliğine katkı sağlamak amacıyla geliştirilen WUBKU'nun güvenliği önem arz etmektedir. Çünkü gerçekleştirilen sistemde, işlemler WUBKU tarafından sağlanan arayüz üzerinden gerçekleştirilmektedir. Arayüz üzerinden gerçekleştirilen işlemler yetkili olarak kabul edildiğinden, WUBKU üzerindeki herhangi bir açıklık, üzerinde barındırdığı tüm uygulamaların kötüye kullanılmasına sebep olabilir. WUBKU geliştirilirken güvenliği için aşağıdaki kriterler göz önünde bulundurulmuştur.

- Kullanıcıdan alınan girdi değerlerinin kontrolü,
- Yapılan işlemlerde, kullanıcının işlemi yapmaya yetkili olup olmadığının kontrolü,
- Oturum kontrolü,
- Kullanıcı yetkilerinin rollere göre belirlenmesi,
- Veritabanı erişimini yapan kullanıcının yetkilerinin kısıtlanması,

WUBKU çalışırken herhangi bir sebepten dolayı çalışması durduğunda, tekrardan çalıştırıldığında, çalışmasına kaldığı yerden devam edebilmektedir. WUBKU'nun bütünlüğü tekrardan sağlamak için kullandığı güvenli veriler veritabanı üzerinde saklanmaktadır. Bu yüzden veritabanı üzerinde bulunan verilerin güvenliği önemli bir durumdur. Veritabanı üzerindeki bir dosyaya ait bilgilerin değişmesi durumunda, o dosyaya ait bir bütünlüğü bozucu bir saldırı olduğunda, veritabanında bulunan bilgiler güvenilir kabul edileceğinden bütünlük tam olarak sağlanamayacaktır. Bu yüzden, WUBKU sürekli olarak saldırı mesajı üretecektir.

WUBKU GPL sürüm 3 ile lisanslanarak herkesin kullanımına sunulmasının yanında herkesin uygulamayı istediği gibi değiştirip özelleştirmesi ve uygulamanın daha geniş kitlelerce kullanılması amaçlanmıştır. WUBKU mevcut sürümü, WUBKU resmi sitesi olan http://www.wubku.com bağlantısından indirilebilmektedir. WUBKU, 2015 yılında Gazi Üniversitesi Mühendislik Fakültesinde gerçekleştirilen ISDFS (Uluslararası Adli Bilişim ve Güvenlik Sempozyumu)'de katılımcılara tanıtılmış ve olumlu dönüt alınmıştır. Ayrıca, WUBKU yaygınlaştırılması çalışmaları kapsamında çoğunluğunu üniversitelerin oluşturduğu 10 kuruma eposta gönderilmiş ve alınan dönütler bir sonraki sürümün özelliklerine katkı sağlamıştır.

Açık kaynak kodlu olan ve GPL sürüm 3 lisansına sahip WUBKU'nun milli işletim sistemi PARDUS'a adapte edilmesi için ULAKBİM ile görüşülmüş ve olumlu dönütler alınmıştır.

4.1. Gereksinimler

Açık kaynak kodlu olan WUBKU, geliştirilirken de açık kaynak kodlu ürünler tercih edilmiştir. Bu ürünler arasından yaygın olarak kullanılan Apache web sunucu, MySQL veritabanı, PHP programlama dili ve Bash uygulaması tercih edilmiştir.

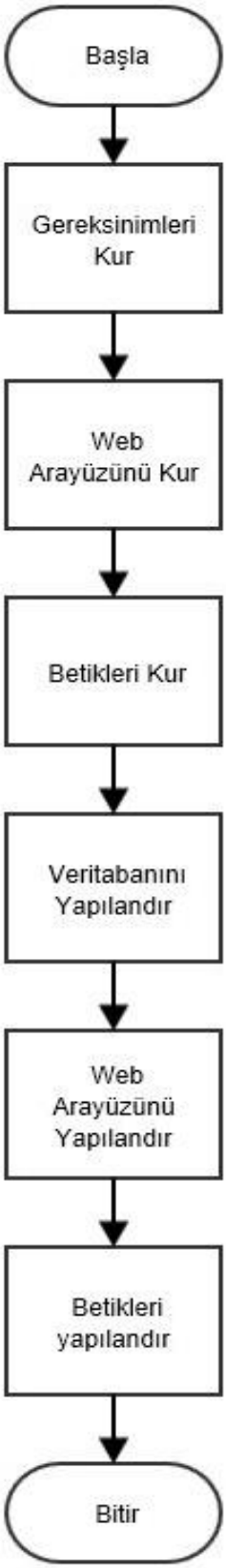
WUBKU kurulumunun yapılabilmesi için ilk olarak gereksinimlerinin kurulması gerekmektedir. İlk olarak, üzerine Apache web sunucu, MySQL veritabanı ve PHP programlama dili kurulabilen, Bash uygulaması ile yazılan betiklerinin çalışabildiği Linux dağıtımı bir işletim sistemi kurulmalıdır. Bu gereksinimlerin ne amaçla kurulduğu ve kullanıldığı aşağıda maddeler halinde verilmiştir.

- Bash uygulaması, bütünlüğü bozulan uygulamaların tespiti ve bütünlüğü bozulan uygulamanın otomatik olarak bütünlüğünün tekrardan sağlanmasında, tekrardan kod yazmaktansa Linux hazır fonksiyonları kullanılmıştır. Bash uygulamasıyla yazılan betikler, web arayüzü üzerinden yapılan işlemlerde; dizin oluşturma, dizin silme, uygulamaların özet değerinin hesaplanması ve bütünlüğün bozulup bozulmadığının kontrolü gibi çeşitli işlemlerde kullanılmaktadır.

- PHP programlama dili üzerinden Linux komutları çalıştırılabildiğinden tercih edilmiştir. PHP kullanılarak tasarlanan web arayüzü üzerinden, kullanıcı tarafından sağlanan yapılandırma bilgilerinin tanımlanması, uygulamaların içerek yönetimi, kullanıcıların yönetimi ve saldırıların takibi gibi işlemlerin yapılmasında kullanılmaktadır.
- MySQL, PHP programlama diliyle bütünleşik çalışabilmesi ve Linux komut satırı üzerinden SQL komutları ile MySQL üzerinde işlem yapılabildiğinden tercih edilmiştir. MySQL web arayüzü tarafından tanımlanan yapılandırma bilgileri, kullanıcı bilgileri, saldırı bilgileri ve uygulama bilgileri gibi verilen saklanması için kullanılmaktadır.
- Apache, hem PHP programlama diliyle bütünleşik çalışabildiğinden hem de Linux dağıtımlarına kolayca kurulabildiğinden tercih edilmiştir. Apache web sunucu, WUBKU web arayüzünün çalıştırılmasında kullanılmaktadır.

4.2. Kurulum

WUBKU'nun düzgün çalışabilmesi için ilk olarak altyapıyı oluşturan Bash uygulaması, PHP programlama dili, MySQL veritabanı ve Apache web sunucusunun kurulması gerekmektedir. Gereksinimlerin kurulmasından sonra, WUBKU web arayüzü ardından da WUBKU betiklerinin kurulması gerekmektedir. Kurulumdan sonra WUBKU arayüz ve betiklerin düzgün olarak çalışabilmesi için veritabanı bağlantı bilgileri, uygulama bilgileri gibi bilgiler tanımlanarak yapılandırılması gerekmektedir. WUBKU kurumu akış şeması Şekil 4.4'te gösterilmektedir. Ayrıca, ayrıntılı kurulum adımları EK-1'de bulunmaktadır.



Şekil 4.4. WUBKU kurulumu akış diyagramı

4.3. Kullanım ve Yapılan İşlemler

WUBKU'nun düzgün çalışması için web arayüzü üzerinden yapılan işlemlere dikkat edilmelidir. Bu yüzden, WUBKU web arayüzü üzerinden yapılan işlemlerin neler olduğu ve işlemlerin nasıl yapılması gerektiği bu bölümde ayrıntılı şekilde anlatılmıştır.

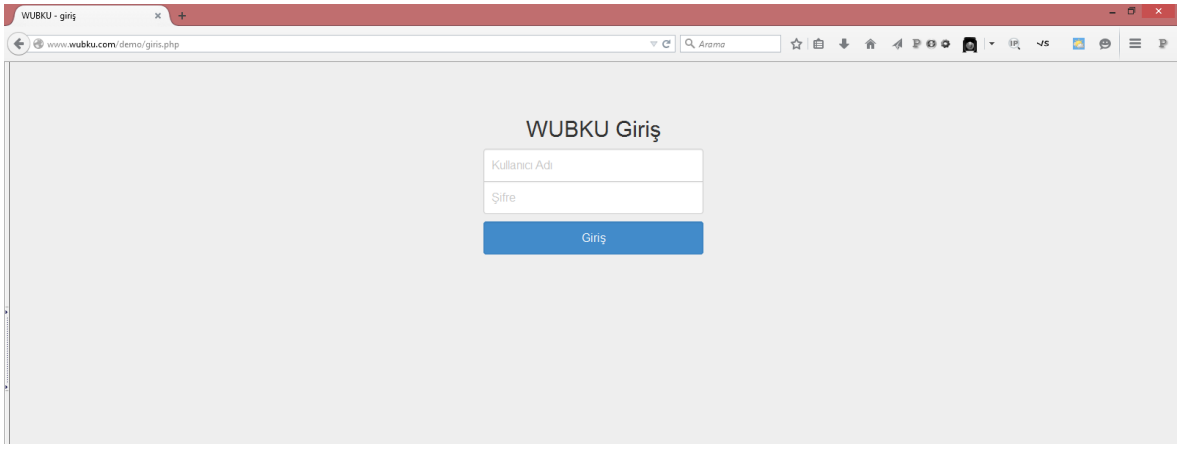
4.3.1. WUBKU çalıştırılması

WUBKU'nun çalıştırılmasında kullanılan komut aşağıda verilmiştir. Ayrıca, sunucu yeniden başladığında WUBKU'nun otomatik olarak başlaması için verilen komut /etc/rc.local dosyasına eklenmelidir.

```
sudo /opt/wubku/bin/wubku.sh /opt/wubku/bin/vtBilgiler.txt >> /var/www/WUBKU.log &
```

4.3.2. WUBKU giriş

WUBKU kurulumun ardından, kurulum ile gelen *yönetici* kullanıcı adı ve *1234* şifresiyle giriş yapılabilmektedir. Giriş işleminde kullanılan kullanıcı adına göre kullanıcı yetkileri otomatik belirlenmektedir. Bu yüzden giriş esnasında kullanıcı rolü seçme işlemi gerekmemektedir. WUBKU giriş ekranı Şekil 4.5'te gösterilmektedir.



Şekil 4.5. WUBKU giriş sayfası

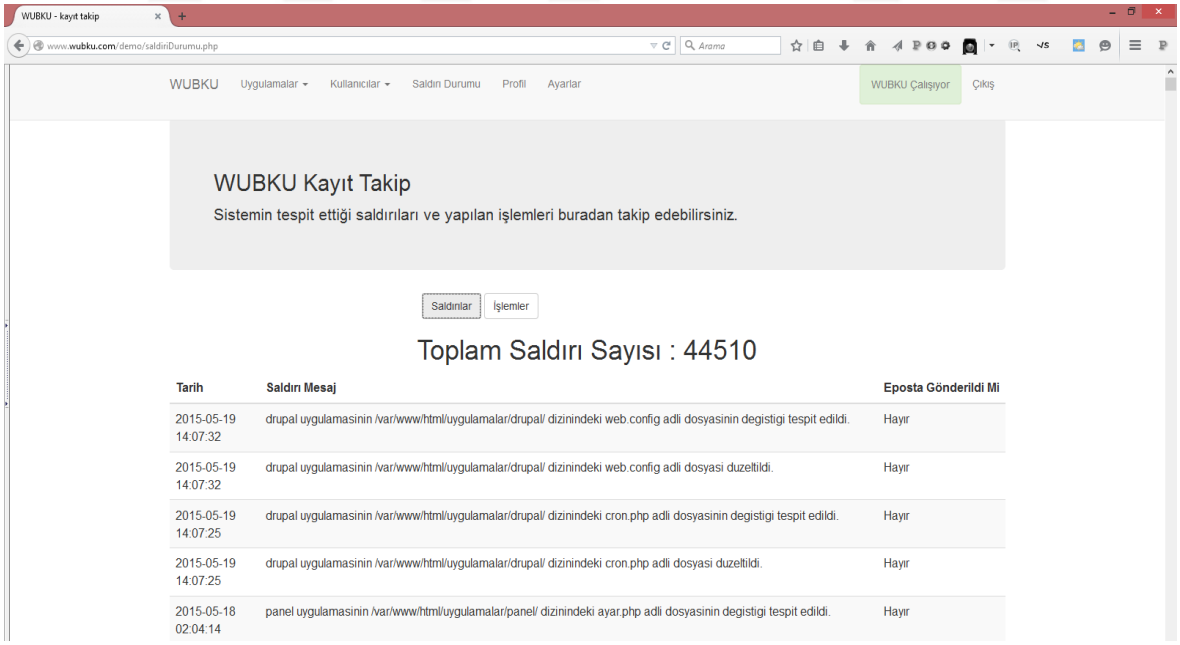
4.3.3. WUBKU yönetici işlemleri

Yönetici rolüne sahip kullanıcı WUBKU üzerinde tam yetkili kullanıcıdır. Yönetici hesabıyla yapılabilen işlemler aşağıda verilmiştir.

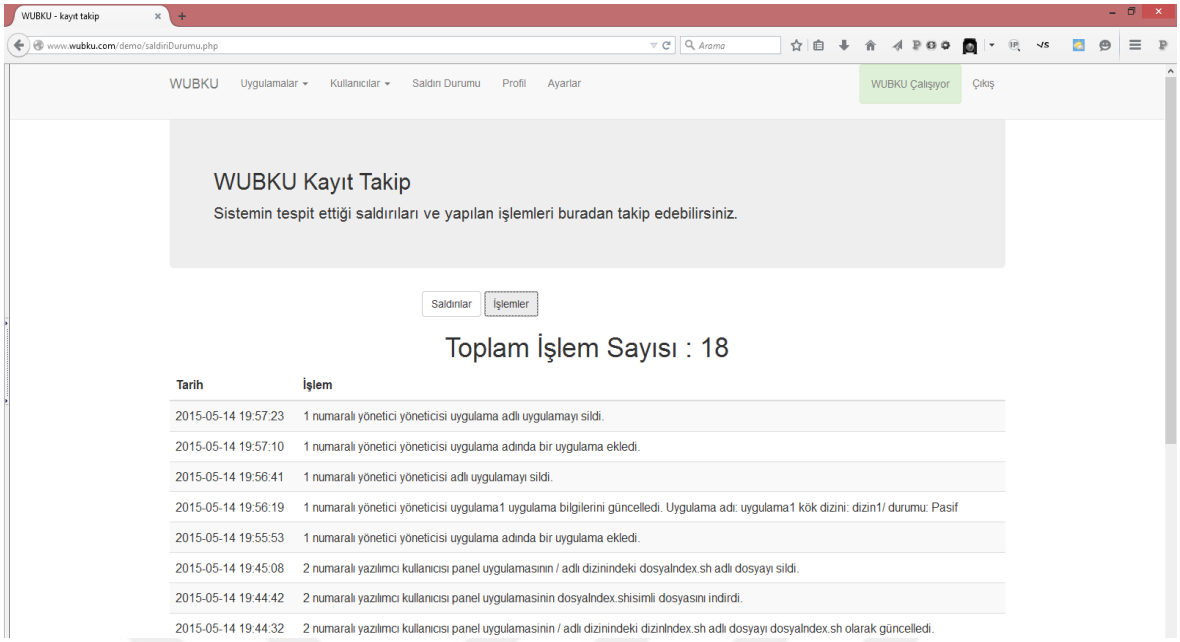
- Uygulama ekleme, silme, düzenleme
- Kullanıcı ekleme, silme, düzenleme
- Tespit edilen saldırılar ve yapılan yetkili işlemlerin takibi,
- Sistemin çalışması için gerekli tanımlama ve bildirim ayarları,

Yönetici rolü saldırı ve işlemlerin takibi

Yönetici rolüne sahip kullanıcılar, sistemin tespit ettiği saldırı uyarılarını ve uygulamalar üzerinde yönetici ve kullanıcı rolüne sahip kullanıcılar tarafından gerçekleşen işlemler takip edilebilmektedir. Yönetici rolüne sahip kullanıcının, kayıtları takip sayfalarının görüntüleri Şekil 4.6 ve Şekil 4.7'de gösterilmektedir.



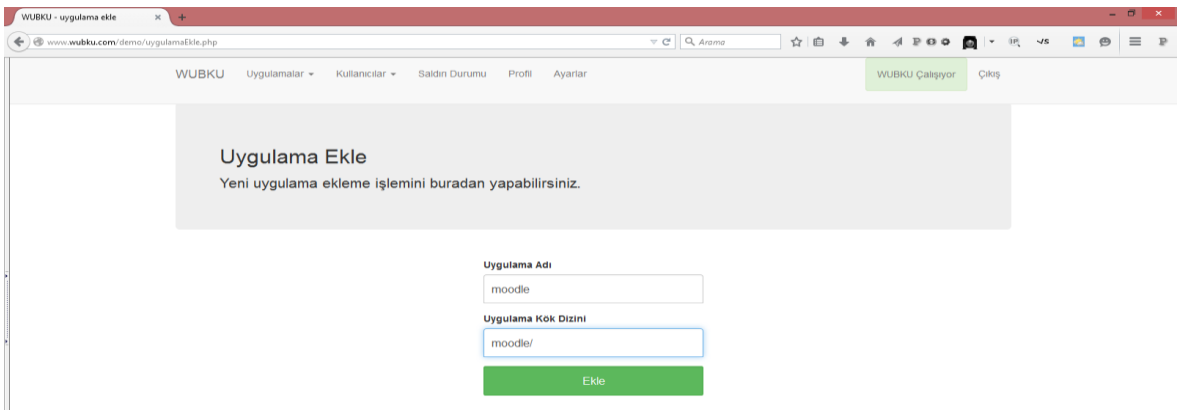
Şekil 4.6. Yönetici rolüne sahip kullanıcının saldırı takip sayfası



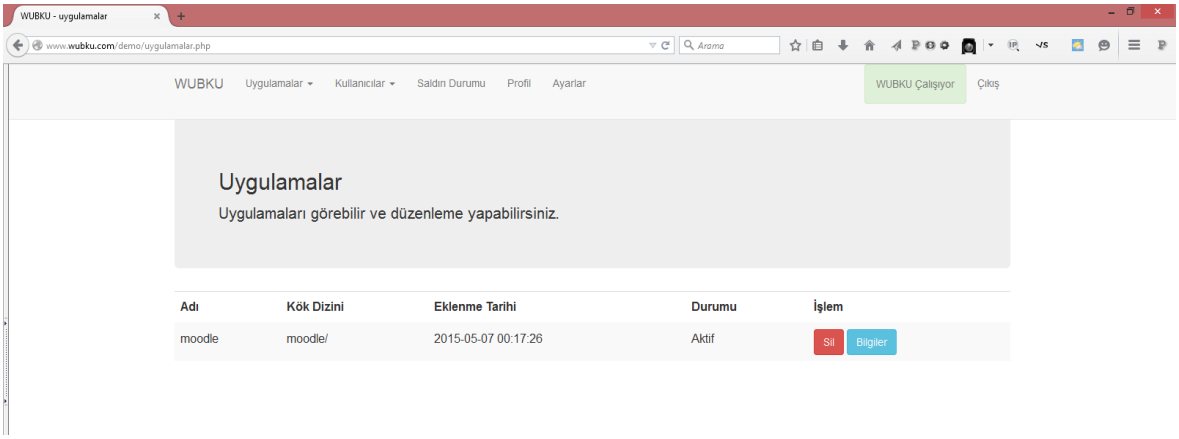
Şekil 4.7. Yönetici rolüne sahip kullanıcının işlem takip sayfası

Yönetici rolü uygulama işlemleri

Yönetici rolüne sahip kullanıcılar, “Uygulamalar” seçeneği üzerinden uygulama ekleme, silme ve düzenleme işlemlerini yapabilmektedir. Uygulama ekleme işleminde “Uygulamanın Adı” ile uygulamaya ait dosya ve dizinlerin bulunacağı “Uygulama Kök Dizini” belirtilmelidir. Burada “Uygulama Kök Dizini” kısmı, “Ayarlar” kısmında belirtildiğinden dizin adı yazılırken belirtilen kök dizininin altında olacak şekilde ve girilen değerin başında “/” ifadesi olmadan dizin adı girilmelidir. Uygulama ekleme işlemini gösteren sayfaların ekran görüntüsü Şekil 4.8’de, mevcut uygulamaları gösteren sayfanın ekran görüntüsü Şekil 4.9’da gösterilmektedir.



Şekil 4.8. Yönetici rolüne sahip kullanıcının uygulama ekleme sayfası

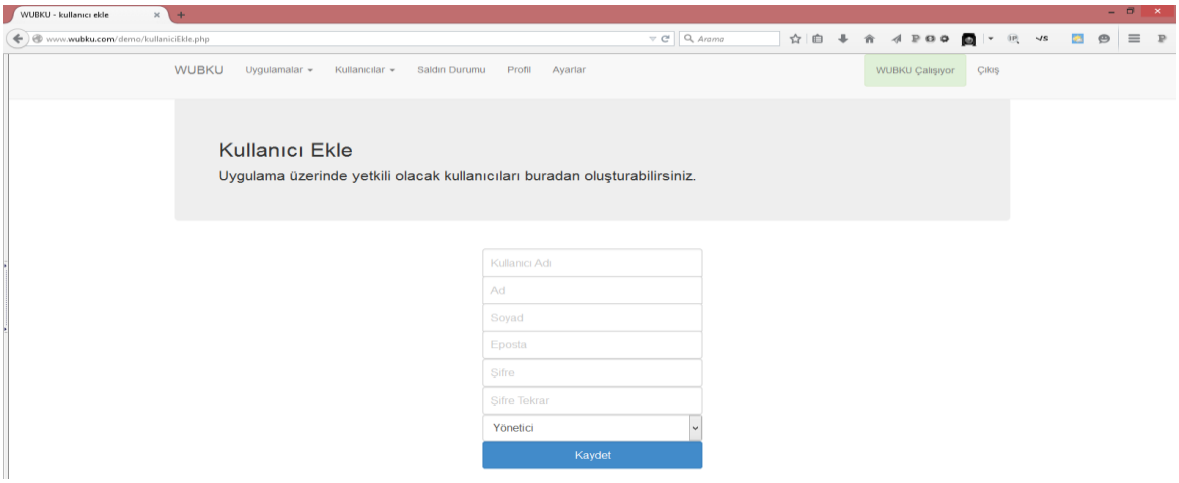


Şekil 4.9. Yönetici rolüne sahip kullanıcının mevcut uygulamaları görüntüleme sayfası

Yönetici rolü kullanıcı işlemleri

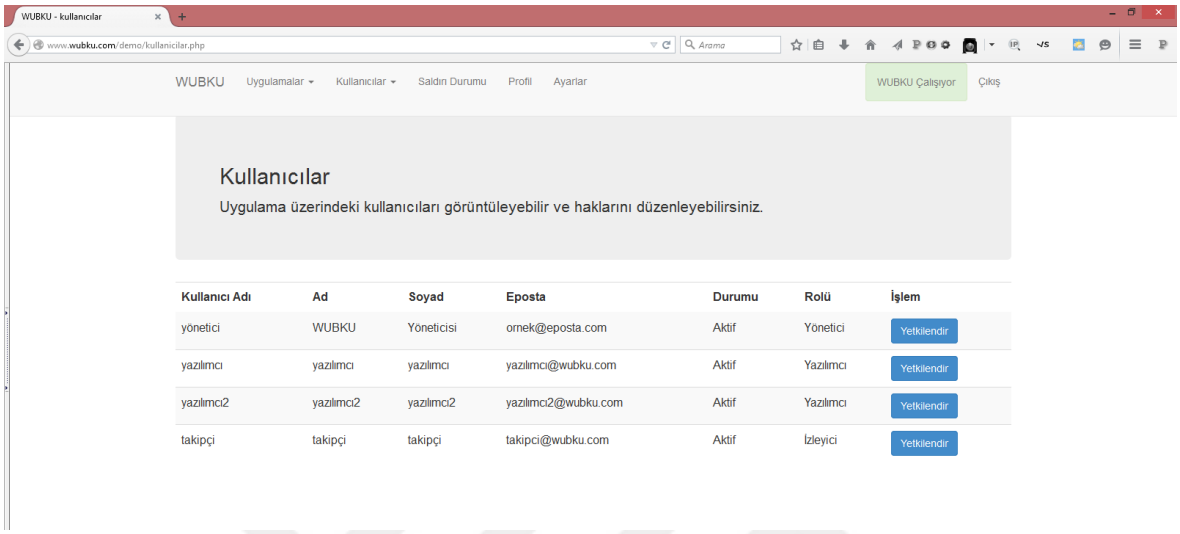
Yönetici rolüne sahip kullanıcılar WUBKU üzerinde kullanıcı ekleme işlemini yapabilirler. Kullanıcı ekleme işleminde aşağıdaki bilgilerin girilmesi gerekmektedir. Kullanıcı ekleme sayfası Şekil 4.10'da gösterilmektedir.

- Kullanıcı adı
- Ad
- Soyad
- Eposta
- Şifre
- Rolü



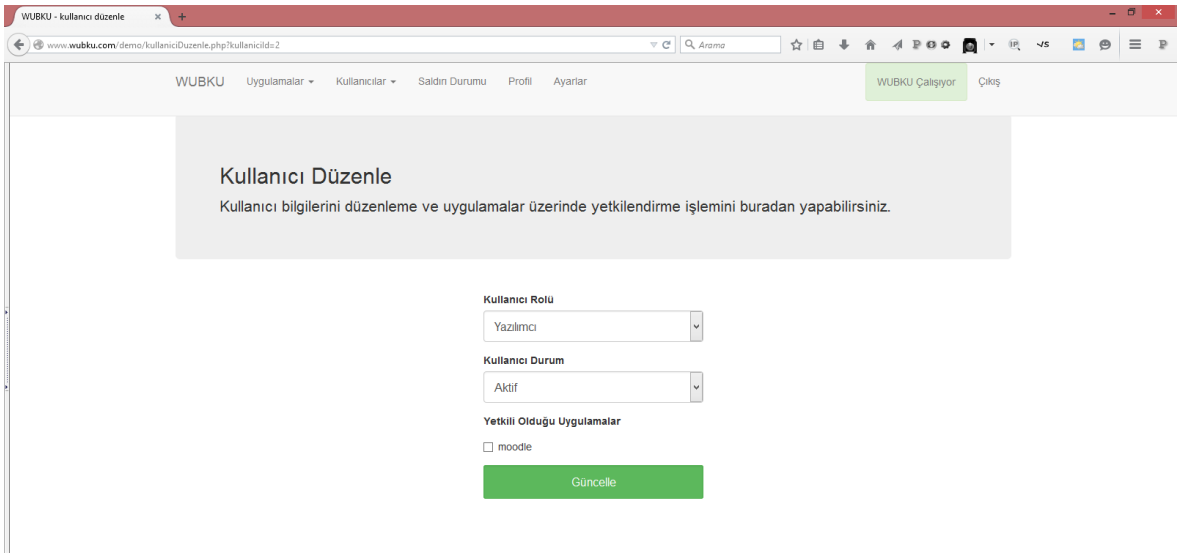
Şekil 4.10. Yönetici rolüne sahip kullanıcı ile yeni kullanıcı ekleme sayfası

WUBKU üzerindeki mevcut kullanıcıların görüntülenebilmesi için, "Kullanıcılar" kısmından "Kullanıcılara Gözat" bağlantısı seçilmelidir. Mevcut kullanıcıların görüntülendiği sayfa Şekil 4.11'de gösterilmektedir.



Şekil 4.11. Yönetici rolüne sahip kullanıcı ile mevcut kullanıcı görüntüleme sayfası

WUBKU üzerindeki mevcut kullanıcıların görüntülendiği sayfada bulunan "Yetkilendir" butonu üzerinden kullanıcı rolü, durumu ve uygulamalar üzerindeki yetkileri düzenlenebilmektedir. Kullanıcı yetkilendirme işleminin yapıldığı sayfa Şekil 4.12'de gösterilmektedir.



Şekil 4.12. Yönetici rolüne sahip kullanıcı ile kullanıcı bilgileri düzenleme sayfası

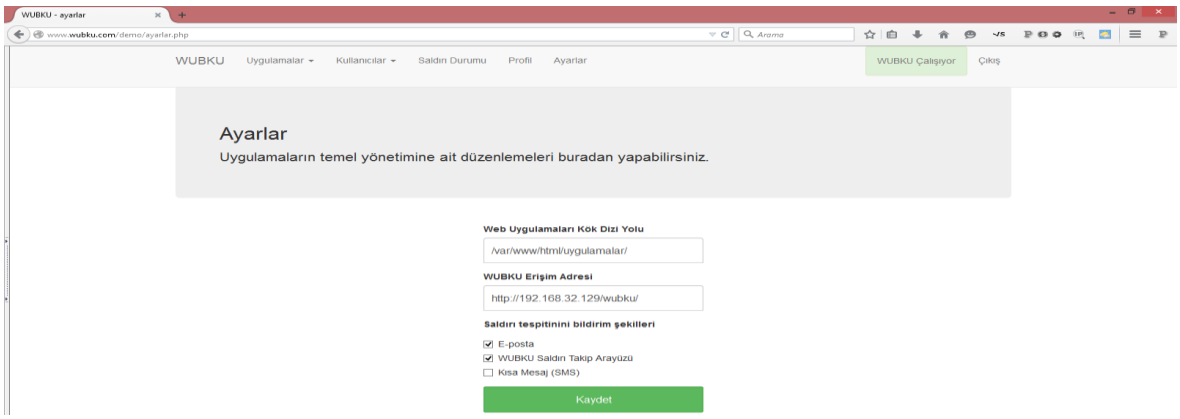
Yönetici Rolü Yapılandırma İşlemleri

WUBKU'nun çalışabilmesi için, "Web Uygulamaları Kök Dizin Yolu" ve "WUBKU Erişim Adresi" bilgilerinin tanımlanması gerekmektedir. "Web Uygulamaları Kök Dizin Yolu" tanımlaması sayesinde hem birden fazla uygulama için bütünlük kontrolü yapılabildiği hem de kök dizinin bütünlük kontrolü üzerinden bütünlüğün bozulup bozulmadığı kontrol edilebilmektedir. "WUBKU Erişim Adresi", herhangi bir uygulamanın bütünlüğünün bozulması durumunda, bütünlüğün tekrardan sağlanabilmesi için ihtiyaç duyulan dosyaların WUBKU üzerinden alınabilmesi için kullanılmaktadır. Bu adrese, WUBKU'nun çalıştığı sunucu üzerinden erişim olduğuna dikkat edilmelidir. WUBKU yapılandırma sayfası Şekil 4.13'te gösterilmektedir. WUBKU üzerinde tespit edilen saldırıların bildirimi üç şekilde yapılabilmektedir. Bunlar;

E-posta; uygulama üzerindeki yetkili kullanıcılara, uygulamanın bütünlüğünün bozulması durumunda, sistem üzerindeki eposta adresi kullanılarak saldırıyı açıklar nitelikte eposta gönderilmesidir.

WUBKU Saldırı Takip Arayüzü; takipçi rolüne sahip kullanıcıların yetkili olduğu uygulamaların bütünlüğünün bozulması durumunda, WUBKU'nun sağladığı arayüz üzerinden yapılan saldırıyı açıklar nitelikteki bilgiler ve uygulama üzerinde gerçekleşen yetkili işlemlerin takibinin yapılabilmesidir.

SMS (Kısa Mesaj Servisi); WUBKU SMS göndermeyi desteklemektedir. Fakat kullanıcıların bunu kendilerinin yapılandırması gerekmektedir.



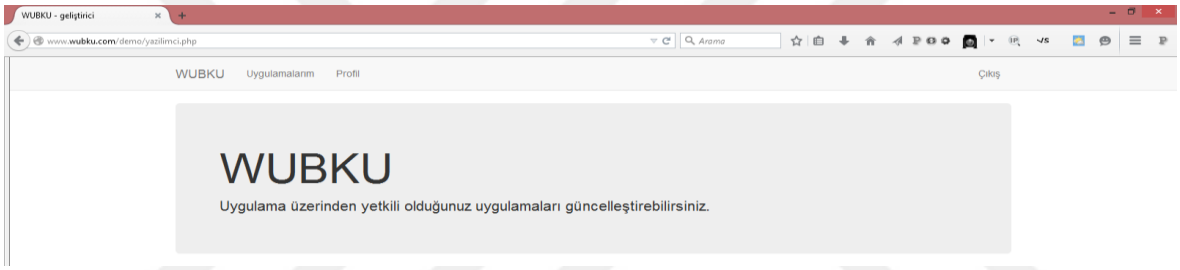
Şekil 4.13. WUBKU ayarları yapılandırma sayfası

4.3.4. WUBKU yazılımcı işlemleri

Yazılımcı rolüne sahip kullanıcı, yönetici tarafından yetkilendirildiği uygulamalar üzerinde işlem yapmaya yetkili kullanıcıdır. Yazılımcı rolü ile aşağıdaki işlemler yapılabilmektedir.

- Uygulama üzerinde dosya ekleme, silme ve düzenleme,
- Uygulama üzerinde dizin ekleme, silme ve düzenleme,
- Kendine ait bilgileri düzenleme,

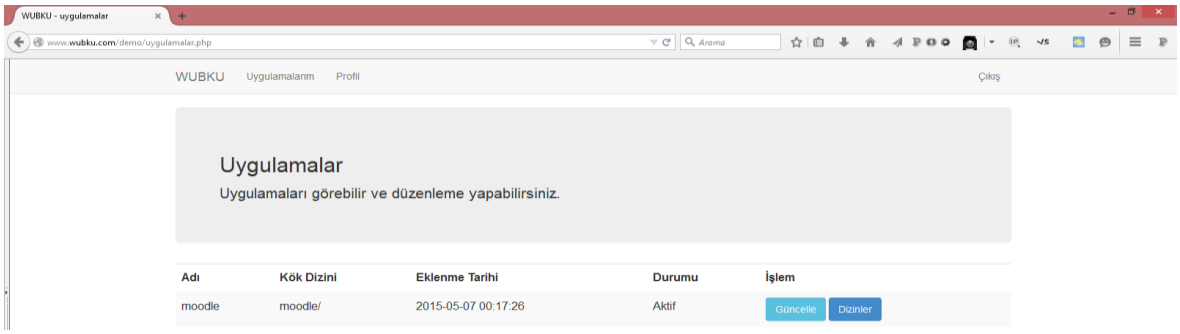
Yazılımcı rolü ile WUBKU'ya giriş yapıldığında açılan sayfa Şekil 4.14'te gösterilmektedir.



Şekil 4.14. Yazılımcı anasayfası

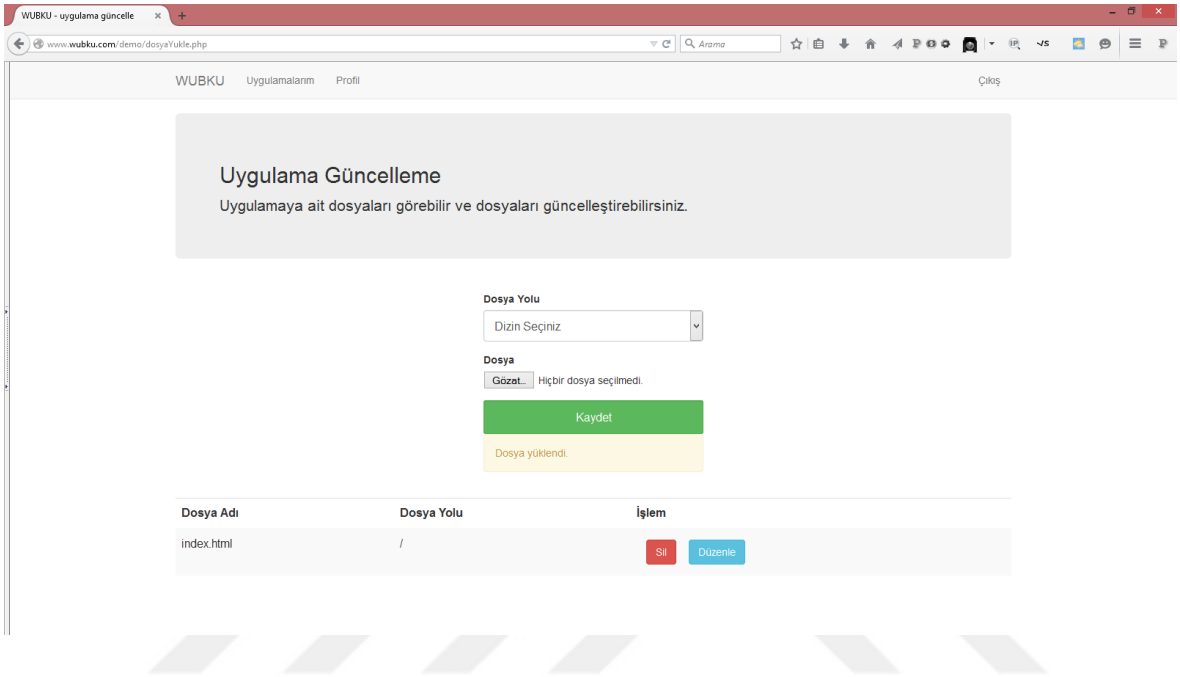
Yazılımcı rolü dosya işlemleri

Yazılımcı rolüne sahip kullanıcılar, yetkili olduğu uygulama üzerinde dosya ekleme, dosya silme ve var olan dosyayı değiştirebilme işlemlerini yapabilmektedir. Yetkili olduğu mevcut uygulamaların görüntülendiği sayfa Şekil 4.15'te gösterilmektedir.



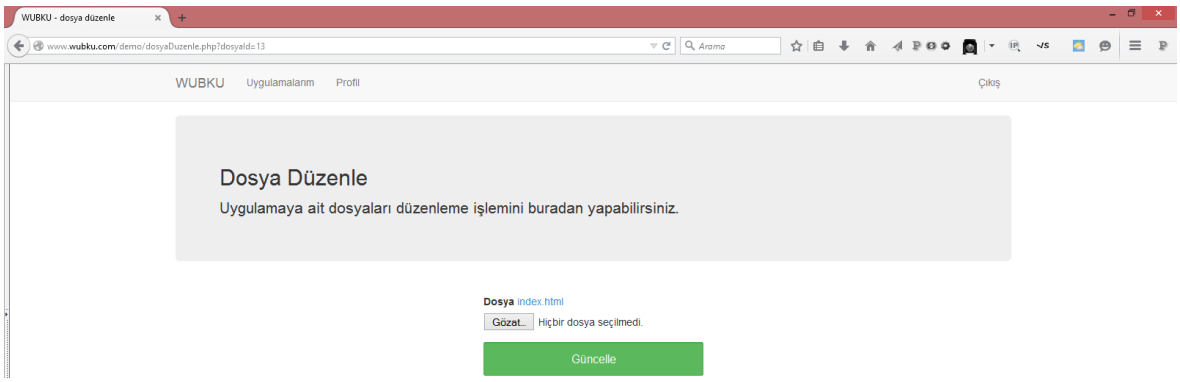
Şekil 4.15. Yazılımcı rolüne sahip kullanıcıların yetkili olduğu uygulamaların görüntülendiği sayfa

Uygulamaların görüntülendiği sayfa üzerinden dosya ekleme/silme/güncelleştirme işlemleri için “Güncelle” butonu kullanılmaktadır. Uygulama güncelleme işlemlerinin yapıldığı sayfa Şekil 4.16’da gösterilmektedir. Bu dosya üzerinden, “Dosya Yolu” seçeneği ile dosya istenen dizine yüklenebilmektedir.



Şekil 4.16. Uygulama üzerindeki mevcut dosya görüntüleme ve dosya ekleme sayfası

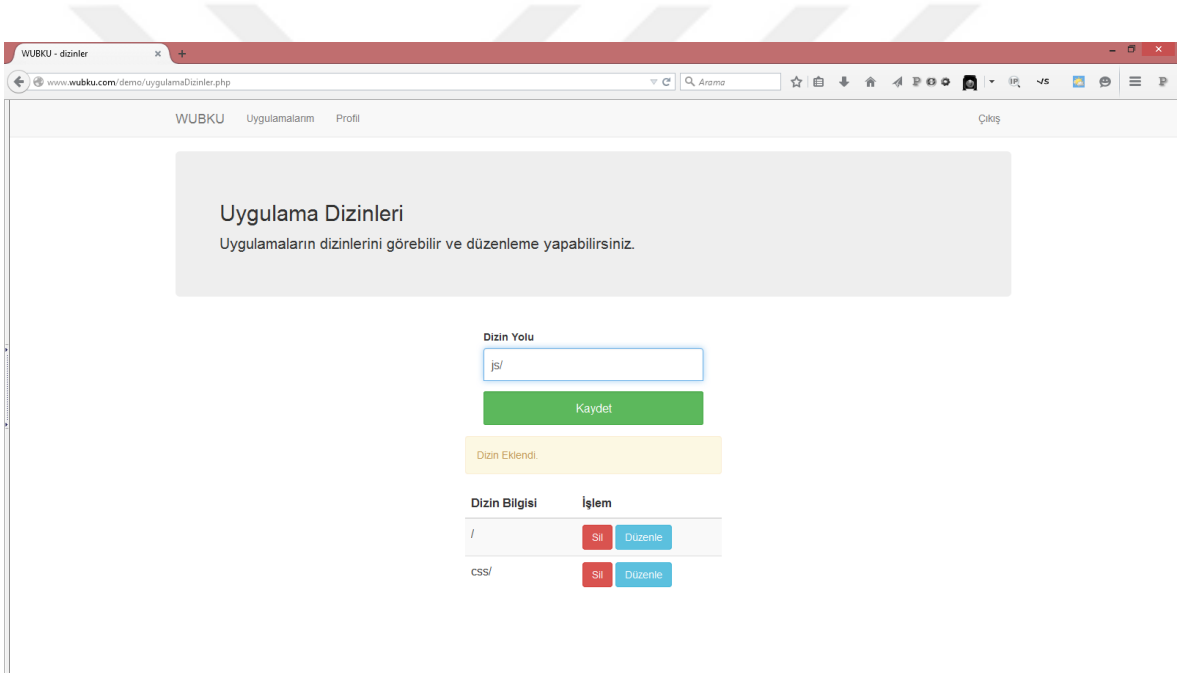
Uygulama güncelleme işlemlerinin yapıldığı sayfa üzerinden mevcut dosyalar düzenlenebilmektedir. Mevcut dosya ile aynı satırda bulunan “Düzenle” butonu üzerinden, mevcut dosya indirilebilmekte ve güncellenebilmektedir. Yine aynı sayfa üzerinden “Sil” butonu ile dosya silinebilmektedir. Mevcut dosya güncelleme sayfası Şekil 4.17’de gösterilmektedir.



Şekil 4.17. Uygulama üzerine eklenen dosyanın indirme ve düzenleme sayfası

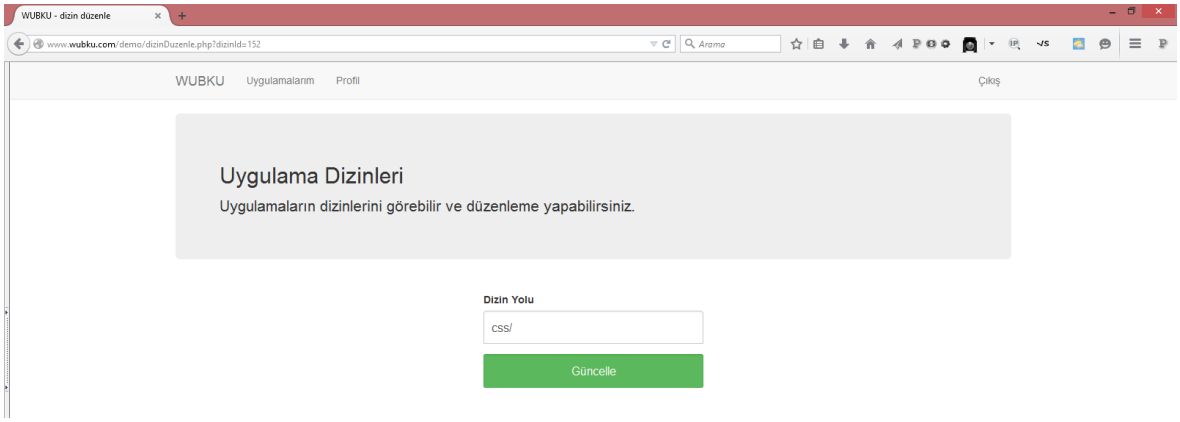
Yazılımcı rolü dizin işlemleri

Yazılımcı rolüne sahip kullanıcılar, yetkili olduğu uygulama üzerinde dizin ekleme, dizin silme ve var olan dizini değiştirebilme işlemlerini yapabilmektedir. Yönetici rolüne sahip kullanıcı tarafından uygulama eklendiğinde kök dizin otomatik olarak oluşturulmaktadır. Bu yüzden kök dizin eklemeye gerek yoktur. İhtiyaç duyulan diğer dizinler ise girilen ifadenin başında "/" karakterinin olmayacak şekilde girilmesine dikkat edilmelidir. Kullanıcın yetkili olduğu uygulamaların görüntülendiği sayfa üzerinde bulunan "Dizinler" butonu üzerinden dizinler üzerinde işlemler yapılabilmektedir. Dizin ekleme ve mevcut dizinlerin görüntülendiği sayfa Şekil 4.18'de gösterilmektedir.



Şekil 4.18. Uygulama üzerine dizin ekleme ve mevcut dizinler sayfası

Uygulama üzerinde bulunan mevcut dizinler, dizinlerin görüntülendiği sayfa üzerinden, dizin isimleri ile aynı satırda bulunan "Düzenle" butonu üzerinden güncellenebilmektedir. Yine aynı sayfa üzerinde bulunan "Sil" butonu ile dizin silinebilmektedir. Dizin güncelleme işleminin yapıldığı sayfa Şekil 4.19'da gösterilmektedir.



Şekil 4.19. Uygulama üzerindeki mevcut dizini düzenleme sayfası

4.3.5. WUBKU takipçi işlemleri

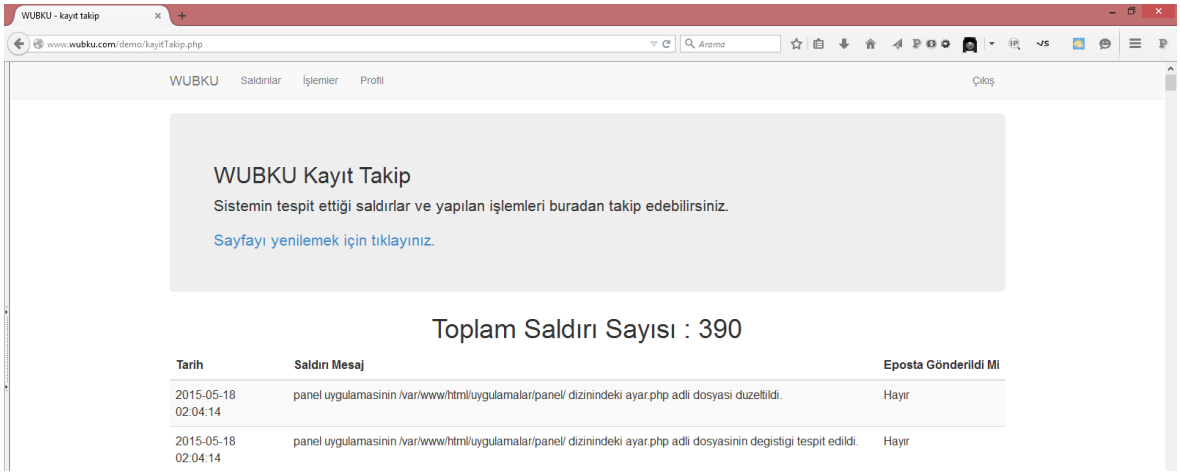
Takipçi rolüne sahip kullanıcı, yönetici tarafından yetkilendirildiği uygulamalar üzerinde tespit edilen saldırıları ve yapılan yetkili işlemleri görüntülemeye yetkili kullanıcıdır. Takipçi hesabı ile aşağıdaki işlemler yapılabilmektedir.

- Uygulama üzerinde tespit edilen saldırıları görüntüleme,
- Uygulama üzerinde yapılan yetkili işlemleri görüntüleme,
- Kendine ait bilgileri düzenleme,

Takipçi ekranı 5 saniyede bir kendini otomatik yenilemektedir ve bu şekilde sürekli güncel durumun takibi sağlanmaktadır.

Takipçi rolü saldırı takip işlemleri

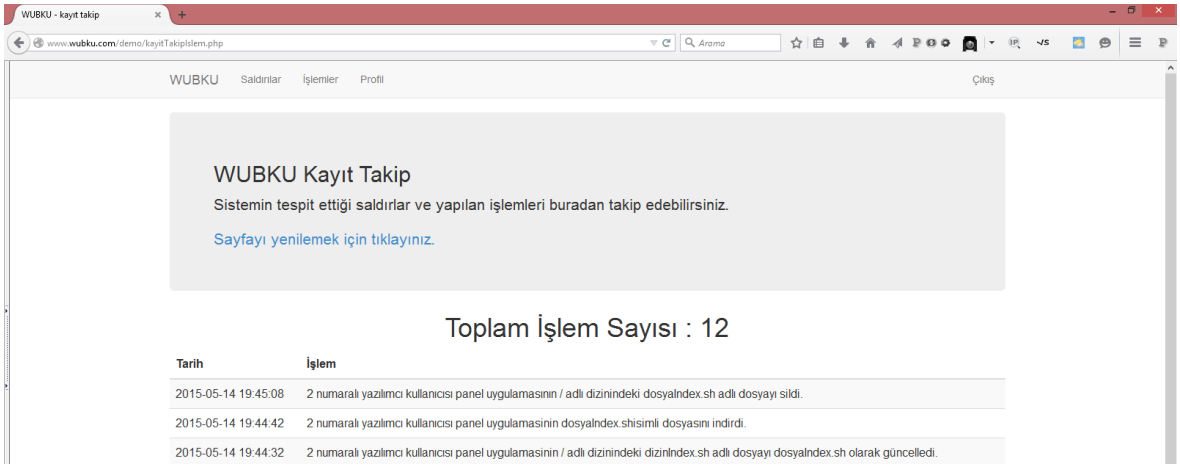
Takipçi rolüne sahip kullanıcı WUBKU'ya giriş yaptığında, tespit edilen saldırıların görüntülendiği sayfa açılmaktadır. Tespit edilen saldırıların görüntülendiği sayfa Şekil 4.20'de gösterilmektedir. Bu sayfa üzerinden, hem bütünlüğü bozucu saldırıyı hem de bütünlüğün tekrardan sağlanmasını açıklar nitelikteki çeşitli bilgiler görüntülenebilmektedir.



Şekil 4.20. Takipçi rolü saldırı takip sayfası

Takipçi rolü yetkili işlem takip işlemleri

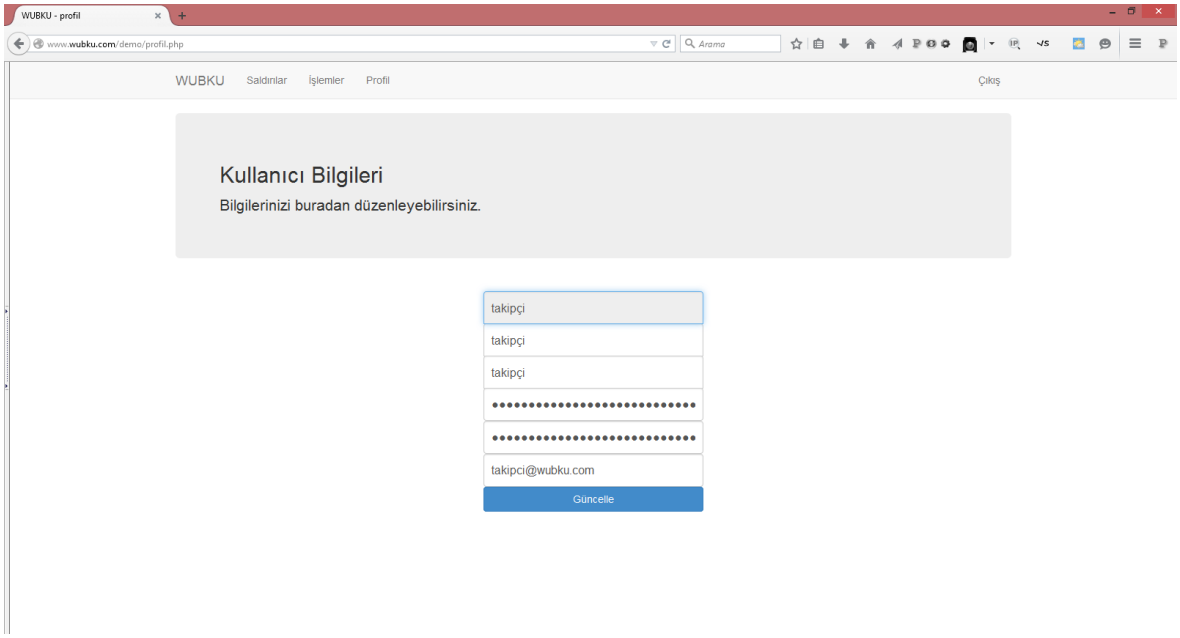
Uygulamalar üzerinde yetkili olarak gerçekleşen, uygulama ekleme, silme, uygulama değiştirme, dosya ekleme, dosya silme, dosya değiştirme, dizin ekleme, dizin silme ve dizin değiştirme işlemlerinin takibi "İşlemler" seçeneği üzerinden görüntülenebilmektedir. Bu sayfa üzerinden, uygulama üzerinde yapılan değişikliğin tarih-saat bilgisi, kullanıcıların uygulamalar üzerinde hangi işlemleri yaptığı bilgileri görüntülenebilmektedir. Uygulamalar üzerinde gerçekleşen yetkili işlemlerin görüntülendiği sayfa Şekil 4.21'de gösterilmektedir.



Şekil 4.21. Takipçi rolü yetkili işlemler takip sayfası

4.3.6. WUBKU profil işlemleri

WUBKU kullanıcıları, WUBKU'ya giriş yaptıktan sonra menü üzerinde bulunan "Profil" seçeneği üzerinden, ad, soyad, şifre ve eposta adresi bilgilerini güncelleyebilmektedir. Profil bilgilerinin güncellenebildiği sayfa Şekil 4.22'de gösterilmektedir.



Şekil 4.22. Yetkili kullanıcı profil bilgilerini görüntüleme ve güncelleme sayfası

5. TEST ÇALIŞMALARI

Bu bölümde geliştirilen WUBKU'nun, uygulamanın bütünlüğünü bozucu bir saldırı olduğunda saldırıyı tespit etme ve saldırı sonucu bütünlüğü bozulan uygulamanın otomatik olarak düzeltilmesinin testi gerçekleştirilmiştir. Yapılan testte, uygulamaların bütünlüğü aşağıdaki işlemler yapılarak bozulmuştur.

- Uygulama üzerine yeni bir dosya ekleme,
- Uyguma üzerinde var olan bir dosyayı silme,
- Uygulama üzerinde var olan bir dosyayı değiştirme,
- Uygulama üzerinde yeni bir dizin oluşturma,
- Uygulama dosyalarının bulunduğu dizini silme,
- Bütün uygulamaların bulunduğu dizini silme,

Test işlemleri, Intel Core i5-450M 2.4 GHz (Gigahertz) 3 MB (Megabayt) Ön Bellek İşlemcisi ve 8 GB (Gigabayt) DDR3 (Çift Veri Hızı) 1066 MHz (Megahertz) bellek donanım özelliklerine sahip Windows 8 işletim sistem sistemi üzerine kurulu sanal sunucu üzerinden gerçekleştirilmiştir. Sanal sunucu donanım özellikleri 2048 MB bellek ve 1 merkezi işlem birimi olarak belirlenmiştir. Sanal sunucu üzerine işletim sistemi olarak Ubuntu 14.04.4 LTS (Uzun Dönem Destek) işletim sistemi kurulmuştur.

Yapılan test çalışması kapsamında, internetten edinilen 4 farklı web uygulaması ile tez kapsamında geliştirilen WUBKU olmak üzere, toplamda 5 web uygulamasıyla test edilmiştir. İnternetten edinilen uygulamalardan iki tanesi kişilerin oluşturup paylaşıma sunduğu uygulamalar arasından seçilmiştir. Diğer ikisi ise gerçek bir uygulama testinin gerçekleştirilmesi amacıyla açık kaynak kodlu olan uygulamalar arasından seçilmiştir. Uygulamalar seçilirken uygulama boyutu, uygulama dizin sayısı ve dosya sayıları dikkate alınmıştır. Test için kullanılan uygulama bilgileri Çizelge 5.1'de verilmiştir.

Çizelge 5.1. WUBKU'nun test edildiği uygulama bilgileri

Sıra Nu.	Uygulama Adı	Dizin Sayısı	Dosya Sayısı	Toplam Boyutu (KB)
1.	Panel	1	10	44
2.	Ticket	21	322	3320
3.	Redmine	66	356	3000
4.	Drupal	146	1087	15584
5.	WUBKU	3	40	624

Test işlemlerinde, Bash uygulamasıyla hazırlanan bir betik ile uygulamaların bütünlüğü bozulmuştur. WUBKU'nun bütünlüğü bozucu durumu ne kadar sürede tespit edebildiğinin hesaplanabilmesi için hazırlanan betik kullanılarak bütünlüğün bozulduğu zaman bilgisi kayıt altına alınmıştır. Uygulamanın bütünlüğünün bozulması durumunda, bütünlüğün bozulduğu zaman bilgisi ile bütünlüğün tekrardan sağlandığı bilgisi WUBKU tarafından kayıt altına alınmaktadır. Bu yüzden testlerde kullanılan tespit zamanı ve düzeltilme zamanı bilgileri için ayrıca kod yazılmamış, WUBKU kayıtları kullanılmıştır. Bu bilgiler kullanılarak bütünlüğün tekrardan sağlanma süresi elde edilmiştir.

Testte kullanılan uygulamalar sırasıyla Panel, Ticket, Redmine, Drupal ve WUBKU olmak üzere WUBKU göç ile WUBKU üzerine aktarılmıştır. Bu şekilde WUBKU göçün de düzgün çalışıp çalışmadığının testi yapılmıştır. Bu uygulamalar arasında Redmine, Ruby programlama dilinde yazılmış olan Tren Rayları (rails) çatısı (framework) kullanılarak geliştirilmiştir [42]. Test için özellikle bu uygulama seçilerek hem farklı programlama dillerinde yazılmış hem de apache üzerinde çalışabilen uygulamaların bütünlüğünün kontrol edildiğiyle otomatik olarak sağlandığının testi yapılmıştır. WUBKU test işleminde daha doğru sonuçlar elde etmek için farklı boyutlardaki dosyaların 10 defa bütünlüğü bozularak test işlemi yapılmıştır. Test işleminde kullanılan dosya boyutları Çizelge 5.2'de verilmiştir.

Çizelge 5.2. Test işlemlerinde kullanılan dosya boyutları

Sıra Nu.	Dosya Boyutu (KB)
1.	0-500
2.	501 – 1000
3.	1001-1500
4.	1501 ve üstü

Test işlemlerinde daha gerçekçi sonuçlar elde etmek amacıyla, kullanılan dosya boyutlarının yukarıda verilen dosya boyutu aralıklarına olabildiğince düzgün dağılım olacak şekilde olmasına dikkat edilmiştir. Ayrıca test işlemi yapılırken farklı uzantılı dosyaların seçilmesi işlemine dikkat edilerek, farklı uzantıya sahip dosyaların bütünlüğünün de kontrol edilebildiği ve otomatik olarak düzeltilip düzeltilemediği test edilmiştir. Test işlemlerinde ilk olarak dosya bütünlüğünün bozulması test edilmiştir. Dosya bütünlüğünün bozulması için kullanılan betik aracılığıyla, dosyanın silinmesi, dosyanın içeriğinin silinmesi ve dosya içeriğine metin eklenmesi işlemleri rastgele seçilerek test gerçekleştirilmiştir. Dosya bütünlüğünün bozulması testinden sonra başka bir betik yardımıyla, uygulama dizinleri içerisinde rastgele isimli dosyalar oluşturularak uygulama bütünlüğü bozulmuş ve uygulama içerisinde yetkisiz eklenen dosyaların tespit edilip edilemediğinin testi yapılmıştır. Bu testten sonra, başka bir betik aracılığıyla uygulama dizinleri içerisinde rastgele isimli dizinler oluşturularak uygulama bütünlüğü bozulmuş ve uygulama içerisinde yetkisiz eklenen dizinlerin tespit edilip edilemediğinin testi yapılmıştır. Dosya ve dizinler üzerinden bütünlüğün bozulması testlerinden sonra uygulama dosya ve dizinlerinin bulunduğu ana dizin silinerek, uygulamanın tamamı silinse bile tespit ve düzeltme yapılıp yapılamadığının testi yapılmıştır. Testlerde son olarak WUBKU üzerinde bulunan 5 uygulamanın da bulunduğu ana dizin silinerek, uygulamalara ait tüm dosya ve dizin yapısı silinse bile, WUBKU tarafından bütünlüğün bozulduğunun testinin yapılabildiği ve uygulamaların bütünlüğünün tekrardan sağlanabildiğinin testi yapılmıştır.

Panel uygulamasının dosya bütünlüğünün bozulması testlerinde kullanılan dosya bilgileri Çizelge 5.3, 5.4, 5.5 ve 5.6'da verilmiştir. Panel uygulamasında toplam dosya sayısı 10 olduğundan farklı boyuttaki dosyaların test işlemlerinde aynı dosyanın bütünlüğü birden fazla bozularak test işlemi yapılmıştır. Panel uygulaması üzerinde yeni dosya ve dizin ekleme testleri, uygulamaya ait tek dizin bulunduğundan aynı dizin üzerinde 10 defa dosya ve dizin ekleme işlemi yapılarak test edilmiştir.

Çizelge 5.3. Panel uygulamasına ait 0 – 500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Bulunduğu Dizin	Dosya Adı
1.	142	/	cikis.php
2.	334	/	ayar.php
3.	409	/	style.css

Çizelge 5.4. Panel uygulamasına ait 501 – 1000 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Bulunduğu Dizin	Dosya Adı
1.	535	/	uyesil.php
2.	846	/	index.php
3.	929	/	giris.php

Çizelge 5.5. Panel uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Bulunduğu Dizin	Dosya Adı
1.	1439	/	uyeler.php

Çizelge 5.6. Panel uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Bulunduğu Dizin	Dosya Adı
1.	1527	/	yeniuye.php
2.	1894	/	uyeduzenle.php
3.	2064	/	panelim.php

Ticket uygulamasının dosya bütünlüğünün bozulması testlerin de kullanılan dosya bilgileri Çizelge 5.7, 5.8, 5.9 ve 5.10'da verilmiştir. Ticket uygulaması dosyaları arasında, farklı boyutlarda yeteri kadar dosya bulunmadığından farklı boyuttaki dosyaların test işlemlerinde aynı dosyanın bütünlüğü birden fazla bozularak test işlemi yapılmıştır.

Çizelge 5.7. Ticket uygulamasına ait 0 – 500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	43	transparent.gif	/csladmin/images/
2.	45	mm_dashed_line.gif	/csladmin/
3.	127	cal.gif	/csladmin/
4.	226	test.php.mno	/_notes/_notes/
5.	285	dwsync.xml	/csladmin/_mmServerScripts/_notes/
6.	285	dwsync.xml	/_mmServerScripts/_notes/
7.	329	cslconn.php	/_notes/Connections/
8.	338	cslconn.php	/Connections/
9.	433	dbcon.php	/_notes/csladmin/
10.	472	Sourcemaster.php.mno	/csladmin/_notes/

Çizelge 5.8. Ticket uygulamasına ait 501 – 1000 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	515	Session.php	/csladmin/
2.	515	Session.php	/_notes/csladmin/
3.	515	Session.php	/_notes/
4.	515	Session.php	/

Çizelge 5.9. Ticket uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	1060	date.gif	/csladmin/
2.	1236	top1.php	/csladmin/
3.	1382	welcome.php	/csladmin/
4.	1382	welcome.php	/csladmin/
5.	1433	top.php	/csladmin/
6.	1433	top.php	/_notes/csladmin/
7.	1433	top.php	/_notes/
8.	1433	top.php	/
9.	1462	newie.css	/includes/
10.	1496	newie.css	/csladmin/includes/

Çizelge 5.10. Ticket uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	6220	check_user.php	/_notes/csladmin/
2.	6358	selectshow1.php	/_notes/csladmin/
3.	6379	selectshowwalkin.php	/csladmin/
4.	6586	diffSoucecodesf.php	/
5.	7656	ShowUpdate.php.bak	/_notes/csladmin/
6.	8194	date-picker.js	/_notes/
7.	13018	showup.php	/csladmin/sourcecodemst/
8.	45380	header1.gif	/_notes/
9.	60871	header-cOMIC1d.JPG	/csladmin/
10.	123267	csl_db.sql	/

Redmine uygulaması açık kaynak kodlu bir proje yönetim uygulamasıdır [42]. Redmine uygulaması, http://www.redmine.org/ bağlantısı üzerinden indirilebilmektedir. Testlerin gerçeğe daha yakın olması amacıyla yaygın olarak kullanılan Redmine uygulaması seçilmiştir. Redmine uygulaması testlerinde kullanılan dosya bilgileri Çizelge 5.11, 5.12, 5.13 ve 5.14'te verilmiştir. Redmine uygulaması test edilirken, farklı uzantıdaki dosyaların seçilmesine dikkat edilmiştir.

Çizelge 5.11. Redmine uygulamasına ait 0 – 500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	112	close.png	/images/
2.	144	bullet_toggle_plus.png	/images/
3.	246	bt_ol.png	/images/jstoolbar/
4.	308	context_menu_rtl.css	/stylesheets/
5.	328	openid-bg.gif	/images/
6.	400	textfield_key.png	/images/
7.	403	bt_img.png	/images/jstoolbar/
8.	469	datepicker.js	/javascripts/
9.	473	dispatch.fcgi	/
10.	479	php.png	/images/files/

Çizelge 5.12. Redmine uygulamasına ait 501 – 1000 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	505	js.png	/images/files/
2.	519	edit.png	/images/
3.	526	wiki_edit.png	/images/
4.	529	ticket_checked.png	/images/
5.	539	xml.png	/images/files/
6.	639	jstoolbar-bs.js	/javascripts/jstoolbar/lang/
7.	721	jstoolbar-hr.js	/javascripts/jstoolbar/lang/
8.	726	jstoolbar-no.js	/javascripts/jstoolbar/lang/
9.	745	jstoolbar-ro.js	/javascripts/jstoolbar/lang/
10.	991	jquery.ui.datepicker-ro.js	/javascripts/i18n/

Çizelge 5.13. Redmine uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	1003	draft.png	/images/
2.	1037	jquery.ui.datepicker-sr.js	/javascripts/i18n/
3.	1094	jquery.ui.datepicker-vi.js	/javascripts/i18n/
4.	1116	jquery.ui.datepicker-ru.js	/javascripts/i18n/
5.	1165	jquery.ui.datepicker-fa.js	/javascripts/i18n/
6.	1182	jquery.ui.datepicker-uk.js	/javascripts/i18n/
7.	1198	jquery.ui.datepicker-el.js	/javascripts/i18n/
8.	1208	jquery.ui.datepicker-bg.js	/javascripts/i18n/
9.	1275	jquery.ui.datepicker-th.js	/javascripts/i18n/
10.	1296	jquery.ui.datepicker-ar.js	/javascripts/i18n/

Çizelge 5.14. Redmine uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	4211	wiki_syntax.html	/help/ar/
2.	4243	wiki_syntax.html	/help/fr/
3.	4466	textile.js	/javascripts/jstoolbar/
4.	5732	attachments.js	/javascripts/
5.	6545	context_menu.js	/javascripts/
6.	7500	gantt.js	/javascripts/
7.	11177	jstoolbar.js	/javascripts/jstoolbar/
8.	12864	wiki_syntax_detailed.html	/help/ar/
9.	47460	application.css	/stylesheets/
10.	337799	jquery-1.8.3-ui-1.9.2-ujs-2.0.3.js	/javascripts/

Drupal uygulaması açık kaynak kodlu bir içerik yönetim uygulamasıdır [43]. Drupal uygulaması, https://www.drupal.org/ bağlantısı üzerinden indirilebilmektedir. Drupal uygulaması testlerinde kullanılan dosya bilgileri Çizelge 5.15, 5.16, 5.17 ve 5.18'de verilmiştir. Drupal uygulaması test edilirken, farklı uzantıdaki dosyaların seçilmesine özen gösterilmiştir.

Çizelge 5.15. Drupal uygulamasına ait 0 – 500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	104	shortcut.admin.css	/modules/shortcut/
2.	176	menu-collapsed-rtl.gif	/themes/garland/images/
3.	191	drupal_autoload_test_interface.inc	/modules/simpletest/tests/ drupal_autoload_test/
4.	230	lock.png	/modules/color/images/
5.	266	syslog.install	/modules/syslog/
6.	305	entity_cache_test_dependency .module	/modules/simpletest/tests/
7.	365	rdf.info	/modules/rdf/
8.	420	path.js	/modules/path/
9.	441	aggregator_test_title_entities.xml	/modules/aggregator/tests/
10.	444	README.txt	/themes/

Çizelge 5.16. Drupal uygulamasına ait 501 – 1000 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	558	shortcut.png	/modules/shortcut/
2.	656	print.css	/themes/bartik/css/
3.	747	taxonomy_test.install	/modules/simpletest/tests/
4.	844	comment_hacks.css.optimized.css	/modules/simpletest/files/ css_test_files/
5.	816	jquery.effects.transfer.min.js	/misc/ui/
6.	918	user-profile-item.tpl.php	/modules/user/
7.	928	update.user.test	/modules/simpletest/tests/ upgrade/
8.	943	powered-blue-80x15.png	/drupal/misc/
9.	968	authorize.js	/misc/
10.	992	node.tpl.php	/themes/garland/

Çizelge 5.17. Drupal uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	1004	README.txt	/themes/stark/
2.	1029	node_access_test.install	/modules/node/tests/
3.	1205	aaa_update_test.1_0.xml	/modules/update/tests/
4.	1323	update_script_test.install	/ modules/simpletest/tests/
5.	1363	jquery.ui.dialog.css	/misc/ui/
6.	1398	dblog.css	/modules/dblog/
7.	1433	form_test.file.inc	/modules/simpletest/tests/
8.	1468	preview.js	/modules/color/
9.	1474	system.admin-rtl.css	/modules/system/
10.	1494	update.aggregator.test	/modules/simpletest/tests/upgrade/

Çizelge 5.18. Drupal uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	2009	powered-blue-88x31.png	/misc/
2.	2331	upgrade.forum.test	/modules/simpletest/tests/ upgrade/
3.	2593	aggregator_test_rss091.xml	/modules/aggregator/tests/
4.	5430	favicon.ico	/misc/
5.	5872	progress.gif	/misc/
6.	7096	dashboard.js	/modules/dashboard/
7.	15149	user.api.php	/modules/user/
8.	26832	openid.inc	/modules/openid/
9.	97210	CHANGELOG.txt	/
10.	115218	system.install	/modules/system/

Tez kapsamında geliştirilen WUBKU uygulamasına ait dosya bilgileri Çizelge 5.19, 5.20, 5.21'de verilmiştir. WUBKU'ya ait 0 – 500 KB boyutunda dosya bulunmadığından bu test yapılamamıştır.

Çizelge 5.19. WUBKU uygulamasına ait 501 – 1000 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	597	signin.css	/css/
2.	902	index.php	/

Çizelge 5.20. WUBKU uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	1202	vtBaglanti.php	/
2.	1316	yazilimci.php	/
3.	1354	yonetici.php	/
4.	1433	oturumKontrol.php	/

Çizelge 5.21. WUBKU uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosya bilgileri

Sıra Nu.	Dosya Boyutu (KB)	Dosya Adı	Bulunduğu Dizin
1.	1696	dosyaIndir.php	/
2.	1756	baslik.php	/
3.	2043	dosyalar.php	/
4.	2549	guvenliWebTakip.php	/
5.	3293	fonksiyonlar.php	/
6.	4235	kullaniciEkle.php	/
7.	5975	dosyaYukle.php	/
8.	31308	smtp.php	/
9.	93117	jquery.js	/js/
10.	129989	bootstrap.css	/css/

Uygulamalar sırasıyla WUBKU göç aracılığıyla WUBKU üzerine aktarılmıştır. Ardından uygulamaların bütünlüğü bozularak, saldırı zamanı, saldırı tespit zamanı ve saldırı sonucu

verilen zararın düzeltilme zamanı bulunmuştur. Panel uygulaması için farklı boyutlardaki dosyalar için yapılan test sonuçları Çizelge 5.22, 5.23, 5.24 ve 5.25'te verilmiştir.

Çizelge 5.22. Panel uygulamasına ait 0 – 500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	cikis.php	/	11:38:40	11:38:40	11:38:40
2.	ayar.php	/	11:38:55	11:38:55	11:38:55
3.	style.css	/	11:39:10	11:39:10	11:39:10
4.	cikis.php	/	11:39:25	11:39:25	11:39:25
5.	ayar.php	/	11:39:40	11:39:40	11:39:40
6.	style.css	/	11:39:55	11:39:56	11:39:56
7.	cikis.php	/	11:40:10	11:40:10	11:40:10
8.	ayar.php	/	11:40:25	11:40:26	11:40:26
9.	style.css	/	11:40:40	11:40:41	11:40:41
10.	cikis.php	/	11:40:55	11:40:55	11:40:55
Ortalama Süreler:				~0 Sn.	~0 Sn.

Çizelge 5.23. Panel uygulamasına ait 501 – 1000 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	uyesil.php	/	11:41:10	11:41:11	11:41:11
2.	index.php	/	11:41:25	11:41:25	11:41:25
3.	giris.php	/	11:41:40	11:41:40	11:41:40
4.	uyesil.php	/	11:41:55	11:41:56	11:41:56
5.	index.php	/	11:42:10	11:42:10	11:42:10
6.	giris.php	/	11:42:25	11:42:26	11:42:26
7.	uyesil.php	/	11:42:40	11:42:41	11:42:41
8.	index.php	/	11:42:55	11:42:56	11:42:56
9.	giris.php	/	11:43:10	11:43:11	11:43:11
10.	uyesil.php	/	11:43:25	11:43:26	11:43:26
Ortalama Süreler:				~1 Sn.	~0 Sn.

Çizelge 5.24. Panel uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	uyeler.php	/	11:43:40	11:43:41	11:43:41
2.	uyeler.php	/	11:43:55	11:43:56	11:43:56
3.	uyeler.php	/	11:44:10	11:44:11	11:44:11
4.	uyeler.php	/	11:44:25	11:44:26	11:44:26
5.	uyeler.php	/	11:44:40	11:44:41	11:44:41
6.	uyeler.php	/	11:44:55	11:44:56	11:44:56
7.	uyeler.php	/	11:45:10	11:45:11	11:45:11
8.	uyeler.php	/	11:45:25	11:45:26	11:45:26
9.	uyeler.php	/	11:45:40	11:45:41	11:45:41
10.	uyeler.php	/	11:45:55	11:45:56	11:45:56
Ortalama Süreler:				~1 Sn.	~0 Sn.

Çizelge 5.25. Panel uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	uyeduzenle.php	/	11:46:10	11:46:11	11:46:11
2.	yeniuye.php	/	11:46:25	11:46:26	11:46:26
3.	panelim.php	/	11:46:41	11:46:41	11:46:41
4.	uyeduzenle.php	/	11:46:56	11:46:56	11:46:56
5.	yeniuye.php	/	11:47:11	11:47:11	11:47:11
6.	panelim.php	/	11:47:26	11:47:27	11:47:27
7.	uyeduzenle.php	/	11:47:41	11:47:41	11:47:41
8.	yeniuye.php	/	11:47:56	11:47:56	11:47:56
9.	panelim.php	/	11:48:11	11:48:11	11:48:11
10.	uyeduzenle.php	/	11:48:26	11:48:26	11:48:26
Ortalama Süreler:				~0 Sn.	~0 Sn.

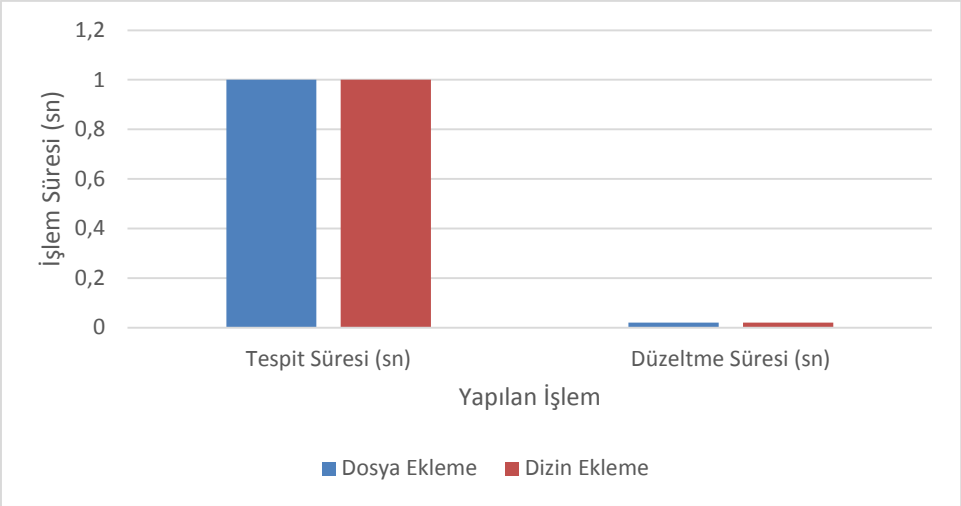
Panel uygulaması için yapılan dosya ve dizin ekleme test sonuçları Çizelge 5.26 ve 5.27'de verilmiştir.

Çizelge 5.26. Panel uygulaması kök dizinine yeni dosya ekleme test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	31392	/	11:48:41	11:48:42	11:48:42
2.	30186	/	11:48:56	11:48:57	11:48:57
3.	26061	/	11:49:11	11:49:12	11:49:12
4.	16973	/	11:49:26	11:49:27	11:49:27
5.	9672	/	11:49:41	11:49:42	11:49:42
6.	677	/	11:49:56	11:49:57	11:49:57
7.	8366	/	11:50:11	11:50:12	11:50:12
8.	25224	/	11:50:26	11:50:26	11:50:26
9.	3782	/	11:50:41	11:50:42	11:50:42
10.	7359	/	11:50:56	11:50:56	11:50:56
Ortalama Süreler:				~1 Sn.	~0 Sn.

Çizelge 5.27. Panel uygulaması kök dizinine yeni dizin ekleme test sonuçları

Sıra Nu.	Dizin Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	30099	/	11:51:11	11:51:12	11:51:12
2.	22398	/	11:51:26	11:51:27	11:51:27
3.	30002	/	11:51:41	11:51:41	11:51:41
4.	26266	/	11:51:56	11:51:57	11:51:57
5.	12214	/	11:52:11	11:52:11	11:52:11
6.	20587	/	11:52:26	11:52:27	11:52:27
7.	11154	/	11:52:41	11:52:41	11:52:41
8.	11213	/	11:52:56	11:52:57	11:52:57
9.	25019	/	11:53:11	11:53:12	11:53:12
10.	21109	/	11:53:26	11:53:26	11:53:26
Ortalama Süreler:				~1 Sn.	~0 Sn.



Şekil 5.1. Panel uygulaması dosya ve dizin ekleme testi ortalama tespit ve düzeltilme Süreleri

Ticket uygulaması için farklı boyutlardaki dosyalar için yapılan test sonuçları Çizelge 5.28, 5.29 ve 5.30, 5.31'de verilmiştir.

Çizelge 5.28. Ticket uygulamasına ait 0 – 500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltme Zamanı
1.	transparent.gif	/csladmin/images/	13:26:05	13:26:19	13:26:19
2.	mm_dashed _line.gif	/csladmin/	13:26:35	13:26:54	13:26:54
3.	cal.gif	/csladmin/	13:27:05	13:27:25	13:27:25
4.	test.php.mno	/_notes/_notes/	13:27:35	13:28:01	13:28:01
5.	dwsync.xml	/csladmin/ _mmServerScripts/ _notes/	13:28:05	13:28:23	13:28:23
6.	dwsync.xml	/_mmServerScripts/_notes/	13:28:35	13:29:02	13:29:02
7.	cslconn.php	/_notes/Connections/	13:29:05	13:29:26	13:29:26
8.	cslconn.php	/Connections/	13:29:35	13:29:48	13:29:48
9.	dbcon.php	/_notes/csladmin/	13:30:05	13:30:28	13:30:29
10.	Sourcemaster. php.mno	/csladmin/_notes/	13:30:35	13:30:54	13:30:54
Ortalama Süreler:				~20 Sn.	~0 Sn.

Çizelge 5.29. Ticket uygulamasına ait 501 – 1000 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	Session.php	/csladmin/	13:31:05	13:31:21	13:31:21
2.	Session.php	/_notes/csladmin/	13:31:35	13:32:03	13:32:03
3.	Session.php	/_notes/	13:32:05	13:32:30	13:32:31
4.	Session.php	/	13:32:35	13:32:51	13:32:51
5.	Session.php	/csladmin/	13:33:05	13:33:22	13:33:22
6.	Session.php	/_notes/csladmin/	13:33:35	13:33:59	13:33:59
7.	Session.php	/_notes/	13:34:05	13:34:28	13:34:29
8.	Session.php	/	13:34:35	13:34:49	13:34:49
9.	Session.php	/csladmin/	13:35:05	13:35:21	13:35:21
10.	Session.php	/_notes/csladmin/	13:35:35	13:35:59	13:35:59
Ortalama Süreler:				~20 Sn.	~0 Sn.

Çizelge 5.30. Ticket uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	date.gif	/csladmin/	13:36:05	13:36:21	13:36:21
2.	top1.php	/csladmin/	13:36:35	13:36:53	13:36:53
3.	welcome.php	/csladmin/	13:37:05	13:37:20	13:37:20
4.	welcome.php	/csladmin/	13:37:35	13:37:50	13:37:50
5.	top.php	/csladmin/	13:38:05	13:38:20	13:38:20
6.	top.php	/_notes/csladmin/	13:38:35	13:38:59	13:38:59
7.	top.php	/_notes/	13:39:05	13:39:28	13:39:28
8.	top.php	/	13:39:35	13:39:48	13:39:48
9.	newie.css	/includes/	13:40:05	13:40:25	13:40:25
10.	newie.css	/csladmin/includes/	13:40:35	13:40:52	13:40:52
Ortalama Süreler:				~18 Sn.	~0 Sn.

Çizelge 5.31. Ticket uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	check_user.php	/_notes/csladmin/	13:41:05	13:41:29	13:41:29
2.	selectshow1.php	/_notes/csladmin/	13:41:35	13:42:02	13:42:02
3.	selectshowwalkin.php	/csladmin/	13:42:06	13:42:27	13:42:27
4.	diffSoucecodesf.php	/	13:42:36	13:43:00	13:43:00
5.	ShowUpdate.php.bak	/_notes/csladmin/	13:43:06	13:43:32	13:43:32
6.	date-picker.js	/_notes/	13:43:36	13:43:38	13:43:38
7.	showup.php	/csladmin/ sourcecodemst/	13:44:06	13:44:25	13:44:25
8.	header1.gif	/_notes/	13:44:36	13:44:36	13:44:36
9.	header-cOMIC1d.JPG	/csladmin/	13:45:06	13:45:22	13:45:22
10.	csl_db.sql	/	13:45:36	13:45:59	13:45:59
Ortalama Süreler:				~18 Sn.	~0 Sn.

Şekil 5.2. Ticket uygulamasına ait farklı boyuttaki dosyaların ortalama tespit ve düzeltilme süreleri

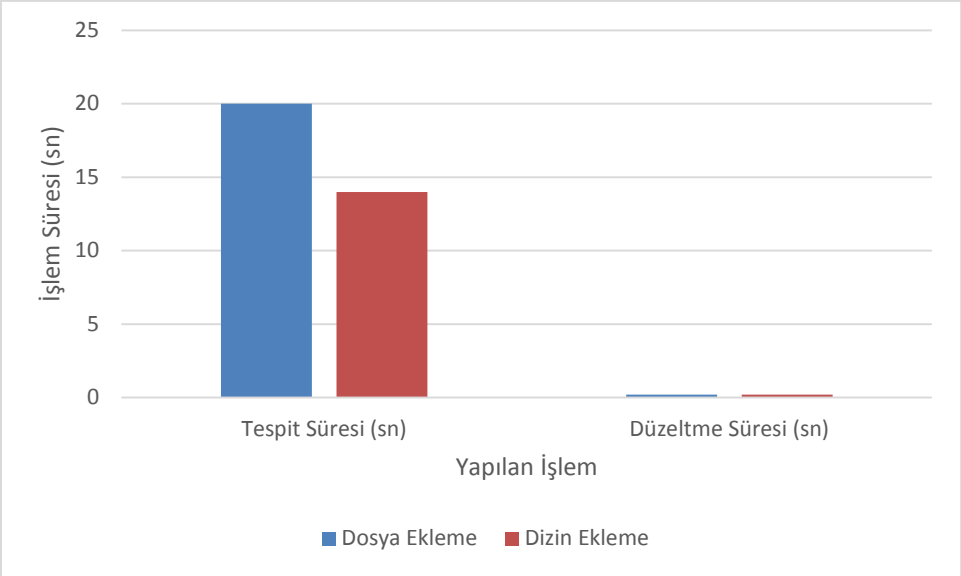
Ticket uygulaması için yapılan dosya ve dizin ekleme test sonuçları Çizelge 5.32 ve 5.33'te verilmiştir.

Çizelge 5.32. Ticket uygulaması dizinlerinde yeni dosya ekleme test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	10230	/csladmin/	19:30:16	19:30:28	19:30:28
2.	29648	/csladmin/includes/	19:31:16	19:31:22	19:31:23
3.	23527	/csladmin/_notes/	19:32:16	19:32:43	19:32:43
4.	28442	/csladmin/_mmServerScripts/	19:33:16	19:33:52	19:33:52
5.	11098	/csladmin/sourcecodemst/	19:34:16	19:34:24	19:34:24
6.	9125	/includes/	19:35:16	19:35:26	19:35:26
7.	26216	/_notes/	19:36:16	19:36:50	19:36:50
8.	26058	/_notes/csladmin/	19:37:16	19:37:45	19:37:45
9.	16917	/_mmServerScripts/	19:38:16	19:38:39	19:38:39
10.	6949	/_mmServerScripts/_notes/	19:39:16	19:39:35	19:39:35
Ortalama Süreler				~20 sn	~0 sn

Çizelge 5.33. Ticket uygulaması dizinlerinde yeni dizin ekleme test sonuçları

Sıra Nu.	Dizin Adı	Dizin Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	24705	/	17:45:52	17:45:59	17:45:59
2.	11208	/Connections/	17:46:37	17:46:53	17:46:53
3.	7436	/csladmin/Connections/	17:47:22	17:47:48	17:47:48
4.	11488	/csladmin/images/	17:48:07	17:48:15	17:48:15
5.	8191	/csladmin/_mmServerScripts /_notes/	17:48:52	17:49:11	17:49:11
6.	13823	/csladmin/sourcecodemst/ Connections/	17:49:37	17:49:39	17:49:39
7.	9160	/csladmin/sourcecodemst/ _mmServerScripts/	17:50:22	17:50:35	17:50:35
8.	16467	/csladmin/sourcecodemst/ _mmServerScripts/_notes/	17:51:07	17:51:31	17:51:31
9.	3281	/_notes/Connections/	17:51:52	17:52:00	17:52:00
10.	6252	/_notes/_notes/	17:52:37	17:52:57	17:52:57
Ortalama Süreler:				~14 Sn.	~0 Sn.



Şekil 5.3. Ticket uygulaması dosya ve dizin ekleme testi ortalama tespit ve düzeltilme süreleri

Redmine uygulaması için farklı boyutlardaki dosyalar için yapılan test sonuçları Çizelge 5.34, 5.35, 5.36 ve 5.37'de verilmiştir.

Çizelge 5.34. Redmine uygulamasına ait 0 – 500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	close.png	/images/	14:31:26	14:31:40	14:31:40
2.	bullet_toggle_plus.png	/images/	14:32:11	14:32:27	14:32:27
3.	bt_ol.png	/images/ jstoolbar/	14:32:56	14:33:15	14:33:15
4.	context_menu_rtl.css	/stylesheets/	14:33:41	14:34:01	14:34:01
5.	openid-bg.gif	/images/	14:34:26	14:34:43	14:34:43
6.	textfield_key.png	/images/	14:35:11	14:35:31	14:35:31
7.	bt_img.png	/images/ jstoolbar/	14:35:56	14:36:18	14:36:18
8.	datepicker.js	/javascripts/	14:36:41	14:37:05	14:37:05
9.	dispatch.fcgi	/	14:37:26	14:37:48	14:37:48
10.	php.png	/images/files/	14:38:11	14:38:29	14:38:29
Ortalama Süreler:				~19 Sn.	~0 Sn.

Çizelge 5.35. Redmine uygulamasına ait 501 – 1000 KB büyüklüğündeki dosyaların test sonuçları

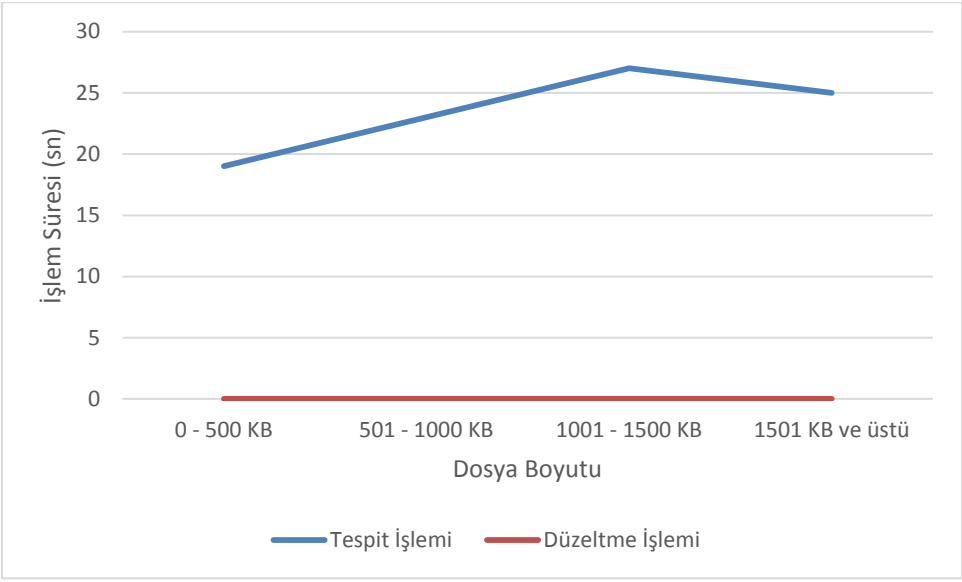
Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	js.png	/images/files/	14:38:56	14:39:15	14:39:15
2.	edit.png	/images/	14:39:41	14:39:59	14:39:59
3.	wiki_edit.png	/images/	14:40:26	14:40:47	14:40:47
4.	ticket_checked.png	/images/	14:41:11	14:41:32	14:41:32
5.	xml.png	/images/files/	14:41:56	14:42:13	14:42:13
6.	jstoolbar-bs.js	/javascripts/jstoolbar/lang/	14:42:41	14:43:07	14:43:07
7.	jstoolbar-hr.js	/javascripts/jstoolbar/lang/	14:43:26	14:43:51	14:43:51
8.	jstoolbar-no.js	/javascripts/jstoolbar/lang/	14:44:11	14:44:37	14:44:37
9.	jstoolbar-ro.js	/javascripts/jstoolbar/lang/	14:44:56	14:45:22	14:45:22
10.	jquery.ui.datepicker-ro.js	/javascripts/i18n/	14:45:41	14:46:09	14:46:09
Ortalama Süreler:				~23 Sn.	~0 Sn.

Çizelge 5.36. Redmine uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	draft.png	/images/	14:46:26	14:46:44	14:46:44
2.	jquery.ui. datepicker-sr.js	/javascripts/i18n/	14:47:11	14:47:40	14:47:40
3.	jquery.ui. datepicker-vi.js	/javascripts/i18n/	14:47:56	14:48:25	14:48:25
4.	jquery.ui. datepicker-ru.js	/javascripts/i18n/	14:48:41	14:49:09	14:49:09
5.	jquery.ui. datepicker-fa.js	/javascripts/i18n/	14:49:26	14:49:54	14:49:54
6.	jquery.ui. datepicker-uk.js	/javascripts/i18n/	14:50:11	14:50:40	14:50:40
7.	jquery.ui. datepicker-el.js	/javascripts/i18n/	14:50:56	14:51:27	14:51:27
8.	jquery.ui. datepicker-bg.js	/javascripts/i18n/	14:51:42	14:52:09	14:52:09
9.	jquery.ui. datepicker-th.js	/javascripts/i18n/	14:52:27	14:52:52	14:52:53
10.	jquery.ui. datepicker-ar.js	/javascripts/i18n/	14:53:12	14:53:40	14:53:40
Ortalama Süreler:				~27 Sn.	~0 Sn.

Çizelge 5.37. Redmine uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	wiki_syntax.html	/help/ar/	14:53:57	14:54:25	14:54:25
2.	wiki_syntax.html	/help/fr/	14:54:42	14:55:15	14:55:15
3.	textile.js	/javascripts/ jstoolbar/	14:55:27	14:55:50	14:55:50
4.	attachments.js	/javascripts/	14:56:12	14:56:36	14:56:36
5.	context_menu.js	/javascripts/	14:56:57	14:57:20	14:57:20
6.	gantt.js	/javascripts/	14:57:42	14:58:05	14:58:05
7.	jstoolbar.js	/javascripts/ jstoolbar/	14:58:27	14:58:50	14:58:50
8.	wiki_syntax_ detailed.html	/help/ar/	14:59:12	14:59:43	14:59:43
9.	application.css	/stylesheets/	14:59:57	15:00:19	15:00:19
10.	jquery-1.8.3-ui-1.9.2 -ujs-2.0.3.js	/javascripts/	15:00:42	15:01:05	15:01:05
Ortalama Süreler:				~25 Sn.	~0 Sn.



Şekil 5.4. Redmine uygulamasına ait farklı boyuttaki dosyaların ortalama tespit ve düzeltilme süreleri

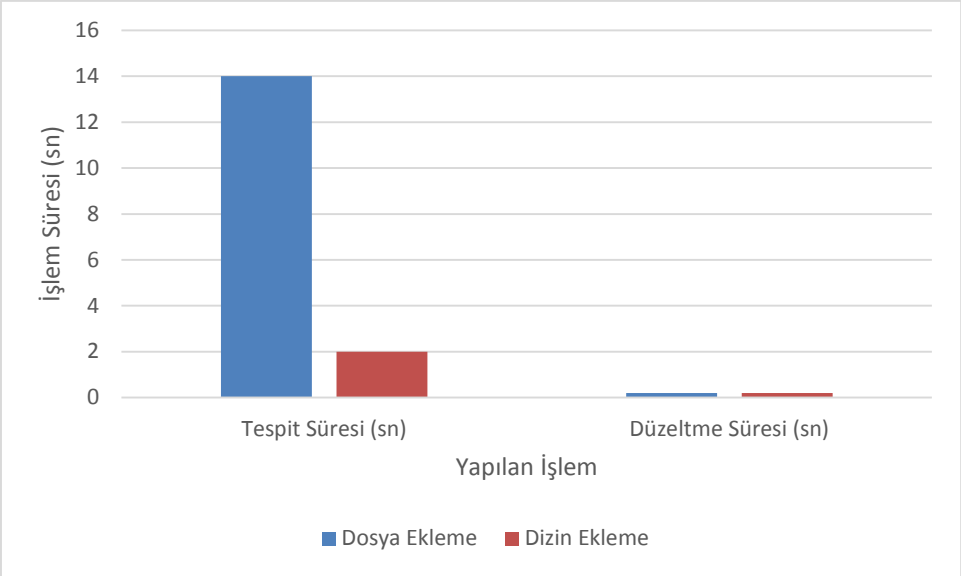
Redmine uygulaması için yapılan dosya ve dizin ekleme test sonuçları Çizelge 5.38 ve 5.39'da verilmiştir.

Çizelge 5.38. Redmine uygulaması dizinlerinde yeni dosya ekleme test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	14027	/	15:01:27	15:01:35	15:01:35
2.	31115	/images/jstoolbar/	15:02:12	15:02:23	15:02:23
3.	18092	/stylesheets/	15:02:57	15:03:12	15:03:12
4.	734	/stylesheets/jquery/	15:03:42	15:03:57	15:03:57
5.	21920	/themes/classic/	15:04:27	15:04:43	15:04:43
6.	15134	/themes/classic/images/	15:05:12	15:05:28	15:05:28
7.	2303	/themes/alternate/	15:05:57	15:06:14	15:06:14
8.	14757	/themes/alternate/stylesheets/	15:06:42	15:06:58	15:06:58
9.	4716	/javascripts/jstoolbar/	15:07:27	15:07:41	15:07:41
10.	21517	/help/	15:08:12	15:08:20	15:08:20
Ortalama Süreler:				~14 Sn.	~0 Sn.

Çizelge 5.39. Redmine uygulaması dizinlerinde yeni dizin ekleme test sonuçları

Sıra Nu.	Dizin Adı	Dizin Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	13967	/images/	15:08:57	15:08:59	15:08:59
2.	2742	/images/files/	15:09:42	15:09:44	15:09:44
3.	2403	/stylesheets/jquery/images/	15:10:27	15:10:28	15:10:28
4.	30611	/themes/classic/stylesheets/	15:11:12	15:11:14	15:11:14
5.	25648	/javascripts/	15:11:57	15:11:59	15:11:59
6.	29837	/javascripts/jstoolbar/lang/	15:12:42	15:12:44	15:12:44
7.	31457	/javascripts/i18n/	15:13:27	15:13:29	15:13:29
8.	6049	/help/sr-yu/	15:14:12	15:14:14	15:14:14
9.	25579	/help/az/	15:14:57	15:14:59	15:14:59
10.	5629	/help/mk/	15:15:42	15:15:44	15:15:44
Ortalama Süreler:				~2 Sn.	~0 Sn.



Şekil 5.5. Redmine uygulaması dosya ve dizin ekleme testi ortalama tespit ve düzeltilme süreleri

Drupal uygulaması için farklı boyutlardaki dosyalar için yapılan test sonuçları Çizelge 5.40, 5.41, 5.42 ve 5.43'te verilmiştir.

Çizelge 5.40. Drupal uygulamasına ait 0 – 500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	shortcut.admin.css	/modules/shortcut/	03:45:24	03:45:50	03:45:50
2.	menu-collapsed-rtl.gif	/themes/garland/images/	03:47:54	03:48:49	03:48:49
3.	drupal_autoload_test_interface.inc	/modules/simpletest/tests/drupal_autoload_test/	03:50:24	03:51:06	03:51:06
4.	lock.png	/modules/color/images/	03:52:54	03:53:49	03:53:49
5.	syslog.install	/modules/syslog/	03:55:24	03:56:03	03:56:03
6.	entity_cache_test_dependency.module	/modules/simpletest/tests/	03:57:54	03:58:29	03:58:29
7.	rdf.info	/modules/rdf/	04:00:24	04:01:03	04:01:03
8.	path.js	/modules/path/	04:02:54	04:02:59	04:02:59
9.	aggregator_test_title_entities.xml	/modules/aggregator/tests/	04:05:24	04:05:54	04:05:54
10.	README.txt	/themes/	04:07:54	04:08:50	04:08:50
Ortalama Süreler:				~38 Sn.	~0 Sn.

Çizelge 5.41. Drupal uygulamasına ait 501 – 1000 KB büyüklüğündeki dosyaların test sonuçları

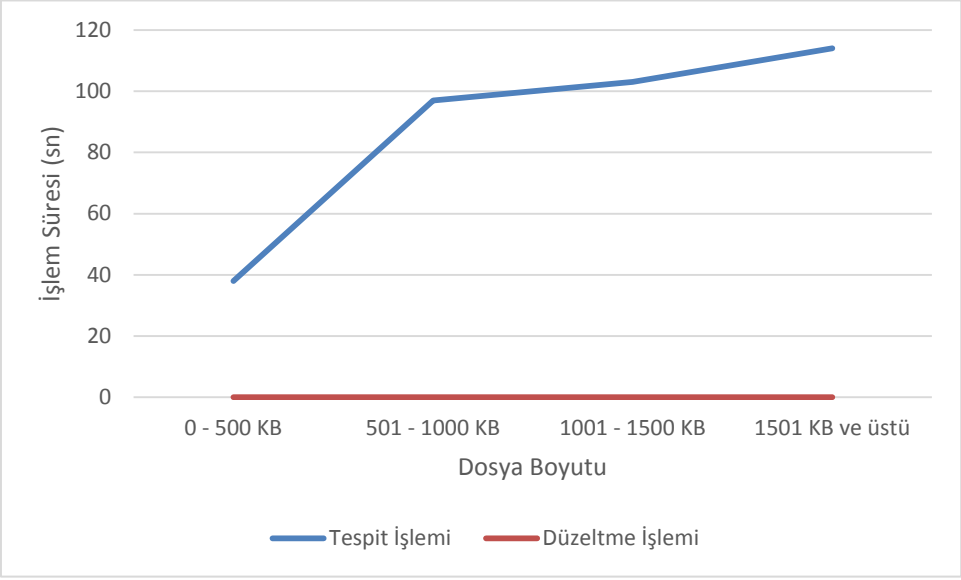
Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	shortcut.png	/modules/shortcut/	05:09:38	05:09:46	05:09:46
2.	print.css	/themes/bartik/css/	05:13:38	05:16:00	05:16:00
3.	taxonomy_test.install	/modules/simpletest/tests/	05:17:38	05:18:30	05:18:31
4.	comment_hacks.css.optimized.css	/modules/simpletest/files/css_test_files/	05:21:38	05:23:47	05:23:47
5.	jquery.effects.transfer.min.js	/misc/ui/	05:25:38	05:26:58	05:26:58
6.	user-profile-item.tpl.php	/modules/user/	05:29:38	05:32:28	05:32:28
7.	update.user.test	/modules/simpletest/tests/upgrade/	05:33:38	05:35:43	05:35:43
8.	powered-blue-80x15.png	/drupal/misc/	05:37:38	05:39:11	05:39:11
9.	authorize.js	/misc/	05:41:38	05:42:13	05:42:14
10.	node.tpl.php	/themes/garland/	05:45:38	05:47:53	05:47:53
Ortalama Süreler:				~97 Sn.	~0 Sn.

Çizelge 5.42. Drupal uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	README.txt	/themes/stark/	05:49:38	05:50:47	05:50:47
2.	node_access_ test.install	/modules/node/tests/	05:53:38	05:56:52	05:56:52
3.	aaa_update_ test.1_0.xml	/modules/update/ tests/	05:57:38	05:59:42	05:59:42
4.	update_script_ test.install	/ modules/simpletest/ tests/	06:01:38	06:03:08	06:03:08
5.	jquery.ui.dialog.css	/misc/ui/	06:05:38	06:06:35	06:06:35
6.	dblog.css	/modules/dblog/	06:09:38	06:12:27	06:12:27
7.	form_test.file.inc	/modules/simpletest/ tests/	06:13:38	06:15:56	06:15:56
8.	preview.js	/modules/color/	06:17:38	06:19:32	06:19:32
9.	system.admin-rtl .css	/modules/system/	06:21:38	06:22:22	06:22:22
10.	update.aggregator .test	/modules/simpletest/ tests/upgrade/	06:25:38	06:26:07	06:26:07
Ortalama Süreler:				~103 Sn.	~0 Sn.

Çizelge 5.43. Drupal uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	powered-blue-88x31.png	/misc/	06:29:38	06:29:51	06:29:51
2.	upgrade.forum.test	/modules/simpletest/tests/upgrade/	06:33:38	06:36:07	06:36:07
3.	aggregator_test_rss091.xml	/modules/aggregator/tests/	06:37:38	06:39:30	06:39:30
4.	favicon.ico	/misc/	06:41:38	06:43:01	06:43:01
5.	progress.gif	/misc/	06:45:38	06:46:11	06:46:11
6.	dashboard.js	/modules/dashboard/	06:49:38	06:52:53	06:52:53
7.	user.api.php	/modules/user/	06:53:38	06:56:35	06:56:35
8.	openid.inc	/modules/openid/	06:57:38	06:59:50	06:59:50
9.	CHANGELOG.txt	/	07:01:38	07:04:23	07:04:23
10.	system.install	/modules/system/	07:05:38	07:06:58	07:06:58
Ortalama Süreler:				~114 Sn.	~0 Sn.



Şekil 5.6. Drupal uygulamasına ait farklı boyuttaki dosyaların ortalama tespit ve düzeltilme süreleri

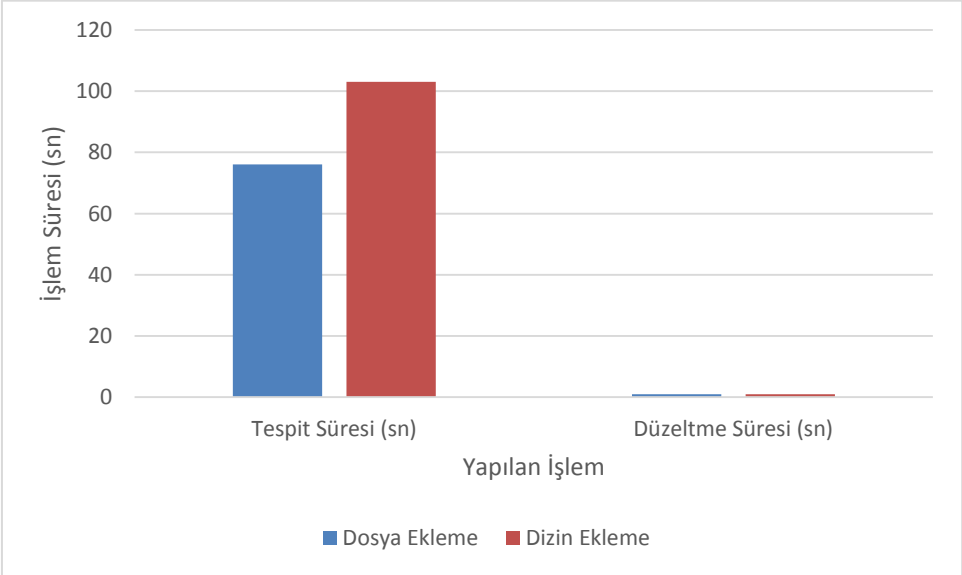
Drupal uygulaması için yapılan dosya ve dizin ekleme test sonuçları Çizelge 5.44 ve 5.45'te verilmiştir.

Çizelge 5.44. Drupal uygulaması dizinlerinde yeni dosya ekleme test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	24500	/scripts/	18:08:08	18:09:39	18:09:39
2.	7578	/modules/update/tests/themes/ update_test_basetheme/	18:12:08	18:13:53	18:13:53
3.	16434	/modules/field/modules/ field_sql_storage/	18:16:08	18:17:31	18:17:31
4.	30748	/modules/block/tests/themes/ block_test_theme/	18:20:08	18:21:57	18:21:57
5.	15106	/modules/simpletest/files/ css_test_files/css_subfolder/	18:24:08	18:26:30	18:26:30
6.	205	/modules/simpletest/tests/ themes/test_theme/templates/	18:28:08	18:30:27	18:30:27
7.	30446	/modules/simpletest/tests/ psr_4_test/src/Tests/Nested/	18:32:08	18:34:10	18:34:10
8.	24882	/modules/simpletest/tests/ _system_listing_incompatible_test/	18:36:08	18:38:34	18:38:34
9.	2548	/modules/simpletest/tests/ psr_0_test/lib//psr_0_test/	18:40:08	18:43:14	18:43:14
10.	13528	/profiles/testing/modules/ _system_listing_incompatible_test/	18:44:08	18:48:01	18:48:01
Ortalama Süreler:				~76 sn	~0 sn

Çizelge 5.45. Drupal uygulaması dizinlerinde yeni dizin ekleme test sonuçları

Sıra Nu.	Dizin Adı	Dizin Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	856	/	20:53:10	20:53:15	20:53:15
2.	3605	/modules/	20:58:10	21:00:57	21:00:57
3.	16400	/modules/shortcut/	21:03:10	21:04:56	21:04:56
4.	30403	/modules/update/tests/themes/	21:08:10	21:08:59	21:08:59
5.	16089	/modules/update/tests/themes/ update_test_subtheme/	21:13:10	21:16:51	21:16:51
6.	16562	/modules/simpletest/files/ css_test_files/css_subfolder/	21:18:10	21:21:11	21:21:11
7.	31874	/modules/simpletest/tests/ _system_listing_compatible_test/	21:23:10	21:25:25	21:25:25
8.	30290	/modules/simpletest/tests/ psr_0_test/lib/psr_0_test/Tests/	21:28:11	21:29:51	21:29:51
9.	11404	/modules/simpletest/tests/psr_0_test/ lib/psr_0_test/Tests/Nested/	21:33:11	21:34:12	21:34:12
10.	29180	/profiles/testing/modules/ _system_listing_compatible_test/	21:38:11	21:38:21	21:38:21
Ortalama Süreler:				~103 sn	~0 sn



Şekil 5.7. Drupal uygulaması dosya ve dizin ekleme testi ortalama tespit ve düzeltilme süreleri

WUBKU uygulaması için farklı boyutlardaki dosyalar için yapılan test sonuçları Çizelge 5.46, 5.47 ve 5.48'de verilmiştir.

Çizelge 5.46. WUBKU uygulamasına ait 501 – 1000 KB büyüklüğündeki dosyaların test sonuçları

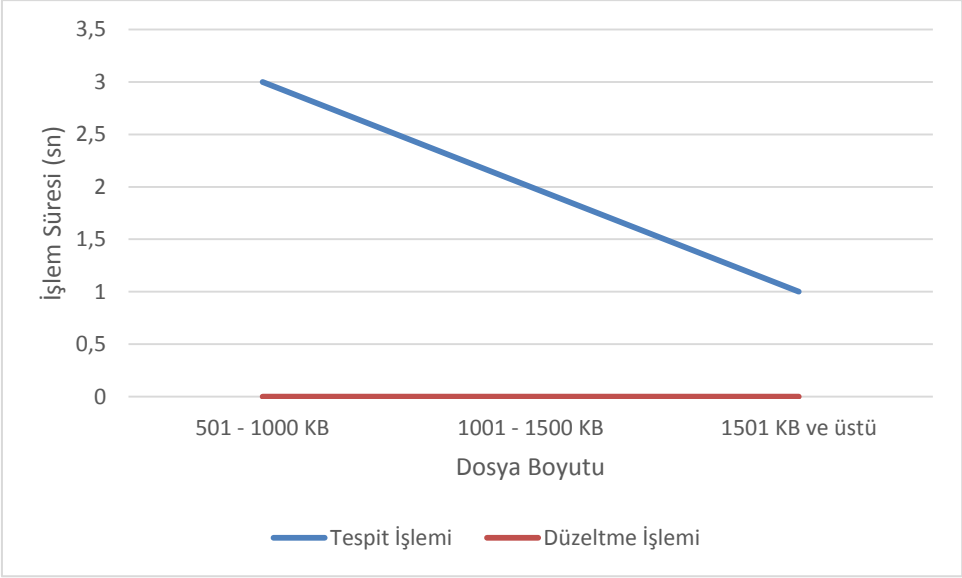
Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	signin.css	/css/	22:03:17	22:03:19	22:03:19
2.	index.php	/	22:04:17	22:04:18	22:04:18
3.	signin.css	/css/	22:05:17	22:05:19	22:05:19
4.	index.php	/	22:06:17	22:06:19	22:06:19
5.	signin.css	/css/	22:07:17	22:07:19	22:07:19
6.	index.php	/	22:08:17	22:08:19	22:08:19
7.	signin.css	/css/	22:09:17	22:09:19	22:09:19
8.	index.php	/	22:10:17	22:10:19	22:10:19
9.	signin.css	/css/	22:11:17	22:11:19	22:11:19
10.	index.php	/	22:12:17	22:12:19	22:12:19
Ortalama Süreler:				~2 Sn.	~0 Sn.

Çizelge 5.47. WUBKU uygulamasına ait 1001 – 1500 KB büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	vtBaglanti.php	/	22:13:17	22:13:20	22:13:20
2.	yazilimci.php	/	22:14:17	22:14:20	22:14:20
3.	yonetici.php	/	22:15:17	22:15:19	22:15:19
4.	oturumKontrol.php	/	22:16:17	22:16:20	22:16:20
5.	vtBaglanti.php	/	22:17:17	22:17:20	22:17:20
6.	yazilimci.php	/	22:18:17	22:18:20	22:18:20
7.	yonetici.php	/	22:19:17	22:19:19	22:19:19
8.	oturumKontrol.php	/	22:20:17	22:20:20	22:20:20
9.	vtBaglanti.php	/	22:21:17	22:21:20	22:21:20
10.	yazilimci.php	/	22:22:17	22:22:20	22:22:20
Ortalama Süreler:				~3 Sn.	~0 Sn.

Çizelge 5.48. WUBKU uygulamasına ait 1501 KB ve üstü büyüklüğündeki dosyaların test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	dosyaIndir.php	/	22:23:17	22:23:20	22:23:20
2.	baslik.php	/	22:24:17	22:24:20	22:24:20
3.	dosyalar.php	/	22:25:17	22:25:20	22:25:20
4.	guvenliWebTakip.php	/	22:26:18	22:26:20	22:26:20
5.	fonksiyonlar.php	/	22:27:18	22:27:20	22:27:20
6.	kullaniciEkle.php	/	22:28:18	22:28:19	22:28:19
7.	dosyaYukle.php	/	22:29:18	22:29:20	22:29:20
8.	smtp.php	/	22:30:18	22:30:20	22:30:20
9.	jquery.js	/js/	22:31:18	22:31:20	22:31:20
10.	bootstrap.css	/css/	22:32:18	22:32:19	22:32:19
Ortalama Süreler:				~2 Sn.	~0 Sn.



Şekil 5.8. WUBKU uygulamasına ait farklı boyuttaki dosyaların ortalama tespit ve düzeltilme süreleri

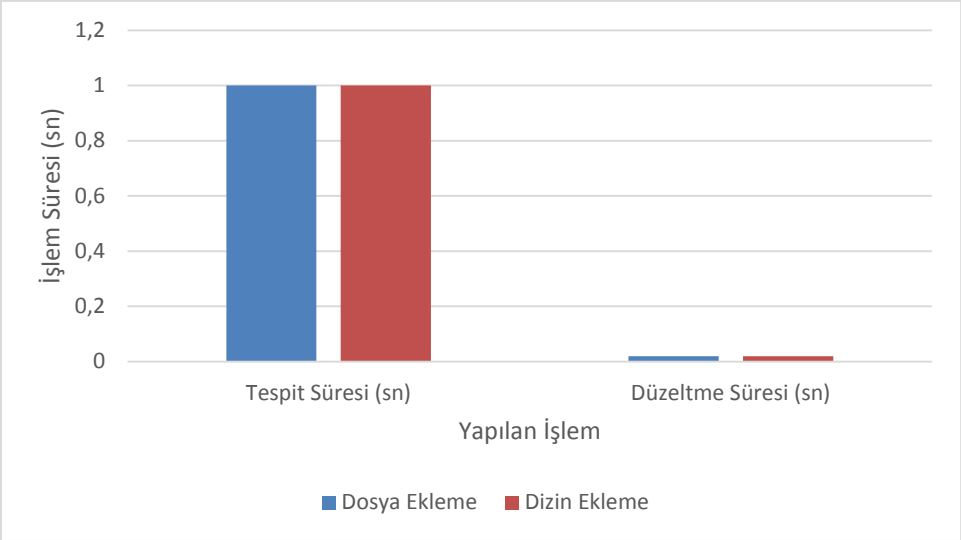
WUBKU uygulaması için yapılan dosya ve dizin ekleme test sonuçları Çizelge 5.49 ve 5.50'de verilmiştir.

Çizelge 5.49. WUBKU uygulaması dizinlerinde yeni dosya ekleme test sonuçları

Sıra Nu.	Dosya Adı	Dosya Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	29219	/	22:33:18	22:33:19	22:33:19
2.	4261	/css/	22:33:33	22:33:34	22:33:34
3.	22884	/js/	22:33:48	22:33:49	22:33:49
4.	5797	/	22:34:03	22:34:04	22:34:04
5.	30762	/css/	22:34:18	22:34:20	22:34:20
6.	17288	/js/	22:34:33	22:34:35	22:34:35
7.	15758	/	22:34:48	22:34:49	22:34:49
8.	864	/css/	22:35:03	22:35:05	22:35:05
9.	2872	/js/	22:35:18	22:35:20	22:35:20
10.	13067	/	22:35:33	22:35:34	22:35:34
Ortalama Süreler:				~1 Sn.	~0 Sn.

Çizelge 5.50. WUBKU uygulaması dizinlerinde yeni dizin ekleme test sonuçları

Sıra Nu.	Dizin Adı	Dizin Yolu	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	22254	/	22:35:48	22:35:49	22:35:49
2.	3984	/css/	22:36:03	22:36:04	22:36:04
3.	17247	/js/	22:36:18	22:36:19	22:36:19
4.	20077	/	22:36:33	22:36:34	22:36:34
5.	2467	/css/	22:36:48	22:36:49	22:36:49
6.	4414	/js/	22:37:03	22:37:04	22:37:04
7.	4070	/	22:37:18	22:37:19	22:37:19
8.	24766	/css/	22:37:33	22:37:34	22:37:34
9.	14347	/js/	22:37:48	22:37:49	22:37:49
10.	32199	/	22:38:03	22:38:04	22:38:04
Ortalama Süreler:				~1 Sn.	~0 Sn.



Şekil 5.9. WUBKU uygulaması dosya ve dizin ekleme testi ortalama tespit ve düzeltilme süreleri

WUBKU uygulama içerisindeki dosya ve dizinlerin bütünlük kontrolünü yanında, uygulamanın kök dizinin silinmesi ya da bütün uygulamaların bulunduğu kök dizinin silinmesi durumunda bile uygulamaların bütünlüğünü tekrardan sağlayarak, uygulamaların düzgün çalışmasını sağlamaktadır. Panel uygulaması kök dizinin silinmesi test sonuçları Çizelge 5.51'de verilmiştir.

Çizelge 5.51. Panel uygulamasının kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	11:53:41	11:53:41	11:53:43
2.	11:54:11	11:54:11	11:54:13
3.	11:54:41	11:54:41	11:54:43
4.	11:55:11	11:55:12	11:55:13
5.	11:55:41	11:55:41	11:55:43
6.	11:56:11	11:56:11	11:56:14
7.	11:56:41	11:56:41	11:56:43
8.	11:57:11	11:57:12	11:57:13
9.	11:57:41	11:57:41	11:57:43
10.	11:58:11	11:58:12	11:58:13
Ortalama Süreler:		~0 Sn.	~2 Sn.

Ticket uygulaması kök dizinin silinmesi test sonuçları Çizelge 5.52'de verilmiştir.

Çizelge 5.52. Ticket uygulamasının kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	13:56:06	13:56:07	13:56:52
2.	13:58:06	13:58:07	13:58:53
3.	14:00:06	14:00:08	14:00:54
4.	14:02:07	14:02:08	14:03:00
5.	14:04:07	14:04:08	14:04:55
6.	14:06:07	14:06:08	14:07:01
7.	14:08:07	14:08:08	14:09:02
8.	14:10:07	14:10:08	14:10:58
9.	14:12:07	14:12:08	14:13:04
10.	14:14:07	14:14:08	14:15:04
Ortalama Süreler:		~1 Sn.	~51 Sn.

Redmine uygulaması kök dizinin silinmesi test sonuçları Çizelge 5.53'te verilmiştir.

Çizelge 5.53. Redmine uygulamasının kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	15:16:28	15:16:29	15:17:30
2.	15:19:28	15:19:29	15:20:22
3.	15:22:28	15:22:29	15:23:23
4.	15:25:28	15:25:30	15:26:28
5.	15:28:28	15:28:29	15:29:29
6.	15:31:28	15:31:30	15:32:31
7.	15:34:28	15:34:30	15:35:28
8.	15:37:28	15:37:30	15:38:30
9.	15:40:28	15:40:30	15:41:35
10.	15:43:28	15:43:30	15:44:34
Ortalama Süreler:		~2 Sn.	~59 Sn.

Drupal uygulaması kök dizinin silinmesi test sonuçları Çizelge 5.54'te verilmiştir.

Çizelge 5.54. Drupal uygulamasının kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	18:12:33	18:12:35	18:15:42
2.	18:17:33	18:17:35	18:20:47
3.	18:22:34	18:22:36	18:25:33
4.	18:27:34	18:27:35	18:30:38
5.	18:32:34	18:32:36	18:35:42
6.	18:37:34	18:37:36	18:41:06
7.	18:42:34	18:42:36	18:45:58
8.	18:47:35	18:47:37	18:50:54
9.	18:52:35	18:52:36	18:55:58
10.	18:57:35	18:57:37	19:00:58
Ortalama Süreler:		~2 Sn.	~194 Sn.

WUBKU uygulaması kök dizin silinmesi test sonuçları Çizelge 5.55'te verilmiştir.

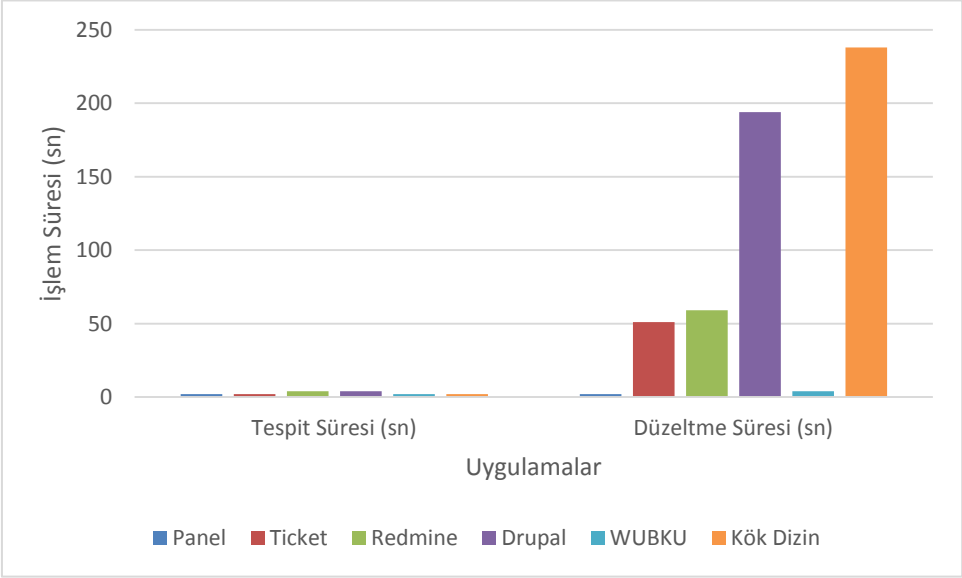
Çizelge 5.55. WUBKU uygulamasının kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	22:38:18	22:38:19	22:38:23
2.	22:41:18	22:41:19	22:41:22
3.	22:44:18	22:44:19	22:44:23
4.	22:47:18	22:47:19	22:47:23
5.	22:50:18	22:50:19	22:50:23
6.	22:53:18	22:53:19	22:53:23
7.	22:56:18	22:56:20	22:56:24
8.	22:59:18	22:59:19	22:59:23
9.	23:02:18	23:02:20	23:02:24
10.	23:05:18	23:05:19	23:05:23
Ortalama Süreler:		~1 Sn.	~4 Sn.

Bütün uygulamaların bulunduğu kök dizinin silinmesi test sonuçları Çizelge 5.56'da verilmiştir.

Çizelge 5.56. Bütün uygulamaların bulunduğu kök dizinin silinmesi test sonuçları

Sıra Nu.	Saldırı Zamanı	Tespit Zamanı	Düzeltilme Zamanı
1.	23:08:18	23:08:18	23:11:22
2.	23:18:19	23:18:19	23:21:55
3.	23:28:19	23:28:19	23:32:08
4.	23:38:19	23:38:19	23:42:22
5.	23:48:19	23:48:20	23:52:20
6.	23:58:20	23:58:20	00:02:21
7.	00:08:20	00:08:20	00:12:30
8.	00:18:20	00:18:20	00:22:42
9.	00:28:20	00:28:20	00:32:34
10.	00:38:21	00:38:21	00:42:43
Ortalama Süreler:		~0 Sn.	~238 Sn.



Şekil 5.10. Bütün uygulamaların kök dizininin silinmesi test sonuçları

5.1. Test Çalışmalarının Değerlendirmesi

WUBKU birbirinden farklı 5 web uygulaması ile test edilmiştir. Mevcut WUBKU sürümünün gereksinimlerinden dolayı web uygulamaları seçilirken, Linux işletim sistemi ve Apache web sunucu üzerinde çalışan uygulamalar tercih edilmiştir. Uygulamaların seçiminde dikkat edilen kriterler aşağıda maddeler halinde verilmiştir.

- Uygulama boyutu,
- Uygulama dosya sayısı,
- Uygulama dizin sayısı,

Test çalışmalarında seçilen bu kriterlerin, bütünlüğün bozulduğunun tespiti ve tekrardan sağlanmasındaki etkisi araştırılmıştır. Elde edilen bulgulardan,

- Uygulama dosya ve dizin sayısının bütünlüğün bozulmasının tespit süresini doğru orantıda etkilediği,
- Uygulama dosya ve dizin sayısının, uygulamanın kök dizininin silinmesi durumunda, uygulamanın düzeltime süresini doğru orantıda etkilediği,
- Uygulamanın toplam boyutu ile bütünlüğü bozulan dosya boyutunun düzeltilme süresini önemsenecek derecede etkilemediği sonuçları elde edilmiştir.

Aynı web sunucusu üzerinde birden fazla web uygulaması çalıştırılabilmektedir. Web sunucuların bu özellikleri göz önünde bulundurularak, WUBKU'nun mevcut sürümüne birden fazla web uygulaması için bütünlük kontrolü sağlama özelliği kazandırılmıştır. Test sonuçlarında, birbirinden farklı 5 web uygulamasının testi aynı web sunucu üzerinde yapılarak, bu özellik test edilmiştir. Yapılan testlerde, aynı web sunucusu üzerinde çalışan birden fazla web uygulaması için düzgün ve efektif olarak çalıştığı sonucu elde edilmiştir.

Yapılan test çalışmalarında, WUBKU'nun aşağıdaki işlemlerin yetkisiz olarak yapılması durumunda, bütünlüğün bozulduğunun tespitini yapabildiği ve otomatik olarak düzeltebildiği sonucu elde edilmiştir.

- Uygulama üzerine yeni bir dosya ekleme,
- Uyguma üzerinde var olan bir dosyayı silme,

- Uygulama üzerinde var olan bir dosyayı değiştirme,
- Uygulama üzerinde yeni bir dizin oluşturma,
- Uygulama dosyalarının bulunduğu dizini silme,
- Bütün uygulamaların bulunduğu dizini silme,

Apache web sunucu üzerinde PHP haricinde yazılmış uygulamalarda çalışabilmektedir. WUBKU'nun mevcut sürümünün PHP harici uygulamalar içinde bütünlük kontrolü yapabildiği ve bütünlüğün tekrardan sağlanabildiğinin testi -Ruby programlama dilinde yazılmış olan Tren Rayları çatısı kullanılarak geliştirilen- Redmine uygulaması üzerinden gerçekleştirilmiştir. Yapılan testlerde, WUBKU farklı programlama dili kullanılarak geliştirilen uygulamalar içinde düzgün ve efektif çalıştığı görülmektedir.

Web uygulamaları farklı uzantıdaki dosyaları kullanabilmektedir. Web uygulamaların bu özelliği göz önünde bulundurularak, WUBKU farklı dosya türleri içinde bütünlük kontrolü yapabilecek ve bütünlüğü tekrardan sağlayabilecek şekilde geliştirilmiştir. Test çalışmalarında kullanılan dosyaların farklı uzantılarda seçilmesiyle, farklı uzantıdaki dosyalar içinde bütünlük kontrolü yapılabildiği ve bütünlüğün tekrardan sağlanabildiğinin testi gerçekleştirilmiştir. Yapılan testlerde, WUBKU farklı dosya türleri içinde de düzgün ve efektif çalıştığı görülmektedir.

Uygulamalar üzerinde yapılan yetkisiz işlemlerin tespiti ve bütünlüğün tekrardan sağlanmasının önemli olmasının yanında uygulamaların üzerinde gerçekleşen yetkili işlemlerin düzgün takibi de önemlidir. Herhangi bir durumda geriye dönük uygulama üzerinde gerçekleşen yetkili işlemlerin takibinde, WUBKU'nun mevcut sürümünde yapılan yetkili işlemler ile ilgili aşağıdaki bilgiler kayıt altına alınmaktadır. WUBKU'nun yapılan işlemleri kayıt altına aldığının testi yapılmış ve aşağıdaki bilgileri kayıt altına aldığı görülmüştür.

- İşlemin kimin tarafından yapıldığı,
- İşlemin yapıldığı zaman bilgisi,
- Hangi uygulama üzerinde işlem yapıldığı,
- Hangi uygulama üzerinde ne tür bir işlem yapıldığı,

Yapılan test çalışmasında, WUBKU'nun bütünlüğü bozucu saldırıları FP uyarı üretmeden %100 oranında tespit ettiği ve bütünlüğü tekrardan sağlayabildiği görülmüştür. Ayrıca, tespit ettiği durumlar ve yaptığı işlemler ile ilgili hatasız ve eksiksiz şekilde kayıt tuttuğu ve uyarı ürettiği görülmüştür.

6. SONUÇLAR

Bu tez çalışmasında, web uygulamalarının güvenlik durumu, web uygulama güvenliği açıklıkları ve bu açıklıkların istismar edilmesine karşı geliştirilen güvenlik çözümleri incelenmiştir. İnceleme sonucunda elde edilen bilgiler doğrultusunda, bütünlük kontrolü üzerinden çalışan bir sistem önerilmiştir. Önerilen sistemin gerçekleştirilmesiyle, web uygulamalarının bütünlüğünü kontrol eden, bütünlüğü bozulduğunda uygulamanın bütünlüğünü otomatik olarak tekrardan sağlayan ve yaptığı tespit ve işlemlerle ilgili uyarı üreten bir web uygulama güvenlik çözümü sunulmuştur. Literatürdeki çalışmalar değerlendirildiğinde,

1. Web uygulamalarında bütünlük kontrolü üzerinden güvenlik çözümü bulunmadığı,
2. Web uygulama güvenliği konusundaki çalışmaların daha çok uygulama kodu üzerindeki kusurların tespiti ve uygulamaların güvenlik testleri üzerine yoğunlaştığı,
3. Çözümlerin çok sayıda ve çok çeşitli olması ve her çözümün uygulamasının olmadığı,
4. Web uygulama güvenliği konusunda birçok çözüm önerisi bulunmasına rağmen, web uygulama güvenliğinin sağlanmasının zor bir süreç olduğu anlaşılmıştır.

Yapılan inceleme, araştırma ve analizler doğrultusunda yapılan çalışma değerlendirildiğinde,

1. Web uygulama güvenliğinde, uygulama kodu güvenliği, uygulamaların güvenlik testlerinin yapılması ve yazılım yaşam döngüsüne güvenliğin bütünleştirilmesi gibi çok çeşitli çözümlerin olduğu,
2. Web uygulama güvenliği çözümlerinin güvenli web uygulaması geliştirme ve var olan uygulamaların güvenlik testleri üzerine yoğunlaştığı,
3. Web uygulama güvenliği konusunda yapılan akademik çalışmalar incelendiğinde, web uygulamasına yapılan bir saldırı sonucu bozulan bütünlüğün otomatik olarak tespitine ve düzeltilmesine yönelik bir çalışma olmadığı ve bütünlük kontrolü üzerinden yapılan bu çalışmanın bir ilk olduğu,
4. Web uygulama güvenliği konusunda tek bir çözüm ile güvenliğin tam olarak sağlanamayacağı anlaşılmıştır.

Bu tez çalışması kapsamında, belirtilen problemlere çözüm olarak önerilen WUBKU uygulamasında, bütünlüğü bozucu saldırıların tespit başarısı %100'dür. Bununla beraber bütünlüğü bozulan uygulamanın bütünlüğünün bozulduğunun tespiti ve otomatik düzeltme işlemini efektif olarak yaptığı görülmüştür. Yapılan testlerde bütünlüğü bozulan uygulamanın tespiti ve düzeltilmesindeki en iyi zaman değeri ~0 sn iken en uzun tespit süresinin ~114 sn ve düzeltilme süresinin ise kök dizinin silinmesi durumundaki ~238 sn olduğu görülmüştür. Yapılan testler sonucunda bütünlüğü bozulan uygulamanın tespit edilmesinde çeşitli etkenlerin olduğu görülmüştür. Bu etkenlerin koruma sağlanan,

- Uygulamanın boyutu,
- Uygulamanın dizin sayısı,
- Uygulamanın dosya sayısı
- Bütünlüğü bozulan dosyanın veritabanın da sorgulama zamanına bağlı olduğu görülmüştür.

Bir web uygulamasında, web uygulamasının üzerinde çalıştığı sunucu ya da sunucu ile iletim katmanı üzerindeki bir açıklığın istismar edilmesi ya da sunucu yönetimi üzerinde yetkili bir kişinin yetkisini kötüye kullanarak yetkisiz işlemler gerçekleştirebilmektedir. Bu gibi uygulama bütünlüğünü bozmaya yönelik yetkisiz işlemler, WUBKU tarafından saldırı olarak kabul etmektir. WUBKU, yapılan saldırı sonucu uygulamanın bütünlüğünün bozulduğunun tespitini yapabilmekte ve bozulan bütünlüğü otomatik olarak tekrardan sağlayabilmektedir.

WUBKU resmi web sitesine http://www.wubku.com adresi üzerinden erişilebilmektedir. Resmi web sitesinde, WUBKU'nun kurulum ve kullanım dökümanları ile WUBKU hakkındaki görüşlerin ifade edilebildiği herkese açık iletişim alanı bulunmaktadır.

WUBKU'ya ilerleyen zamanlarda ilk olarak kazandırılması planlanan özellikler aşağıdaki maddelerde belirtilmiştir.

- Tarama periyodunun yönetici tarafından belirlenebilmesi,
- İstenen dosya ve dizinler için sadece bütünlük kontrolü,
- İstenen dizinlerin bütünlük kontrolünden hariç tutulması,

- Web dosyalarının web uygulamalarından bağımsız bir dizin üzerinden çalıştırılması,
- PARDUS'a adapte edilebilecek bir çalışmaya dönüştürülmesi,

Önerilen güvenlik sisteminin gerçekleştirilmesinde bazı problemlerle karşılaşılmıştır. Bu problemler aşağıdaki maddelerde belirtilmiştir.

- Bütünlük kontrolünün anormallik tabanlı mı yoksa imza tabanlı mı olacağı,
- Bütünlüğü doğrulanmış dosya ve dizin bilgilerinin nerede saklanacağı,
- Uygulama üzerindeki yapılan işlemlerin yetkili kişi tarafından yapılıp yapılmadığının nasıl belirleneceği,
- Uygulamalara ait imzaların nasıl oluşturulacağı,

WUBKU mevcut sürümü düzgün şekilde çalışmakla beraber farklı durumlarda uygulanmanın çalışmasında bir takım problemlerle karşılaşılabilir. Olası problemler aşağıdaki maddelerde belirtilmiştir.

- Dosya boyutu çok büyük uygulamalar için sunucu üzerinde disk alanı problemi oluşturma olasılığı,
- Veritabanında saklanan uygulama bilgilerinin yetkisiz olarak değiştirilmesi,
- Yapılandırma işlemlerindeki yanlış ya da eksik işlem sonucunda uygulamaların silinmesi olasılığı ve koruma sağlayan WUBKU'nun kendisinin silinmesi durumu,

Bu tez kapsamında yapılan çalışmanın web uygulamalarındaki güvenlik durumu, web uygulamaları üzerinde bulunan açıklıkların neler olduğunu, bu açıklıkların istismar edilmesinin nasıl sonuçlar doğuracağını ortaya koymasıyla web uygulama güvenliği konusunda farkındalığın artmasına katkı sağlayacağı değerlendirilmektedir. Bunun yanında, yapılan çalışmanın web uygulama güvenliğine özgün bir sistemle katkı sağlaması, önerilen sistemin GPL sürüm 3 ile lisanslanarak herkesin kullanımına sunulması ve WUBKU'nun http://www.wubku.com adresinden indirilip kullanılabilmesinin önemli bir katkı sağlayacağı değerlendirilmektedir. Elde edilen bilgiler ışığında önerilen güvenlik çözümü ve bu çözümün WUBKU olarak gerçekleştirilmesinin varlığı hem ülkemizde böyle bir projenin olması açısından hem de web uygulama güvenliğine katkı sağlayacağının değerlendirilmesinden önem arz ettiği değerlendirilmektedir. Ayrıca,

literatür çalışmalarından farklı olarak, saldırının verdiği zararı tespit edip otomatik olarak düzeltmesiyle güvenlik çözümlerine farklı bir bakış açısı kazandıracağı değerlendirilmektedir.

KAYNAKLAR

1. Teodoro, N., Serrão, C. (2011). Web application security: Improving critical web-based applications quality through in-depth security analysis. *In Information Society (i-Society), 2011 International Conference on IEEE*, London, 457-462.

2. Wang, X., Wang, L., Wei, G., Zhang, D., Yang, Y. (2010). Hidden web crawling for SQL injection detection. *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on IEEE*, Beijing, 14-18.

3. Fonseca, J., Vieira, M., Madeira, H. (2014). Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection. *IEEE Transactions on, Dependable and Secure Computing*, 11(5), 440-453.

4. İnternet: Sutherland, B. Web Application Security: The Overlooked Vulnerabilities. Third Brigade, Inc. URL: http://www.webcitation.org/query?url=http%3A%2F%2Frepo.hackerzvoice.net%2Fdepot_cehv6%2FCEHv6%2520Module%252017%2520Web%2520Application%2520Vulnerabilities%2FWeb_Application_Security_TBrigade.pdf&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

5. Atashzar, H., Torkaman, A., Bahrololum, M., Tadayon, M. H. (2011). A survey on web application vulnerabilities and countermeasures. *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on IEEE*, Seogwipo, 647-652.

6. Munir, R. F., Ahmed, N., Razzaq, A., Hur, A., Ahmad, F. (2011). Detect HTTP Specification Attacks Using Ontology. *In Frontiers of Information Technology (FIT) IEEE*, Islamabad, 75-78.

7. Scholte, T., Balzarotti, D., Kirda, E. (2012). Have things changed now? An empirical study on input validation vulnerabilities in web applications. *Computers & Security*, 31(3), 344-356.

8. Xie, J., Chu, B., Lipford, H. R., Melton, J. T. (2011). ASIDE: IDE support for web application security. *Proceedings of the 27th Annual Computer Security Applications Conference, ACM*, 267-276.

9. Priya, R. L., Lifna, C. S., Jagli, D., Joy, A. (2014). Rational Unified Treatment for Web application Vulnerability Assessment. *Circuits, Systems, International Conference on, Communication and Information Technology Applications (CSCITA), 2014, IEEE*, Mumbai, 336-340.

10. Fonseca, J., Seixas, N., Vieira, M., Madeira, H. (2014). Analysis of field data on web security vulnerabilities. *IEEE Transactions on, Dependable and Secure Computing*, 11(2), 89-100.

11. Scholte, T., Robertson, W., Balzarotti, D., Kirda, E. (2012). Preventing input validation vulnerabilities in web applications through automated type analysis. *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual IEEE*, Izmir, 233-243.

12. Monga, M., Paleari, R., Passerini, E. (2009). A hybrid analysis framework for detecting web application vulnerabilities. *Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems, IEEE Computer Society*, 25-32.

13. İnternet: WASC Threat Classification Version 2.00. Web Application Security Consortium. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fprojects.webappsec.org%2Ff%2FWASC-TC-v2_0.pdf&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

14. Møller, A., Schwarz, M. (2014). Automated detection of client-state manipulation vulnerabilities. *ACM Transactions on Software Engineering and Methodology (TOSEM), 23*(4), 29.

15. İnternet: Application Vulnerability Trends Report : 2014. CENZIC, Inc. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.trustwave.com%2FResources%2FLibrary%2FDocuments%2FCenzic-Application-Vulnerability-Trends-2014%2F&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

16. İnternet: 2014 Vulnerability Statistics Report. Edgescan. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.bccriskadvisory.com%2Fwp-content%2Fuploads%2FEdgescan-Stats-Report.pdf&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

17. İnternet: 2014 Website Security Statistics Report. WhiteHat Security, Inc. URL: http://www.webcitation.org/query?url=http%3A%2F%2Finfo.whitehatsec.com%2Frs%2Fwhitehatsecurity%2Fimages%2Fstatsreport2014-20140410.pdf&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

18. İnternet: 2014 Global Report on the Cost of Cyber Crime. Ponemon Institute. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fh20195.www2.hp.com%2Fv2%2Fgetpdf.aspx%2F4AA5-5207ENW.pdf%3Fver%3D1.0&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

19. İnternet: Top 10 2013 – Top 10. Open Web Application Security Project. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.owasp.org%2Findex.php%2FTop_10_2013-Top_10&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

20. İnternet: Web Application Security Consortium. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.webappsec.org%2F&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

21. Johari, R., Sharma, P. (2012). A survey on web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection. *Communication Systems and Network Technologies (CSNT), 2012 International Conference on IEEE*, Rajkot, 453-458.

22. Halfond, W. G., Orso, A., Manolios, P. (2008). WASP: Protecting Web applications using positive tainting and syntax-aware evaluation. *IEEE Transactions on, Software Engineering*, 34(1), 65-81.

23. Shar, L. K., Tan, H. B. K. (2012). Predicting common web application vulnerabilities from input validation and sanitization code patterns. *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on IEEE*, Essen, 310-313.

24. Takamatsu, Y., Kosuga, Y., Kono, K. (2012). Automated detection of session management vulnerabilities in web applications. *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on IEEE*, Paris, 112-119.

25. Zhang, L., Gu, Q., Peng, S., Chen, X., Zhao, H., Chen, D. (2010). D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms. *ICSEA*, Nice, 501-507.

26. Yu, F., Tung, Y. Y. (2014). Patcher: An Online Service for Detecting, Viewing and Patching Web Application Vulnerabilities. *System Sciences (HICSS), 2014 47th Hawaii International Conference on IEEE*, Waikoloa, HI, 4878-4886.

27. Qianqian, W., Xiangjun, L. (2014). Research and design on Web application vulnerability scanning service. *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on IEEE*, Beijing, 671-674.

28. Eshete, B., Villafiorita, A., Weldemariam, K., Zulkernine, M. (2013). Confeagle: Automated Analysis of Configuration Vulnerabilities in Web Applications. *Software Security and Reliability (SERE), 2013 IEEE 7th International Conference on IEEE*, Gaithersburg, MD, 188-197.

29. Razzaq, A., Hur, A., Haider, N., Ahmad, F. (2009). Multi-layered defense against web application attacks. *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on IEEE*, Las Vegas, NV, 492-497.

30. Teodoro, N., Serrao, C. (2011). Web application security: Improving critical web-based applications quality through in-depth security analysis. *Information Society (i-Society), 2011 International Conference on IEEE*, London, 457-462.

31. Razzaq, A., Hur, A., Shahbaz, S., Masood, M., Ahmad, H. F. (2013). Critical analysis on web application firewall solutions. *Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on IEEE*, Mexico City, Mexico, 1-6.

32. Li, X., Xue, Y. (2014). A survey on server-side approaches to securing web applications. *ACM Computing Surveys (CSUR)*, 46(4), 54.

33. Ghosh, R., Verma, S., Kumar, R., Kumar, S., Ram, S. (2015). Design of Hash Algorithm Using Latin Square. *Procedia Computer Science*, 46, 759-765.

34. Rao, P. V., Reddy, K. T., Varma, P. S. (2015). ASH-512: Design and implementation of cryptographic hash algorithm using co-ordinate geometry concepts. *Journal of Information Security and Applications*, 20, 47-60.

35. Federal Information Processing Standards Publication. (2012) *Secure hash standard (SHS), FIPS PUB*, 180-4.

36. Wang, X., Lai, X., Feng, D., Chen, H., Yu, X. (2005). Cryptanalysis of the Hash Functions MD4 and RIPEMD. *Advances in Cryptology–EUROCRYPT 2005*, Springer Berlin Heidelberg, 1-18.

37. Wang, X., Feng, D., Lai, X., Yu, H. (2004). *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, *IACR Cryptology ePrint Archive*, 2004, 199.

38. Wang, X., Yu, H., Yin, Y. L. (2005). Efficient collision search attacks on SHA-0. *In Advances in Cryptology–CRYPTO 2005*, Springer Berlin Heidelberg, 1-16.

39. Wang, X., Yin, Y. L., Yu, H. (2005). Finding collisions in the full SHA-1. *Advances in Cryptology–CRYPTO 2005*, Springer Berlin Heidelberg, 17-36.

40. Knudsen, L. R., Mathiassen, J. E. (2005). Preimage and collision attacks on MD2. *Fast Software Encryption*, Springer Berlin Heidelberg, 255-267.

41. Gueron, S., Johnson, S., Walker, J. (2011). SHA-512/256. *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on IEEE*, Las Vegas, NV, 354-358.

42. İnternet: Overview – Redmine. Redmine. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.redmine.org%2F&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

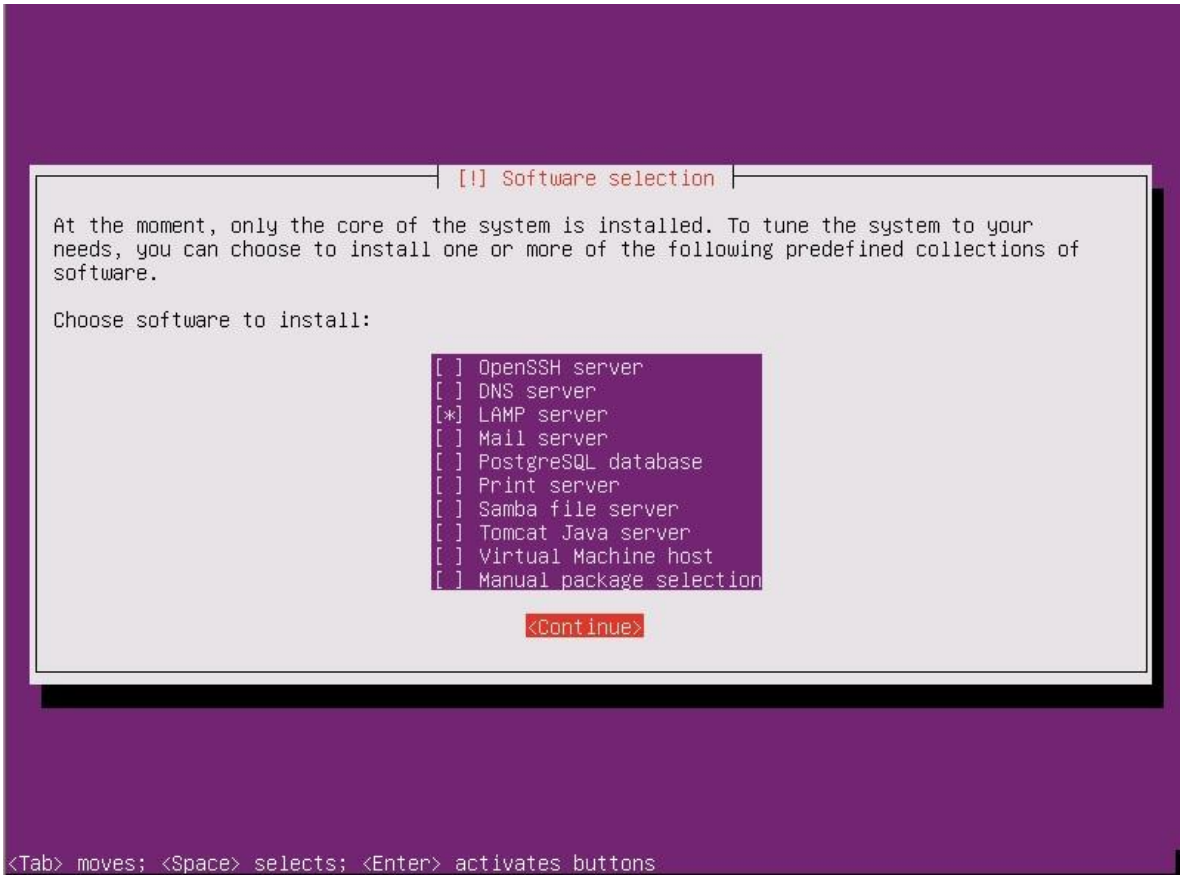
43. İnternet: Drupal – Open Source CMS | Drupal.org. Drupal. URL: www.webcitation.org/query?url=https%3A%2F%2Fwww.drupal.org%2F&date=2015-07-23 Son Erişim Tarihi: 23-07-2015

EKLER

EK-1. WUBKU kurulum işlemleri

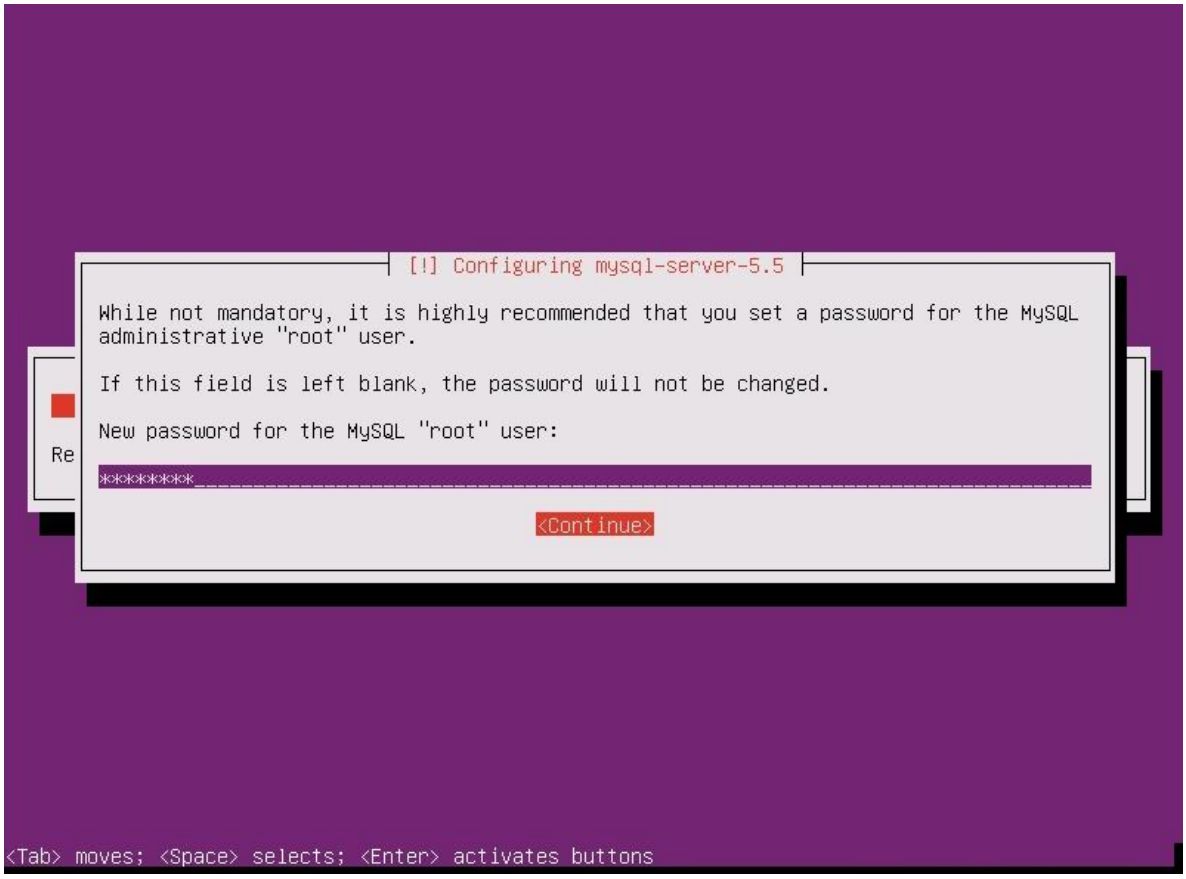
1. Gereksinimlerin Kurulumu

WUBKU'nun çalışabilmesi için Bash uygulaması, PHP programlama dili, MySQL veritabanı ve Apache web sunucu kurulumu gerekmektedir. Linux işletim sisteminde üç gereksinim LAMP (Linux Apache MySQL PHP) server olarak geçmektedir. LAMP Server işletim sistemi kurulması esnasında "Software selection" aşamasında gelen seçenekler arasından seçilerek kurulumu kolay şekilde yapılabilmektedir. LAMP server kurulum ekran görüntüleri Şekil 1.1, 1.2 ve 1.3'te gösterilmektedir.



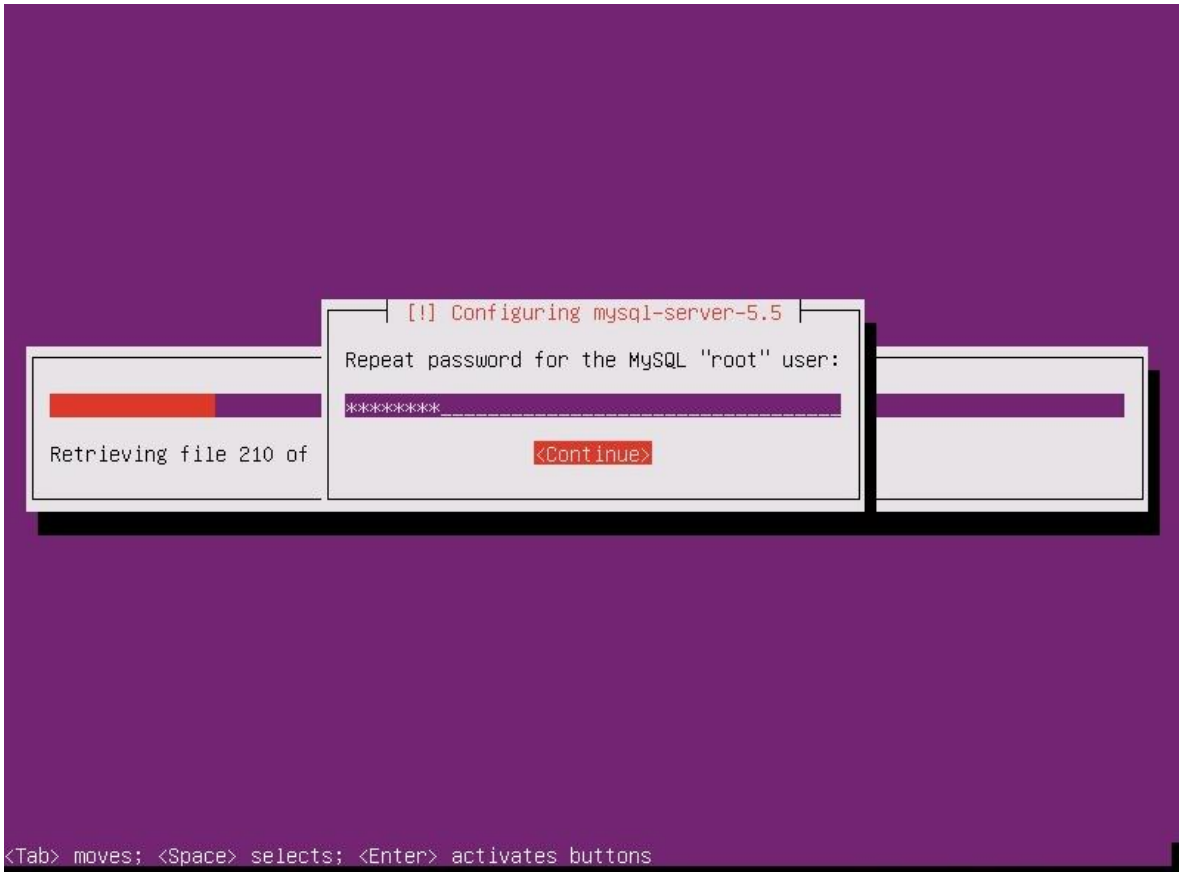
Şekil 1.1. LAMP Server kurulumu için LAMP Server seçeneğinin seçilmesi

EK-1.(devam) WUBKU kurulum işlemleri



Şekil 1.2. MySQL veritabanı için yönetici şifresinin belirlenmesi

EK-1.(devam) WUBKU kurulum işlemleri



Şekil 1.3. MySQL veritabanı için yönetici şifresinin doğrulanması

Ubuntu Server işletim sistemi üzerine gerekli gereksinimler, işletim sistemi kurulumu esnasında kurulabildiği gibi, kurulu işletim sistemi üzerine aşağıdaki komut kullanılarak da kurulabilmektedir.

```
sudo apt-get install lamp-server^
```

EK-1.(devam) WUBKU kurulum işlemleri

2. WUBKU Arayüz ve Betiklerin Kurulumu

Gereksinimlerin kurulmasının ardından, internet üzerinden erişilebilir olan WUBKU resmi sitesinin (http://www.wubku.com) İndir bağlantısı kullanılarak WUBKU kurulum dosyalarının indirilmesi gerekmektedir. Kurulum dosyaları sıkıştırılmış klasörden çıkartıldıktan sonra, kurulum dosyaları içerisindeki betikler klasörünün içeriği /opt/wubku/bin klasörün altına taşınmalıdır. Bunun işlemin yapılması için gerekli komutlar aşağıda verilmiştir.

```
sudo mkdir -p /opt/wubku/bin
sudo mv wubku/betikler/* /opt/wubku/bin
chmod +x /opt/wubku/bin/*.sh
```

Kurulum dosyaları içerisindeki web klasörünün içeriği /var/www/ klasörün altına taşınmalıdır. Bunun işlemin yapılması için gerekli komutlar aşağıda verilmiştir.

```
sudo mkdir -p /var/www/html/wubku
sudo mv wubku/web/* /var/www/html/wubku
```

EK-1.(devam) WUBKU kurulum işlemleri

3. Veritabanı Yapılandırması

Kurulum dosyaları arasında bulunan wubku.sql dosyası, arayüz tarafından kullanılacak olan temel yapılandırma bilgilerini içermektedir. Ayrıca, WUBKU tarafından koruma sağlanacak web uygulamalarına ait bilgiler ve tespit edilen işlemler ile yapılan yetkili işlemlerin saklanmasında kullanılmaktadır. Veritabanının oluşturulması için gerekli komutlar aşağıda verilmiştir. Komutlarda kalın olarak yazılı ifadeler değişkenlerdir ve isteğe bağlı olarak değiştirilebilmektedir.

```
echo "create database veritabaniAdi;" | mysql -u root -p

mysql -u root -p -D veritabaniAdi < wubku/wubku.sql

echo "grant create, insert, select, delete, update on veritabaniAdi.* to veritabaniKullanicisi@localhost identified by 'veritabaniSifresi'" | mysql -u root -p
```

EK-1.(devam) WUBKU kurulum işlemleri

4. WUBKU Arayüz ve Betiklerin Yapılandırması

WUBKU arayüzünün veritabanı bağlantısının sağlanabilmesi için, arayüz dosyalarının bulunduğu dizin içerisindeki vtBaglanti.php dosyasında bulunan $kullaniciAdi, $sifre ve $veritabaniAdi değişkenleri, bir önceki adımda veritabanı yapılandırmasında kullanılan bilgiler ile aynı olduğuna dikkat edilmelidir.

```
$sunucu          ="localhost";
$kullaniciAdi    ="wubkuVeritabaniKullanicisi";
$sifre           ="wubkuVeritabaniSifresi";
$veritabaniAdi   ="wubkuVeritabani";
```

WUBKU tarafından kullanılan betiklerin veritabanına erişim sağlayabilmeleri için, betiklerin bulunduğu dizin içerisindeki vtBaglanti.txt dosyasında bulunan vtSifresi, vtKullaniciAdi, vtAdi değişkenlerinin bir önceki adımda veritabanı yapılandırmasında kullanılan bilgiler ile aynı olduğuna dikkat edilmelidir.

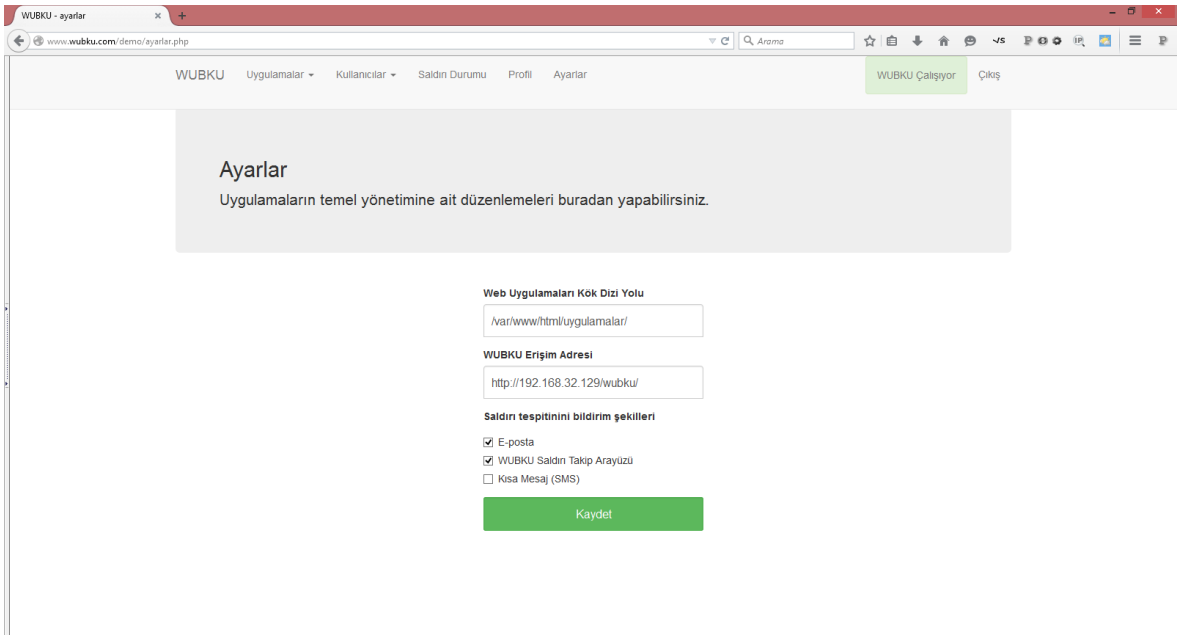
```
vtSifresi=veritabaniSifresi

vtKullaniciAdi=veritabaniKullanicisi

vtAdi=veritabaniAdi
```

Herhangi bir dosyanın bütünlüğünün bozulması durumunda, bütünlüğü doğrulanmış dosyanın WUBKU üzerinden indirilebilmesi için arayüz dosyalarının bulunduğu dizin içerisindeki dosyaIndir.php dosyasında bulunan $baglantiIPAdresi değişkeni WUBKU'nun web arayüzüne erişim IP adresi ile aynı olmalıdır. Ayrıca, WUBKU'nun çalıştığı web sunucundan, tanımlanan IP adresi kullanılarak WUBKU arayüzüne erişim sağlanabildiğine dikkat edilmelidir.

EK-1.(devam) WUBKU kurulum işlemleri

```
$baglantiIPAdresi = "IPAdresi";
```

WUBKU'nun düzgün çalışabilmesi için temel bilgilerin arayüz üzerinden tanımlanması gerekmektedir. İlk kurulumdan sonra, WUBKU web arayüzüne ***http://ipAdresi/wubku/giris.php*** bağlantısından ulaşılabilmektedir. Burada kurulum ile gelen kullanıcı adı "yönetici" ve şifre "1234" bilgileri ile giriş yapılabilmektedir. Yönetici olarak giriş yapıldıktan sonra, ***Ayarlar*** bağlantısı üzerinden "Web Uygulamaları Kök Dizin Yolu" ve "WUBKU Erişim Adresi" bilgileri tanımlanmalıdır. Yine bu sayfa üzerinden, WUBKU tarafından üretilen uyarıların bildirim şeklide belirlenebilmektedir. Ayarlar sayfasının ekran görüntüsü Şekil 1.4'te gösterilmektedir.



Şekil 1.4. WUBKU ayarlar sayfası görünümü

EK-1.(devam) WUBKU kurulum işlemleri

Eposta atılması için gerekli bilgiler, arayüz dosyalarının bulunduğu dizin içerisindeki epostaGonder.php dosyasında tanımlanması gerekmektedir. Tanımlamada değiştirilmesi gereken alanlar aşağıda verilmiştir.

```
$mail->Host= 'smtp.gmail.com';
$mail->From = 'wubku@gmail.com';
$mail->FromName = 'WUBKU';
$mail->SMTPSecure = 'tls';
$mail->Port = '587';
$mail->Username = 'wubku@gmail.com';
$mail->Password = 'epostaŞifresi';
$mail->SetFrom('wubku@gmail.com', 'WUBKU');
$mail->Subject = "WUBKU - Uygulamanızda saldırı tespit edildi";
```

WUBKU'nun tespit ettiği saldırıları eposta atabilmesi için ***/etc/crontab*** dosyası içeriğine aşağıdaki satır eklenmelidir.

```
*/1 *   * * *   root    curl  http://wubkuErisimAdresi/epostaGonder.php
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı :Aydoğdu, Durmuş
Uyruğu :T.C.
Doğum tarihi ve yeri :31.08.1989, Bakırköy
Medeni hali :Bekâr
Telefon :0(507) 319 60 70
Faks :-
e-mail :aydogdu.durmus@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek Lisans	Gazi Üniversitesi/ Bilgisayar Mühendisliği	2015
Lisans	Gazi Üniversitesi/ Bilgisayar Mühendisliği	2013
Lise	H. Nazif Kurşunoğlu Anadolu Meslek Lisesi	2007

İş Deneyimi

Yıl	Yer	Görev
2009-Halen	Deniz Kuvvetleri Komutanlığı Bilgi Sistemleri Güvenlik Personeli	

Yabancı Dil

İngilizce

Yayınlar

Aydoğdu, D., Gündüz S., Sağıroğlu, Ş. (2015). Web Uygulama Güvenliği Açıklıkları ve Güvenlik Çözümleri Üzerine Bir Araştırma. *The 3rd International Symposium on Digital Forensics and Security*, Ankara, Turkey, 144-149.

Hobiler

Ata binme, Yüzme, Tenis, Kayak