TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AÇIK KAYNAK WEB UYGULAMA GÜVENLİK DUVARI MODSECURITY'NİN KULLANILABİLİRLİK ANALİZİ

YÜKSEK LİSANS TEZİ

Murat ALAGÖZ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgi Güvenliği

Tez Danışmanı: Prof. Dr. Ali Aydın SELÇUK

ARALIK 2020

Fen Bilimleri Enstitüsü Onayı

……..
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

…….
Prof. Dr. Oğuz ERGİN
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 181111021 numaralı Yüksek Lisans Öğrencisi **Murat ALAGÖZ'**ün ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"AÇIK KAYNAK WEB UYGULAMA GÜVENLİK DUVARI MODSECURITY'NİN KULLANILABİLİRLİK ANALİZİ"** başlıklı tez **18.12.2020** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı :	**Prof. Dr. Ali Aydın SELÇUK** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Kemal BIÇAKCI (Başkan)** İstanbul Teknik Üniversitesi	.....
	Dr. Öğretim Üyesi Özgür ERGÜL Gazi Üniversitesi	.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Murat ALAGÖZ

ÖZET

Yüksek Lisans Tezi

AÇIK KAYNAK WEB UYGULAMA GÜVENLİK DUVARI
MODSECURITY'NİN KULLANILABİLİRLİK ANALİZİ

Murat Alagöz

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Bilgi Güvenliği

Danışman: Prof. Dr. Ali Aydın Selçuk

Tarih: Aralık 2020

Saldırganların ağ ve sistem güvenlik açıklarını istismar etmeye çalışması, güvenlik duvarı ve saldırı tespit sistemleri gibi ürünlere olan talebi artırmaktadır. Siber güvenlik ekiplerinin varlıklarını daha iyi yönetmesi ve güvenlik ürünlerinin çeşitlenmesi ile bilgi güvenliği ihlallerindeki artışın evrilerek, uygulama seviyesine doğru kaydığı gözlemlenmektedir. İyi tasarlanan güvenlik ürünleri sistem yöneticileri tarafından daha fazla tercih edilmekte ve güvenliği sağlamakta etkin rol oynamaktadır. Ancak ürünlerin geliştirilmesinde kullanılabilirlik ilkeleri göz ardı edildiğinde, kullanıcılar ürünleri hatalı yapılandırarak bekledikleri faydayı elde edememekte, hatta savunulacak sistemi daha korunmasız hale getirebilmektedir.

Güvenlik duvarı halen olmazsa olmaz güvenlik cihazı olarak görülmektedir. İlk çıktığı günden bu zamana kadar çeşitli yetenekler kazanmıştır. Saldırıların odak noktasında bulunan web uygulamaları, saldırı yüzeyinin önemli bölümünü oluştururken, sadece güvenlik duvarı veya saldırı tespit sistemleri web uygulamalarını korumakta yetersiz kalmaktadır. İstemcilerin sunucuya ulaşmadan önce bir güvenlik cihazında trafiğin sonlandırılarak, sadece belirlenen isteklerin sunucuya ulaşması istenmekte, böylelikle

web uygulama güvenlik duvarı ihtiyacı ortaya çıkmaktadır. Web uygulama güvenlik duvarı, web uygulamalarının güvenliğini sağlayan ve uygulama katmanına kadar çalışan güvenlik duvarı türüdür. Uygulama ile istemci arasında konuşlandırılır ve iki yönlü trafiği analiz edebilmektedir. ModSecurity, açık kaynaklı bir web uygulama güvenlik duvarıdır. Tüm web uygulamaları için temel koruma düzeyi sağlamayı amaçlayan Açık Web Uygulaması Güvenlik Projesi'nin (OWASP) Çekirdek Kural Kümesi (CRS) ile güvenlik kontrollerini verilen kural setlerine göre yapmaktadır.

Kullanılabilirlik çalışmalarıyla kullanılabilirliği olumsuz etkileyen hususlar tespit edilerek çözüm sağlamak mümkün olmakla beraber, en sık görülen web uygulama zafiyetlerinden olan yanlış yapılandırmaların azaltılması mümkündür. Bu çalışmada ModSecurity'nin hibrit yöntemle kullanılabilirlik denetimi ve çalışması gerçekleştirilmiştir. Birinci aşamada sezgisel değerlendirme ve bilişsel gözden geçirme teknikleri ile ModSecuirty değerlendirilmiş, elde edilen bulgular ışığında üç farklı konuda iyileştirme sağlayacak kural-kontrol ve geribildirim mekanizması tasarlanarak betik oluşturulmuştur. İkinci aşamada betiğin kullanılabilirliğe katkısının ölçümlenebilmesi için önce bulut sistemlerde deney ortamı hazırlanmış, geleceğin sistem yöneticileri olan bilgisayar mühendisliği öğrencileri ile pandemi şartları göz önünde bulundurularak uzaktan kullanıcı çalışması yapılmıştır. Sonuç olarak; tasarlanan mekanizmanın ModSecurity'nin doğru yapılandırılmasına olumlu katkıda bulunduğu, sistem yöneticilerinin kural girerken yaptığı söz dizimi hataları nedeniyle web sunucusunun hizmet dışı kalma süresini azalttığı tespit edilmiştir.

Anahtar Kelimeler: ModSecurity, Web uygulama güvenlik duvarı, Kullanılabilirlik, Kullanılabilir güvenlik.

ABSTRACT

Master of Science

USABILITY ANALYSIS OF AN OPEN SOURCE WEB APPLICATION FIREWALL MODSECURITY

Murat Alagöz

TOBB University of Economics and Technology
Institute of Natural and Applied Science
Department of Computer Engineering
Information Security

Supervisor: Prof. Dr. Ali Aydın Selçuk

Date: December 2020

Attacks try to exploit network and system vulnerabilities, increased the demand for products such as firewalls and intrusion detection systems. With the development of security products and the better management of cyber security teams, it is observed that the increase in information security violations has evolved and shifted towards the application layer. Well-designed security products are more preferred by system administrators and play an active role in providing security. However, when the usability principles are ignored in the development of the products, the users cannot obtain the benefit they expect from the products, they can even configure the products incorrectly and make the system to be defended more vulnerable.

The firewall is still considered a must-have security device. He has gained various talents since his first release. While web applications, which are at the focal point of attacks, constitute a significant part of the attack surface, only firewalls or intrusion detection systems are insufficient to protect web applications. Before the clients reach the server, the traffic is terminated on a security device and only the specified requests are requested to reach the server, thus the need for the web application firewall arises. A web application firewall is a type of firewall that secures web applications and works

down to the application layer. It is deployed between the application and the client and can analyze two-way traffic. ModSecurity is an open source web application firewall. With the Core Rule Set (CRS) of the Open Web Application Security Project (OWASP), which aims to provide a basic level of protection for all web applications, ModSecurity performs security checks according to the given rule sets.

Although it is possible to provide solutions by detecting issues that negatively affect usability through usability studies, it is possible to reduce the wrong configurations, which are one of the most common web application weaknesses. In this study, the usability control and operation of ModSecurity with the hybrid method was carried out. In the first stage, ModSecuirty was evaluated using heuristic evaluation and cognitive review techniques, and a script was created by designing a rule-control and feedback mechanism that will provide improvement in three different subjects in the light of the findings. In the second stage, in order to measure the contribution of the script to usability, first an experiment environment was prepared in cloud systems, and remote user work was carried out with computer engineering students, who are the system administrators of the future, taking into account the pandemic conditions. As a result; It has been determined that the designed mechanism contributes positively to the correct configuration of the web application firewall ModSecurity and reduces the downtime of the web server due to syntax errors made by system administrators while entering the rule.

Keywords: ModSecurity, Web application firewall, Usability, Usable security.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Prof. Dr. Kemal BIÇAKCI'ya, kıymetli tecrübelerinden faydalandığım Bilgisayar Mühendisliği Bölümü öğretim üyelerine, sektörel tecrübeleriyle katkı sağlayan misafir öğretim görevlilerine, yüksek lisans eğitimi süresince desteğini hiçbir zaman esirgemeyen M.Serkan TOK'a, oyun zamanlarından feragat eden ancak bunun farkında bile olmayan canım kızım Adel'e ve her zaman yanımda ve arkamda olan eşim Lyuba'ya teşekkür ederim.

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

KISALTMALAR

API : Uygulama programlama arayüzü (application programming interface)
ASP : Etkin sunucu sayfaları (active server pages)
CLI : Komut satırı arayüzü (command line interface)
DoS : Servis dışı kalma (denial of service)
DVWA : Lanet zafiyetli web uygulaması (Damn vulnerable web application)
ftp : Dosya aktarım iletişim kuralı(file transfer protocol)
ftps : Güvenli dosya aktarım iletişim kuralı(file transfer protocol)
go : Go programlama dili (golang)
GUI : Grafik kullanıcı arayüzü (graphical user interface)
HTML : Hiper Metin İşaretleme Dili (Hypertext Markup Language)
http : Hiper-Metin Transfer Protokolü(Hyper-Text Transfer Protocol)
https : Güvenli Hiper-Metin Transfer Protokolü(Hyper-Text Transfer Protocol Secure)
IDS : Saldırı Tespit Sistemleri (Intrusion Detection Systems)
IIS : İnternet bilgi servisi (Internet Information Service)
IP : İnternet protokolü (Internet Protocol)
IPS : Saldırı önleme sistemleri (Intrusion prevention systems)
IPSec : Internet protokolü güvenliği (internet protocol security)
ISO : Uluslararası Standartlar Teşkilatı (International Organization for Standardization)
OSI : Open Systems Interconnection
OWASP : Açık web uygulama güvenliği projesi (Open web application security project)
Owasp-crs : OWASP Çekirdek kural seti (Core ruleset)
PGP : E-posta güvenliği programı (Pretty good privacy)
PHP : Üstünyazı Önişlemcisi (Hypertext Preprocessor)
RFC : TCP/IP nin tanımlanmasında kullanılan standart numaralara sahip dokümanlardır. (Request for comments)
Rx : Düzenli ifadeler (Regular expression)
SQL : Yapılandırılmış sorgu dili (Structured Query Language)
SSH : Güvenli kabuk (secure shell)
TLS : Taşıma katmanı güvenliği (transport layer security)
URL : Tekdüzen Kaynak Bulucu (Uniform Resource Locator)
VPN : Sanal özel ağ (Virtual private network)
WAF : Web uygulama güvenlik duvarı (Web application firewall)

RESİM LİSTESİ

Sayfa

1. GİRİŞ

Milyarlarca insanı birbirine bağlayan internet, modern bilgi toplumunun temel direği olarak nitelendirilmektedir. Dünya çapında yaklaşık olarak 4.57 milyar tekil kullanıcının internet vasıtasıyla birbirine bağlı olduğu, bunun da dünya nüfusunun %59'unu oluşturduğu bildirilmiştir [1].

Bilgi teknolojileri hayatın her alanına nüfuz ederken, kullanılan cihazların birbirleriyle ve insanlarla olan iletişiminin güvenli hale getirilmesi büyük çaba, kaynak ve iş gücü gerektirmektedir. Bankacılık, enerji, sağlık, sanayi sistemleri ve hatta sosyal medya gibi başkalarıyla veri paylaşımı faaliyetleri web uygulamaları kullanılarak yapılabilmektedir. Birçok alanda ortak kullanım sunan bu uygulamalar, farklı ihtiyaçları karşılarken çok değişik teknolojiler kullanmaktadır. Web uygulamaları, php, asp, python, java, go, ruby gibi çok çeşitli dillerde geliştirilebilmektedir. İhtiyaç duyulan veriler mssql, mysql, oracle, postgresql gibi çeşitli veri tabanı sistemleri yardımıyla saklanmakta ve anlık olarak erişim sağlayarak kullanıcıların deneyimlerini daha iyi seviyeye getirmektedir. Ancak diğer yandan yazılımların karmaşık ve birbiriyle ilişkili olması, uygulama güvenliğini sağlamayı zorlaştırmaktadır [2].

Web uygulamaları siber saldırıların birincil hedefi haline gelirken, web uygulamalarındaki zafiyetlerin istismar edilmesi finansal ve itibari kayıplara neden olmaktadır [3]. Açık Web Uygulama Güvenliği Projesi (OWASP) güvenli uygulama geliştirme ve sürdürmeye yönelik çalışma yapan bir topluluktur. Web uygulamalarına yönelik en kritik güvenlik riskleri hakkında geniş bir fikir birliğini temsil eden OWASP İlk 10 ile sorunlu alanlardan en çok yapılan on tanesine karşı korunmak için eğitim ve farkındalık sağlamaktadır [4].

İlk zamanlarda web uygulamalarının güvenlik sorumluluğu, sadece geliştiriciler veya uygulama sahiplerine ait olarak görülmekte iken, son zamanlarda daha proaktif bir savunma anlayışına dayanan web uygulama güvenlik duvarları bu sorumluluğu paylaşmaktadır [5]. Sistem yöneticileri tarafından kullanılan komut satırı arabirimi (CLI), grafik kullanıcı arabirimi (GUI) ve uygulama programlama (API) arabirimi

arasından, CLI'ın en sık tercih edildiği çalışmalarda tespit edilmiştir [6][7][8] . Fakat deneyimli sistem yöneticilerinin bu seçimi komut satırının kullanılabilirlik açısından bazı dezavantajları beraberinde getirdiği gerçeğini değiştirmemektedir [7]. Özellikle komut satırı ile yönetilen güvenlik sistemleri günlük kullanıcıların uğraşamayacağı kadar zaman alıcıdır veya karmaşıktır, kullanmaya istekli olan kişilerin bile hata yapmalarına yol açmaktadır. Gerekli görevleri makul sürede, makul bir çaba ile yerine getirememek, büyük çoğunlukla güvenlik mekanizmalarının kullanımının reddedilmesine veya sistem yöneticilerinin hata yapmasına yol açmaktadır [9].

OWASP tarafından 2003 yılından beri yayınlanan İlk 10 projesinin amacı; geliştiricileri, tasarımcıları, web uygulama mimarlarını ve yöneticilerini en yaygın görülen web uygulaması güvenlik zayıflıklarının sonuçları hakkında eğitmektir. İlk 10, yüksek sorunlu alanlara karşı korunabilmek için temel teknikler sağlar ve iyileştirmelerin nasıl olacağı konusunda rehberlik yapar. 2017 yılında yayınlanan son sürümünde, web uygulamalarının karşı karşıya oldukları en yaygın görülen risk veya sorunlardan altıncısı yanlış güvenlik yapılandırmasıdır [10]. Güvenlik sistemlerinde kullanılabilirlik sorunları yüzünden yapılan hatalı yapılandırmalar vahim sonuçlar doğurabilmektedir.

Bu yüksek lisans tezinin amacı; ModSecurity açık kaynak web uygulama güvenlik duvarının kullanılabilirlik teknikleri ile analizini gerçekleştirmek, elde edilen bulgular ışığında ModSecurity'nin kullanılabilirliğinin artırılması için gerekli önlemler sunarak alınan önlemlerin kullanıcı çalışması ile doğrulayarak kullanılabilirliğini artıracak öneriler sunmaktır.

2. LİTERATÜR TARAMASI

2.1 Güvenlik Duvarı ve WAF Araştırmaları

Srokosz ve diğ.(2018) akademik araştırmaları inceleyerek web uygulamalarını koruma yöntemlerinin çok tartışılmadığını, çoğunlukla güvenlik duvarlarında tespit edilemeyen standart dışı saldırıların ne şekilde yapıldığına odaklanıldığını bildirmiştir [11].

Clincy ve Shariar (2018) çalışmasında Web Uygulama Güvenlik Duvarının gerekliliğinin yanı sıra, olumlu veya olumsuz politika tabanlı saldırı algılama modellerinin güçlü ve zayıf yönleri olduğunu belirtmektedir. Web sunucusunun varsayılan yapılandırmasını kullanmanın güvenlik duvarı olmasına rağmen açıklıklara yol açtığını öne sürerek, çözüm olarak ise uygulama güvenlik testleri yapılmasını önermiştir [12].

Rietz ve diğ. (2016) kötü niyetli veya bilinmeyen aktif içerikleri tespit etmek ve engellemek için uygulama protokolünü, belge yapılarını ve programlama dillerini ayrıştırabilen saldırı tespit analiz birimini güvenlik duvarına bütünleştirmeyi önermiştir. Çalışmada HTML sayfaları ve javascript modellemelerine odaklanılmıştır ancak geniş web içeriğinde bulunan diğer teknolojilere değinilmemiştir [13]. Ghanbariyan ve diğ.(2015)[14] web uygulamalarında kullanılan parametreler her ne kadar doğru olarak kullanılsa bile HTML ve HTTP RFC [15][16][17][18]'lerinde belirtilen uyarlamaların dışında kullanım olduğunda WAF tarafından isteklerin engellendiğini tespit etmiştir [14].

ModSecurity, güvenlik kurallarını web uygulamaların ihtiyaçlarına uygun olarak özelleştirme yeteneği sağlamakta ve OWASP ModSecurity Çekirdek Kural Seti (CRS) ile birlikte yaygın olarak kullanılmaktadır. Sobola ve diğ. (2020) yaptığı çalışmada Modsecurity'nin OWASP ilk 10 listesindeki risklere karşı performansını değerlendirmiştir. ModSecurity ve kural seti olarak CRS'nin web saldırılarını tespit yetenekleri ve yoğun trafiğe (DoS) maruz kaldığında performans açısından etkinliğinin anlaşılmasına katkıda bulunmuş, çeşitli saldırı vektörlerinin birleştirilerek

güvenlik duvarını başarılı bir şekilde atlatmak için uygulanabileceği yeni yolların olduğunu göstermektedir [19].

Razzaq ve diğ. (2013) çalışmasında açık kaynak kodlu olarak ve ticari pazarda bulunan çeşitli web uygulama güvenlik duvarlarını sistemlerin güvenliği penceresinden karşılaştırmıştır. Bu sistemlerin tek başına yeterli olmayacağı, mutlaka geleneksel güvenlik cihazlarına ek olarak konumlandırılması gerektiğini belirtmiştir [3]. Tirumula ve diğ.(2015) ücretsiz ve açık kaynak saldırı tespit sistemleri ile ilgili yaptığı çalışmada kullandığı açık kaynak araçlardan birisi olarak IronBee web uygulama güvenlik duvarını inceleyerek web saldırılarında küçük ve orta ölçekli web uygulamalarında başarılı sonuçlar elde ettiğini tespit etmiştir [20].

Beckerle ve Martucci (2013) çalışmasında, erişim denetimi kural kümelerinin ne kadar kullanılabilir olduğunu ölçebilecek güvenlik ve kullanılabilirlik ölçütlerini tanıtmıştır. Geliştirdikleri yaklaşım ile farklı kural kümelerini karşılaştırmak için tek tip ve bilimsel bir yöntem sunarak, sonuçların uygulanmasının kural setlerini önemli ölçüde iyileştirebileceğini göstermiştir [21].

Smetters ve Good (2009) çalışmasında, kullanıcıların verileri üzerinde hangi düzeyde kontrole ihtiyaç duyduklarını anlamak için kurumsal bir ortamdaki dosya paylaşımı uygulamalarında ne tür erişim politikaları oluşturulduğunu incelemiştir. Politikaların kullanılabilirliği artırmak ve hatayı azaltmak amacıyla erişim kontrol sistemleri ve yönetim araçları için bir dizi iyileştirme önermiştir [22].

Wool(2004) tarafından yapılan çalışmada, kurumsal güvenlik duvarlarının genellikle kötü yazılmış kural setlerini uyguladığını, kural seti kalitesini iyileştirmek için bazı yararlı gözlemler sunmuştur [23].

Alfayyadh ve diğ. (2010), Microsoft Windows XP işletim sisteminde çalışan dört popüler son kullanıcı güvenlik duvarını, Jøsang ve diğ.(2007) tarafından önerilen sekiz kullanılabilirlik ilkesine göre bilişsel gözden geçirme ve değerlendirme yöntemi kullanarak analiz etmiştir. Güvenlik duvarlarının tasarım sürecine kullanılabilirlik uzmanlarının dâhil edilmesini ve kullanıcılarla kullanılabilirlik testleri yapılmasını önermektedir [24][25].

Mansmann ve diğ. (2012) güvenlik duvarının kullanıcı arayüzünde bulunan kullanılabilirlik sorunlarını sistem yöneticilerinden geri bildirim alarak incelenmiştir. Karmaşık kurallarda görünürlük sorunları olduğunu öne sürerek, güvenlik duvarı kural

kümelerini ve nesne gurubu tanımlarını anlamaya yönelik görselleştirme aracı tasarlamıştır. Yaptığı iki vaka çalışması sonucunda, yapılandırma dosyasında kural arama ve bulmanın görselleştirilmesi, renk bağlantısı ve etkileşim yoluyla düzenleyebilmesinin büyük fayda sağladığı tespit edilmiştir [26].

Zhang ve diğ. (2007) tarafından yapılan çalışmada, ücretsiz kural setlerinde bulunan yanlış yapılandırma ayarları ele alınmıştır. On iki güvenlik duvarı yöneticisinin katılımıyla kullanıcı çalışması yapılmıştır [27].

Raja ve diğ.(2009) Güvenlik Duvarı Kullanıcılarının Zihinsel Modellerini İyileştirme çalışmasında; tasarımcıların kişisel güvenlik duvarının ara yüz basitliğini sağlamak için kullanıcının yaptığı eylemin karmaşıklığının gizlenmemesi gerektiğini vurgulamıştır [28]

Bhatt ve diğ.(2008) çalışmasında, sistem yöneticilerini genellikle sınırsız teknik bilgi ve beceriye sahip özel bir kullanıcı türü olarak kabul etmekte, ortalama kullanıcının gerekli görevleri tamamlamasını zorlaştıracak veya imkânsız hale getirecek herhangi bir yazılım eksikliğinin üstesinden gelebilecek özel yetenekli kişiler olarak varsaymaktadır [29]. Avishai Wool. ve diğ. (2010)'da tespit edildiği üzere, kullanılabilirlik özelliklerinin bazılarını göz ardı etmenin açıkça hatalı bir yaklaşım olduğu ve güvenlik duvarı kurallarındaki yanlış yapılandırma sorunlarının çözülmesine yardımcı olmadığı belirtilmiştir [30].

Voronkov ve diğ.(2017) güvenlik duvarı yapılandırması, yöneticilerin veya son kullanıcıların günlük işlerinde uğraşmaları gereken karmaşık ve hataya açık bir süreç olduğunu, güvenlik duvarlarının yanlış yapılandırılmasının, ağda çok sayıda güvenlik açığına yol açacağını, bu nedenle güvenlik duvarı yapılandırma sürecinin kullanılabilirlik çalışmalarından büyük ölçüde yararlanabileceğini öne sürmektedir [9].

Voronkov ve diğ. (2020) güvenlik duvarı kural setlerinin kullanılabilirliği ile ilgili yaptığı çalışma güvenlik duvarı kural kümelerinin yönetilebilirliğini etkileyen dört kullanılabilirlik özelliğini tanıtmış, sekiz sistem yöneticisi ile yapılan bir pilot çalışma yaparak sonuçların belirlemiştir. Algılanan bilişsel çabanın, güvenlik duvarı kural kümesindeki kuralların görünümünden etkilenip etkilenmediğini test etmek için bir kullanıcı çalışması tasarlanmıştır. Etkiyi önce göstermiş ve sonrasında ölçmüştür.

Sonuç olarak yapılan iyileştirmenin ortaya koyduğu ölçütlerle çok güçlü bir şekilde ilişkili olduğunu göstermiştir [31].

2.2 Kullanılabilirlik Araştırmaları

Kullanılabilir güvenlik denilince ilk akla gelen, Whitten ve Tygar (1999) tarafından yapılan "Why Johnny can't encrypt" çalışmasıdır. Kullanıcı hatalarının birçoğunun güvenlik hatalarına yol açtığı veya katkıda bulunduğu öne sürülerek, elektronik posta güvenliğini sağlamak için PGP 5.0 ile vaka çalışması tasarlanmış, kullanıcı testiyle 90 dakika içerisinde kullanıcıların şifreli bir e-posta göndermesi istendiğinde çoğunluğunun bunu yapamadığı ortaya koyulmuştur [32].

Juristo ve diğ. (2007) yazılımların kullanılabilirliğini iyileştirmek için kullanılabilirlik modellerinin kullanılmasını ve geliştirme süreci boyunca çok sayıda kullanıcının katılımı ile kullanılabilirlik çalışmalarının yapılması gerektirdiğini vurgulamıştır. Geri bildirim özelliğini; sistem durumu, etkileşim, uyarı ve uzun işlem geri bildirimi olarak dörde ayırarak hangi amaçlara hizmet edeceğini tanımlamıştır [33].

Jain ve diğ. (2019) kullanıcıların sağlıklı gıda seçimi yapmalarına yardımcı olan karar destek programı geliştirmişlerdir. Karar yardımı için iki geri bildirim tasarımı oluşturarak, birinde açılır pencere biçiminde diğerinde ise açılan pencere ile birlikte sesli geri bildirim sağlamıştır. Sonuç olarak; iki geri bildirim tasarımının karar verme sürecinde bilişsel çabayı önemli ölçüde azaltarak kullanılabilirliği artırdığını tespit etmişlerdir [34].

Kung ve diğ. (2020) çalışmasında sanal klavye girişinde geribildirim modellerinin kullanılabilirliğini araştırmıştır. Kullanıcıların doğru tuşları bulmalarına yardımcı olmak için, son derece ayırt edilebilir bir dizi titreşim modeli tasarlanmış ve sanal klavyenin farklı bölgeleri ile ilişkilendirilmiş ve kullanıcı çalışması yapmıştır. Uygun sayıda uyaran modelin daha yüksek yazma hızı, daha yüksek yazma etkinliği ve daha düşük hata oranı sağladığını göstermiştir [35].

3. TEKNİK ARKA PLAN

3.1 Güvenlik Duvarı

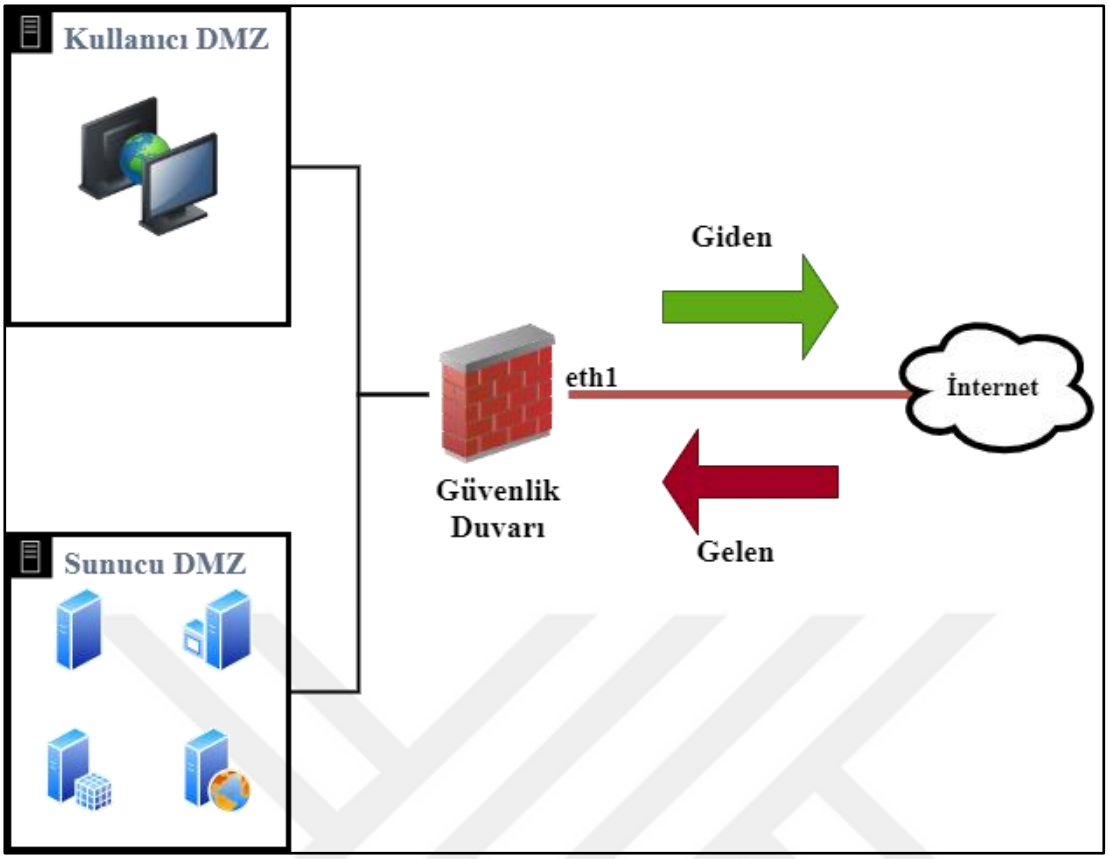
Firewall (Ağ Güvenlik Duvarı)deyimi bilgi çağı öncesinde yangının yapıların arasında yayılmasını önlemek için oluşturulan yapılara verilen isim iken günümüzde bu deyim bilgi teknolojilerinde kullanılan bir terim haline gelmiştir. Türk Dil Kurumu tarafından Ağ Güvenlik duvarı olarak çevrilmiştir [36] ve günümüzde güvenlik duvarı olarak kullanılmaya devam etmektedir.

Güvenlik duvarı, ağlar arasında hareket eden trafiği verilen kural setleri vasıtasıyla inceleyerek yetkisiz erişimleri engelleyen güvenlik cihazıdır. Tasarımdaki amacı trafiğin denetlenmeden her iki yönde de güvenlik duvarını geçememesidir. Günümüzde üretilen yönlendirici, modem, kablosuz istasyonlar, anahtarlama cihazları gibi ağ cihazları büyük çoğunlukla güvenlik duvarı ile birlikte sunulmakta, işletim sistemlerinde bir özellik olarak beraberinde gelebilmektedir. Windows işletim sistemine sahip bilgisayarlarda yazılım olarak kullanılabildiği gibi, Linux işletim sistemi çekirdeği seviyesinde kural setlerini işletebilmektedir.

İnternetin doğuşundan bu yana güvenlik duvarı yapısal olarak çok büyük değişime uğramıştır. Güvenlik duvarı olarak açık kaynak ve ücretsiz yazılımlar bulunurken diğer yandan, satın alındığında ise daha gelişmiş ara yüzler, içerik filtreleme, uygulama katmanı güvenliği ve VPN gibi servislerde dâhil edilerek çok geniş yelpazede ürünler bulunabilmektedir.

Güvenlik duvarı yöneticisi tarafından yönetilen kural listesi vasıtasıyla ağlar arasında hareket eden paketlere ne olacağı belirlenir. Güvenlik duvarından bahsederken birkaç terimi açıklığa kavuşturmak faydalı olacaktır.

Ağ bakış açısıyla Şekil 3.1'de verilen ağ topolojisiyle tanımlama yapacak olursak, eth1 arayüzüne gelen paketler (inbound), çıkan ya da giden paketler ise (outbound) olarak adlandırılır.



Şekil 3.1 : Basit bir ağ topolojisinde gelen ve giden trafik.

Kural söz dizemi genellikle Şekil 3.2'de görüldüğü gibidir. Gelen paketin bilgilerinden; kaynak ip ve port, hedef ip ve port ve protokol bilgileri kural kümesiyle satır satır karşılaştırılır. Bu inceleme neticesinde gelen paket eşleştiği ilk kuralda bulunan eyleme göre üç farklı şekilde uygulanabilir; düşürme (drop), reddetme (deny), kabul etme (allow). Kural kümesindeki son satır hiçbir kurala uymayan bir pakete ne yapılacağını belirler.

<src_addr> <src_port> <dst_addr> <dst_port> <protocol> <action>
<kaynak ip> <kaynak port> <hedef ip> <hedef port> <protokol> <eylem>

Şekil 3.2 : Güvenlik kuralı söz dizemi.

Kurumsal çevrelerdeki güvenlik duvarları yetkisiz trafiğin çoğunu dışarıda tutar. İzinsiz giriş tespit sistemleri, neler olduğu konusunda farkındalık ve iyileştirme fırsatları sağlar. Kullanıcı trafiği, IPsec tabanlı VPN'ler, SSH, TLS ve şifreli e-posta gibi teknolojilerle kriptografik olarak korunur ve gelen paketlerin veya bağlantıların kimlik doğrulaması yetkili varlıkları ve verileri ayırt etmek için kullanılır [37]. Sonuç

olarak zamanla ortaya çıkan çeşitli ihtiyaçları karşılamak amacıyla güvenlik duvarları da geliştirilmiş ve çeşitlendirilmiştir.

ISO Uluslararası Standartlar Teşkilâtı tarafından ağ cihazlarının birbiriyle nasıl iletişim kuracaklarını belirleyen bir model olarak OSI modeli 1984 yılında çıkartılmıştır. Kısa sürede yaygın şekilde kabul görerek, ağ üzerinde hareket eden verilerin kapsülleme yöntemiyle nasıl hareket ettiğini açıklamaktadır [38].

Güvenlik duvarı çeşitleri çalıştığı katmana, kullanım sahasına, çalışma şekline hatta bütçeye göre bile sınıflandırılabilmektedir. Bu çalışmada en yaygın görülen özelliğine, trafiğin hangi OSI katmanlarına göre denetlendiğine uygun olarak ele alınmıştır

3.1.1 Paket filtreleme güvenlik duvarı

Paket filtreleme güvenlik duvarı yalnızca OSI modelinin ağ katmanında çalışır ve uygulama protokolleri arasında ayrım yapmaz. Paketin IP'sindeki ve protokol başlıklarındaki alanları inceleyerek, tek tek paketleri kabul edip etmeyeceğine karar verir. İşlem gereksinimleri düşük olduğu ve performansı gözle görülür derecede etkilemediği için ilk zamanlarda iyi bir seçenek olarak görülmüştür. Paket filtreleme güvenlik duvarı, IP paketlerini kaynak ve hedef IP adresi ile kaynak ve hedef bağlantı noktasına göre filtreler. Yalnızca paket başlıkları incelendiğinden, saldırganlar tarafından basit sahtekârlık teknikleriyle atlatılabilir. Günümüzde bu tür bir güvenlik duvarı basit ve sınırlı olarak kabul edilmektedir.

3.1.2 Devre seviyesi ağ geçidi

Devre seviyesi ağ geçidi, OSI referans modellerinin taşıma katmanında çalışır. Paket seviyesinde filtreleme yerine devre seviyesinde filtreleme uygular. Bir oturum açılmadan ve veri alışverişi yapılmadan önce, taşıma katmanındaki bağlantıların (yani devrelerin) geçerliliğini izin verilen bağlantılar tablosuna göre kontrol eder. Geçerli bir oturumu tanımlayan kurallar, örneğin hedef ve kaynak adreslerini ve bağlantı noktalarını, günün saatini, kullanılan protokolü, kullanıcıyı ve şifreyi belirler. Bir oturuma izin verildiğinde başka kontrol yapılmaz.

3.1.3 Uygulama seviyesi ağ geçitleri

Uygulama ağ geçidi veya uygulama düzeyi ağ geçidi, istemci ile sunucu arasında aracı görevi gören proxy aracılığıyla uygulanır. Korunan ağ içindeki istemci, internet gibi

daha az güvenli ağlardan gelen hizmetleri kullanmak isteyebilir. İstemcinin kimlik doğrulaması onaylandıktan sonra yürürlükteki güvenlik politikaları tarafından izin verilen hizmetler proxy sunucusu tarafından ileriye aktarılır. Böylelikle tüm veri alışverişleri proxy sunucusu tarafından gerçekleştirilir. Hedeflenen uygulama protokolüne karşılık gelen paketler, uygulamaya özel verileri incelemek için özelleştirilmiş filtreye yönlendirilir. Bu sayede paketlerin tamamen engellenmesine değil, paketin içindeki verilerin değiştirilmesine neden olabilir [37].

3.1.4 Yeni nesil güvenlik duvarı

Durum bilgisi içeren çok katmanlı inceleme güvenlik duvarı, OSI modelinin yedi katmanının tümünü inceleyen karmaşık filtreleme biçimi kullanır. Her paket bütünüyle incelenir ve izin verilen paket durumları veya kurallar ile karşılaştırılır. Paketin başlığına ve içeriğine göre ayrıntılı erişim kontrolü kararları verebilir. Uygulama ve veriye dayalı tehditleri yönetmede üstünlük sağlarken izinsiz giriş algılama ve önleme teknolojisini içerir. Başlıca özellikleri; şifreli trafik denetimi, uygulama kontrolleri, kimlik tabanlı kontroller, veri ihlali/sızıntısı koruması içerik filtrelemedir [39].

Güvenlik duvarı, iç ağı yetkisiz erişimden koruyabilecek cihazdır. Yalnızca paketleri filtreleyen basit yöntemden amacına, kaynaklarına ve hedeflerine bağlı olarak trafiğe izin vermeye veya trafiği engellemeye karar verebilen karmaşık paket denetçilerine kadar yıllar boyunca ağ ortamına dâhil olmuştur [40]. Fakat güvenlik duvarı nihayetinde kısıtlamaları olan bir güvenlik sistemidir ve tek başına bütün ağın güvenliği sağlaması beklenemez [41].

Artık güvenlik duvarları, saldırı tespit sistemleri (IDS) ve izinsiz giriş önleme sistemleri (IPS) gibi geleneksel güvenlik ürünlerinin günümüzün web uygulamalarını tehlikelerden korumak için yeterli olmadığının farkına varılmaktadır. Güvenlik duvarları; uygulama katmanı protokol verilerini saldırı işaretleri için yeterince analiz etmez, saldırı tespit sistemleri (IDS); algılanan bir saldırıyı durdurmak için herhangi bir işlem yapmaz sadece durumsal farkındalık yaratır, son olarak izinsiz girişi önleme sistemleri (IPS) ise, HTTP trafiğinin yeterince anlayamamaktadır [42].

Birçok kuruluş, uygulamalarını güvenlik açıklarına karşı test etmenin ve ardından sonuçları anlama, önceliklendirme sonrasında ise kodda iyileştirme yapmanın düşünülenden çok daha zor olduğunu ve uzmanlık gerektirdiğini keşfetmiştir. Gerekli

uzmanlığa sahip olmayan kuruluşlar, uygulamalarının maruz kaldığı riski genellikle tam olarak anlayamaz ve bu nedenle çok fazla öncelik vermezler.

Web uygulama sahipleri, güvenlik açıkları için web uygulamalarının değerlendirilmesini, sonuçların güvenlik uzmanları tarafından analiz edilerek uygun şekilde önceliklendirme yapılmasını ister. Güvenlik açıklarının tanımlanarak geliştirme ekibi tarafından düzeltilmesini sağlamaya yönelik tüm çabalar, kuruluşları yine de kabul edilemez derecede riske maruz bırakmaktadır. Ticari yazılım satıcıları, geliştirilen uygulamaların güncellemelerini çabuk yamalayamadığından, kodlama ve test için zaman gerektirdiğinden, web uygulamalarındaki güvenlik açıklarının giderilmesi hemen gerçekleşmez. Sonuç olarak yama işleminin zaman alması; web uygulamalarının korumasız olduğu sürece saldırganların güvenlik açığından yararlanılabileceği önemli bir fırsat yaratmaktadır.

3.2 WAF Ve ModSecurity

WAF'lar, saldırıları engellemek için web trafiğinin uygulama katmanına özgü ayrıntılı analizini yaparak, belirli teknolojilerle ağ savunmalarındaki boşluğu ortadan kaldırır ve koruduğu web uygulamasındaki açıklıkları azaltmaktadır.

Yapılan ankete göre, Web uygulamaları internette birincil saldırı vektörü olmaya devam etmekte ve azalmaya dair hiçbir işaret göstermemektedir [43]. Saldırganlar, siteler arası komut dosyası oluşturma, SQL enjeksiyonu ve hedeflenen diğer birçok sızma tekniğine yönelik ağ saldırılarını daha fazla kullanmaktadır. Web uygulamalarında bulunan güvenlik açıkları, zayıf giriş doğrulaması, güvenli olmayan oturum yönetimi, yanlış yapılandırılmış sistem ayarları ve hatta işletim sistemlerindeki veya web sunucusu yazılımındaki zafiyetler de dâhil olmak üzere pek çok açıklığı kullanarak karmaşık saldırılar gerçekleştirebilmektedir [44].

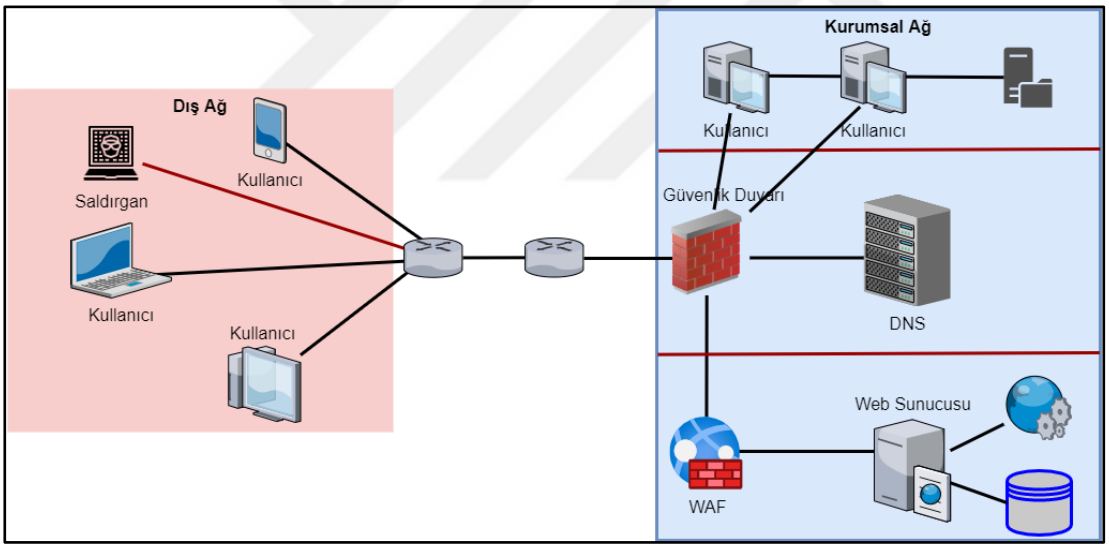
Çözüm olarak ise güvenli kod geliştirme, yapılandırma hatalarını en aza indirme ve uygulama katmanında çalışan güvenlik cihazı kullanma gibi çeşitli metodolojiler ve teknikler kullanılmaktadır [45].

3.2.1 WAF

Web uygulama katmanında güvenlik sorunlarını sağlayacak protokol ve standartlar yoktur. Güvenli kod geliştirme tamamen geliştiriciye bağlıdır. Geliştiriciler ise

güvenlik risklerini çeşitli sebepler ile önemsememektedirler. Dolayısıyla web uygulamalarında güvenlik açıkları bırakırlar ve saldırganlar bundan kolayca yararlanabilir. WAF'lar, kullanıcı giriş alanlarından URL'lere ve başlıklara kadar her şeyi inceler, ayrıca kullanıcı oturumlarını ve tanımlama bilgilerini izler ve sunucudan hassas verilerin sızmasını engeller.

Web uygulama güvenlik duvarı (WAF), web uygulamalarına gelen trafiği görüntüleme, filtreleme ve gerektiğinde iki taraflı olarak engelleme görevi üstlenir. WAF, web uygulamasına gelen şüpheli istekleri incelerken aynı zamanda zararsız isteklerin sorunsuz bir şekilde çift yönlü olarak iletilmesini sağlamaktadır [46]. Şekil 3-3'de verilen topoloji incelendiğine, web uygulaması kurumsal bir ağda segmentasyon yaparak konuşlandırılmıştır. Web uygulamasının dış ağa hizmet vermesi istendiğinde, bütün istekler web uygulamasına gidecek şekilde güvenlik cihazları yapılandırılır. WAF ise web sunucusuna gelen isteklerin arasında şüpheli istekleri engellemek için konumlandırılmaktadır.



Şekil 3.3 : Topolojide WAF'ın konumlandırılması.

WAF, http isteğini web uygulamasına göndermeden analiz eder ve istenmeyen web trafiğini engeller. WAF tarafından sunulan önemli diğer bir özellik, web uygulamalarında açığa çıkan zafiyetlere yönelik yama uygulanamadığında, keşfedilen güvenlik açığından koruyan sanal yamalamadır (virtual patch).

Sanal yama terimi ilk olarak Saldırı Önleme Sistemi (IPS) satıcıları tarafından kullanılmaya başlanmıştır. Web uygulamasına özgü bir terim değildir ancak kullanımdan ötürü bu şekilde bilinmektedir. Sanal yama, uygulamada bulunan

güvenlik açıklığından yararlanma girişimlerini belirleyebilen ve engelleyebilen politikadır. WAF'a bu politika atandığında, trafik analiz edildiğinden ve saldırıları engellediğinden, kötü amaçlı trafik hiçbir zaman web uygulamasına ulaşmaz. Sonuç olarak uygulamanın kaynak kodunu değiştirmeden, yama sayesinde korunmasıdır.

Uygulama sahipleri kaynak kodundaki açıklıkları neden düzeltemedikleri göz önünde bulundurulursa, sanal yamanın önemi daha iyi anlaşılabilir. Uygulamalar, satıcı tarafından sağlanan yama yayınlanana kadar veya bir yama test edilip çalışan bir sistem güncellenene kadar risk altında kalmaya devam eder. Sanal yama, çevrimdışı olması imkânsız olan kritik sistemler için koruma sağlar, acil yama uygulamak için harcanan zamanı ve parayı azaltır. Kuruluşların normal yama döngülerini sürdürmelerine olanak tanır.

Sayılan bu sebeplerden dolayı, web uygulamalarında bilinen ve bilinmeyen sorunları önlemek için WAF kullanılmaktadır. Farklı teknikler kullanarak kötü amaçlı sorgu veya isteklerin yapılmasını engeller. Uygulamaları farklı tehditlere karşı koruyabilir, ancak tam koruma sağlamaz. Uygulama tasarımsal olarak güvenliği tehdit eden modül barındırıyorsa, WAF'ın güvenlik açısından yanlış bir şey olduğunu belirlemesi çok zordur.

Açık kaynak ve farklı yazılım üreticileri tarafından çeşitli WAF çözümleri sağlanmaktadır. Bu çözümler genel olarak farklı teknikler kullanarak korumaya sağlarken hepsinin temel ve ortak özelliği ise koruduğu web uygulamasına göre yapılandırılmasıdır [3].

3.2.2 ModSecurity

ModSecurity, web sunucularını ve istemcilerini saldırılardan korumak için kullanılan web uygulaması güvenlik duvarıdır. İlk sürümü Kasım 2002 yılında Ivan Ristic tarafından yazılarak Apache Lisansı 2.0 altında yayımlanmıştır Başlangıçta sadece Apache sunucuları ile çalışabilmekte iken, zaman içerisinde Nginx ve IIS desteği de eklenmiştir. Web uygulaması ile aynı sunucuda gömülü veya ters vekil sunucu olarak çalışabilmektedir. Bu çalışmanın yapıldığı versiyonunda grafik ara yüz veya öğrenme mekanizması bulunmamaktadır [47].

2015 yılında yapılan bir araştırmada, dünyada çapında en sık kullanılan web sitelerinin %49'u Apache, %22'si NginX ve %12'sinin ise Microsoft IIS sunucuları olduğu bildirilmiştir [48]. ModSecurity, açık kaynaklı olduğu için esnek yönetim

özellikleriyle başka modüller ile birleştirilebilir, isteğe göre değiştirilebilir ve farklı web uygulamalara sahip sunucuları koruyacak şekilde yapılandırılabilmektedir [3] . Bu tez çalışmasında ModSecurity'nin ele alınmasının iki sebebi bulunmaktadır. Birincisi, ModSecurity günümüzde yaygın olarak kullanılan, açık kaynaklı ücretsiz bir web uygulaması güvenlik duvarıdır. İkincisi, esnek ve kararlı çalışan kural diline sahiptir ve karmaşık güvenlik açıklarını azaltmasına olanak tanıyan bir dizi benzersiz özelliğe sahiptir. Ancak kullanmadan önce uygun şekilde yapılandırılması ve ayarlanması gerekmektedir.

ModSecurity güvenlik duvarının varsayılan yapılandırmasında herhangi bir kural seti olmadığı için OWASP tarafından geliştirilen Çekirdek Kural Setiyle OWASP-CRS kullanımı yaygındır [49]. Sunucuya gelen ve giden trafiği günlüğe kaydeder, verilen kurallar kümesine göre istekleri ve yanıtları kontrol ederek engellemek için kullanılabilir. OWASP-CRS'deki kurallar, yanlış yapılandırılmış veya hatalı biçimlendirilmiş HTTP trafiğini, yaygın web uygulaması saldırı tekniklerini, sunucudan çıkan hassas verileri ve bir dizi başka denetimi kontrol eder. Birçok web saldırı türüne karşı hazır kurallar yazılmış olması nedeniyle güvenlik uzmanları tarafından tercih edilmektedir[19].

Güvenlik kural motoru SecRuleEngine tarafından trafik, kurallarla karşılaştırılarak kuraldaki parametrelere göre denetim yapılır. SecRuleEngine üç farklı çalışma modu bulunmaktadır. Açık (On) olarak ayarlanmışsa kurallar işlenir, kapalı (Off) olarak ayarlanmışsa hiçbir kural işleme alınmaz iken, sadece tespit et (DetectionOnly) değerinde ise kurallar işlenir, ancak trafikte hiçbir denetim yapılmaz. Yapılandırma dosyasındaki ilgili bölüm Resim 3.1'de gösterilmiştir.

```
# -- Rule engine initialization ----------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine DetectionOnly
```

Resim 3.1 : Modsecurity güvenlik kural ayarı satırı.

İlk kurulumda varsayılan değerler ayarlı olduğundan ModSecurity kullanılmadan önce yapılandırılmalıdır. ModSecurity'nin istek gövdelerine (POST verilerini içeren SecRequestBodyAccess) veya yanıtlara (ModSecurity'nin veri sızıntısını belirlemesini

sağlayacak SecResponseBodyAccess) erişim için yapılandırılması gerekir. Yapılandırma dosyasındaki ilgili bölüm Resim 3.2'de gösterilmiştir.

```
# -- Request body handling ------------------------------------------

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
```

Resim 3.2 : ModSecurity'nin gelen istekleri denetleme ayarı.

ModSecurity, sunucudan yapılan istekleri günlüğe kaydetmek için kullanılabilir. SecAuditEngine direktifi: açık, kapalı veya sadece ilgililer(relevantOnly) değerlerini alabilir. Sadece ilgili denetim loglarında hangi durum kodlarında kayıt alınacağı, SecAuditRelevantStatus'daki düzenli ifade ile belirtilir. Varsayılan olarak http durum kodlarından 404 hariç olarak 4xx veya 5xx durum kodlarını günlüğe kaydedilir. Yapılandırma dosyasındaki ilgili bölüm Resim 3.3'de gösterilmiştir.

```
# -- Audit log configuration ----------------------------------------

# Log the transactions that are marked by a rule, as well as those that
# trigger a server error (determined by a 5xx or 4xx, excluding 404,
# level response status codes).
#
SecAuditEngine RelevantOnly
SecAuditLogRelevantStatus "^(?:5|4(?!04))"
```

Resim 3.3 : Modsecurity'nin denetim log ayarı.

Düşürme(drop), engelleme(block) veya reddetme(deny) kuralıyla eşleşen trafik günlüğe kaydedilmektedir.

ModSecurity gömülü ve ters proxy olarak iki farklı şekilde kullanılabilmektedir. Her iki yöntemin de fayda ve mahsurları bulunmaktadır. Gömülü olarak kullanımda, web sunucusu ile aynı işletim sisteminde çalışmakta, topolojik veya mimaride değişiklik gerektirmemektedir. Bu yöntemin zayıf tarafı ise, web sunucusu ile aynı kaynakları kullanmasıdır.

Ters vekil sunucu olarak kullanımında ise, web sunucu ile istemci arasında konumlandırılarak, sunucuya gelen bütün istekleri karşılar ve filtreleyerek sunucuya iletir. Vekil sunucu olarak çalışığı için topolojik çalışırma gerektirir ancak ayrı güvenlik katmanı oluşturur. Aynı ağda bulunan birçok web uygulamasını aynı anda sunucuya özel yapılandırma ile koruyabilir. Ters proxy olarak kullanımındaki dez

avantajı ise, ayrı bir katman oluşturduğu için birden fazla proxy kullanarak yük dengelemesi yapılması gerekliliğidir.

Bu çalışmada ModSecurity Apache web sunucusu ile birlikte kullanılmıştır. Kural veya yapılandırma değişiklikleri metin düzenleyici ile yapılmaktadır. Yapılan değişiklik sonrasında komut satırında *apachectl configtest* komutu çalıştırılarak hata kontrolü yapılabilmektedir. Söz dizemi hatası yoksa Resim 3.4'deki bildirim, hata varsa Resim 3.5'deki uyarı mesajı çıkmakta ve hatanın nerede olduğu gösterilmektedir.

```
root@modseclab-1:/home/mrtalagoz# apachectl configtest
Syntax OK
```

Resim 3.4 : Yapılandırma kontrolü sonrası söz dizemi hatası olmadığı bildirimi.

```
root@modseclab-1:/home/mrtalagoz# apachectl configtest
AH00526: Syntax error on line 11 of /etc/apache2/sites-enabled/000-default.conf:
Rules error. File: /etc/apache2/modsecurity.d/rules/customrules.conf. Line: 8. Column: 56.
Invalid input:  @rx zeus" \\
Action 'configtest' failed.
The Apache error log may have more information.
```

Resim 3.5 : Yapılandırma kontrolü sonrası söz dizemi hatasının yeri ve uyarısı.

Kural kontrolü yapıldıktan sonra *systemctl restart apache2* komutu ile yapılan yeni değişiklikler devreye alınmalıdır. Yeniden başlatma sonrasında herhangi bir hata ile karşılaşılmadığında servis yeniden başlatılmakta ve hiçbir uyarı gösterilmemektedir. Yapılandırma, kural söz dizeminde veya yeniden başlatma esnasında hata bulunması durumunda Resim 3.6'da gösterilen hata mesajı gösterilmekte ve web sunucusu yeniden başlatma başarısız olduğu için erişim kesilmektedir.

```
root@modseclab-1:/home/mrtalagoz# systemctl restart apache2
Job for apache2.service failed because the control process exited with error code.
See "systemctl status apache2.service" and "journalctl -xe" for details.
```

Resim 3.6: Yeniden başlatma hata mesajı.

3.2.3 Kural söz dizimi ve örnek kural incelemesi

ModSecurity denetim mantığını ve politikalarını tam olarak istenilen şekilde uygulanmasını sağlayacak kural diline sahiptir. Kural dili başlangıçta çok basit görünebilir, ancak güçlü ve esnektir. Bu bölümde, kural dilinin tüm özelliklerine genel bakış açısı ile sunulacaktır.

Bir kuralın söz dizimi Resim 3.7'de görüldüğü gibi beş bölümden oluşmaktadır. Bütün parametreler ve açıklamalarının detayları Mosecurity el kitabında [47] bulunmaktadır.

Kural söz dizimi açıklanırken her bölümün sonunda OWASP-CRS'den alınan ve Resim 3.8'de verilen örnek kural açıklanmıştır.

Resim 3.7: ModSecurity kural söz dizemi.

```
SecRule ARGS "@rx
^(?i:file|ftps?|https?):\/\/(?:\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})" \
"id:931100,\
phase:2,\
block,\
capture,\
t:none,\
msg:'Possible Remote File Inclusion (RFI) Attack: URL Parameter using IP
Address',\
logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}:
%{MATCHED_VAR}',\
tag:'paranoia-level/1',\
tag:'OWASP_CRS',\
tag:'capec/1000/152/175/253',\
ctl:auditLogParts=+E,\
ver:'OWASP_CRS/3.3.0',\
severity:'CRITICAL',\
setvar:'tx.rfi_score=+%{tx.critical_anomaly_score}',\
setvar:'tx.anomaly score pl1=+%{tx.critical anomaly score}'"
```

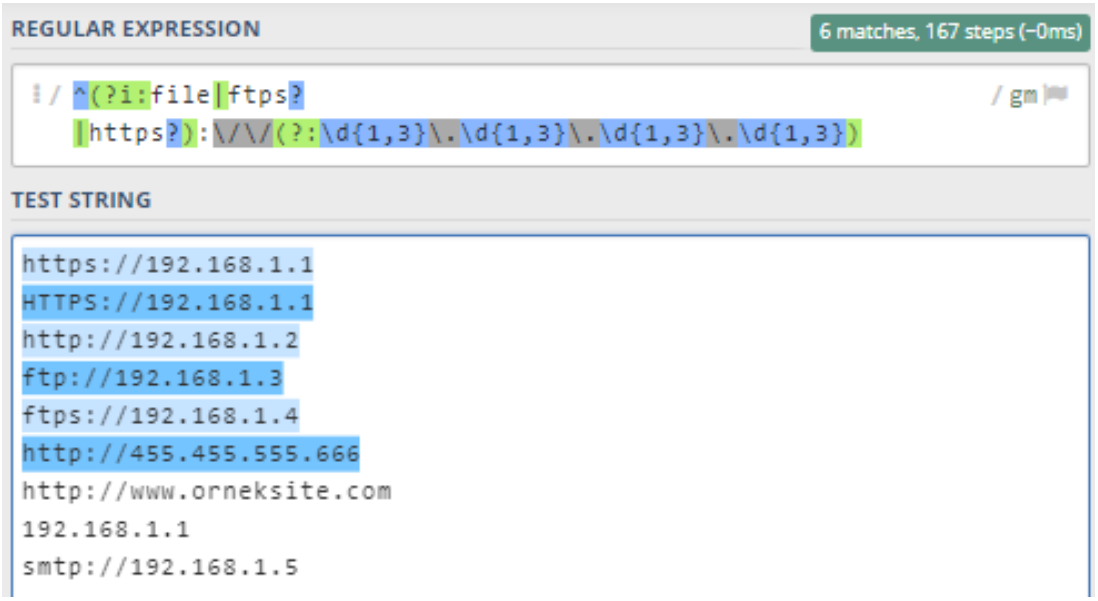
Resim 3.8 : Örnek kural [49].

Direktifler: İlk bölümde; kuralın o satıra ne ile ilgili olduğunu bildirecek direktif belirtilmelidir. 15 farklı direktif bulunmaktadır. Örnek kuralda "SecRule" direktifi kullanılmıştır. Bu kelime satırdaki kuralın güvenlik kuralı olduğunu belirtmektedir.

Değişkenler: ModSecurity'de değişkenler, HTTP isteğinin incelenecek kısımlarını tanımlamak için kullanılır. ModSecurity'nin temel özelliklerinden biri, ham verileri önceden işleyerek ve kuralların algılama mantığına odaklanmasını kolaylaştırmasıdır. Bir kuralda birden fazla değişken belirtilebilir. Toplam 121 farklı değişken bulunmaktadır. Tek bir kuralda birden fazla değişken kullanılabilir. Örnek kuralda http isteğinde argümanların incelenmesi için "ARGS" kelimesi kullanılmıştır. Boru (pipe-|) işareti ile biden fazla argümanın denetlenmesi sağlanabilmektedir.

Operatörler: 38 farklı operatör tanımlanabilmektedir. Bu operatörler kural satırındaki aranacak kelime, zaman ya da mantıksal kıyaslamaların tanımlanmasını

sağlamaktadır. Örnek kuralda "@rx" yani düzenli ifade operatörü çağrılmıştır. Düzenli ifade operatörü sayesinde çok geniş yelpazede, yüksek doğrulukta ve hızlı bir şekilde isteğin belirli yerlerinde istenen ifade aranabilmektedir. Düzenli ifade operatöründen sonra gelen ifade ise "file, ftp veya ftps, http veya https" kelimeleri ile başlayan bir ifade ve hemen arkasından "://" sonrasında ise bir IP notasyonu yani üç nokta ile ayrılmış 4 adet 1-999 arası rakam aranmaktadır. Üçüncü parti bir yazılım ile düzenli ifadenin kontrol edilmiş hali Resim 3.9'de verilmiştir. Mavi renk ile boyalı olan satırlar yazılan ifadeyle eşleşen satırlardır. Yazan ifade büyük ya da küçük harfli olup olmaması fark etmediği açıkça görülmektedir. Diğer yandan IP notasyonuna uymayan 6'ncı satır ise yine bu imzaya takılacaktır.



Resim 3.9 : Düzenli ifade açıklaması.

Dönüştürücü fonksiyonlar: Son bölümde dönüştürücü fonksiyonlar ve eylemler bir arada kullanılabilmektedir. Toplam 101 adet farklı seçenek bulunmaktadır. HTTP isteğinin hepsini küçük harfe çevirmek ve kötü karakterleri temizlemek gibi işlemler burada tanımlanmaktadır. Bütün kullanılabilecek dönüştürücü fonksiyonlar açıklamaları Modsecurity HandBook kitabında [47] detaylı bir şekilde anlatılmıştır. Örnekteki son bölümde "t:none" dizemi ile dönüştürücü fonksiyonun ara belleği temizlenmektedir.

Eylemler: Eylemler güvenlik duvarında kullanıldığı gibidir.

4. KULLANILABİLİRLİK DENETİMLERİ

Kullanılabilirliği göz önünde bulundurarak sistem, ürün ve hizmetleri geliştirme ve değerlendirmenin amacı; kullanıcıların kullanım esnasındaki davranışlarını dikkate alarak, memnun ve verimli bir şekilde belirlenen sistem, ürün veya hizmetler vasıtasıyla hedeflerine ulaşmalarını sağlamaktır [50]. İnsan-Bilgisayar Etkileşimi (HCI) ise bir bilgisayar programı veya teknolojisinin tasarımına ve tasarlanan bu progrmaın kullanıcı ile etkileşiminin nasıl gerçekleştirildiğine odaklanan bir çalışma alanıdır [51].

Kullanıcı deneyimi alanında, laboratuvar tabanlı kullanılabilirlik çalışmaları gibi denenmiş ve gerçek yöntemlerden, denetlenmemiş çevrimiçi değerlendirmelere kadar çok çeşitli araştırma yöntemleri bulunmaktadır [52]. Bu bölümde, ModSecurity'nin kullanılabilirlik değerlendirmesi sezgisel değerlendirme ve bilişsel gözden geçirme yöntemleriyle gerçekleştirilmiştir.

4.1 Sezgisel Değerlendirme Tekniği

4.1.1 Yöntem

Sezgisel değerlendirme, sezgisel çözümleme ölçütleri aracılığıyla kullanılabilirlik sorunlarını tespit edilebilmesi için kullanılan bir yöntemdir. Öğrenmesi diğer yöntemlere nazaran kolaydır ve kullanılabilirliğe etki eden ciddi sorunların tespit edilebilmesi mümkündür. Kullanıcılar sezgisel değerlendirme sürecinde kendilerine verilen basit görevleri kişisel deneyimlerine ve verilen sezgisel değerlendirme ölçütlerine uygun olarak, sistem arayüzünde serbest bir şekilde gezinerek yerine getirirler. Bu tekniği, değerlendirilen sistemin uzmanlarıyla yapabilir ancak kullanılabilirlik alanında uzman kişiler ile yapıldığında daha faydalı çıktılar elde edilmesi mümkündür [53].

Literatürde sezgisel değerlendirme ölçütlerinin sayısı üzerinde farklı görüşler bulunmaktadır [54] [55]. Bu çalışmada Nielsen tarafından önerilen on değerlendirme ölçütü esas alınmış ve Çizelge 4.1 sunulmuştur [54].

Çizelge 4.1 : Sezgisel değerlendirme ölçütleri [54].

Nu.	Sezgisel Değerlendirme Ölçütleri
1	Sistem durumunun görünürlüğü.
2	Sistem ve gerçek dünya arasındaki eşleşme bulunması
3	Kullanıcı kontrolü ve özgürlük sağlaması
4	Tutarlılık ve standartlar.
5	Hata önleme.
6	Hatırlama yerine tanıma.
7	Esneklik ve kullanım verimliliği.
8	Estetik ve minimalist tasarım.
9	Kullanıcıların hataları tanımasına, teşhis etmesine ve hatalardan kurtulmasına yardımcı olma.
10	Yardım ve belgeler.

Sistemin kullanılabilirliğinin değerlendirilmesi için üç ya da beş uzmanın yeterli olduğu Nielsen (1994) tarafından bildirilmektedir [52]. Bu yüzden ModSecurity web uygulama güvenlik duvarının sezgisel değerlendirmesine, dört güvenlik duvarı yöneticisi ve bilgi teknolojileri alanında çalışan bir kullanılabilirlik uzmanı katılmıştır. Her bir uzmana bulut üzerinde barındırılan web sunucusu üzerine kurulu bir web uygulamasına gömülü olarak yapılandırılmış ModSecurity web uygulama güvenlik duvarı tahsis edilerek SSH üzerinden erişim yetkisi verilmiştir. Uzmanlara sezgisel değerlendirme esnasında kullanması için; yapılandırma dosyalarının dizinleri, üç farklı senaryo, SSH protokolü üzerinden sunucuya erişebilmesi için yapılandırılmış terminal programı temin edilmiştir. Çalışmanın amacı ModSecurity'nin kullanılabilirliğinin değerlendirmesi olduğu için sisteme bağlanma, kural ve yapılandırma dosyalarının dizinleri hazırlanarak verilmiştir. Her uzman ile video konferans programı yardımıyla bağlanılarak kısa bir bilgilendirme yapılmış, sisteme uzaktan bağlantı sağladıkları teyit edilmiştir.

4.1.2 Bulgular

ModSecurity web uygulama güvenlik duvarı üzerinde icra edilen sezgisel değerlendirme sonucunda; S1'de kural girişinin söz dizemi kontrolü olduğu ancak anlamsal hata kontrolü bulunmadığı tespit edilmiştir. Böylelikle güvenlik kuralının söz dizemi doğru olsa bile zararlı isteğin sunucuya erişip erişmediği görülememektedir. S3'de hata yapıldığında herhangi bir geri alma mekanizması bulunmadığı tespit edilmiştir. Kural yazma işleminin çalışan bir sunucuda yapılmaması, test ortamında yazıldıktan ve kontrol edildikten sonra devreye alınması tavsiye edilmektedir. S5'de söz dizimi hatasından korumaya yönelik kontrol mekanizması vardır ancak hatayı önlemeye yönelik bir otomatik kontrol sağlamaktadır. Uzmanlar tarafından elde edilen diğer bulguların özeti Çizelge 4.2'de sunulmuştur.

Çizelge 4.2 : Sezgisel değerlendirme bulguları.

Madde	Elde Edilen Bulgular
S-1	Herhangi bir geri bildirim mekanizması yoktur. Söz dizimi hataları dışında semantik hatalara veya başarılı kural girişlerini onaylamaya yönelik geri bildirim yoktur. Sistem görünürlüğü zayıftır.
S-3	Sistem belirli dosyalar içerisine metin editörü yardımıyla yapılandırılmaktadır. Hata anında aynı dosyaya girip hatalı metni silmek dışında hatanın geri alınmasına yönelik bir mekanizma yoktur.
S-5	Hata önleme mekanizması olmadığı gibi söz dizimi hatalarında web sunucusunun devre dışı kalması gibi hatanın etkisini üstel arttıracak bir işleyiş bulunmaktadır.
S-6	Tanımaya yönelik hiçbir talimat bulunmamaktadır. Tüm dosya yolları, kural formatları anımsanmalı, anımsanmadığı takdirde sistemin dışındaki kaynaklardan (web siteleri, kitap vb. dokümanlar) faydalanılmalıdır.
S-7	Sistem doğrudan Linux işletim sisteminin sağladığı komut satırından yönetilmekte, herhangi bir hızlandırıcı, kısa yol, gelişmiş komut satırı vb. uygulamalar bulunmamaktadır.
S-8	Sistem herhangi bir diyalog, bilgilendirme içermemektedir.
S-9	Sadece söz dizimi hatalarında kullanıcıya ikaz bildirilmekte ve ikazda satır numarası verilerek satırın gözden geçirilmesi istenmektedir. Hatanın ne olduğu net ifade edilmemektir.
S-10	Sisteme dair bir yardım menüsü, işletme kılavuzu desteği bulunmamaktadır.

4.2 Bilişsel Gözden Geçirme Tekniği

Bilişsel gözden geçirme, yeni kullanıcının hedef sistemdeki görevleri gerçekleştirirken karşılaşabileceği kullanılabilirlik sorunlarını tespit edebilmek için tasarlanan bir değerlendirme yöntemidir. En büyük yararı hızlı ve maliyeti etkin bir şekilde tamamlanabilmesidir [57] [60].

4.2.1 Yöntem

Bu yöntemde öncelikle kullanıcıdan yerine getirmesi beklenen ana görev seçilir. Bu görevin gerçekleştirilmesi için gereken alt adımlar aşamalandırılarak her birisinde belirlenen dört soruya [58] cevap aranır. Verilen cevaplara göre tavsiye iyileştirmeler yazılır, araştırmacılardan sonuçlar toplanır. Çalışmada kullanılan görevler Çizelge 4.3'te, bilişsel değerlendirme soruları Çizelge 4.4'te sunulmuştur.

Çizelge 4.3 : Bilişsel gözden geçirme tekniğinde kullanılan görevler.

Görev	Açıklama
G-1	ModSecurity çalışır hale getirme.
G-2	Kural ekleme
G-3	Bir kuralı hariç tutma

Çizelge 4.4 : Bilişsel gözden geçirme tekniğindeki sorular [58].

Madde	Açıklama
1	Kullanıcılar doğru süreci üretmeye mi çalışıyor? Kullanıcı hedefine ulaşmak için bu adımın gerekli olduğunu biliyor mu?
2	Kullanıcılar sonucu elde etmek için yapmaları gereken doğru eylemi görüyorlar mı?
3	Kullanıcılar eylemlerinin doğru sonucu elde etmek için yapmaları gereken eylem olduğunun farkında mı? (Yoksa bunun yerine farklı bir eylem mi seçiyor?)
4	Kullanıcılar geri bildirimi anlayabiliyor mu? Kullanıcılar doğru eylemi gerçekleştirdiyse, amaçlanan sonuçlarına doğru ilerleme kaydettiklerini söyleyebiliyor mu?

4.2.2 Bulgular

Yapılan çalışma ile bilişsel gözden geçirme neticesinde, kullanıcıların yapılandırma ve elde edilen özet bulgular Çizelge 4.5'te sunulmuştur.

Çizelge 4.5 : Bilişsel gözden geçirme sonucunde elde edilen bulgular.

Madde	Elde Edilen Bulgular
B-1	Kullanıcılar görevleri yerine getirmek için hangi dosya yolundaki yapılandırma dosyasını açması gerektiğini bilmiyor.
B-2	"Doğru eyleme yönelik herhangi bir yönlendirme, yardım sekmesi, ipucu veya örnek eylem yok.
B-4	Söz dizimi hatalarında bildirilen hata kodu dışında semantik hatalara veya eylemin doğru gerçekleştirildiğini onaylamaya yönelik bir geribildirim mekanizması yok.

4.3 İyileştirmeler ile Kural Kontrol ve Geri Besleme Mekanizmasının Tasarımı

Sezgisel değerlendirme ve bilişsel gözden geçirme teknikleri neticesinde elde edilen bulgular incelendiğinde sorun sahası iki farklı bölüme ayrılmıştır. Bu düşüncenin ana sebebi kullanıcı çalışması sonuçlarını mümkün olduğunca ölçülebilir hale getirmektir. Bu maksatla S-6, B-1 ve B-2 maddelerinde elde edilen bulguları sabit tutabilmek için iyileştirmeler hazırlanmış, S-1, S-5, S-8, S-9 ve B-4 maddelerini kapsayacak şekilde iyileştirme tasarlanarak kullanıcı çalışması ile sonuçlarının ölçümlenecektir.

4.3.1 Kullanıcı çalışmasında sabit tutulacak iyileştirmeler

Sistem yöneticileri ilk defa sisteme bağlandıklarında varsayılan ayarların hangi dizinde olduğu, sistemde hangi programların çalıştığı, hangi servislerin hizmet verdiğini anlaması için biraz zamana ihtiyacı olduklarını tespit edilmiştir. Bu süre sistem yöneticisinin tecrübesi ile ters orantılı olduğu için kullanıcı çalışması öncesinde bu sorunun öncelikle çözülmesi gerektiğini planlanmıştır. Bunun için ModSecurity'in kurulduğu dizin, kural ve yapılandırma dosyalarının dizinleri, kural girmek için gerekli metin editörünün komutu, örnek kural ve kural formatı eğitim dkümanı hazırlanarak eklenmiştir. Ayrıca bütün kural parametreleri sınıflandırılarak tablo haline getirilmiştir.

Kullanıcıların bulut sistemdeki sunuculara bağlanırken uğraşacağı emek ve çabayı en aza indirmek, ancak aynı zamanda gerekli güvenlik tedbirlerini de alarak, çalışmanın sağlıklı sonuçlar vermesini temin etmek için açık anahtar yöntemi ile yetkilendirmenin yapılacağı üçüncü parti terminal programı çalışmaları yapılmıştır.

4.3.2 Kural kontrol ve geri besleme mekanizmasının tasarımı

S-1, S-5, S-8, S-9 ve B-4 maddeleri kullanılabilirliğini yüksek seviyede etkilediğinden ve sistemin servis dışı kalmasını sağlayarak erişilebilirliği etkileyebileceğinden dolayı bir kural kontrol ve geri besleme mekanizması tasarlanmıştır. Diğer maddeler tezin kapsamına girmediği değerlendirildiğinden incelenmeyerek kapsam dışı bırakılmıştır.

Güvenlik duvarına yeni kural girilmesi sonucunda söz dizimi ve semantik hataları kullanıcıya geribildirim ile bildirecek, söz dizimi hatalarında web sunucusunun devre dışı kalmasını önleyecek ve doğru girilen kural sonucunda kullanıcıya başarılı kural girişi geribildirimi sunacak bir betik hazırlanmış, akış diyagramı Şekil 4.1'de gösterilmiştir.

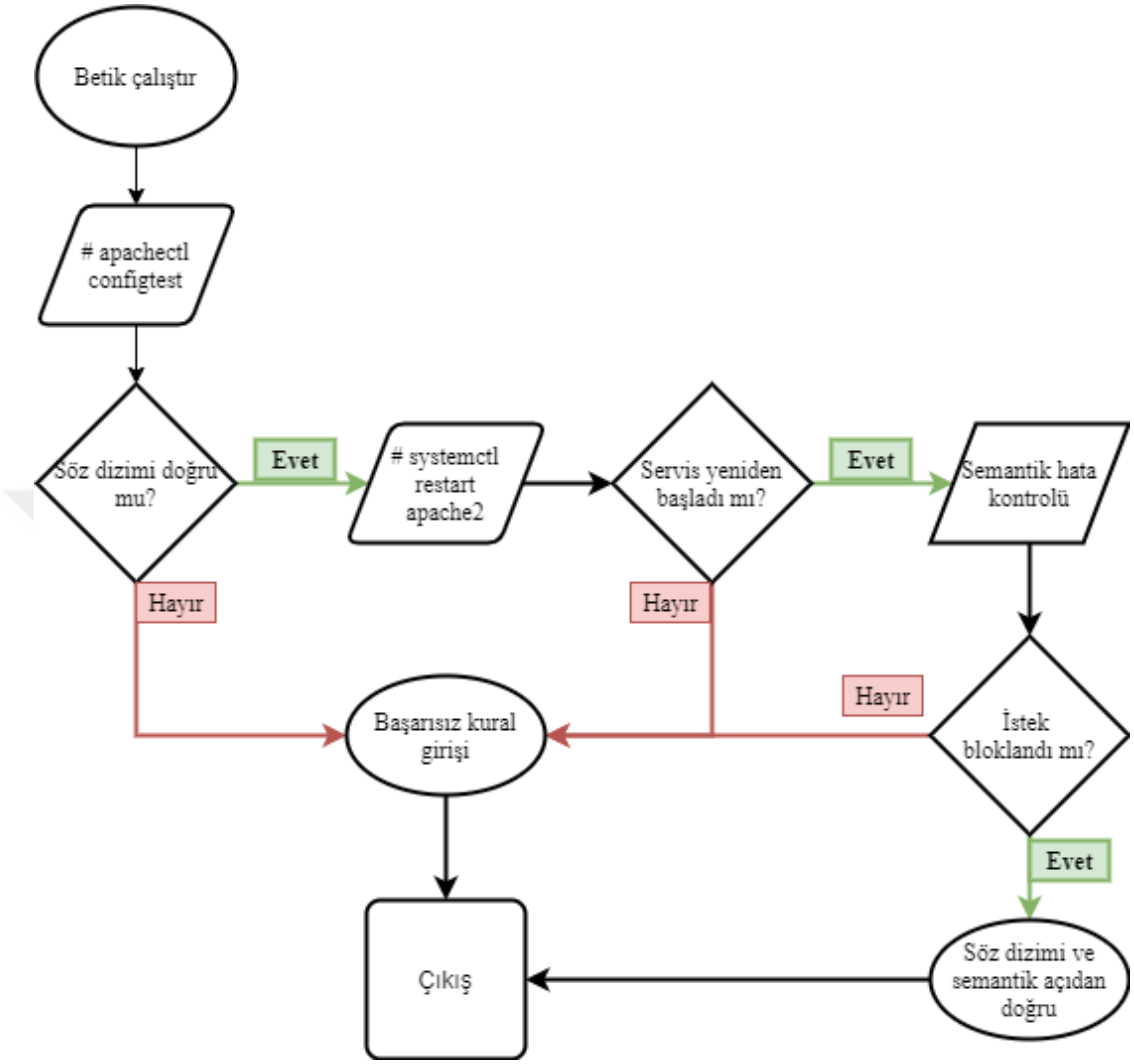
Betik çalıştırıldığında, öncelikle yapılandırma dosyalarında sözdizimi kontrolü yapılmaktadır. Sözdizimi hatası tespit edildiği takdirde kullanıcıya Resim 4.1'deki uyarı mesajı gösterilmekte, kuralın derlenmesi önlenerek web sunucunun hizmet dışı kalmadan erişilebilirliğini sürdürmesi sağlanarak, kullanıcıya yazdığı kuralı kontrol etmesi gerektiğini bildiren geri bildirim gösterilmektedir.

```
root@modseclab-1:/home/tester1# /etc/check/1.sh
AH00526: Syntax error on line 11 of /etc/apache2/sites-enabled
/000-default.conf:
Rules error. File: /etc/apache2/modsecurity.d/modsecurity.conf
. Line: 8. Column: 27. Invalid input:  DetectionOnl
Action 'configtest' failed.
The Apache error log may have more information.
ÖZÜR DiLERiM,KURALINIZI KONTROL EDiNiZ, HEMEN ÜSTTE AYRINTILI
AÇIKLAMA BULUNMAKTADIR.
root@modseclab-1:/home/tester1#
```

Resim 4.1 : Kural kontrol ve geri besleme mekanizması söz dizimi hata uyarısı.

Sözdizimi kontrolünde hata bulunmaması durumunda yazılan kural devreye alınarak semantik hata kontrolüne geçilmektedir. Bu safhada güvenlik duvarına girilen kuralın engellemesi gereken zararlı istek web sunucusuna gönderilmekte, sunucudan dönen cevaba göre kuralın isteği engelleyip engellemediği tespit edilmektedir. İstek

engellenmediyse kuralda semantik hata olduğu değerlendirilerek kullanıcıya Resim 4.2 sunulan başarısız kural geribildirimi gösterilmektedir.

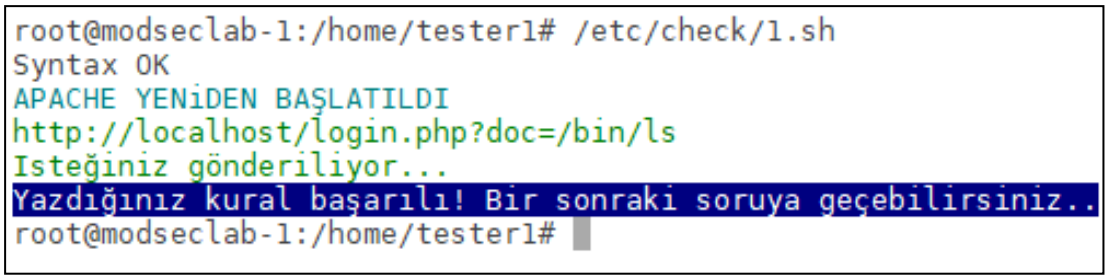


Şekil 4.1 : Kural kontrol ve geri besleme mekanizması akış diyagramı.

```
root@modseclab-1:/home/tester1# /etc/check/1.sh
Syntax OK
APACHE YENiDEN BAŞLATILDI
http://localhost/login.php?doc=/bin/ls
Isteğiniz gönderiliyor...
Kural eksik veya hatalı..Kuralı kontrol ediniz...
root@modseclab-1:/home/tester1#
```

Resim 4.2 : Kural kontrol ve geri besleme mekanizması semantik hata uyarısı.

İstek WAF tarafından engellenirse kuralda semantik hata olmadığı değerlendirilerek kullanıcıya Resim 4.3'de sunulan başarılı kural geribildirimi gösterilmektedir.



Resim 4.3 : Kural kontrol ve geri besleme mekanizmasının doğru kural bildirimi.

Kural-kontrol ve geri bildirim mekanizmasının içerisinde bulunan ve kuralın semantik kontrolü için zararlı web isteği gönderen betikte sadece kullanıcı testine dâhil edilmiş görevlere yönelik zararlı istekler kullanılmaktadır.

Kumar ve Lim (2019) Mirai benzeri botnet saldırılarında ele geçirilen nesnelerin ürettiği paketleri iki boyutlu (zaman-cihazdan cihaza iletişim) olarak inceleyerek botnetler tarafından ele geçirilen nesnelere dair imzalar tespit etmiş ve bir bot tespit algoritması geliştirmiştir. Bu algoritma ile ağ trafiğini izleyerek botnet tespiti yapabilecek sentinel (nöbetçi) cihazların ağlara kurulmasını önermişlerdir [41].

Shafi ve Basit (2019) yazılım tabanlı ağ mimarisi ile farklı alt ağları birbirine bağlamayı ve her bir alt ağda blok zincir kullanarak yönetici tarafından belirlenen güvenlik kurallarının doğrulanarak nesnelere iletilmesini ve bu sayede dağıtık hizmet dışı bırakma saldırıları düzenleyen botnetlere yeni cihazların katılımının önlenmesini önermiştir [42].

5. KULLANICI ÇALIŞMASI

Kullanıcı çalışması yapabilmek maksadıyla, TOBB Ekonomi ve Teknoloji Üniversitesi İnsan Araştırmaları Değerlendirme Kuruluna başvuru yapılmıştır. 11 Ağustos 2020 tarihli ve Sayı: 27393295-100 numaralı yazı ile "Açık Kaynak Web Uygulama Güvenlik Duvarı (ModSecurity)'nin Kullanılabilirlik Araştırması" etik yönden uygun görülerek, izin verilmiştir. Kullanıcı çalışması, TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği öğrencilerinin pandemi şartları sebebiyle uzaktan katılımı ile gerçekleştirilmiştir. Katılım öncesinde öğrencilerin dört farklı güne planlaması yapılmış, konferans bağlantıları e-posta adreslerine gönderilmiştir.

5.1 Çalışma Senaryosu

Çalışmada giriş anketi, eğitim, ModSecurity kullanıcı testi, çıkış anketi ve veri analizi olarak toplam beş aşama bulunmaktadır.

5.1.1 Giriş anketi

Açık kaynak bir anket uygulaması üzerinde yayınlanan giriş anketine katılımcılar davet edilmiştir. Açık rıza alınarak yaş, cinsiyeti, eğitim durumu gibi demografik veriler toplanmış ve siber güvenlik alanında deneyim tespit etmeye yönelik sorular sorulmuştur. Anket soruları Ek-1'dedir.

5.1.2 Eğitim

Yapılan ön çalışma neticesinde elde edilen bulgular ışığında, katılımcılara web uygulama güvenlik duvarının çalışma prensibi, ModSecurity'nin sözdizimi, aktif hale getirilmesi ve kural yazımı ile ilgili 45 dakikalık eğitim videosu hazırlanarak açık kaynak video paylaşım sitesinde yayınlanmıştır. Hazırlanan video içeriğinin testte icra edilecek görevleri de kapsaması sağlanarak kullanıcıların bilgi eksiği bulunmadan teste katılımı garanti altına alınmıştır.

Kullanıcıların buluttaki sunuculara uzaktan güvenli bağlantı kurmalarını sağlamak maksadıyla terminal programı hazırlanmıştır. Kullanıcılara test öncesinde bir makine tahsis edilerek kullanıcı adı verilmiş ve terminal bağlantısı yapması sağlanmıştır. Terminal programını bilmeyen kullanıcılar için eğitim videosu ve eğitim materyali hazırlanarak kullanıcıların kullanımına açılmıştır.

Sezgisel değerlendirme ve bilişsel gözden geçirme sonrasında elde edilen bulgulardan, S-6, B-1 ve B-2 maddelerine uygun olarak, yapılandırma dosyalarının dizinleri, kuralların bulunduğu dosyalar, kural kontrol ve geri besleme betiğinin nasıl çalıştırılması gerektiği, Linux sunucu kullaımıyla ilgili temel hatırlatmalar, kural formatı ve bir adet örnek kural verilmiştir. Dağıtılan belge Ek-2'tedir.

5.1.3 Görevler

Görevler hazırlanırken birinci bölümde yapılan çalışmadan istifade edilerek, sistem yöneticisinin karşılaşabileceği durumlara uygun görevler hazırlanmıştır. Durumların daha kolay anlaşılabilmesi için günlük hayatta karşılaşılabilecek bir senaryo çizilmiştir. Toplam beş kural belirlenmiştir, soruların detaylı hali Ek-3'tedir.

Çizelge 5.1 : Kullanıcı çalışmasında verilen görevler.

Görev	Açıklama
1	Konfigurasyon dosyasında gerekli değişiklikleri yaparak ModSecurity'i aktif hale getirme.
2	User Agent'ı "Zeus" olan bütün istemcileri engelleme.
3	Dizin aşımı zafiyetinin sanal yamasını yapma.
4	"ShellShock" zafiyetinin sanal yamasını yapma.
5	Belirlenen bir kurala exception tanımlama.

5.1.4 Kullanıcı çalışması

Kullanıcı çalışması için telekonferans yöntemi ile katılan kullanıcılara öncelikle çalışmanın amacı ve kapsamı açıklanmıştır. Daha sonra giriş ve çıkış anketi, eğitim videosu linkleri, terminal programı, verilen görevler ve kural yazmak için gerekli olabilecek tüm parametrelerin bulunduğu bir eğitim dökümanı kullanıcılar ile paylaşılmıştır.

Tüm kullanıcıların giriş anketini doldurmasını müteakip, eğitim videolarını izlemeleri istenmiştir. Eğitim videosu biten kullanıcılar uzak sunucuya bağlanarak beş adet görevi yapmışlardır. Görevleri biten kullanıcılar terminal çıktıları alınarak, çıkış anketine yönlendirilmiş sonrasında çalışma bitirilmiştir.

5.1.5 Çıkış anketi

Açık kaynak bir anket uygulaması üzerinde yayınlanan çıkış anketine katılımcılar davet edilmiştir. Açık rıza alınarak katılımcıların ModSecurity ve gerçekleştirilen kullanıcı testi hakkındaki algı ve düşüncelerini tespit etmek amacıyla çeşitli tip ve ölçekte sorular sorulmuştur. Çıkış Anketi soruları Ek-4'tedir.

Katılımcılara ModSecurity kullanıcı testi öncesinde görev adımlarını ve kural girişi esnasında bilinmesi gereken sözdizimi kelimelerini içeren eğitim materyali dağıtılmıştır. Test grubuna dağıtılan dokümana kontrol grubuna dağıtılandan farklı olarak "kural-kontrol ve geribildirim mekanizmasının kullanımı" hakkında ilave bilgi verilmiştir.

5.2 Teknik Altyapı

Katılımcılara ModSecuirty3 ve zafiyetli web sunucu uygulaması olan DVWA kurulu Ubuntu 18.04 Linux dağıtımlı sanal makineler bulut ortamında hazırlanarak tahsis edilmiştir. Sistemin yönetim paneli Resim 5.1'degösterilmiştir. Katılımcıların SSH protokolü ile uzaktan güvenli erişimi için bulut sistemde gerekli yapılandırılmalar yapılmıştır.

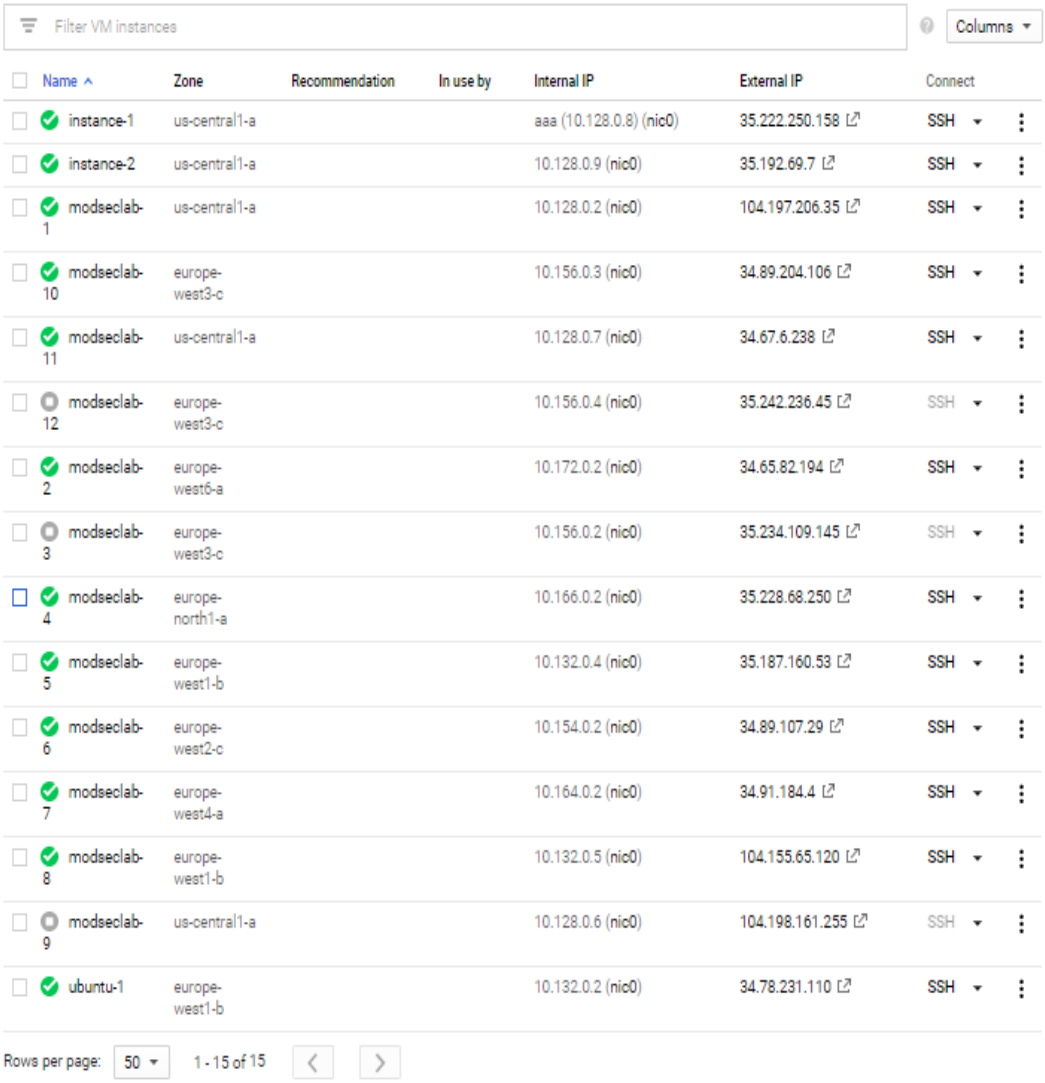
Katılımcılar kendilerine dağıtılan dokümanlardan faydalanarak görevleri yerine getirmiş, talep ettikleri takdirde video konferans uygulaması üzerinden bağlantı sorunları ile ilgili teknik destek sağlanmıştır. Hiçbir şekilde kullanıcılara kural girmesi ile ilgili yardım sağlanmamıştır. Görevleri tamamladığını beyan eden katılımcıların günlük (log) dosyaları veri analizinde kullanılmak üzere e-posta yolu ile toplanmıştır.

5.3 Veri Analiz Yöntemi

Kontrol grubu ve test grubu arasında başarılan görev sayıları, çalışma süresince web sunucuyu devre dışı bırakma süreleri ve görevleri tamamlama (sistemde kalış) süreleri arasında farklılık olup olmadığını tespit etmek amacıyla; verinin dağılımına ve

niteliğine uygun olarak %95 güven düzeyinde istatistiksel testler gerçekleştirilmiştir. Tüm analizler IBM SPSS Statistics V.22 yazılımı ile gerçekleştirilmiştir.

Resim 5.1 : ModSecurity sanal makineler yönetim paneli.



Katılımcıların grup bazında; doğru başardıkları görev sayıları normal dağılıma uymadığından Mann-Whitney U testi ile web sunucuyu hizmet dışı bırakma süreleri normal dağılıma uymadığından Mann-Whitney U testi ile analiz edilmiştir.

5.4 Katılımcı Profili

Kullanıcı testine TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği bölümü öğrencilerinden 26 kişi katılmıştır. Katılımcılara ait demografik dağılım Çizelge 5.2'de, katılımcıların siber güvenlik deneyimlerine dair bulgular Çizelge 5.3'de sunulmuştur.

Çizelge 5.2 : Katılımcıların demografik dağılımı.

Değişken	**Kategori**	**f**	**%**
Cinsiyet	Erkek	15	57,69
	Kadın	9	34,61
	Belirtmeyen	2	7,69
Yaş	18-20	2	7,69
	21-23	21	80,76
	24-26	3	11,53
Eğitim durumu	Lisans Öğrenci	26	100
İş durumu	Çalışıyor	3	11,53
	Çalışmıyor	23	88,47

Çizelge 5.3 : Katılımcıların siber güvenlik alanında deneyimleri.

Değişken	**Kategori**	**f**	**%**
Siber güvenlik bilgi seviyeniz konusunda kendinizi değerlerlendiriniz. (1 tecrübem yok – 5 çok tecrübeliyim)	1	12	46,15
	2	13	50
	3	1	3,85
	4	0	0
	5	0	0
Daha önce herhangi bir ağ güvenlik cihazı yönettiniz mi? Süre belirtiniz.	Yönetmedim	23	88,47
	0-1 yıl	2	7,69
	1-3 yıl	1	3,85
Aşağıdaki siber güvenlik ile ilgili konulardan aşina olduklarınızı işaretleyiniz.	Sızma testi	4	15,38
	Kriptoloji	7	26,92
	Ağ güvenliği	7	26,92
	İşl. Sistemi güvenliği	3	11,53
	Zararlı yazılım analizi	1	3,85
	Hiçbiri	12	46,15

5.5 Bulgular

ModSecurity web uygulama güvenlik duvarını "kural-kontrol ve geribildirim mekanizması" ile kullanan test grubu ve güvenlik duvarını değişiklik yapılmadan kullanan kontrol grubu arasında görevlerin başarılması, web sunucuyu hizmet dışı bırakma süresi ve görevleri tamamlama süresi bakımından anlamlı düzeyde fark olup olmadığına yönelik icra edilen istatistiksel analizlerin sonuçları aşağıda açıklanmıştır. Analizlere ilave olarak tespit edilmiş bir kısım bulgular da paylaşılmıştır. İstatistiksel analizlerin özeti Çizelge 5.4'te sunulmuştur.

Görev Başarı Durumu: Grup değişkenine göre başarılan görev sayılarına dair veriler normal dağılıma uymadığından Mann Whitney U testi ile analiz edilmiştir. Test grubundaki katılımcıların ortalama 4,53 görevi, kontrol grubundaki kullanıcıların ortalama 3,61 görevi doğru tamamladığı ve %95 güven düzeyinde $p=0,01<0.05$ istatistiksel açıdan iki grubun doğru tamamlanan görev sayıları arasında anlamlı fark bulunduğu gözlemlenmiştir.

Web Sunucu Hizmet Dışı Bırakma Süresi: Grup değişkenine göre kullanıcı testinde hatalı kural girişi nedeniyle web sunucuyu devre dışı bırakma süreleri normal dağılıma uymadığından Mann Whitney U testi ile analiz edilmiştir. Test grubundaki katılımcıların ortalama 2 dakika, kontrol grubundaki kullanıcıların ortalama 17 dakika web sunucusunu devre dışı bıraktığı ve %95 güven düzeyinde $p=0,01<0.05$ istatistiksel açıdan iki grubun web sunucuyu hizmet dışı bırakma süreleri arasında anlamlı fark bulunduğu gözlemlenmiştir.

Görevleri Gerçekleştirme Süresi: Grup değişkenine göre kullanıcıların sistemde toplam kalış (görevleri tamamlama) süreleri normal dağılıma uyduğundan bağımsız t testi ile analiz edilmiştir. Test grubundaki katılımcıların ortalama 52 dakika 20 saniye, kontrol grubundaki kullanıcıların ortalama 51 dakika 25 saniyede görevleri tamamladığı ve %95 güven düzeyinde $p=0,873>0.05$ istatistiksel açıdan iki grubun sistemde toplam kalış (görevleri tamamlama) süreleri arasında anlamlı fark bulunmadığı gözlemlenmiştir.

Diğer Bulgular: Çıkış anketinde katılımcılara ModSecurity'nin genel kullanımına ve gerçekleştirilen görevlere yönelik LIKERT ölçekli "zorluk seviyesi nedir (1 en düşük-5 en yüksek)" soruları sorulmuş ve yanıtlar analiz edilmiştir. Zorluk seviyesi ile ilgili sorulara test grubunun yanıt ortalaması 1.79, kontrol grubunun yanıt ortalaması

1.96'dır. Verilerin normal dağılıma uyması nedeniyle %95 güven aralığında bağımsız t testi icra edilmiş ve p= 0,423>0,05 yanıtlar arasında istatistiksel açıdan anlamlı fark bulunmadığı tespit edilmiştir.

Çizelge 5.4 : İstatiksel analizlerin özeti.

Analiz Konusu	**Test (Ort.)**	**Kontrol (Ort.)**	**Test Tipi**	**p**	**Sonuç**
Doğru Tamamlanan Görev Sayısı	4,53	3,61	Mann-Whitney U	0,01	Anlamlı fark var.
Web Sunucuyu Hizmet Dışı Bırakma Süresi (dk:sn)	02:00	17:00	Mann-Whitney U	0,01	Anlamlı fark var.
Görevleri Tamamlama Süresi (dk:sn)	52:20	51:25	Bağımsız t testi	0,873	Anlamlı fark yok.

Giriş anketinde katılımcılardan Linux işletim sistemi kullanım deneyim süreleri sorulmuş ve diğer bulgular ile kıyaslanmıştır. Test grubunda yer alan ve daha önce Linux işletim sistemi kullanmadığını beyan eden iki kullanıcının tüm görevlerde başarılı olduğu; kontrol grubunda yer alan ve daha önce Linux işletim sistemi kullandığını beyan eden iki katılımcının tüm görevlerde başarısız olduğu tespit edilmiştir.

Çıkış anketinin sonunda katılımcılardan açık uçlu olarak alınan ModSecurity geliştirme önerilerinde kontrol grubundaki katılımcıların ModSecurity tarafından üretilen söz dizimi hata mesajlarında hatanın yerini gördüklerini ancak tam olarak neyin hata olduğunun ifade edilmediğini, hata bildiriminin geliştirilmeye ihtiyaç duyulduğunu beyan ettikleri görülmüştür.

6. SONUÇ VE ÖNERİLER

Çalışmanın ilk bölümünde gerçekleştirilen kullanılabilirlik denetimi neticesinde ModSecurity üzerinde tespit edilen kusurların komut satırından yönetilen diğer uygulamalarla benzer nitelikte olduğu değerlendirilmektedir. Bilişsel gözden geçirme safhasında dört sorudan üçünde, sezgisel değerlendirme neticesinde toplam on ölçütten sekizinde iyileştirme yapılması gerektiği tespit edilmiştir. Çalışma kapsamında bilişsel gözden geçirme B4 maddesi, sezgisel değerlendirme S1, S5, S8 ve S9 ölçütlerinde iyileştirme yapılması uygun görülerek bir kural-kontrol ve geribildirim mekanizması hazırlanmıştır ve kullanıcı testine tabi tutulmuştur.

İyileştirmenin güvenlik duvarı ile kullanımında iki temel amacın gerçekleştirilmesine dikkat edilmiştir;

1. Kullanıcıların güvenlik duvarına doğru kural girme ve doğru yapılandırma davranışını pekiştirme,

2. Kullanıcıların girdikleri kurallardaki sözdizimi hataları sebebiyle web sunucunun devre dışı kalma süresini azaltması.

Kullanıcı testi sonrasında yapılan istatistiksel test sonuçları bu iki amacın gerçekleştirildiğini destekler niteliktedir.

Kural-kontrol ve geri besleme mekanizmasını kullanmayan kontrol grubunda başarılan görev sayılarının ortalaması 3.61 iken bu sayı mekanizmayı kullanan test grubunda 4,53'tir ve başarılı görev sayılarında yaklaşık %25 artış sağlanmıştır.

Güvenlik duvarının kullanımında kontrol grubunun ortalama 17 dakika süreyle web sunucuyu hizmet dışı bıraktığı tespit edilirken bu süre test grubu için 2 dakikadır ve web sunucusunun hizmet dışı kalma süresi yaklaşık %88 azalmıştır.

Kullanıcıların ModSecurity'nin kullanım zorluğuna yönelik değerlendirmeleri kıyaslandığında kontrol grubu 5 üzerinden 1.96 genel ortalama ile zorluk seviyesini bildirirken test grubunda bu rakam 1.79'dur. Bu durum iyileştirmenin kullanıcılara

ilave bir işlem yükü getirmediği ve güvenlik duvarının kullanımını zorlaştırmadığı sonucunu destekler niteliktedir.

Test grubundaki Linux bilmeyen kullanıcıların görevlerin tamamını başarması; üretilen eğitim materyallerinin ve yapılan iyileştirmenin deneyimsiz kullanıcıların doğru bir şekilde güvenlik duvarını yapılandırmasına destek olduğunu ispatlar niteliktedir.

Çalışma kapsamında ModSecurity'nin uzman kişiler tarafından yapılan kullanılabilirlik değerlendirmesi safhasındaki bulguların çıkış anketinde katılımcılardan toplanan ModSecurity geliştirme önerileri ile benzer içerikte olması kullanılabilirlik değerlendirmesi ve kullanıcı testi safhalarının birbiri ile çelişmeyen, tutarlı, gerçek hayata yakın ve birbirini destekler bir metodolojide uygulandığı sonucunu desteklemektedir.

Bu çalışmada ModSecurity web uygulama güvenlik duvarı kullanılabilirlik değerlendirmesine tabi tutularak kullanılabilirlik kusurları ortaya çıkarılmış, güvenlik duvarına girilen kuralları kontrol edecek ve kullanıcıya geribildirim sağlayacak bir iyileştirme önerisi sunulmuştur. Söz konusu iyileştirme bir kullanıcı testi yardımıyla değerlendirilmiştir. İyileştirmenin sistemin kullanılabilirlik seviyesini arttırdığı, güvenlik duvarı üzerinde doğru kural girme oranını arttırdığı, kullanıcı hataları nedeniyle web sunucusunun hizmet dışı kalma süresini azalttığı tespit edilmiştir.

Artan siber tehditlerle birlikte gelecekte de web sunucularının güvenliğini sağlamak amacıyla ModSecurity web uygulama güvenlik duvarının kullanılmaya artarak devam edeceği öngörülmektedir. Bu nedenle ModSecurity üzerinde kullanılabilirlik çalışmalarının devam etmesinin web uygulama güvenliği alanında büyük yarar sağlayacağı değerlendirilmektedir. Yardım menüsü ve işletme kılavuzu eksikliği, yetersiz bilgilendirme mesajları, hızlandırıcı kısa yol veya özgün gelişmiş komut satırı bulunmayışı ile çalışma kapsamına alınmayan diğer kullanılabilirlik kusurları üzerinde yapılacak iyileştirmelerle ModSecurity'nin kullanılabilirlik düzeyine anlamlı katkılar sağlanabileceği, böylece yanlış yapılandırmanın azaltılarak uygulama güvenliğinin artırılabileceği değerlendirilmektedir.

KAYNAKLAR

[1] "World Internet Users Statistics and 2020 World Population Stats." https://www.internetworldstats.com/stats.htm Alındığı tarih: 09.11.2020.

[2] **Clincy V.** and **Shahriar H.,** (2018). "Web Application Firewall: Network Security Models and Configuration," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 835–836, doi: 10.1109/COMPSAC.2018.00144.

[3] **Razzaq A., Hur A., Shahbaz S., Masood M.**, and **Ahmad H. F.**,(2013) "Critical analysis on web application firewall solutions," pp. 1–6, 2013, doi: 10.1109/isads.2013.6513431.

[4] "(No Title)." https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf Alındığı tarih: 09.11.2020.

[5] **Prandl S., Lazarescu M.**, and **Pham D. S.** , (2015). "A study of web application firewall solutions," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9478, pp. 501–510, doi: 10.1007/978-3-319-26961-0_29.

[6] **D. Botta** *et al.*,(2007). "Towards understanding IT security professionals and their tools," *ACM Int. Conf. Proceeding Ser.*, vol. 229, pp. 100–111, , doi: 10.1145/1280680.1280693.

[7] **Nag A. K., Dasgupta D.,** and **Deb K.**, (2014). "An Adaptive Approach for Active Multi-Factor Authentication," *9th Annu. Symp. Inf. Assur.*, pp. 39–47, [Online]. http://www.albany.edu/iasymposium/proceedings/2014/ASIA14Proceedings.pdf#page=49. Alındığı tarih: 09.11.2020.

[8] **Takayama L.** and **Kandogan E.**,(2006). "Trust as an Underlying Factor of System Administrator Interface Choice." In CHI'06 extended abstracts on Human factors in computing systems (pp. 1391-1396).

[9] **Voronkov A., Iwaya L. H., Martucci L. A.,** and **Lindskog S.**,(2017). "Systematic literature review on usability of firewall configuration," *ACM Comput. Surv.*, vol. 50, no. 6, doi: 10.1145/3130876.

[10] "OWASP Top 10-2017," 2003. [Online]. Available: https://github.com/OWASP/Top10/issues. Alındığı tarih: 09.11.2020

[11] **Srokosz M., Rusinek D.,** and **Ksiezopolski B.**, "A new WAF-based architecture for protecting web applications against CSRF attacks in malicious environment," (2018). *Proc. 2018 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2018*, vol. 15, pp. 391–395, doi: 10.15439/2018F208.

[12] **Clincy V. , Shahriar H.**, (2018). "Web Application Firewall: Network Security Models and Configuration," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 835–836, doi: 10.1109/COMPSAC.2018.00144.

[13] **Rietz R., Konig H., Ullrich S.,** and **Stritter B.,** (2016)."Firewalls for the Web 2.0," *Proc. - 2016 IEEE Int. Conf. Softw. Qual. Reliab. Secur. QRS 2016*, pp. 242–253, doi: 10.1109/QRS.2016.36.

[14] **Ghanbari Z., Rahmani Y., Ghaffarian H.,** and **Ahmadzadegan M. H.**, (2016). "Comparative approach to web application firewalls," in *Conference Proceedings of 2015 2nd Int. Conf. on Knowledge-Based Engineering and Innovation, KBEI 2015*, Mar., pp. 808–812.

[15] "RFC 1866 - Hypertext Markup Language - 2.0." https://tools.ietf.org/html/rfc1866 Alındığı tarih: 09.11.2020.

[16] "RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1." https://tools.ietf.org/html/rfc2616 Alındığı tarih: 09.11.2020.

[17] "RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content." https://tools.ietf.org/html/rfc7231 Alındığı tarih: 09.11.2020.

[18] "RFC 7992 - HTML Format for RFCs." https://tools.ietf.org/html/rfc7992 Alındığı tarih: 09.11.2020.

[19] **Sobola, T. D., Zavarsky, P., & Butakov, S.**, (2020). "Experimental Study of ModSecurity Web Application Firewalls," pp. 3–7. In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) (pp. 209-213). IEEE.

[20] **Tirumala, S. S., Sathu, H., & Sarrafzadeh, A.,** (2015). "Free and open source intrusion detection systems: A study.," *In 2015 International Conference on Machine Learning and Cybernetics (ICMLC) (Vol. 1, pp. 205-210). IEEE.*

[21] **Beckerle M.** and **Martucci L. A.,** (2013)."Formal definitions for usable access control rule sets from goals to metrics," *SOUPS 2013 - Proc. 9th Symp. Usable Priv. Secur.*, doi: 10.1145/2501604.2501606.

[22] **Smetters D. K.** and **Good N.,** (2009). "How users use access control," *SOUPS 2009 - Proc. 5th Symp. Usable Priv. Secur.*, doi: 10.1145/1572532.1572552.

[23] **Wool A.,** (2004). "A Quantitative Study of Firewall Configuration Errors," *Computer (Long. Beach. Calif).*, vol. 37, no. 6, pp. 62–67, doi: 10.1109/MC.2004.2.

[24] **Alfayyadh B., Ponting J., Alzomai M.,** and **Jøsang A.**, (2010) "Vulnerabilities in personal firewalls caused by poor security usability," *Proc. 2010 IEEE Int. Conf. Inf. Theory Inf. Secur. ICITIS 2010*, pp. 682–688, doi: 10.1109/ICITIS.2010.5689490.

[25] **Jøsang A., AlFayyadh B., Grandison T., AlZomai M.,** and **McNamara J.,** (2007). "Security usability principles for vulnerability analysis and risk assessment," *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, pp. 269–278, 2007, doi: 10.1109/ACSAC.2007.14.

[26] **Mansmann F., Göbel T.,** and **Cheswick W.**, (2012). "Visual analysis of complex firewall configurations," *ACM Int. Conf. Proceeding Ser.*, pp. 1–8, doi: 10.1145/2379690.2379691.

[27] **Zhang M., Xu B.,** and **Lu S.**, (2015)"An intelligent framework to detect network intrusion," *ICEIEC 2015 - Proc. 2015 IEEE 5th Int. Conf. Electron. Inf. Emerg. Commun.*, pp. 20–25, doi: 10.1109/ICEIEC.2015.7284478.

[28] **Raja F., Hawkey K.,** and **Beznosov K.**, (2009) "Revealing hidden context: Improving mental models of personal firewall users," *SOUPS 2009 - Proc. 5th Symp. Usable Priv. Secur.*, 2009, doi: 10.1145/1572532.1572534.

[29] **Bhatt S., Okita C.,** and **Rao P.** -Hewlett-Packard, "Fast, Cheap, and In Control: A Step Towards Pain Free Security!"[Online]. Available: http://www.hpl.hp.com/techreports/2007/. Alındığı tarih: 09.11.2020

[30] **Wool, A.** (2010). Trends in firewall configuration errors: Measuring the holes in swiss cheese. IEEE Internet Computing, 14(4), 58-65.

[31] **Voronkov A., Martucci L. A.,** and **Lindskog S.**, (2020). "Measuring the Usability of Firewall Rule Sets," *IEEE Access*, vol. 8, no. 4, pp. 27106–27121, doi: 10.1109/ACCESS.2020.2971093.

[32] **Whitten, A., & Tygar, J. D.** (1999). Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *USENIX Security Symposium* (Vol. 348, pp. 169-184).

[33] **Juristo N., Moreno A. M.,** and **Sanchez-Segura M. I.**, (2007) "Guidelines for eliciting usability functionalities," *IEEE Trans. Softw. Eng.*, vol. 33, no. 11, pp. 744–758, doi: 10.1109/TSE.2007.70741.

[34] **Jain, P., Djamasbi, S., & Hall-Phillips, A.** (2020). The Impact of Feedback Design on Cognitive Effort, Usability, and Technology Use.

[35] **Kung C. H., Hsieh T. C.,** and **Smith S.**, (2020). "Usability study of multiple vibrotactile feedback stimuli in an entire virtual keyboard input," Appl. *Ergon.*, vol. 90, no. September 2020, p. 103270, doi: 10.1016/j.apergo.2020.103270.

[36] "(No Title)." http://tdk.gov.tr/wp-content/uploads/2016/11/Bilgi-Güvenliği-Belgesi-2018.pdf Alındığı tarih: 09.11.2020.

[37] **Van Oorschot, P. C.** (2020). Computer Security and the Internet: Tools and Jewels. Springer Nature.

[38] “ISO/IEC 7498-1:1994(en), Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model — Part 1.” https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-1:ed-1:v2:en Alındığı tarih: 09.11.2020.

[39] **Neupane K., Haddad R.,** and **Chen L.,** (2018). “Next Generation Firewall for Network Security: A Survey,” *Conf. Proc. - IEEE* vol. 2018-April, pp. 1–6, doi: 10.1109/SECON.2018.8478973.

[40] **Ali F. A. B. H.,** (2011).“A study of technology in firewall system,” in *ISBEIA 2011 - 2011 IEEE Symposium on Business, Engineering and Industrial Applications*, , pp. 232–236, doi: 10.1109/ISBEIA.2011.6088813.

[41] **Mao H., Zhu L.,** and **Li M.**, (2012). “Current state and future development trend of firewall technology,” in *2012 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012*, doi: 10.1109/WiCOM.2012.6478472.

[42] **Barnett R.**, (2009). “WAF Virtual Patching Challenge : Securing WebGoat with ModSecurity Two Separ ate Appr oaches to Remediation Securing the Code Web Application Firewalls,”

[43] **Sampaio D.** and **Bernardino J.**, (2017). “Evaluation of Firewall Open Source Software,”, doi: 10.5220/0006361203560362.

[44] **Luo A., Huang W.,** and **Fan W.**, (2019)“A CNN-based Approach to the Detection of SQL Injection Attacks,” *Proc. - 18th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2019*, pp. 320–324, doi: 10.1109/ICIS46139.2019.8940196.

[45] **Pubal J.**, (2020) “SANS Institute Information Security Reading Room Web Application Firewalls,”.

[46] “Web Application Firewall Evaluation Criteria,” 2005. Available: http://www.webappsec.org. Alındığı tarih: 09.11.2020.

[47] “ModSecurity Handbook Second Edition: The book that will tell you everything you need to know about the popular open source web application firewall (Feisty Duck).” https://www.feistyduck.com/books/modsecurity-handbook/ Alındığı tarih: 09.11.2020.

[48] **O’Leary, M.** (2019). Apache Apache and ModSecurity. In Cyber Operations (pp. 721-788). Apress, Berkeley, CA.

[49] “OWASP ModSecurity Core Rule Set.” https://owasp.org/www-project-modsecurity-core-rule-set/ Alındığı tarih: 09.11.2020.

[50] “ISO - ISO 9241-210:2010 - Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.” https://www.iso.org/standard/52075.html Alındığı tarih: 09.11.2020.

[51] **Dix A., Finlay J., Abowd G. D.,** and **Beale R.**, "Human-computer interaction," 2004. www.hcibook.com. Alındığı tarih: 09.11.2020.

[52] **Nielsen J.**,(1994). "Usability inspection methods," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol., pp. 413–414, 1994, doi: 10.1145/259963.260531.

[53] **Nielsen J.**, (1992). "Finding usability problems through heuristic evaluation," in *Conference on Human Factors in Computing Systems - Proceedings*, 1992, pp. 373–380, doi: 10.1145/142750.142834.

[54] **Nielsen J.**, (1994). "Enhancing the explanatory power of usability heuristics," in *Conference on Human Factors in Computing Systems - Proceedings*, , pp. 152–158, doi: 10.1145/191666.191729.

[55] **Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N.** (2016). "Designing the user interface: strategies for effective human-computer interaction". Pearson. Alındığı tarih: 09.11.2020.

[56] **Weinschenk, S., & Barker, D. T.** ,(2000). Designing effective speech interfaces. Wiley.

[57] **Polson, P. G., Lewis, C., Rieman, J., & Wharton, C.** (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. International Journal of man-machine studies, 36(5), 741-773.

[58] **Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C.** (2002). Cognitive walkthrough for the web. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 463-470).

EKLER

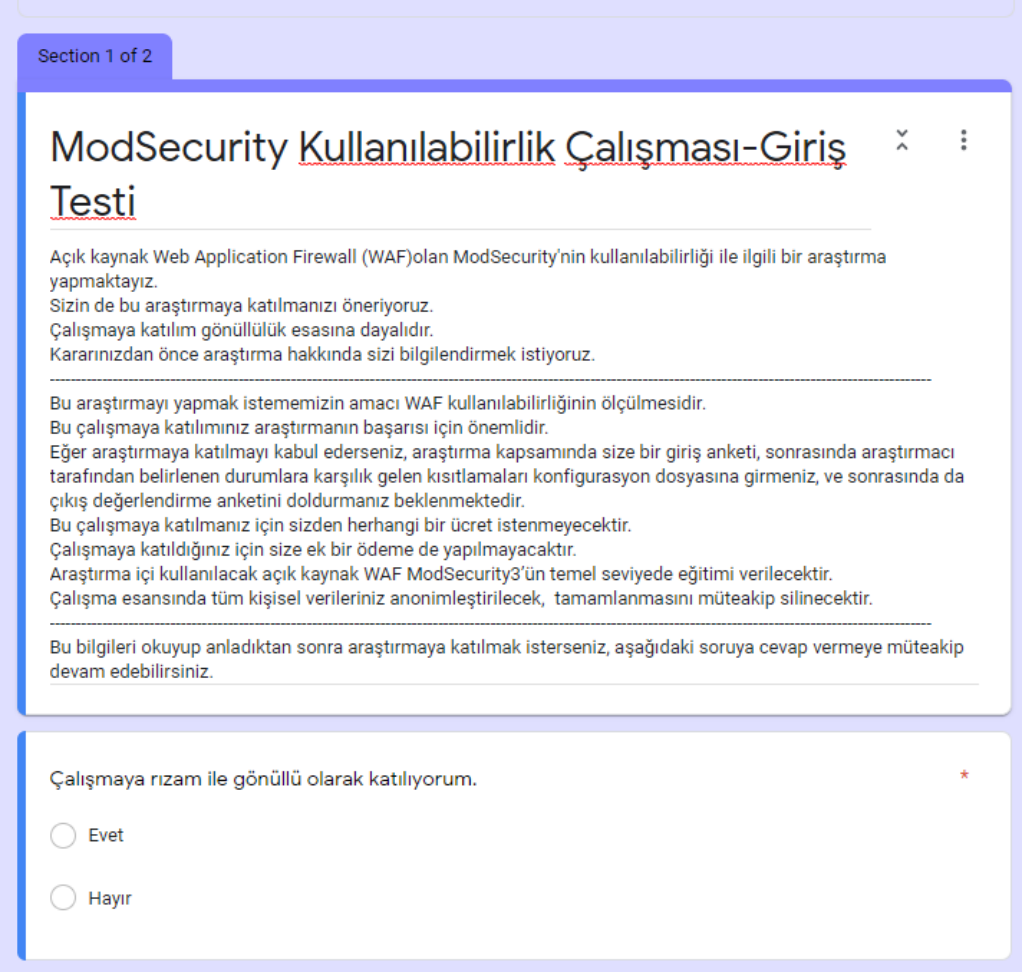
EK 1: Kullanıcı Çalışması Giriş Anketi.

EK 2: Kullanıcı Çalışması İçin Verilen Gerekli Bilgiler.

EK 3: Kullanıcı Çalışması İçin Hazırlanan Soru Kâğıdı.

EK 4: Kullanıcı Çalışması Çıkış Anketi.

EK-1 KULLANICI ÇALIŞMASI GİRİŞ ANKETİ



Resim Ek-1.1: Katılımcılara görüntülenen aydınlatılmış onam formu.

1. Cinsiyetinizi seçiniz.

☐ Kadın ☐ Erkek ☐ Cevap vermek istemiyorum.

2. Yaşınıza uygun aralığı seçiniz.

☐ 18'den küçük
☐ 18-20
☐ 21-23
☐ 24-26
☐ 27-29
☐ 30-35
☐ 36-40
☐ 41-45
☐ 45 üzeri

3. Eğitim durumunuzu seçiniz.

☐ Lise Mezun
☐ Lisans Öğrenci
☐ Lisans Mezun
☐ Yüksek Lisans Öğrenci
☐ Yüksek Lisans Mezun
☐ Doktora Öğrenci
☐ Doktora Mezun
☐ Diğer. (Lütfen belirtiniz.)

4. Çalışma durumunuzu açıklayan en yakın seçeneği işaretleyiniz.

☐ Öğrenciyim.
☐ Öğrenciyim aynı zamanda bilişim ile ilgili bir alanda çalışıyorum
☐ Bilişim ile ilgili bir alanda çalışıyorum.
☐ Çalışmıyorum.
☐ Diğer. (Lütfen belirtiniz.)

5. Bilgisayar bilimleri ile ilgili ne tür eğitim/eğitimler aldınız? Uygun olan birden fazla seçeneği işaretleyebilirsiniz.

☐ Hiç bir eğitim almadım
☐ Ön Lisans
☐ Lisans
☐ Yüksek Lisans
☐ Hizmet içi eğitim
☐ Online sertifika/kurs eğitimleri/Kişisel ilgi çaba veya gayretler
☐ Hiçbir eğitim almadım
☐ Diğer. (Lütfen belirtiniz.)

6. Aşağıdaki açık kaynak işletim sistemi dağıtımlarından kullandıklarınızı işaretleyiniz.

☐ Hiç birisini kullanmadım
☐ Linux
☐ FreeBSD
☐ NetBSD
☐ OpenBSD
☐ Open Solaris
☐ FreeDos
☐ GNU
☐ Diğer. (Lütfen belirtiniz.)

7. Aşağıdaki Linux dağıtımlarından kullandıklarınızı işaretleyiniz.

☐ Hiç kullanmadım.
☐ Android
☐ CentOs
☐ Fedora
☐ Kali Linux
☐ Oracle Linux
☐ Pardus
☐ Red Hat Linux
☐ SUSE Linux
☐ Ubuntu
☐ Diğer. (Lütfen belirtiniz.)

8. Aşağıdaki Siber Güvenlik ile ilgili aşina olduğunuz konuları işaretleyiniz.

☐ Uç sistem / cihaz güvenliği
☐ Zararlı yazılım analizi
☐ Ağ güvenliği
☐ Sızma testi
☐ Kriptoloji
☐ İşletim sistemi güvenliği
☐ Diğer. (Lütfen belirtiniz)

9. Siber güvenlik bilgi seviyeniz konusunda kendinizi değerlendiriniz.

Hiç bilgim yok				Çok iyi
1	2	3	4	5
☐	☐	☐	☐	☐

10. Bilgisayarınızda çoğunlukla hangi web tarayıcıyı kullanırsınız? Komut satırı arayüzünden hiçbir programı yönettiniz mi? Süre belirtiniz.

☐ Daha önce hiç yönetmedim.
☐ 0-1 Yıl
☐ 1-3 Yıl
☐ 4-10 Yıl
☐ 10 Yıl ve Fazlası

11. Daha önce herhangi bir network güvenlik cihazı (Firewall, NAC, IPS/IDS vb.) yönettiniz mi? Süre belirtiniz.

☐ Daha önce hiç yönetmedim

☐ 0-1 Yıl

☐ 1-3 Yıl

☐ 4-10 Yıl

☐ 10 Yıl+

12. Aşağıdaki ağ güvenliği araçlarından daha önce kullandığınız araçları işaretleyiniz.

☐ Saldırı Tespit Sistemi

☐ Saldırı Önleme Sistemi

☐ Güvenlik Duvarı

☐ Uygulama güvenlik duvarı

13. Daha önce Güvenlik Duvarı (Firewall) kuralı yazdınız mı?

☐ Evet. ☐ Hayır.

14. Daha önce ModSecurity web uygulama güvenlik duvarı kullandınız mı?

☐ Evet. ☐ Hayır.

15. Bunun dışındaki görüş ve önerilerinizi buraya yazabilirsiniz

EK 2: KULLANICI ÇALIŞMASI İÇİN VERİLEN GEREKLİ BİLGİLER

1. Modsecurity Konfigurasyon Dosyası:

/etc/apache2/modsecurity.d/modsecurity.conf

2. Yeni kuralların Yazılacağı Dosya:

/etc/apache2/modsecurity.d/rules/customrules.conf

3. Önceden hazırlanan kuralların bulunduğu dosya:

/etc/apache2/modsecurity.d/rules/genericrules.conf

4. Exception kurallarının bulunduğu dosya

/etc/apache2/modsecurity.d/rules/customexceptions.conf

5. Modsecurity üzerinde bütün işlemlerinizi root olarak gerçekleştirin.

6. Kontrol ve geri besleme scripti kullanımı:

Her sorudan sonra ceabınızın doğruluğunu kontrol etmek için soru numarası ile başlayan scripti çalıştırın. Örnek olarak 3. Soru için: /etc/check/3.sh

7. Metin editörü olarak nano kullanabilirsiniz. Kapatmak için ctrl+X sonrasında kaydet kapat yapmanız yeterli. nano /etc/apache2/modsecurity.d/modsecurity.conf

8. Herhangi bir aşamada hata aldığınızda öncelikle hata mesajının neden olduğunu irdeleyin. Soru sormaktan çekinmeyiniz.

9. 2. Sorudan itibaren gerekli kurallara 1002 den başlayarak id numarası veriniz.

10. Size gönderilen Excel Dosyasında kural formatına uygun kullanabileceğiniz parametreler sınıfladırılmıştır.

Directives: 5.1, Variables: 5.2-5.8, Operators: 5.9-5.12, Actions: 5.13-5.19

Kural Formatı

DIRECTIVES VARIABLES OPERATOR [TRANSFORMATION_FUNCTIONS, ACTIONS]

Örnek Kural:

```
SecRule REQUEST_URI "@rx admin-login.php" \
    "id:999999, \
    deny”
```

EK 3: KULLANICI ÇALIŞMASI İÇİN HAZIRLANAN SORU KAĞIDI

1. Konfigurasyon Dosyasında gerekli değişikliği yaparak ModSecurity'i aktif hale getirin.
2. SİEM'den web sitenizin çok yavaş cevap verdiği ve yapılan analiz sonucunda REQUEST HEADER'ın USER AGENT bilgisinde "zeus" bilgisi bulunan çok sayıda istekle, tahmini olarak botnet tarafından, kaynaklarınızın tüketildiği bilgisi verilmiştir. Http Request Headerında bulunan User Agent'ı Zeus olan bütün istemcileri bloklayın.

   Kural için girmeniz gereken regex zeus tur.

   Ayrıca Yazdığınız kuralın action bölümüne bu bilgileri ekleyin.

   t:none, t:lowercase,

   tag:'Zeus Botnet'

   tag:'User-Agent Block',tag:'Custom Rule'

3. Web siteniz genelinde "Path Traversal" zaafiyeti olduğu ve bu zaafiyetin argümanlar(ARGS) kullanılarak sömürülebildiği öğrenilmiştir. Zaafiyetin yama geçilenceye kadar WAF tarafından kapatılması istenmiştir. Söz konusu zaafiyeti istismar edebilecek erişim isteklerini bloklayın ve kuralı devreye alın.

   Kural için girmeniz gereken regex (?:^|[\\/])\.\.(?:[\\/]|$)

   Ayrıca Yazdığınız kuralın action bölümüne bu bilgileri ekleyin.

   t:none,t:utf8toUnicode,t:urlDecodeUni,t:removeNulls,t:cmdLine

   msg:'Path Traversal Attack (/../)'

   tag:'attack-lfi', tag:'Custom Rule'

4. Web sitenizde ShellShock zaafiyeti barındırdığı yapılan zaafiyet analizi sonucunda anlaşılmıştır. Açıklığın şimdilik Request_Headers veya Request_Line da geçtiği bilinmektedir. Açıklığın giderilmesi için çalışmalar devam etmektedir ancak bu zamana kadar geçen süre içerisinde sanal yama yapılması istenmiştir. Söz konusu zaafiyeti istismar edebilecek erişim isteklerini bloklayın.

   Kural için girmeniz gereken regex ^\(\s*\)\s+{

   Ayrıca Yazdığınız kuralın action bölümüne bu bilgileri ekleyin.

   capture,

   t:none,t:urlDecode,

   msg:'Remote Command Execution: Shellshock (CVE-2014-6271)',

   logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}: %{MATCHED_VAR}',

   tag:'attack-rce', tag:'Custom Rule'

   ctl:auditLogParts=+E

5. Kural dosyasında bulunan ve Kural ID’si 1000 olan kurala Exception tanımlayın. Bu tanımlamayı kuralda gördüğünüz bir etikete (tag) göre yapınız.

EK-4 KULLANICI ÇALIŞMASI ÇIKIŞ ANKETİ

1. Bu yazılımı kullanmak için daha yetkin hissedebilmek için ne kadar zamana ihtiyacınız olacağını düşünüyorsunuz?

☐ 0-1 Saat
☐ 1-5 Saat
☐ 5+ Saat
☐ Diğer. (Lütfen belirtiniz.)

2. Bu yazılımı kullanmak için daha yetkin hissedebilmek için ne kadar zamana ihtiyacınız olacağını düşünüyorsunuz?

☐ 0-1 Saat
☐ 1-5 Saat
☐ 5+ Saat
☐ Diğer. (Lütfen belirtiniz.)

3. Sistem yöneticisi olarak düşündüğünüzde, ModSecurity'i Web Uygulamalarını ne seviyede koruduğunu düşünüyorsunuz?

Hiç bir koruma sağlamaz. Çok iyi koruma sağlar.

1	2	3	4	5
☐	☐	☐	☐	☐

4. Sistem yöneticisi olarak düşündüğünüzde, sizden WAF ürünü ile ilgili tavsiye istenirse, ModSecurity'i tavsiye etme olasılığınızı puanlayınız.

Kesinlikle tavsiye etmem. Kesinlikle tavsiye ederim.

1	2	3	4	5
☐	☐	☐	☐	☐

Verilen görevleri göz önünde bulundurarak zorluk derecesine göre soruları puanlayınız.

5. ModSecurity'i aktif hale getirme.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

6. Http Request Headerında bulunan User Agent'ı Zeus olan bütün istemcileri bloklaması.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

7. "Path Traversal" zafiyetinin sanal yaması.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

8. "ShellShock" zafiyetinin sanal yamalası.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

9. Exception tanımlanması.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

10. Size sağlanan materyaller göz önünde bulundurulursa, LAB ortamına erişim zorluğunu puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

11. Genel olarak düşündüğünüzde, ModSecurity'nin kullanma zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

12. ModSecurity'nin SYNTAX zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

13. ModSecurity'nin kural yazma zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

14. ModSecurity'nin hata giderme (troubleshooting) zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

15. Verilen görevlerin zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

16. Konfigurasyon dosyalarının dosya yolunun kullanma zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

17. Kural dosyalarının dosya yolunun kullanma zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

18. Komut satırı kullanarak program yönetiminin zorluk derecesini puanlayınız.

Çok Kolay		Orta		Çok Zor
1	2	3	4	5
☐	☐	☐	☐	☐

19. ModSecurity'nin kullanımının daha kolay nasıl olabileceğini düşünüyorsunuz.

20. Eklemek istediğiniz başka husus var mıdır?

ÖZGEÇMİŞ

Ad-Soyad : **Murat ALAGÖZ**

Uyruğu : **T.C.**

Doğum Tarihi ve Yeri : **1986 Kayseri**

E-posta : **mrtalagoz@gmail.com**

ÖĞRENİM DURUMU:

- **Lisans** : 2009, Kara Harp Okulu, Sistem Mühendisliği (Elektronik)
- **Lisans** : 2018, Anadolu Üniversitesi, Açık Öğretim Fakültesi (Sosyoloji)
- **Ön Lisans** : 2018, Trakya Üniversitesi, Tunca MYO (Bilgisayar Prog.)
- **Yüksek Lisans:** 2020, TOBB Ekonomi ve Teknoloji Üniversitesi, Bilgisayar Mühendisliği (Bilgi Güvenliği)

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2009-2018	**Yurt içi - Yurt dışı**	**Brl.K.lığı**
2018-	**K.K.K.lığı Ankara**	**İşltm. Sb.**

YABANCI DİL: İngilizce -İyi.

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

Alagöz, M., Tok M.S.,(2020) Açık kaynak web uygulama güvenlık duvarı ModSecurity'nin kullanılabilirlik analizi. Uluslararası 30 Ağustos Bilimsel Araştırmalar Kongresi, Ankara, Türkiye.