T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DEVSECOPS SÜREÇLERİNİN ÇEVİKLEŞTİRİLMESİ: DİNAMİK LOG ANALİZ YAKLAŞIMI

Baybarshan EKİZ

Yüksek Lisans Tezi

ADLİ BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI

Bilgi Güvenliği

AĞUSTOS 2021

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

DEVSECOPS SÜREÇLERİNİN ÇEVİKLEŞTİRİLMESİ: DİNAMİK LOG ANALİZ YAKLAŞIMI

Tez Yazarı
Baybarshan EKİZ

Danışman
Doç. Dr. Şengül DOĞAN

AĞUSTOS 2021

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Adli Bilişim Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	DevSecOps Süreçlerinin Çevikleştirilmesi: Dinamik Log Analiz Yaklaşımı
Yazarı:	Baybarshan Ekiz
İlk Teslim Tarihi:	9.6.2021
Savunma Tarihi:	10.8.2021

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

İmza

Danışman: Doç. Dr. Şengül DOĞAN — Onayladım
Fırat Üniversitesi Teknoloji Fakültesi Adli Bilişim Mühendisliği Bölümü

Başkan: Dr. Öğr. Üyesi Erhan AKBAL — Onayladım
Fırat Üniversitesi Teknoloji Fakültesi Adli Bilişim Mühendisliği Bölümü

Üye: Dr. Öğr. Üyesi Mehmet BAYGIN — Onayladım
Ardahan Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bolumu

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Doç. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Makine Öğrenmesi Tabanlı Kötücül Yazılımların Tespiti" Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

10.08.2021

Baybarshan Ekiz

ÖNSÖZ

Tez çalışmamın tüm aşamalarında desteğini esirgemeyen ve büyük bir sabırla benden sonuç bekleyen değerli hocam Sayın Doç. Dr. Şengül DOĞAN'a teşekkürlerimi sunarım.

Bütün hayatım boyunca gerek manevi gerekse maddi yardımlarını esirgemeyen, iş hayatımın ve özel hayatımın bütün zorluklarında yanımda olan ve bana sabır gösteren; başta annem ve babam olmak üzere tüm aileme teşekkürü bir borç bilirim.

Baybarshan Ekiz

ELAZIĞ, 2021

İÇİNDEKİLER

Sayfa

ÖZET

DevSecOps Süreçlerinin Çevikleştirilmesi: Dinamik Log Analiz Yaklaşımı

Baybarshan Ekiz

Yüksek Lisans

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Adli Bilişim Mühendisliği Anabilim Dalı
Ağustos 2021, Sayfa: 87

Yazılım sektörü için her geçen gün artan esneklik ve performans artırım ihtiyacı günümüzde DevOps kavramını var etmiştir. Günümüzde DevOps sistemleri, güncel yazılım geliştirme süreçleri için çok büyük önem taşımaktadır. Güvenlik sektörünün de öneminin son yıllarda çok ciddi bir artış göstermesi de DevSecOps kavramının meydana çıkmasını sağlamıştır. Bu bağlamda sistemlerin kendilerine özgü yapılarına göre sürekli işleyen denetim, operasyon ve geliştirme süreçlerine dair otonom süreç bütünü esneklik ve performans artırımının düşük maliyetle sağlanabilmesi için çok kritik önem taşımaktadır.

Bu tez çalışması kapsamında otonom olarak DevSecOps süreçlerini daha çevik bir şekilde tasarlamak amaçlanmıştır. Önerilen DevSecOps modeli, yazılımsal sistemlerin en kaliteli ve güvenli şekilde geliştirilmesi için güvenlik denetim mekanizmalarının sisteme özgü olarak oluşturulabilmesini hedeflemektedir. Önerilen sistem, otonom güvenlik denetim mekanizması için sunucu üzerinde bulunan loglardan faydalanarak öğrenme yapmaktadır. Bu öğrenme kapsamında sistem, üzerinde sömürülmesi muhtemel olabilecek zafiyetlere dair önlem almak üzere istatistiki sonuçlar vermektedir. Önerilen DevSecOps modelini doğru çalışması, ancak anormal durumların ve zafiyet karakteristiklerinin sürekli olarak tespit edilebilir olması durumunda mümkündür.

Yapılan çalışmada önerilen DevSecOps modeline yönelik uygulanabilirliğin sınanması için, Docker kaynak kullanım istatistiklerini gösteren log dosyaları oluşturulmuştur. Bu log dosyaları analiz edilerek anormal durumların ve zafiyetlerin karakteristiğinin tespit edilebilirliğine dair, z-skor ve nokta çift serili korelasyon metotlarıyla analiz yapılarak bulgular oluşturulmuştur. Bulgulara göre varılan sonuç göstermektedir ki anormal durum tespitini ve zafiyet karakteristiğini loglar üzerinden tespit edebilmek mümkündür.

Anahtar Kelimeler: Log, DevSecOps Süreçleri, Güvenlik, Statik

ABSTRACT

Agility of DevSecOps Processes: Dynamic Log Analysis Approach

Baybarshan Ekiz

Master of Science

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences
Digital Forensic Engineering Department
August 2021, Page: 87

The ever-increasing need for flexibility and performance enhancement for the software industry has created the concept of DevOps today. Today, DevOps systems are of great importance for current software development processes. The significant increase in the importance of the security sector in recent years has also led to the emergence of the concept of DevSecOps. In this context, the autonomous process of the control, operation and development processes, which are constantly operating according to the unique structures of the systems, is of critical importance in order to provide flexibility and performance increase at low cost.

Within the scope of this thesis, it is aimed to design autonomous DevSecOps processes in a more agile way. The proposed DevSecOps process aims to create system-specific security control mechanisms in order to develop software systems with the highest quality and security. The proposed system learns from the logs on the server for the autonomous security control mechanism. Within the scope of this learning, it gives statistical results in order to take precautions about the vulnerabilities that may be exploited on the system. The proposed DevSecOps model will work properly only if abnormal conditions and vulnerability characteristics are continuously detectable.

In order to test the applicability of the DevSecOps model proposed in the study, log files that showing Docker resource usage statistics were created. By analyzing these log files, findings were created by analyzing the characteristics of abnormal situations and vulnerabilities with z-score and point biserial correlation methods. The result obtained according to the findings shows that it is possible to detect abnormal condition detection and vulnerability characteristics through logs.

Keywords: Log, DevSecOps Processes, Security, Static

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR

Simgeler

z_i	:i. Değerin Z-Skor'u
σ	:Standart Sapma
μ	:Ortalama
$\bar{x}$	:Ortalama
ss	: Standart Sapma

Kısaltmalar

ZSO	: Zafiyet Sömürü Oranı
EZSO	: Eski Zafiyet Sömürü Oranı
YZSO	:Yeni Zafiyet Sömürü Oranı
DAST	:Dynamic Application Security Test
SAST	:Static Application Security Test
WAF	:Web Application Firewall
IDS	:Intrusion Detection System
DKP	:Deney Karakteristik Puanı
DKL10BS	:Deneyin Korelasyon Listesinde İlk 10'da Bulunma Sayısı
DKL50BS	:Deneyin Korelasyon Listesinde İlk 50'de Bulunma Sayısı
DKL100BS	:Deneyin Korelasyon Listesinde İlk 100'de Bulunma Sayısı
DKL250BS	:Deneyin Korelasyon Listesinde İlk 250'de Bulunma Sayısı

1. GİRİŞ

1.1. Modern Yazılım Sistemlerinde Esneklik, Kalite ve Güvenlik

Günümüzde ideal bir yazılım sistemi; esnek tasarlanmalıdır, kaliteli geliştirilmelidir ve kullanıcılara güvenli olarak sunulmalıdır. Esneklik, kalite ve güvenlik kavramları incelendiğinde, esneklik son 25 yılda ciddi bir artışa sahiptir. Son 25 yıldaki teknoloji ve birey tabanlı ihtiyaçların değişim hızı üstel olarak artmaktadır. Yazılım sektöründeki esneklik ihtiyacı ise teknoloji ve birey tabanlı ihtiyaçların değişim hızıyla paralel olarak yükselmektedir. Esneklik temelinde hızlı ve çevik yazılım üretimini sağlamaktadır. Bu sebeple esneklik, yazılım sektöründe her geçen gün daha da önemli hale gelmektedir. DevOps otonom yapısıyla, esnekliğe olan ihtiyacı, hızlı ve çevik olarak karşılamaktadır.

Yazılımda kalite kriterinin temeli incelenirse, doğruluk ve geçerlilik kavramları ön plana çıkmaktadır. Bir yazılımın yapması gereken iş tanımlıdır, yazılım kendisine tanımlanan işi ne kadar yeterli düzeyde sağlıyor ise, bu yazılım o düzeyde geçerlidir. Bununla beraber yazılım ne kadar performans standartlarına uygun olarak çalışıyor ise, o kadar doğru bir yazılımdır. Tam olarak kullanıcıların ihtiyacını karşılamasına rağmen sistemin gerekenden daha yavaş çalışması durumunda sistem, geçerli fakat doğru olmayan bir sistemdir. Bunlardan dolayı yazılım sistemleri için kalitenin belirleyicisi, doğrulama(verification) ve geçerlemedir(validation)[57].

Doğru ve geçerli yazılımın hızlı ve esnek olarak üretiminin ideal bir yazılım sistemi için kritik olduğu görülmektedir. Farklı bir şekilde ifade etmek gerekirse kaliteli yazılımın, esnek bir şekilde üretilmesi günümüzde yüksek öneme sahiptir. DevOps yaklaşımıyla, hızlı ve çevik olarak yazılımın kalite denetimini sağlanmaktadır. DevOps yaklaşımı ile ilgili olarak her yerde uygulanabilecek tek bir model yoktur, tek bir felsefi görüş vardır. Bu felsefe temelinde; denetim, operasyon ve geliştirme süreçlerini hızlı ve çevik olarak yürütmeye dayanmaktadır. Bu sebeple her kurumun karakteristik yapı farklılığı, farklı DevOps süreçlerini ortaya çıkarmaktadır.

Yazılım sektörü için esneklik ve kalite kadar günden güne önem kazanan diğer ihtiyaç ise güvenliktir. Güvenlik ihtiyacının da karşılanması, dinamik ve esnek bir şekilde DevSecOps süreçleri ile sağlanmaktadır. DevOps perspektifine göre denetim mekanizmaları sadece yazılımın kaliteli ve geçerli olduğuna dair kontrolün çevik ve otonom olarak sağlanmasından sorumludur. DevSecOps süreçleri DevOps perspektifine güvenlik katmanının da eklenmesiyle oluşmaktadır. Fakat DevSecOps güvenlik denetimini de otonom yapısıyla hızlı ve çevik olarak yapmaktadır.

1.2. Günümüzde Dinamik Güvenlik Modellerinin Gereksinimi

Güvenliğin yazılımlarda standardize edilmesi için belirli kontroller bulunmaktadır. Bu kontroller sayesinde standartlara uyan sistemler "güvenli", uymayanlar ise "güvenli değil" olarak tanımlanabilmektedir. Bu güvenlik düzeylerinin belirlenmesini sağlayacak kontrol listeleri, Owasp gibi merkeziyetçi kurumlar tarafından sağlanmaktadır. Bu merkeziyetçi kurumlar tüm yazılımsal sistemlere uygulanabilen standartları belirlemektedir.

Merkeziyetçi yapılar varyasyonların az olduğu sistemler için işleyişin ve yönetimin sağlıklı olduğunu gösterse de, günümüzde yazılım sistemlerinin varyasyonlarının giderek arttığı duruma tam olarak uyum sağlamamaktadır. Araba veya buzdolabı gibi fabrikasyon üretim sistemlerince üretilen ürünler, merkeziyetçi kurumların oluşturduğu standartlara bağlı olarak kaliteli üretilebilmektedir. Fakat günümüzde yazılım sistemlerinin çok büyük bir çoğunluğu biricik bir yapıdadır. Bununla birlikte sürekli bir değişim de göstermektedirler.

Merkeziyetçi olmayan denetim mekanizmalarına yönelik inceleme yapılırsa, dinamizmin merkeziyetçi denetim standartlarıyla uyumsuz olduğu görülecektir. Merkeziyetçi olmayan denetim mekanizmalarına örnek olarak Blockchain incelenirse denetimin, dinamizmden dolayı mümkün olmadığı çok açık olarak görülecektir. Blockchain'i denetlemek için o günün şartlarına göre merkeziyetçi kurumlar tarafından oluşturulan prosedürler 1 hafta içinde şart ve koşulların mutlak değişimi sebebiyle geçerliliğini kaybedecektir. Günümüzde yazılım sistemleri için de aynı durum geçerlidir.

Blockchain işlem dinamizmine dair değişimler yazılım sistemlerinin dinamizmiyle benzer bir yapıdadır. Teknik, toplumsal, insani, müşteri/şirket hedefi ve hukuki olmak üzere birçok gerekçeden dolayı değişim gereksinimi yazılımsal sistemlerde ortaya çıkabilmektedir. Herkese açık olan, internet üzerinden herkesin eriştiği ve sürekli geliştirilen açık kaynak sistemleri tamamıyla merkeziyetçi olarak yönetmek güvenlik ve performansın sürekli azalması sonucunu doğuracaktır. Bu sebeple sistemin karakteristiğini öğrenen algoritmalar aracılığıyla her sisteme özgü olan güvenlik politikaları düzenli aralıklarla güncellenmelidir ve denetlenmelidir.

2. İLGİLİ ÇALIŞMALAR

2.1. Sistemlerde Kar ve Performans Artırımının Otonomlaştırımı

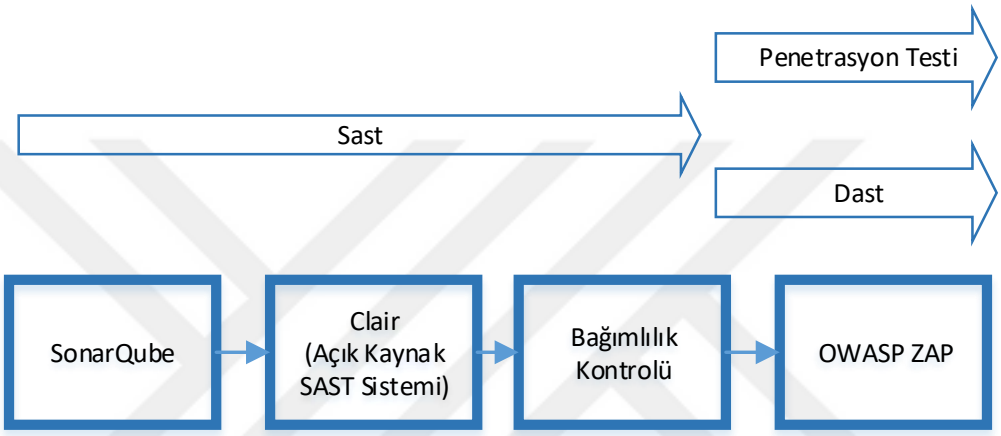
Tso vd. [21] yaptıkları çalışmalarda sanallaştırma sistemlerinin cloud tabanlı olarak ürünleştirilmesine dair inceleme yapmışlardır. Maliyet faktörlerine göre ağ altyapısı için %15'lik, sunucu altyapısı için ise % 45'lik bir sermaye ayrılmaktadır. Bu sistemleri hizmet olarak pazarlayan kişilerle beraber sistemlerin müşterililerinin kar elde etmesi için istenilen performansı minimum maliyetle sağlayabilmek gerekmektedir. Bu sebeple dinamik olarak yükün analiz edilmesi sonucu sistemlerin kaynak dağılımının değişebilmesi gerekmektedir. Ancak bu yolla müşteri ve hizmet sağlayıcı tarafların kar ve performansı optimum düzeyde dengelenerek arttırılmaktadır.

Yu vd. [24] Kişilerin Facebook üzerindeki davranışlarına göre zararlı olabilecek bir davranışı oluşturma durumlarına dair öngörünün yapılabilmesi amaçlanmıştır. Bu bağlamda kullanıcı verileri analiz edilmiştir ve riskli kullanıcı davranışları gruplanabilmiştir. Gruplama sayesinde benzer davranışlara sahip kullanıcılar risk grubuna dahil edilebilmiştir. Grup dahilinde bulunan kullanıcıların büyük çoğunluğu risk taşıması durumu doğrulanmıştır, fakat risksiz olarak kabul edilen kullanıcıların arasında da risk niteliğine sahip olanlar çıkmıştır. Bu bağlamda kullanıcı davranışlarının analizi, sistemde bulunan tüm riskli kullanıcıları tespit edememesine rağmen, riske dair öngörülerinin büyük çoğunluğu doğru olmuştur. Bu öngörüler kapsamında risksiz olarak görülen kullanıcılara Tso vd. [21] yaptığı çalışmalar bağlamında, daha yüksek performansa sahip sunucularla hizmet sağlanabilmesi için altyapı oluşturulabilmektedir.

Liu vd. [22] çok büyük depolama ihtiyacı olan sistemlerin giriş/çıkış oranlarının ölçümlemesi yapılmıştır. Bu ölçümleme sonucu görülmüştür ki büyük kapsama sahip sistemlerde giriş/çıkışın birbirine oranıyla yazılım sistemlerinin kaynak tüketimleri çok büyük bir oranda koordine edilmeksizin yönetilmektedir. Bu durum ise darboğazları beraberinde getirmektedir. Çalışmada giriş çıkış karakteristiğiyle kaynak tüketimi arasındaki davranışsal bütünlüğü oluşturarak darboğazların önüne geçilmesi amaçlanmıştır. Bu amacın gerçekleştirilmesi için giriş/çıkış gürültü verilerinin indirgenmesi ve log verilerinin analiz edilmesi için Oak Ridge ulusal laboratuvarında bulunan dünyanın en büyük 2.süper bilgisayarı olan titan kullanılmıştır. Uygulamalarda görülmüştür ki darboğazların önüne geçilmesi için sistem karakteristiğinin, log tabanlı davranış analizinin sağlanması yoluyla gerçekleştirilmesi yüksek bir başarımla mümkündür.

2.2. DevSecOps Süreçlerinin Çıkış İhtiyacı

Viitasuo vd. [14] Modern bir toplumun parçası olan yazılım sistemlerine DevOps süreçleri kapsamında, güvenliğin de sağlanarak üretilmesi için Şekil 2.1. üzerindeki gibi bütüncül bir entegrasyon önerisinde bulunulmuştur. DevSecOps yaklaşımından yararlanılarak açık kaynak eklentileri güvenlik kriterlerini sürekli sağlayacak şekilde CI/CD entegrasyonu yapılmıştır. Bu sayede sürekli olarak Owasp Top 10 üzerinde tanımlı olan zafiyetlere karşı önlem alınmak üzere zafiyet kontrolü Şekil 2.1 üzerindeki akış aracılığıyla gerçekleşmektedir.



Şekil 2.1. Viitasuo vd. [14] Tarafından Önerilen GoCD Sisteminin Yapısı

Ahmed vd. [16] Güvenlik süreçlerinin geliştirme süreçleriyle birlikte yürütülmesinden ziyade ayrı bir katmanda yürütüldüğü bir süreç olduğu söylenmektedir. Güvenlik ve geliştirme sürecinin bütüncül olarak Hevner ve Chatterjee [56] yaklaşımı kapsamında oluşturulması amaçlanmıştır. Bu yolla hızlı ve güvenli bir sürecin yürütülebileceği öngörülmüştür. Bu kapsamda DevOps ekiplerinin güvenlik süreçleri konusunda beceri sahibi olması ve gerekli eklentileri CI/CD süreçlerine entegre etmesi gerekliliği sağlanmıştır. Yapılan çalışmalar sonucu görülmektedir ki güvenlik ve geliştirme sürecinin bütüncül olarak yürütülmesi durumunda, hem güvenlik artmaktadır hem de maliyet azalmaktadır[56].

Heilmann vd. [17] DevOps süreçlerinin genel olarak esnekliğin sağlanarak kalitenin arttığı görülmektedir. DevSecOps süreçleri, güvenliğinde yüksek seviyede tutulduğu ve DevOps yaklaşımıyla bütüncül bir süreç olarak tanımlanmıştır. Bu tanım kapsamında güvenliğin DevOps ile entegre olduğu DevSecOps süreçlerinin, kalite ölçümlemesinin ne gibi kriterle yapılabileceğine dair araştırmalar yapılmıştır. Bu bağlamda başarımın kaliteyle doğru orantılı olarak artması için DevSecOps süreçlerinin başarım kriterlerinin doğru olarak belirlenmesinin ciddi önem taşıdığı görülmüştür.

Genel olarak DevOps kavramının ortaya çıkışındaki temel gerekçe operasyonel ve geliştirme süreçlerinin kopukluğudur. Günümüzdeki ihtiyaçları karşılamakta en önemli kriter olan esnekliği sağlamak için süreçler Çevik olarak yürütülmektedir. Genel olarak yapılan çalışmalar göstermektedir ki, çevik süreçler günümüzde sadece operasyonel ve geliştirme süreçlerinin bütüncül yapılmasının ötesine geçerek güvenlik süreçlerini de kapsamına almalıdır. Bu kapsamda günümüz için DevSecOps kalitenin en yüksekte tutulup, maliyetin en düşük kaldığı modelleri sunabilmektedir.

2.3. 0.Gün Zafiyetlerinde WAF'ın Yetersiz Kalması

Asselin vd.[30] 0.gün zafiyetlerinin tespitinde WAF gibi sistemlerin yetersizliğinden kaynaklanan güvenlik sorunlarının düzeltilmesi için çalışmalar yapılmıştır. Bu çalışmalar kapsamında anormal durumun tanılanması ve bu tanının zafiyet tespiti için kullanılması gerektiği savunulmaktadır. Bu bağlamda yapılan çalışma göstermektedir ki 0.gün açıkların engellenmesi için, anormal davranışın analiz edilerek öğrenme tabanlı bir yapının kullanımı yoluyla başarılı sonuçlar elde edilebilmektedir.

Khairkar vd. [34] Yapılan çalışmada IDS'in 0. gün açıklarına çözüm bulamaması sorunu ele alınmıştır. Log datalarının kullanımı yoluyla zafiyetin oluşması ve sistem davranış ilişkisinin analiz edilmiştir. Bu analiz kapsamında düzenli güncellemeler ve eski verilerin anlamlandırılmasının, zafiyet tespitini davranış analizi üzerinden yapılması için önem taşıdığı görülmüştür.

Matin vd. [29] Sistemlerin içine sızabilen trojen, solucan ve benzeri zararlı yazılımları tespit etmek için IDS(Saldırı tespit sistemleri), güvenlik duvarı ve antivirusler yeterli değildir. Bu sebeple anormal davranışların takibinin sağlanması ve analiz edilebilmesi için honeypot olarak isimlendirilen gerçek gibi gözüken kopya sistemler kullanılmalıdır. Bu kopya sistemler üzerinden anormal durumlarda sistem davranışına dair öğrenme yapılarak zafiyet sömürüsü yapan zararlı yazılımların gerçek sistemde tespit edilmesi gerektiği savunulmaktadır.

2.4. Zafiyetlerin Otonom Olarak Log Dosyaları Üzerinden Analizi

Gao vd. [26] Log dosyalarının analiz edilmesi yoluyla SQL Injection zafiyetinin öngörülebilirliğine dair analiz çalışmaları yapılmıştır. Temel olarak SQL Injection saldırısının gerçekleşmesi durumunda, log dosyaları üzerinde örüntülerin elde edilmesi ve dinamik olarak gerçek zamanlı bir biçimde bu örüntünün trafik üzerinde eşleştirilmesine dayanmaktadır. Öğrenme ve sınıflandırma algoritmaları aracılığıyla bu metodun yüksek bir başarımla öngörü için yeterli olduğunu göstermiştir.

Gawron vd. [27] Log dosyalarının analizi yoluyla zafiyet karakteristiğinin pasif olarak tespit edilebilmesinin ne düzeyde mümkün olabileceğine dair çalışmalar gerçekleştirilmiştir. Bu çalışmalar için kaynak logları, trafik dataları ve işletim sistemi üzerindeki processler analiz edilmiştir. Çalışmalar sonucu her sistemin karakteristik olarak zafiyetleri farklı da tepki verdiği görülmüştür. Aynı sistem üzerinde ise farklı zamanlarda aynı zafiyetler diğerler zafiyetlere kıyasla ayırt edilebilecek düzeyde bir karakteristik oluşturmaktadır.

Maimon vd. [23] Sistemlerin karakteristik yapısının değişkenliği kullanıcı yapısının değişkenliğiyle bağlantılıdır. Bu bağlamda sistemlerin kullanıcı ve kullanım yapılarına göre saldırgan motivasyonu şekillenmektedir. Bu motivasyonu şekillendiren temel unsurlar ise başlıca SKRAM olarak kısaltılan "beceriler, bilgi, kaynaklar, yetki ve motivasyonlar" bütününden oluşmaktadır. Bütüncül bakılırsa sistemin kullanıcı ve kullanım şekline göre SKRAM yapısı şekillenmektedir. Bu kapsama göre de kurumlar sistemlerine yönelik risk analiz ve önlemlerini almaktadır. Fakat bugün için çevrimiçi sistem yapılarının dinamizmi ve esnekliği, SKRAM yaklaşımını uzun vadede güvenli bir yapının sağlanması için yetersiz kılmaktadır. Bu sebeple dinamik olarak sistemin izlenmesi, analiz edilmesi ve SKRAM modellerinin düzenli olarak saldırgan ve kullanıcı davranışları da göz önünde bulundurularak güncellenmesi önerilmektedir.

2.5. Zafiyet Tespitinin Dinamik ve Esnek Olarak Analizi

Khan vd. [33] 0.gün zafiyet sömürümü durumunda yetersiz kalan IDS sistemlerinden dolayı entropi yoluyla anormal durum tanılama yöntemleri sınanmıştır. Yapılan çalışmalar entropiye göre zafiyet durumunda davranışların analizi üzerinden anormal durumun tanılanmasının SQL Injection ve XSS zafiyetleri üzerinde yapılan denemeler aracılığıyla analiz edilmiştir. Analizler göstermektedir ki, entropinin analiz edilmesi anormal durumun tespiti için başarılı sonuçlar vermektedir.

Liu vd. [32] Anormal durum tespitinin hızlı ve yüksek başarıma sahip olabilmesi için yüksek yoğunluğa sahip log dosyalarının doğru analiz yöntemleri için çalışılmıştır. Çalışma göstermektedir ki filtreleme, iyileştirme, kümeleme davranış analizinin yapılması için kritik önem taşımaktadır

Zhang vd. [31] Güvenlik zafiyetlerinin oluştuğu anormal durumların tespiti için logların analiz yöntemlerinin iyileştirilmesi ve daha kararlı sonuçların elde edilebilmesi için çalışma yapılmıştır. Bu çalışma kapsamında log dataların toplanırken uygulanan filtrelerin dinamikliğinin önemine vurgu yapmaktadır. Bununla beraber anormal davranışların karakteristiğinin zamanla değiştiği için, uyum sağlanması başarımı arttırmak için önemli bir kriterdir. Sonuç olarak görülmüştür ki sistemin daha dinamik olarak davranış analizi yapması durumunda, zafiyet tespitinin başarımını ciddi oranda artmaktadır.

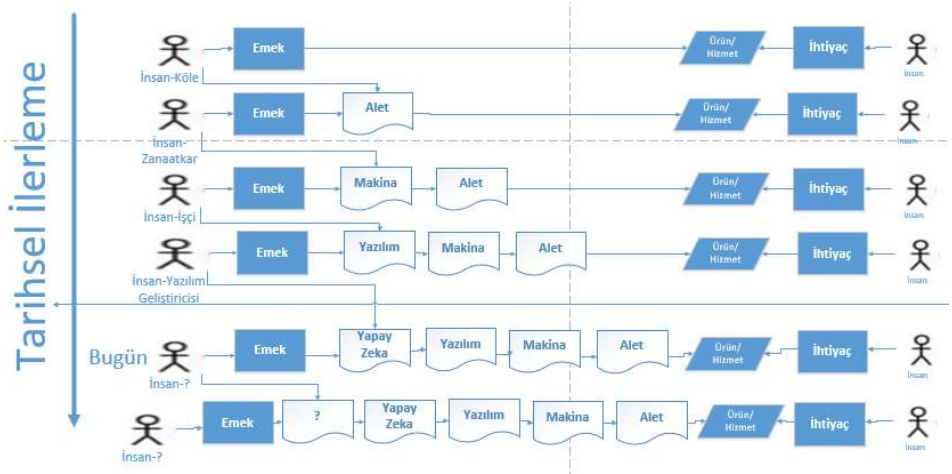
3. MATERYAL VE METOT

3.1. Sorun

Tarihin bilinen ilk zamanlarından günümüze doğru gelindiğinde gelişim hep emek-ihtiyaç ikilisine dayalı ilerlemiştir. Belirli ihtiyaçlar belirli emeklerle karşılanmaktadır. Bu durum geçmişte de, bugün de aynıdır. Fakat daha az emekle daha çok ihtiyacın karşılanmasını sağlayacak kritik kırılımlar genellikle insanın yaptığı işi yapabilecek daha verimli bir alternatif icat edildiğinde gerçekleşir. Bu icatlar devrim niteliği taşıyabilmektedir ve tüm insanlık tarihiyle son 200 yıl kıyaslandığında bu kırılımların daha çok olduğu görülmektedir.

Ardışık olarak tekerlek ve benzeri icatlar köleliği bitirecek ilk süreci başlatmıştır. Bu süreç temelinde Şekil 3.1. üzerinde görüldüğü gibi ilerlemektedir ve gelişimi çok ağır işlese de sonunda sanayi devrimi ve fabrikasyonun yaygınlaşmasıyla kölelik tamamen kaldırılmıştır. Burada "fabrikasyonun yaygınlaşması köleliğin kalkmasını sağlamıştır" gibi direk olarak bağlantı olduğuna dair bir tespit yoktur, sadece tüm insanlık tarihi içinde iki olayın çok yakın zamanda gerçekleştiğine dair bir tespitte bulunulmuştur.

Yazılımsal sistemlerle beraber memurluk, bankacılık ve benzeri otonom süreçleri gerçekleştiren personeller de git gide azalmaktadır. Bugünse, gelecekte yazılım geliştiricilerin, güvenlik sistemlerinde bulunan kişilerin ve benzeri personellerin yerlerini yavaş yavaş bunu yapabilen yapay zeka gibi sistemlerin alacağı görülmektedir. Bu sebeple bugün otonom teknolojilerle yazılım geliştirme süreçleri DevOps gibi yaklaşımlarla ilerletilmeye başlanmıştır. Burada test, kalite, güvenlik, operasyon gibi alanlardaki personellerin işleri DevOps sistemleri tarafından otomatik sağlanmaktadır. Fakat hala DevOps/DevSecOps süreçlerinin belirli alanlarında standart yapılara dayanılarak statik parametreler kullanılmaktadır ve bu alanlar otonom olarak üretilen dinamik parametreler haline getirilmelidir.

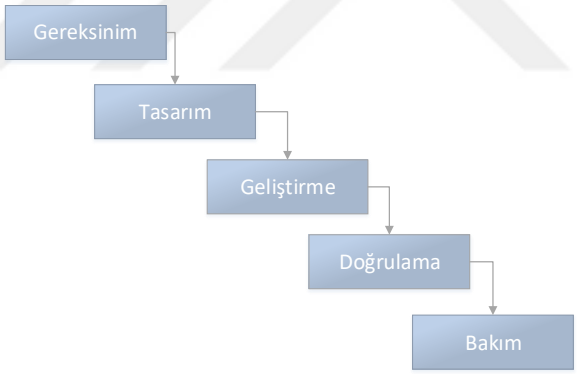


Şekil 3.1. İnsanlık Gelişim Sürecinde Emek ve İhtiyaç İkilisinin Bağlantı Kırılımları

3.1.1. Yazılım Geliştirme Süreçlerindeki Temel Sorunlar

Günümüze kadar birçok yazılım geliştirme süreci ortaya çıkmıştır. Günden güne yazılım geliştirme süreçleri personelin niteliğine, müşterinin niteliğine ve teknolojinin yapısına göre şekillenmiştir. Geçmişte yazılım sisteminin yapması gereken işin net olarak belirlendiği ve bu netliğe göre o işi karşılayacak yazılımı ortaya çıkarmaya yönelik geliştirme süreç modelleri kullanılmaktaydı. Bu nedenle geçmişte önceliğin ağırlıklı olarak doğru analizde olduğu sürekliliğin ise geri plana atıldığı süreçler ön plandadır.

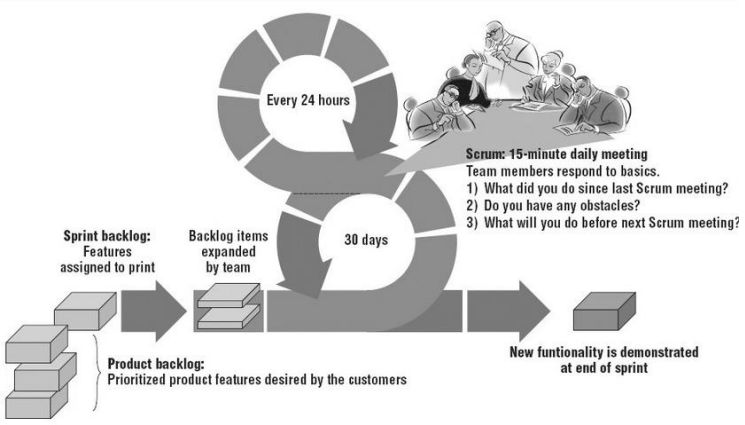
Rekabet ve gereksinimlerin değişkenlik gösterememesinin yanında sistemlerin bugüne göre daha az kompleks olması da geçmişteki yazılım geliştirme modellerinin ortaya çıkmasında rol oymaktadır. Geçmişteki modellere bakıldığında Şekil 3.2. üzerinde görüldüğü gibi şelale modeli veya spiral, V gibi modeller ön plandadır. Bu modeller daha lineer ve tek bir hedef doğrultusunda geliştirildiğinden esnekliği geri plana atan bir yaklaşımı doğurmuştur. Öyle ki uzun bir geliştirme sürecinden geçtikten sonra ara verilip yeni özellikler eklenen bir yazılımın geçmişe dönük olarak düzeltmelerini yapmak yerine, sıfırdan yazılımı geliştirmeye başlamak zaman maliyeti açısından daha tercih edilebilirdir. Tek atım ”One Shot” olarak adlandırılmalarının sebebi de esneklik faktörünün göz ardı edilmesine yönelik başlıca göstergedir[6].



Şekil 3.2. Geçmişte Yaygın Olarak Kullanılan Şelale Modeli

Analiz, Geliştirme, Sistem, Altyapı, Güvenlik ve Kalite gibi gereksinimler geçmişteki yazılım ihtiyacının değişmesinin de etkisiyle farklılaşmıştır. Sürekliliği olan modeller ön plana çıkmıştır. Geçmişteki yazılımlar belirli ihtiyaçları otonom olarak sağlıyordu. Fakat bugün hedef, otonom olarak ihtiyacı karşılayan yazılımları geliştiren yazılımsal sistemleri geliştirmektir. Bu durumda da eski modeller yetersiz kalmaktadır. Gereksinimler değişmiştir, bu gereksinimleri karşılayacak farklı yazılım geliştirme süreç modelleri ortaya konmuştur ve bu modeller ile ilerlenmektedir.

Birbirine entegre altyapıların oluşturulduğu, denetimlerin otomatik sağlandığı, konfigürasyonların durum ve şartlara göre anlık olarak değişebilmesini otonom olarak sağlayan yaklaşımın yaygınlaşması sonucunu DevOps felsefesi doğmuştur. Bu sonuç beraberinde güvenliğin ön plana alındığı DevSecOps süreçlerini de beraberinde getirmiştir. DevOps ve DevSecOps Analiz, Geliştirme, Sistem, Altyapı, Güvenlik ve Kalite noktasında geçmişe göre çok daha farklı sorunlara çözüm bulmaya odaklanmaktadır. Bu odaklanma sürecini de çevik olarak şekil 3.3. üzerindeki gibi çevik süreç modelleriyle ilerletmektedir.



Şekil 3.3. Günümüzde Daha Çok Kullanılan Scrum Modeli [7]

Analiz Kaynaklı Sorunlar

Yazılım geliştirme süreçlerinde her şeyin başlangıcı analizdir. Doğru analiz ancak doğru yazılımsal sistemi ortaya çıkarmaktadır. Bu sebeple analiz süreçlerinin sağlıklı, doğru, açık ve net olarak yapılıp, dokümantasyon haline getirilerek ilgili ekiplere dağıtılması bütüncül sürecin işleyişi için kritiktir. Fakat geçmişte bu analiz hatalarının ve yanlış anlaşılmaların gerektirdiği sebeplerden dolayı yazılımda değişiklik yapılırken, bugün değişime uyumluluk temel bir gereksinim haline gelmiştir. Analizler bu sebeple anlık olarak değişmektedir.

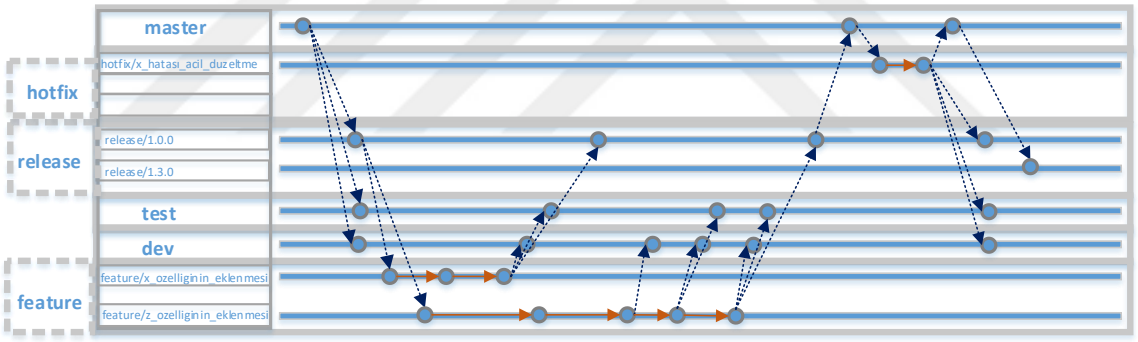
Değişime uyum gereksinimi geliştirme, kalite, güvenlik, sistem, denetim ve yapısal kriterlerinin sürekli değişmesi sonucunu da beraberinde getirmiştir. Buradaki sürekli geliştirme yaklaşımının temel felsefesi, analizin sürekli değişiminden dolayıdır. Günümüzde kullanılan geliştirme süreç modeli analiz sürecinin değişimini gereksinim olarak kabul etmesinden dolayı yazılım mimari modelinin de sürekli gelişim sorununu doğurmuştur. Analiz ve gereksinim; kısa bir süre önce Linux tabanlı 16 GB RAM'e sahip bir sunucunun kapalı bir ağ içinde çalıştığı ve aylık olarak deployment sağlanan bir mimariye uygunken, kısa bir süre içinde değişip 8 GB RAM'e sahip açık ağda çalışan ve günlük deployment sağlanan bir mimariyi gerektirebilir.

Geliştirme Kaynaklı Sorunlar

Analizin belirlendiği bir durumda, yazılım geliştirme süreçlerine başlamak için yeterli kabul edilebilecek bir zemine kavuşmuş olur. Bu zeminde artık bina inşa etmek üzere yapısal çalışmalar başlayabilir. Fakat bugün felsefi olarak bina gibi yazılım yapmak sonunda çok büyük yıkımları da getirmektedir. Bu sebeple geliştirmede de esneklik çok büyük önem kazanmıştır.

Geliştirme ekiplerinin personellerinin görevleri, etki alanları ve ekipleri sürekli değişebilmektedir. Kısa bir süre içerisinde gerçekleştirilmek üzere hedefler belirlenir ve diğer her şey göz ardı edilerek bu hedeflere odaklanılıp belirlenen süre içerisinde işler ekip olarak tamamlanır. Geçmişte versiyonlama sistemleri olan TFS ve SVN gibi yapılar çok lineer kalmaktadır. Çok yönlü ve birbiriyle entegre olabilen modüler versiyonlama bugünün ihtiyaçlarını çok daha ciddi oranda karşılamaktadır.

Bugün geliştirme sürecindeki versiyonlamanın nasıl olacağı, hangi noktada ne gibi kontrollerin gerçekleşmesi gerektiğinin deploymenta yönelik onay kriterlerinin neler olacağı da değiştiği için bunlara da entegre olabilen yazılımlar ortaya çıkmıştır. Şekil 3.4. üzerinde görüldüğü gibi git ve benzeri sistemler daha esnek olarak ihtiyacı karşılamaktadır.



Şekil 3.4. git flow tabanlı geliştirme ve versiyonlama sistemi örneği

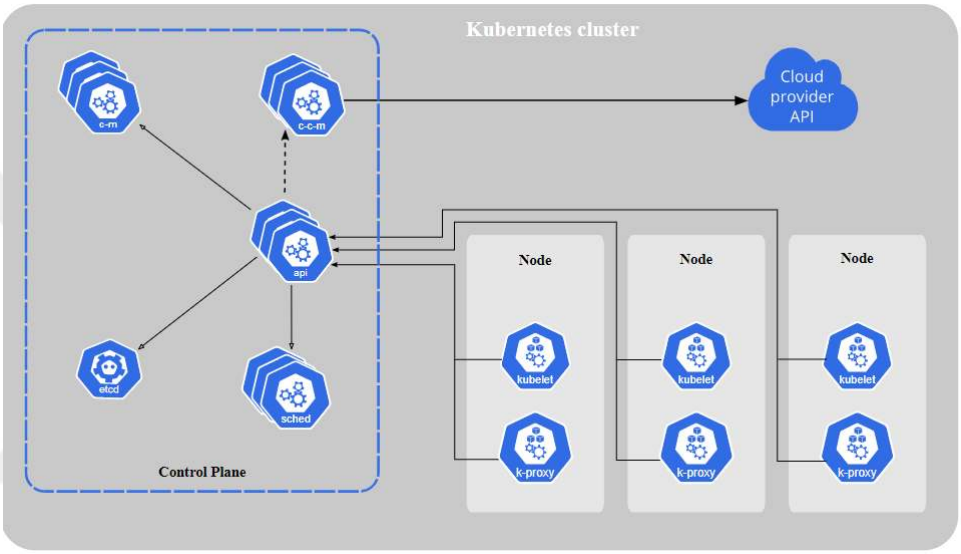
Sistem ve Altyapı Kaynaklı Sorunlar

Bugünkü yazılım sistemlerinin gereksinimlerine göre sistem ve altyapıları sürekli olarak değişkenliklere dirençli olabilmelidir. Aynı yazılımsal sistemin CPU ihtiyacının yükselmesi de, kullanıcı yükünün bir anda artması da muhtemeldir. Bunun yanı sıra tüm altyapı da değişebilir.

Sistemde sorun yaşanması durumunda anlık olarak sorun giderilmeli ve kullanıcılar tarafından bu sorun hissedilmemelidir. Bu gereksinimlerden dolayı geçmişteki hantal sanal sunucu ve raid sistemlerinin yerine daha bütüncül bir sanallaşmayı sağlayan docker ve kubernetes gibi teknolojiler yaygınlaşmıştır. Bu sayede hem verimli hem de esnek bir altyapı ihtiyacı karşılanmıştır. Bu altyapı ise çok esnek olmasından dolayı sistemin karakteristiğinin ve davranışlarının ancak iyi analiz edilmesiyle doğru olarak çalışan kompleks konfigürasyonları beraberinde getirmiştir.

Kompleks konfigürasyonlar güvenliğe yönelik denetim ile bütüncül sağlanmalıdır. Burada güvenlik gereksiniminin net olarak anlaşılması için zararlı olabilecek kullanıcı gruplarının trafiklerine yönelik bilgiler toplanmaktadır. Bu bilgi toplama sürecini genel olarak kurumlarda siber güvenlikten sorumlu personeller takip eder, toplar ve doküman halinde raporlayıp ilgili operasyona yönelik personele sunar.

Bu güvenlik gereksinimlerine göre altyapıyla uyumlu konfigürasyonlar örnek olarak Şekil 3.5. üzerindeki kubernetes örneğindeki gibi sürekli olarak değişebilmektedir. Fakat bu sürekli değişimin içindeki manuel süreçler genel süreci yavaşlatmaktadır.



Şekil 3.5. .Kubernetes Altyapı Örneği [8]

Kalite Kaynaklı Sorunlar

Bugün alt süreçler arttığı gibi kalite kriterleri de artış göstermiştir. Daha kompleks proje yönetim süreçleri, deployment süreçleri, karmaşık alt yapılar, sürekli deployment sağlanma gerekliliği, sunulan ürünün belirli testlere tabi olması ve ürünün yükten bağımsız her ortamda stabil çalışması gibi birçok göz ardı edilemeyecek kriter oluşmuştur. Bu artış ve gereksinimler manuel olarak takip edilebilir olmadığı gibi uyumlu ve esnek olarak sağlanması gerekmektedir.

Kalite ve Hız: Bir öncesine göre her zaman daha hızlı ve kaliteli çıktı üretilebilmelidir. Kalite, ihtiyaçları en doğru şekilde karşılama oranıdır. Doğrulama ve Geçerleme ne kadar yüksek ise kalite o kadar yüksektir.

Doğru/Geçerli Ürün Sağlama: Doğrulamada geliştirilen ürünün doğru bir şekilde geliştirilmesi dikkate alınır. Hız, güvenlik, kod karmaşıklığı ve benzeri standart kalite metriklerine dikkat edilir. Geçerleme ise istenilen ürünün üretildiğine yönelik kontroldür. Bu iki kontrol birbirlerinden bağımsızdır.

Çok hızlı ve güvenli bir yazılım, müşterinin isteklerini sağlamıyor ise bu doğru bir yazılım olsa dahi geçerli değildir. Müşterinin istediği gereksinimlere uygun fakat çok yavaş çalışan, güvenlik açıkları olan ve kullanıcıları ciddi olarak bekleten bir sistem ise geçerli fakat doğru olmayan bir sistem olacaktır. Doğru ve Geçerli ürünün sağlanması da DevOps felsefesinin ilkelerindendir.

Bu noktada kalite ve hız kriterinin doğru ve geçerli bir ürün olarak sağlanmasına yönelik birçok otonom denetim gerekmektedir. Bu denetimlerin doğru metriklerle belirlenmesi kalitenin artışında ve azalışında etkili olacaktır.

Doğru metrik kavramı çok önemlidir, çünkü herhangi bir metriği çok yüksek veya düşük olması değil, doğru olması kalite için önemlidir. Bu sebeple doğru metrik seçimi birçok yapıya yönelik kriterleri sağlayarak seçilmelidir.

Bu seçim manuel olarak statik şekilde verildiği ve kalite kriterleri de sürekli değiştiği için sonradan kaliteye yönelik analiz yapılıp doğru metriğin seçilip değiştirilmesi gerekmektedir. Fakat bu çok uzun bir süreç olduğu için yapılamamaktadır ve kalite tam olarak sağlanmamaktadır.

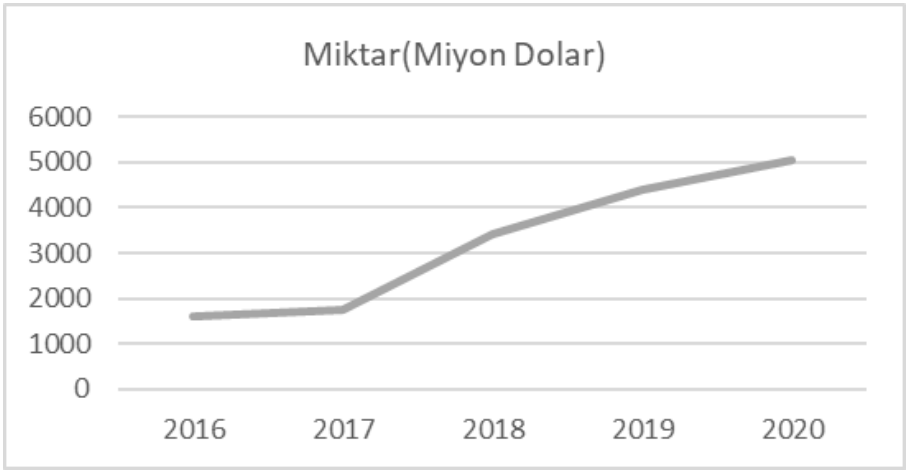
3.1.2. Güvenliğin önemi

Yazılım sistemlerinin içerdiği değerli bilgilerin artması ve yasadışı hacking sektörünün bu etkiyle her geçen gün büyümesiyle güvenlik daha ön plana alınarak, daha kompleks önlemler ve prosedürler kullanılmaya başlanmıştır. Fakat Tablo 3.1. üzerindeki toplam suçların yıllık artışından elde edilen Şekil 3.6. üzerinde de görüldüğü gibi siber suçların son yıllardaki ciddi miktardaki artışı ciddi bir güvenlik sorunu olduğunu göstermektedir.

Bu sorunlar 0. Gün açıklarına hazır olmayan sistemler ve 0.gün açıklarını bulmak üzere çalışan kötü niyetli kitlenin artışıyla doğru orantılı olarak artmıştır. Yeni bir açık türü tespit edildiğinde güncellemelerin gelebilmesi için açık türünün iyi anlaşılması ve testlerden geçmesi ardından ilgili tüm sistemlerin de o güncellemeyi kendi altyapısında sağlaması gerekmektedir. İlgili zafiyet o güne kadar güncelleme yapmayan tüm sistemlerde sömürüye açık olarak kalacaktır. Bu durum ciddi bir sorun teşkil etmektedir.

Tablo 3.1. Son 5 Yılda Suç Şikayet Merkezi Yıllık Raporlarına Göre Siber Suç Türüne Göre Toplam Kaybedilen Para (Crime Complain Center Annual Reports. 2020,2019,2018.2017,2016[9])

Siber Suç Tipi	2016	2017	2018	2019	2020
E-posta Dolandırıcılığı	360.513.961,00	676.151.185,00	1.297.803.489,00	1.776.549.688,00	1.866.642.107,00
Güven Dolandırıcılığı/Romantizm	219.807.760,00	211.382.989,00	362.500.761,00	475.014.032,00	600.249.821,00
Yatırım	138.228.282,00	141.110.441,00	252.955.320,00	300.478.433,00	336.469.000,00
Ödeme Yapmama/Teslim Edilmeme	123.407.997,00	96.844.144,00	183.826.809,00	222.186.195,00	265.011.249,00
Kimlik Hırsızı	95.869.990,00	77.134.865,00	149.458.114,00	221.365.911,00	219.484.699,00
Yanıltma	73.092.101,00	66.815.298,00	148.892.403,00	196.563.497,00	216.513.728,00
Emlak/Kiralama	60.484.573,00	60.942.306,00	117.711.989,00	160.305.789,00	213.196.082,00
Kişisel Veri İhlali	59.139.152,00	57.861.324,00	100.429.691,00	124.292.606,00	194.473.055,00
Teknik Destek	58.917.398,00	57.207.248,00	92.271.682,00	120.102.501,00	146.477.709,00
Kredi kartı dolandırıcılığı	57.688.555,00	56.231.333,00	88.991.436,00	111.491.163,00	129.820.792,00
Kurumsal Veri İhlali	56.004.836,00	53.450.830,00	83.357.901,00	107.498.956,00	128.916.648,00
Devlet Kimliğine Bürünme	48.187.993,00	38.883.616,00	70.000.248,00	100.602.297,00	109.938.030,00
Diğer	47.875.765,00	29.703.421,00	64.211.765,00	66.223.160,00	101.523.082,00
Gelişmiş Ücret	40.517.605,00	23.853.704,00	63.126.929,00	57.836.379,00	83.215.405,00
Gasp	31.679.451,00	16.835.001,00	60.214.814,00	55.820.212,00	70.935.939,00
İş	22.005.655,00	15.302.792,00	53.225.507,00	54.041.053,00	62.314.015,00
Piyango/Çekilişler/Miras	21.283.769,00	14.810.080,00	48.241.748,00	53.398.278,00	61.111.319,00
Kimlik Avı/Vishing/Smishing/Pharming	15.811.837,00	14.580.907,00	45.487.120,00	48.642.332,00	54.241.075,00
Fazla ödeme	13.725.233,00	12.569.185,00	38.697.026,00	42.618.705,00	51.039.922,00
Fidye yazılımı	12.278.714,00	12.467.380,00	21.903.829,00	20.242.867,00	29.157.405,00
Sağlıkla İlgili	11.213.566,00	5.766.550,00	20.000.713,00	19.866.654,00	29.042.515,00
Sivil mesele	7.806.416,00	5.536.912,00	15.802.011,00	12.371.573,00	24.915.958,00
Yanlış beyan	6.829.467,00	5.003.434,00	15.172.692,00	10.293.307,00	19.707.242,00
Kötü Amaçlı Yazılım/Korku Yazılımı/Virüs	3.853.351,00	2.344.365,00	7.411.651,00	8.965.847,00	6.904.054,00
Taciz/Tehdit Şiddet	2.431.261,00	1.466.195,00	4.474.792,00	7.598.198,00	6.547.449,00
IPR/Telif Hakkı/Sahte	1.932.021,00	1.405.460,00	*3.621.857	2.214.383,00	5.910.617,00
Hayır Kurumu	1.660.452,00	925.849,00	2.052.340,00	2.009.119,00	4.428.766,00
Kumar	1.635.321,00	809.746,00	1.684.179,00	1.772.692,00	3.961.508,00
Yeniden Nakliye	995.659,00	598.853,00	1.006.379,00	1.458.118,00	3.095.265,00
Çocuklara Karşı Suçlar	290.693,00	46.411,00	926.953,00	1.128.838,00	660.044,00
Hizmet Reddi/TDos	219.935,00	20.147,00	265.996,00	975.311,00	512.127,00
Hacktivist	55.500,00	79.173,00	77.612,00	129.000,00	50,00
Toplam	$1.595.444.269,00	$1.758.141.144,00	$3.412.183.899,00	$4.384.057.094,00	$5.046.416.677,00



Şekil 3.6. Son 5 Yılda Suç Şikayet Merkezi Yıllık Raporlarına Göre Toplam Siber Suç Üzerinden Kaybedilen Toplam Para (Crime Complain Center Annual Reports. 2020,2019,2018.2017,2016[9])

3.1.3. Owasp Top 10 Hazırlanma Sistemi

OWASP derneği 2003'ten beri birçok aktivitesinin yanı sıra zafiyetlerin gruplandığı ve sıralandığı top 10 listesini 2-3 yılda bir hazırlamaktan sorumludur. Owasp Top 10 listesi bilgi güvenliği sektöründeki gelişmelere ve değişikliklere göre güncellenen bir listedir.

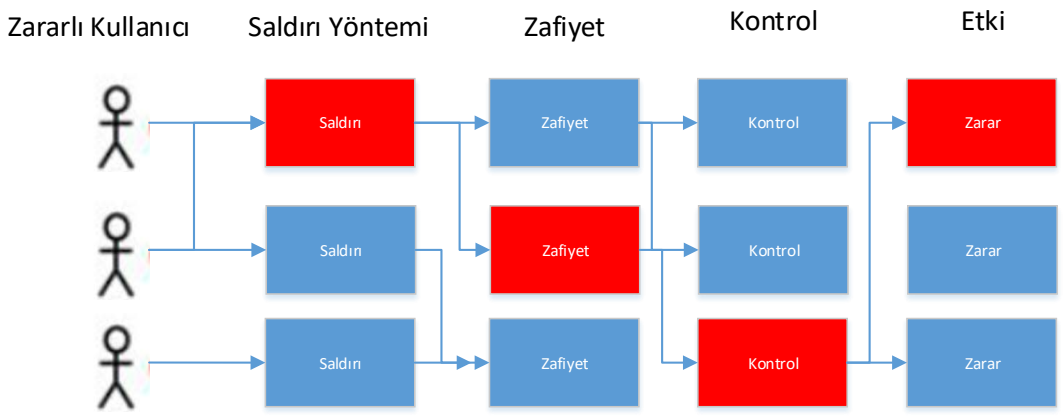
Dünyanın en önemli güvenlik akreditasyon yapılarından biridir. Owasp Top 10 kuruluşlarının çoğu için önemli bir kontrol listesini sağlamayı hedeflemektedir. Bu kontrol listesi sayesinde belirli bir standardizasyon birçok kurum için sağlanmış olacaktır.

Siber Güvenlik personellerine göre genellikle bir kuruluşun Owasp Top 10 ile uyumlu olması, ne düzeyde güvenli bir sisteme sahip olduğuna dair kritik bir göstergedir. Bu sebeple sorgulanmaksızın birçok kurum tarafından direkt olarak Owasp Top 10 kontrolü yapılmaktadır. Ardından zafiyetlerin tespit edilmesi durumunda önlemler sisteme eklenmektedir [5, 13, 11, 10].

Bir Owasp Top 10 listesi birçok faktöre göre oluşturulmaktadır. Bu faktörler farklı sektörden olan kurumların zafiyet istatistikleri ve resmi kurumlar tarafından yayınlanan raporlarıdır. Bu kapsamda zafiyetlerin oluşturdukları zarara göre genel riskleri ve bu sömürünün miktarı, ilgili zafiyetin önem düzeyini belirlemektedir.

Önem düzeyine dair istatistiki verilere göre komite bütüncül bir karar vererek 2-3 yılda bir Owasp Top 10 listesini tamamlar. Kurum Yöneticilerinin Sağladığı İstatistikler: Owasp dahilinde komiteler vardır. Bu komitelerle bağlantılı olan bölgesel sorumlular bulunur.

Bölge sorumlularıyla bağlantılı olan kurum ve güvenlik yöneticilerinin sağladığı istatistiklere dayalı olarak genel bir veri elde etmektedir. Bu veri komiteye düzenli olarak sunulmaktadır.



Şekil 3.7. Saldırı yöntemlerinin öncelik düzeyinin belirlendiği zarar sonuçlarına dair Owasp Komisyonunun Önem Düzey Kriteri[5]

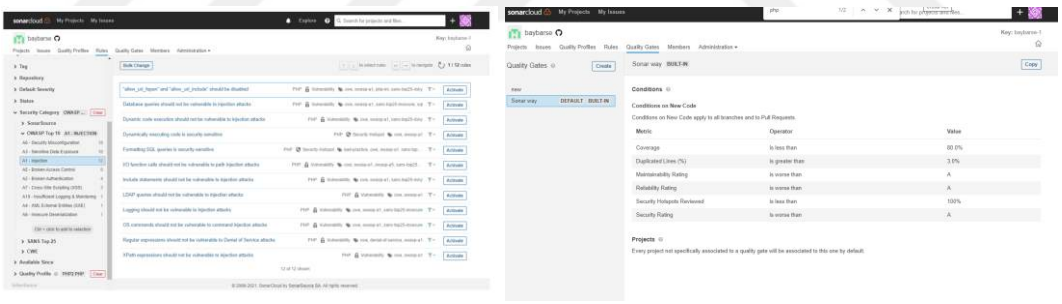
Resmi kurumlar tarafından yayınlanan genel rapor, istatistikler ve komite araştırmaları Owasp Top 10 çok genel ve bütüncül bir yapı haline getirmektedir. Neredeyse herhangi bir şekildeki, herhangi bir web tabanlı yazılım altyapısı kullanan tüm sektörlerdeki kuruluşların aynı zafiyetlerden mustarip olabileceğini kesin kabul ederek Şekil 3.7. üzerinde gösterilen yaklaşıma göre Owasp Top 10 listesini komite hazırlanmaktadır.

Bu durum standardizasyon için önem teşkil etse dahi, her kurum için gerçekçi anlamda en ideal ve doğru denetim metriklerini vermemektedir. Sonuç olarak ise yazılım yöneticileri performanstan feragat etmek ile standardizasyondan feragat etmek arasında kalmaktadır.

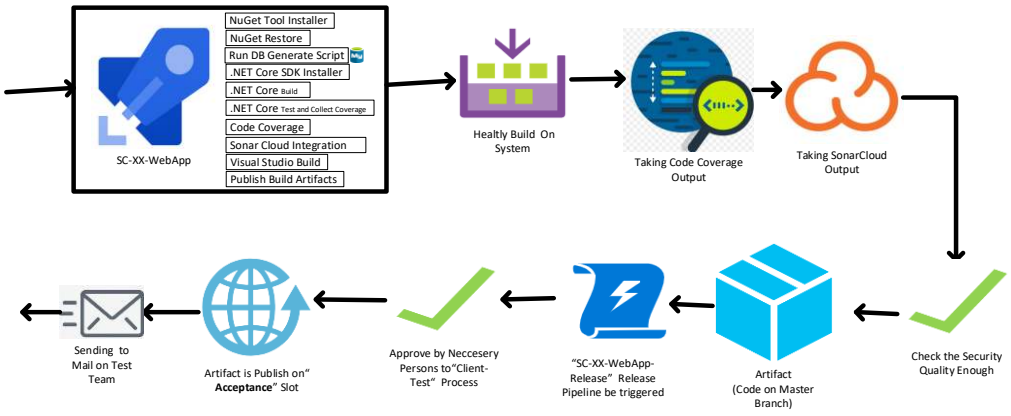
3.1.4. SAST Tabanlı DevSecOps Yaklaşım Yolu ve Hıza Olan Etkileri

SAST statik olarak kodların zafiyetlerinin tarandığı ve kodların güvenlik düzeyine dair çıktı üreten sistemlerin genel adıdır. Günümüzde Sonarcloud gibi birçok SAST sistemi bulunmaktadır. Bu sistemler otomatik veya manuel olarak tarama yaparak belirli bir güvenlik seviyesine göre deployment için olumlu ya da olumsuz çıktı üretirler.

DevSecOps'un diğer görevlerinin yanı sıra bu noktada alacağı karar, bu çıktıların nasıl ve neye göre olacağı ve hangi ortama hangi seviyedeki çıktı düzeyi deployment için yeterli olacağına göre değişmektedir.



Şekil 3.8. Sonarcloud Kural ve Metrik Seviyesi Belirleme Ekranları



Şekil 3.9. DevSecOps akışının SonarCloud kontrolüne yönelik örnek bir bölüm

Şekil 3.8. üzerinde başlangıçta hangi kuralların aktif ve pasif olacağı belirlenmektedir. Bu duruma göre sürekli olarak aynı zafiyet tipleri aynı metrik seviyeleriyle kontrol edilmesi sağlanır. Kontrol başarılıysa deployment Şekil 3.9. üzerindeki örnek olarak verilen akışa benzer bir akışa göre sağlanabilmektedir.

Fakat burada kaliteyi düşüren, değişen yazılımsal yapıya rağmen sabit kalan metrik ve kurallardır. Bunun sonucu olarak Tablo 3.2. üzerinde görüldüğü gibi sadece kurum için kritik olan gerekçeler göz önünde bulundurularak önlem almak yerine, bir süre sonra Owasp'ın kritik olarak belirlediği fakat kurum için kritik olmayan gerekçeler kalmaktadır.

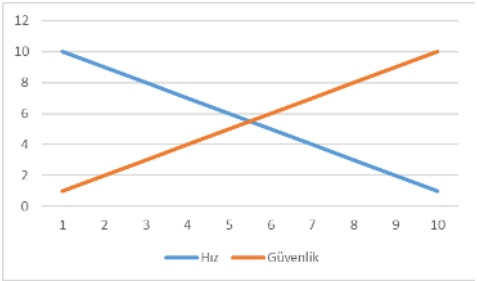
Bunun yanı sıra kurum için önemli olan fakat Owasp için önemli olmayan kriterler zamanla yazılım yapısının değişmesinden dolayı kontrol edilip, gerekmesi durumunda eklenmelidir. Bir süre sonra gerekli önlemlerin olmadığı gereksiz önlemlerin olduğu bir kontrol sistemi zamanla hız performansından feragat edilecektir. Güvenlik seviyesi, sürekli değişen bir yazılım sisteminin çok eski bir versiyonuna göre eski ile aynı metrik değerleri baz alınarak belirleneceği için sonarcloud'a göre olmasa da "gerçek" olarak düşmektedir.

Tablo 3.2. Kurum Riskleri ve Owasp'ın Belirlediği Risklerin Kritiklik Çaprazlama Tablosu

		OWASP Risk Düzeyine Göre Önlemler	
		Kritik	Kritik Değil
Kurum Risk Önlemleri	Kritik	X	X
	Kritik Değil	X	

Bunun manuel çözümü sürekli olarak belirli personellerin zaman ayırıp güvenlik departmanından gelen raporları dikkate alarak sistemi ve trafiği analiz edip metrikleri ve zafiyetlere yönelik kontrolleri düzenlemesidir.

Fakat bu çözüm insan faktörünü barındırdığı için her zaman stabil olması mümkün olmayan ve zaman maliyeti çok yüksek bir çözümdür. [1] kaynağına göre güvenlik ve hız ilişkisi Dinamik olmadığı için güvenlik düzeyini arttırmak aynı oranda hızı düşürmek anlamına gelmektedir. Hız sağlanmak istense bile gerekli seviyede güvenlik sağlanamamaktadır. Bu çelişkiden dolayı kalite sürekli olarak Şekil 3.10. üzerinde görüldüğü gibi en iyi düzeyde sağlanamamaktadır.



Şekil 3.10. Hız ve Güvenlik Önlem İlişkisi

3.2. Yaklaşım

Sorunların özet olarak günümüzde DevSecOps süreçlerinin manuel yürütülmesinden ve statik yapılardan kaynaklanan gerekçelere bağlı sonuçların ortaya çıkması olarak görebiliriz. En büyük sorun yazılım ne kadar değişse de sürekli aynı zafiyetlerin önemli ve önemsiz olarak görülmesiydi. Burada sistemin yapısının düzenli olarak anlaşılması ve hangi zafiyetlerinin önemli/önemsiz olacağına dair verilen kararın sürekli değişken nitelikte olması gerekmektedir. Bu önem düzeyi saldırganların ve saldırıların neler olduğunu bilerek ancak mümkün olabilmektedir.

Saldırganlar ve saldırı türleri değişkenlik göstermekle beraber sisteme verdiği zarar da değişkenlik göstermektedir. Düzenli olarak sistem üzerindeki bu saldırılar verdikleri/verebilecekleri zarara göre güvenlik test sonuçlarını etkileyen metrikler güncellenmelidir.

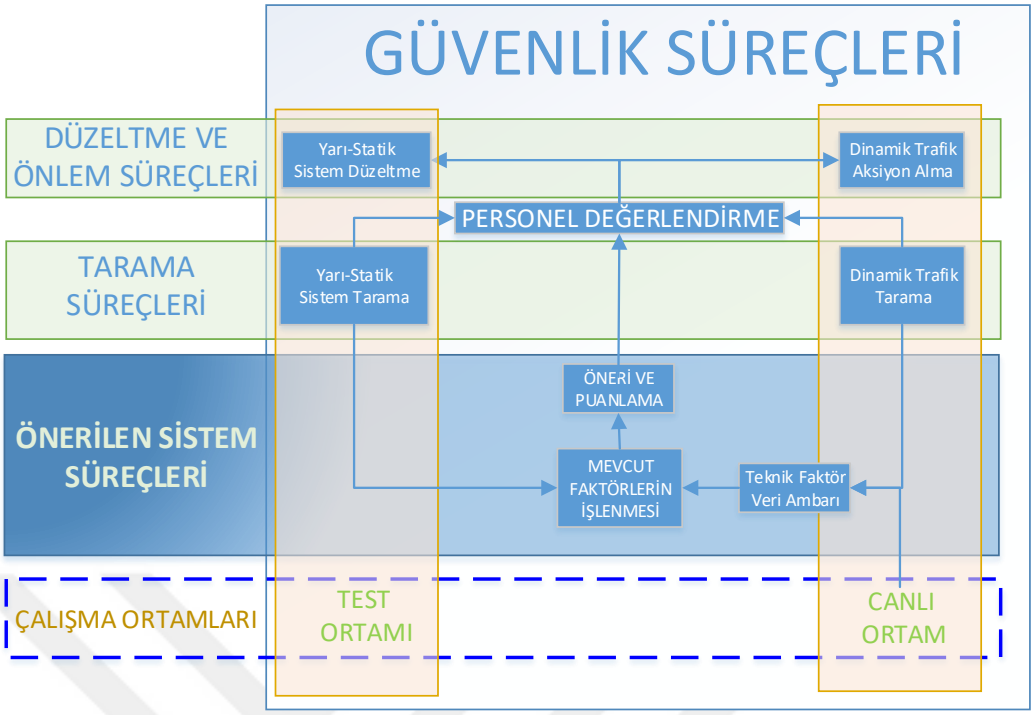
Bu güncelleme manuel olarak çok uzun sürdüğü için daha otonom bir sistem tarafından düzenli olarak güncellenmelidir. Sistem bu zafiyetleri anlık olarak sunucu davranışına göre tanımalıdır. Bu tanıma için belirli aralıklarla davranış değişimlerinin düzeyini tespit etmek üzere zararlı kullanıcılar honeypot benzeri sunucularda gözlenmelidir.

Güvenlik ekipleri tarafından sağlanan zafiyet sömürülerinin olduğu zamanlar ve zafiyet tipinin ne olduğu verisi elde edilmelidir. Bu verilere göre sunucu davranışı ve zafiyet tipi eşleştirilerek, ilerde zafiyet varlığı durumunda aynı davranışın yakalanması sağlanabilecektir. Belirli zaman aralıklarıyla davranışların zafiyetlerle eşleşme düzeyi iyileştirilecektir.

Güncel olarak elde edilen zafiyet ve sömürü düzeyleri geçmişten mevcut zamana kadar olan süreçte güncel veriyi ağırlıklandırarak zafiyetin sömürü oranına göre SAST test sonuçlarıyla değerlendirilebilecektir.

Temel olarak önerilecek yaklaşıma göre bir sistem tasarlanırsa, risk yönetiminin insan denetiminde olacağı göz ardı edilmemelidir. Bu tanımaya göre SAST test sonuçlarını etkileyecek metrik ve zafiyet tipi güncellemenin otonom yapılması çok sağlıklı olmayacaktır. Asıl sorun olan metrik ve zafiyet seçimine yönelik analizin zaman maliyetidir.

Bu maliyetin indirilmesi ve kısa bir sürede kısa bir inceleme yaparak uzun zaman aralıklarıyla zafiyetlerin ve metriklerin güncellenmesinin personel tarafından sağlanabileceği bir öneri sistemi en ideal sistem olarak görülmektedir. Bu sistem temel olarak Şekil 3.11. üzerinde görülmektedir.



Şekil 3.11. DevSecOps Güvenlik Süreçlerine Dair Önerilen Yaklaşım

3.2.1. Sistem Karakteristiğinin Tespit Yöntemi

Sistemin hangi girdilerde hangi çıktılar verdiğini tespit edebiliriz. Sistem üzerinde kullanıcı davranışları sistemin yükünü, yani sistemin girdisini göstermektedir. Buna dayanarak kullanıcı girdilerini sistematik olarak yaparsak bu yük testi olarak adlandırılmaktadır.

Mevcut öneriye dair araştırma verilerini elde etmek için yük testleri girdilerdir, çıktılar ise sunucu loglarıdır. Yük ve loglar arasında doğrusal bir bağlantı bulunmaktadır. Yapılan deneylerde görülmüştür ki yükün artışı, azalışı veya yapısal değişimi log verilerini de arttırır, azaltır ve yapısal olarak değiştirir.

Bununla beraber log verilerinin analiz edilmesi yoluyla kullanıcı davranışlarını tespit etmek sistem zafiyetlerinin hangilerinin sömürüldüğüne yönelik tespiti de sağlamaktadır. Karakteri oluşturan mekanizma aynı zafiyetleri hedef alan kötü niyetli kullanıcılardır. Hangi kullanıcı davranışının hangi sistem davranışını oluşturduğunu anlayabilmek sistemin karakterini anlamak demektedir.

Burada kullanıcı davranışını anlamak hangi zafiyetin sömürüleceğini de anlamak anlamına gelmektedir. Sistem karakteri, sistem altyapısı ve teknolojisi ne kadar farklılık gösterse de değişmeyecektir.

Şekil 3.12. Sistem Davranışının Oluşum Süreci

Değişen sistem, altyapı ve teknolojiden kaynaklanan bir şekilde davranış olacaktır. Örnek olarak aynı yapıdaki process Linux yapısındaki RAM kullanımı, Windows yapısı ile mimari farklılıklardan dolayı aynı olmamaktadır[2]. Bunula beraber dile göre de ciddi bir kaynak kullanım farklılığı aynı algoritmik akışla gerçekleştiren yapılar için farklıdır[3].

Bu farklılıklar birçok teknik farklılıkla orantılıdır. Bu durum göstermektedir ki, zafiyetin tespiti durumunda belirli bir örüntüyü sürekli olarak o zafiyetle eşleştirmek doğru sonucun uzun vadede alınmasına engel teşkil etmektedir.

Bu sebeple 2-4 hafta gibi sürelerde güncelleme yapılarak bu güncelleme sonucundaki veriler ve zafiyet oranları göz önünde bulundurularak öneride bulunan sistem daha sağlıklı çalışmaktadır. Kullanıcı profili ve sistemde hedeflenebilecek sömürü tipleri aynı kalabilecektir.

Sektörel değişimlerde bu değişimde rol oynamaktadır ve içerdiği datalara göre riskleri de değişebilmektedir. Örnek olarak bankacılık sektörü için injection kesinlikle kabul edilemeyecek büyük bir riskken ve pazaryeri uygulamaları için bu kadar büyük bir risk değildir.

3.2.2. Güvenlik Önlemlerinin Belirlenmesi

Sistemin nasıl sömürüleceğinin bilinmesi sonucunda sömürünün ne olduğuna dair veriyi alabiliriz. Bu sonucunda sisteme yönelik ne gibi alanlarda sömürünün olabileceğine dair analiz edilebilir data elde ederiz. Bu noktada datanın sürekli güncelliği kritiktir.

Temelinde sistem profesyonel bir saldırıya hazır olabilecek kadar güvenliyken, dışarıdan en düşük seviyedeki saldırganın bile sisteme sızabileceğine dair bir algının oluşturulması da stratejik önem taşımaktadır. Zaten kolayca sızılabilen bir sistem olduğu için maliyetli ve ağır yöntemler kullanmak gereksiz görüleceği için, profesyonel bir saldırgan dahi olsa ortalama bir saldırıda bulunacaktır. Bu durum sonucunda yapılacak saldırı tipi artık sistem tarafından bilinecektir ve bu zafiyetlere dair önlem alınarak ilerde aynı alan üzerinden çok profesyonelce bir saldırı dahi yapılsa başarısız olması kuvvetle muhtemel olacaktır.

Sistem insan davranışlarının rastsallığıyla paralel bir rastsallıkta güvenlik önlemi almaktadır. Böyle bir sisteme sağlıklı olarak saldırı yapılması için 2 yol bulunmaktadır:1.Dünya üzerindeki tüm saldırganları organize edip belirli bir süre, belirli açıkların sistem ve risk yönetiminden sorumlu kişinin hatasıyla tarafından göz ardı edilmesini sağlayarak o açıkların bir anda sömürülmesini sağlamak. 2.Direk olarak içeriden bir personel ile anlaşmak.

Bu yollar incelenecek olursa 1.Durumun olma ihtimali; dünya üzerindeki tüm insanların aynı ülkede yaşama, aynı düşünce yapısında olma ve aynı şirkette çalışma ihtimalini oluşturan unsurlar bir saldırının olma ihtimalini oluşturan unsurlar ile aynı olduğu için mümkün değildir. Bununla beraber 1. veya 2. durum yine de gerçekleşse dahi mevcut güvenlik sistemleri de bu muhtemel sorunlara hukuki güvenceler dışında çözüm bulamamaktadır, mevcut sisteminde personel sözleşmesi ve benzeri hukuki güvencelerle korunabileceği mümkün olacağı için aynı şartlar altında önerilen sistem daha makuldür.

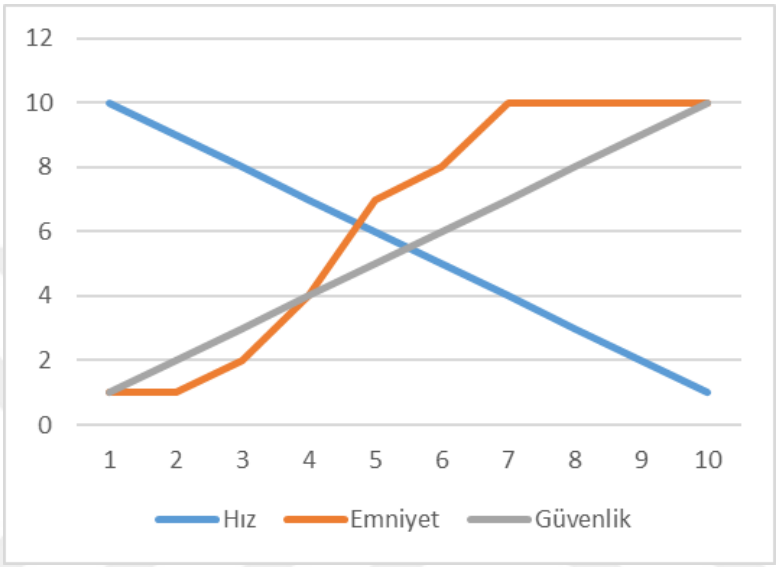
Önerilen güvenlik yaklaşımına göre Owasp için önemli olanlara değil, Owaspa göre sınıflanmışlara bakılıp sistemde sömürülenlerin sömürü seviyesine göre ne gibi alanlarda kurum hedef alınmakta olduğu anlaşılmalıdır. Bu hedef alandaki bir 0. Gün açığı durumunda zincirleme risklere yol açmaması için kritik olarak görülmelidir.

Sisteme yönelik kötü niyetli kullanıcı grubu 0.gün zafiyetlerine yönelik alanda daha çok risk teşkil etmektedir ve çalışmaktadır[4]. Bununla beraber Owasp'ın kritik gördüğü ve kurum içinde risk grubunda olmayan yapılar bulunabilmektedir.

Güvenlik yöneticileri risk ve sömürü durumuna göre analiz yaparak Owasp için kritik görülen zafiyetleri kritik seviyeden çıkarabilmektedir. Tablo 3.3. üzerinde görüldüğü gibi bir yaklaşım önerilmektedir. Bu yaklaşım emniyeti arttırırken performansı da optimum düzeyde tutmaktadır.

Tablo 3.3. Önerilen Yaklaşım

		OWASP Risk Düzeyine Göre Önlemler	
		Kritik	Kritik Değil
Kurum Risk Önlemleri	Kritik	X	X
	Kritik Değil		



Şekil 3.13. Güvenlik, Emniyet ve Hız İlişkisi[1]

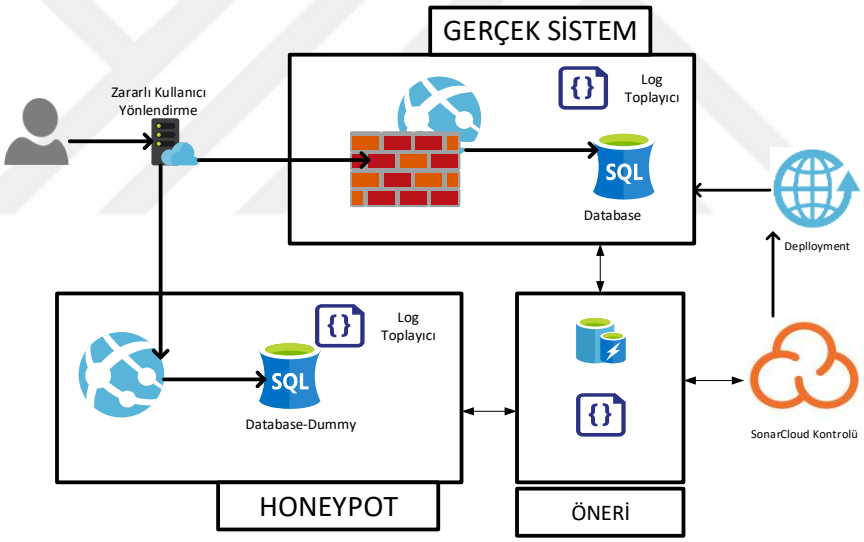
[1] kaynağına göre emniyet önemlidir ve güvenliği üst düzeyde tutmak yerine emniyet dikkate alınmalıdır bu yolla Şekil 3.13. üzerinde görüldüğü gibi performans arttırılmalıdır. Önerilen sistem Şekil 3.12 ile de görülen yaklaşımına göre izlenen sistem 10 düzeyinde emniyeti gerekli güvenlik önlemlerini 7 düzeyinde alarak performansı en yüksek seviyede en doğru güvenlik önlemiyle sağlamaktadır.

3.2.3. Mimari Yapı

Topolojik Mimari

Sistem başlangıçta veri toplayabilmek amacıyla zararlı kullanıcıları ve hangi zafiyeti sömürmek üzere istek yaptığını tespit edebilir, fakat bu kullanıcılar Honeypot'a yönlendirilirler. Bu kullanıcıların zafiyet sömürü davranışlarının karşılığı olan log davranışları kayıt edilir. Ardından bu loğların ve zafiyetlerin zaman damgaları kıyaslanarak analiz edilir. Bu sayede sistem davranışı başlangıç aşaması için tespit edilmiş olur.

Farklı bir yaklaşım yolu olarak tüm zafiyetler bilinçli bir şekilde stabil bir sisteme uygulanır. Bu uygulama sayesinde hangi zafiyet olsa da, zafiyetin sömürüsü durumunda sistemin nasıl davranıldığı tespit edilebilecektir. Normal kullanıcı davranış kayıtları da zafiyet sömürü kayıtlarıyla orantılı olarak yük olarak verildiğinde normal bir trafiğin olduğu sunucuda zafiyetlerin ayrılabilmesine yönelik tespit bu yolla yapılabilecektir. Deneyler genel olarak bu yolla yapılıp analiz sağlanacaktır, fakat önerilen topoloji Şekil 3.14 şeklindedir.



Şekil 3.14. Önerilen Topoloji

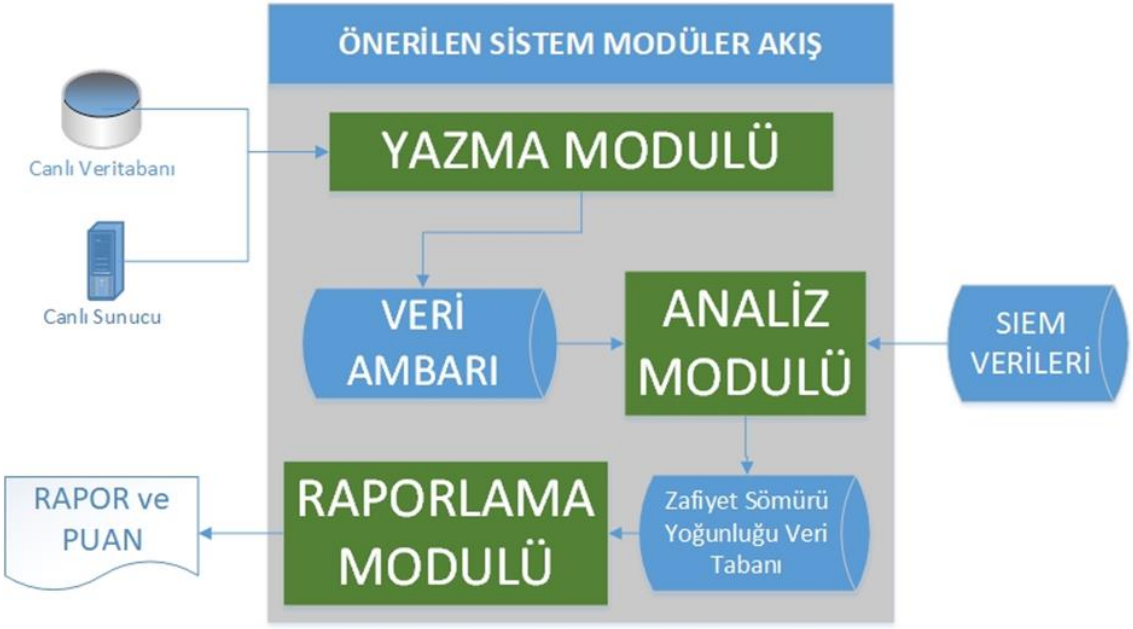
Şekil 3.14. üzerinde honeypot aracılığıyla davranış logları 10 saniyede bir toplanmaktadır. Ardından önerilen sisteme günlük olarak yollanmaktadır. Önerilen sistemde honeypot davranış veriler anlaşılır olarak işlenmektedir. Sonuç olarak zafiyetlerin olması durumunda hangi zafiyetin ne gibi kaynak kullanımına tekil olarak sahip olduğu tespit edilir ve saklanır. Gerçek sistem üzerinden de davranış verileri öneri sistemine alınır, anlaşılır olarak saklanılır.

Haftalık olarak gerçek sistemden alınan davranış verileri, işlenen honeypot davranışlarına göre yorumlanır ve hangi zafiyetin sömürülmek istendiğine dair istatistikler elde edilir. Genel olarak deployment sürecindeki sonarcloud kontrolünde bu istatistikler dikkate alınarak güvenlikten sorumlu personellere öneri verebilecek bir sistem oluşmuş olmaktadır.

Algoritmik Yapı ve Veri Akış Mimarisi

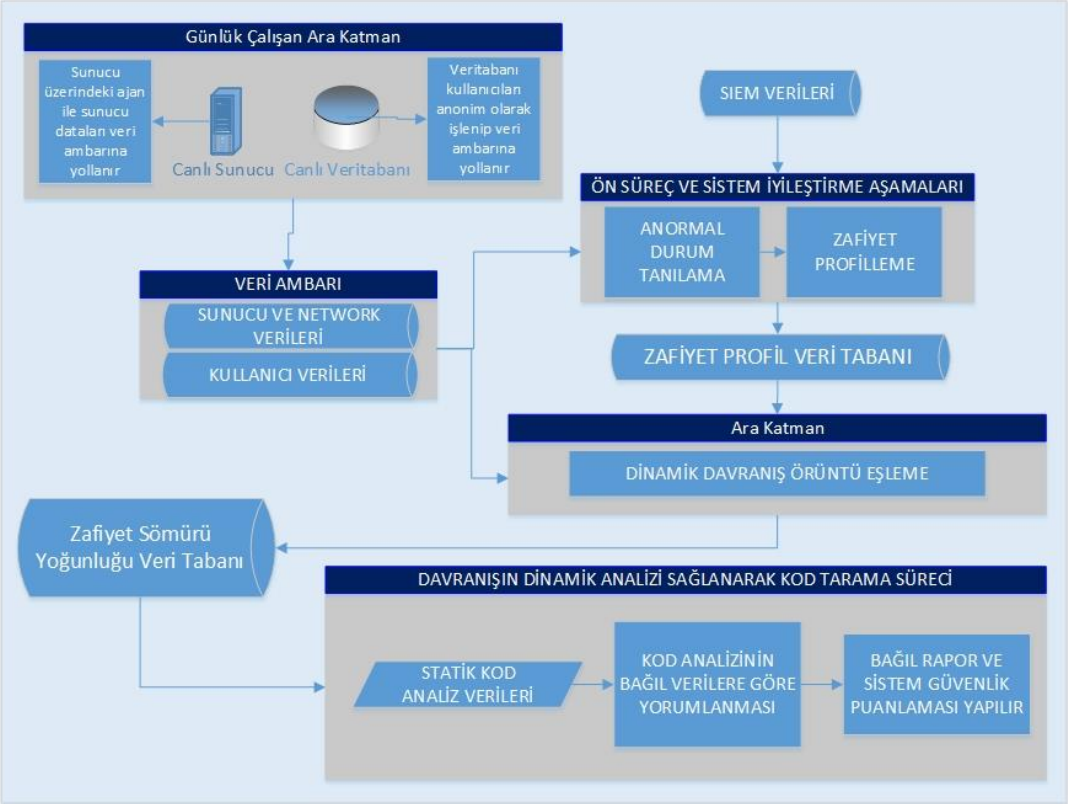
3 adet temel modül bulunmaktadır. Önerilen sistemde bunlar yazma, analiz ve raporlama modülleridir. Yazma ve Raporlama modülleri basit ve düzenli çalışan scriptlerden meydana gelmektedir. Bu 2 modül kompleks değildir, sadece belirli dizinlere yazma/okuma yapmaktadır ve belirli sunuculara yollama/alma istediğinde bulunmaktadır.

Önerilen sistemin temeli Analiz modülü üzerinedir. Analiz modülü içerisinde birçok veri ambarının verilerine Şekil 3.15. üzerinde görüldüğü gibi filtreleme, dönüşüm, sınıflandırma algoritmaları uygulanmaktadır. Belirli zaman aralıklarında Şekil 3.17. üzerinde görüldüğü gibi iyileştirme yapmaktan da sorumludur.



Şekil 3.15. Önerilen Sistemin Modüler Akışı

Şekil 3.15. modüllerinin iç akışları 3.3.Veri Ambarı Oluşturma ve 3.4.Zafiyet Sömürü Yoğunluk Verilerinin Çıkarılması bölümlerinde detaylı olarak açıklanacaktır. Anormal Durum Tanısı bölümde Şekil 3.17. üzerindeki akışa yönelik analizler gerçekleştirilecektir.

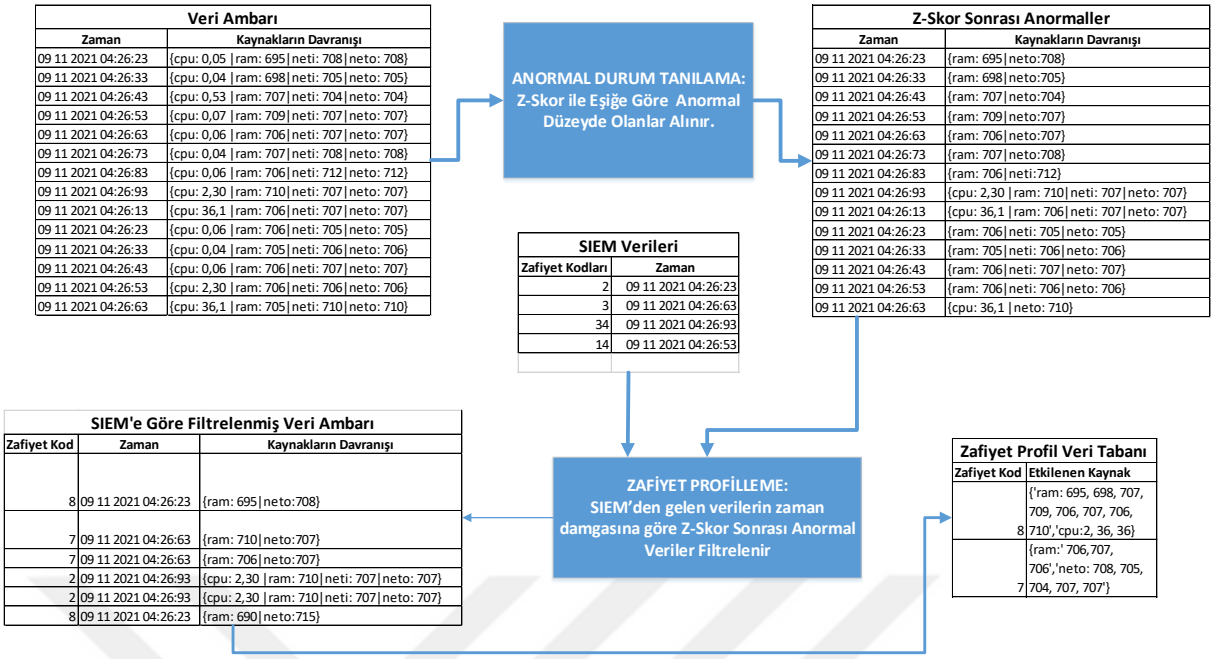


Şekil 3.16. Analiz Modülü Detaylı Veri Akışı

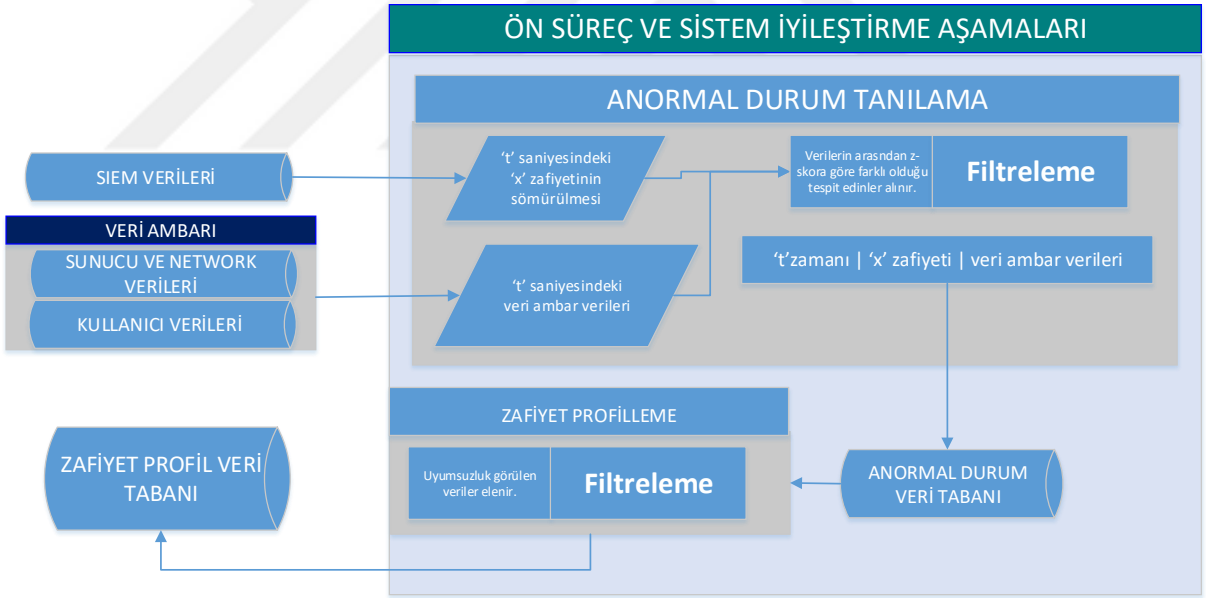
Şekil 3.16. ya göre günlük olarak çalışan bir ara katman olarak tanımlanan yapı üzerinde sürekli bir ajan çalışmaktadır. Bu ajan veri ambarlarına gerekli görülen verileri kayıt eder. Bu veri ambarlarındaki veriler ön süreç ve sistem iyileştirme modülü aracılığıyla anormal durum tespiti SIEM üzerinden alınan zafiyet ve zaman damgası verileri kullanılarak yapılır.

Buradaki anormal çıkarımı için durum için z-skordan faydalanılacaktır. Bu anormal durumlar ayrılarak hangi zafiyet ile daha çok eşleştiğini anlamak üzere eşik değerlerine göre en faydalı kaynaklar ayrıştırılacaktır. Bu aşamaya zafiyet profilleme aşaması olarak isimlendirilecektir. Bu zafiyet profili "ZAFİYET PROFİL VERİ TABANI"na kayıt edilir. Bu sayede zafiyetin gerçekleşmesi durumunda hangi kaynağın nasıl davrandığına dair örüntü eşleştirmesi yapılabilecektir. Bu eşleştirme için Nokta çift serili korelasyon yönteminden faydalanılacaktır.

Bu kullanım "Ara Katman" olan "DİNAMİK DAVRANIŞ ÖRÜNTÜ İŞLEME" üzerinde veri ambarındaki güncel veriler kullanılarak yapılmaktadır. Sonrasında "Zafiyet Sömürü Yoğunluğu Veri Tabanı"na kayıt işlemi gerçekleştirilecektir. "Zafiyet Sömürü Yoğunluğu Veri Tabanı" hangi zafiyet tipinin ne kadar sömürüldüğüne yönelik verileri içermektedir. Ardından güvenlik personelleri faydalanmak üzere "DAVRANIŞ DİNAMİK ANALİZİ SAĞLANARAK KOD TARAMA SÜRECİ"ni gerçekleştirebilecektir.

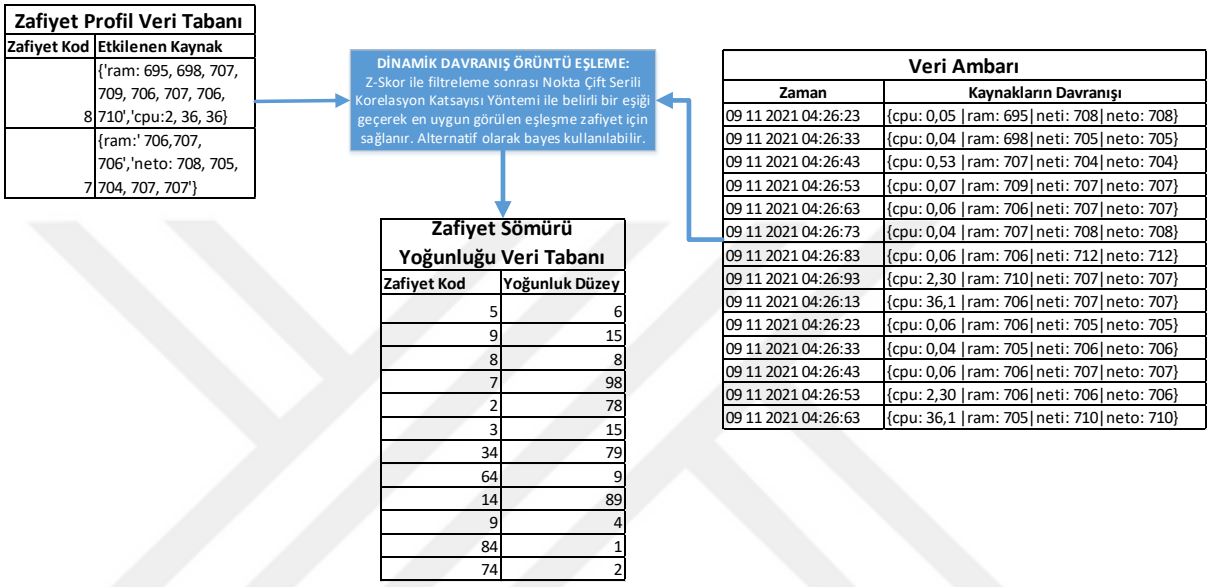


Şekil 3.17. Ön Süreç ve Sistem İyileştirme Aşamalarının Örnek Verilerle Zafiyet Profil Veri Tabanının Oluşturulma Sürecine Yönelik Akışı



Şekil 3.18. Ön Süreç ve Sistem İyileştirme Aşamalarının Algoritması

Şekil 3.17. ve Şekil 3.18. üzerindeki sistem sayesinde zafiyet profilinin yapısı çözümlenebilmektedir. Bu profilin başlangıçta oluşturulması için 1-2 haftalık bir süreyle günlük olarak Honeypot'a tanılanmak istenen saldırılar yapılmalıdır. Ardından Şekil 3.14. topolojisine uygun olarak sistem işleyecektir. Haftalık olarak Zafiyet Profil Veri Tabanını güncellemek için Şekil 3.17. sürecinin işletilmesi gerekmektedir. Bu sayede Şekil 3.19. üzerinde gösterildiği gibi "Dinamik Davranış Örüntü Eşleme" daha doğru sonuçlara varacaktır.

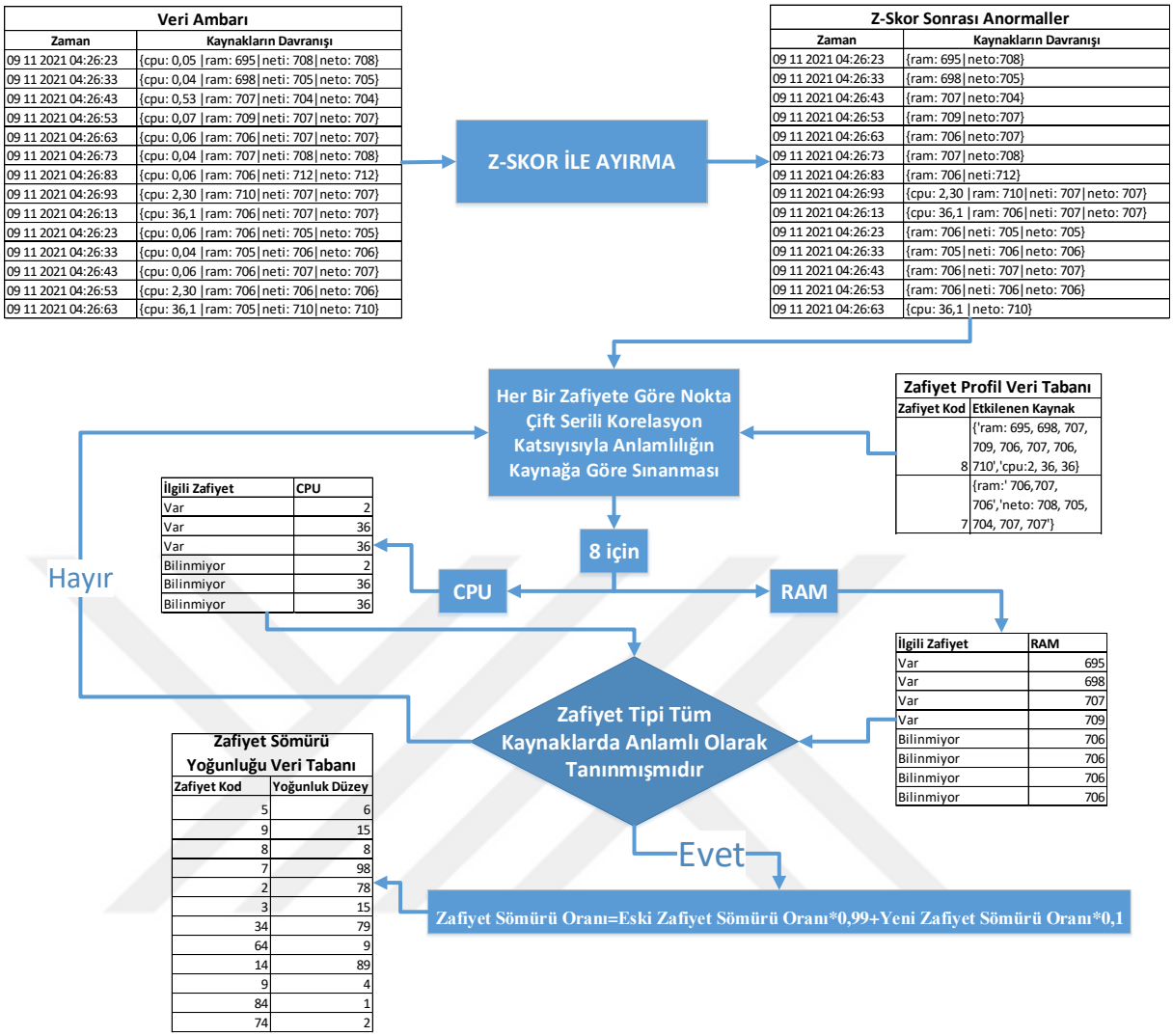


Şekil 3.19. Dinamik Davranış Örüntü Eşleşme Veri Akış Süreci

Şekil 3.19.üzerinde "Zafiyet Sömürü Yoğunluğu Veri Tabanı" saatlik olarak güncellenmektedir. Bunun sayesinde dinamik olarak mevcut zafiyetlerin hangisinin sömürüldüğü bilgisine erişilebilecektir. Bu kapsam doğrultusunda güvenlik yöneticileri anlık olarak kolay bir şekilde ilgili zafiyetin sömürü durumunu görebileceklerdir.

2-4 haftalık sürelerde bir metrik ve zafiyet ayarlamasını sağlayabileceklerdir. Burada bir diğer kritik nokta sistem "Zafiyet Sömürü Oranı" güncellemesi yaparken yeni durumları ekleyip eskisini de tamamen yok etmemek için 3.1 denklemiyle güncelleme yapılmaktadır. Bu sayede zamanla belirli zafiyetler sıklaşırsa o zafiyet sömürü oranı daha sağlıklı olarak alınabilecektir.

$$ZSO = EZSO*0,99+ YZSO *0,1 \quad (3.1)$$



Şekil 3.20. Detaylı Olarak Dinamik Davranış Örüntü Eşleşme Veri Akış Süreci

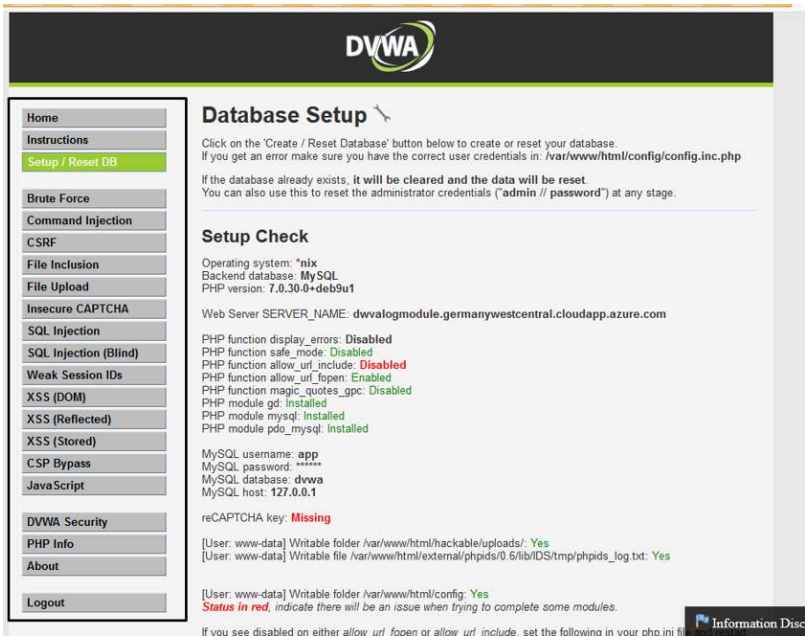
3.2.4. Deney Uygulamaları

Deney Ortam Yapısı

Dinamik Davranış Görüntü Eşlemenin ispatına yönelik deneyleri içerek verilerin toplanmasına yönelik deneyler gerçekleştirilmiştir. Ardından bu deneyler ile Normal Yük ve Zafiyet Yükü ayrımının mümkünlüğüne dair incelemeler yapılacaktır. Sonrasında ise zafiyet tiplerinin ayrıla bilirliğine dair incelemeler de eklenerek veri işleme süreçlerine tabi tutulacaktır.

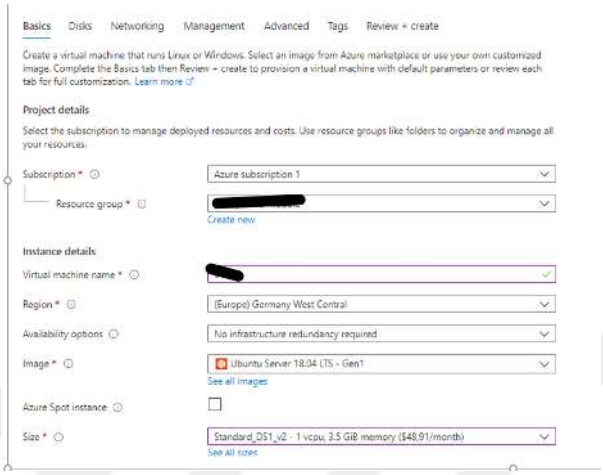
Linux: İşletim sistemi çekirdeği olan Linux çekirdeğine dayanan bir açık kaynaklı Unix benzeri işletim sistemleri ailesidir. Linux genellikle bir Linux dağıtımında paketlenir. Dağıtımlar, çoğu GNU projesi tarafından sağlanan Linux çekirdeğini ve destekleyici sistem yazılımını ve kitaplıklarını içerir. Birçok Linux dağıtımı, adlarında "Linux" kelimesini kullanır, ancak Özgür Yazılım Vakfı, bazı tartışmalara neden olan GNU yazılımının önemini vurgulamak için "GNU /Linux" adını kullanır. Seçilme gerekçesiyse Linux'un stabil olması, üzerinde rahatça log toplanabilmesi ve docker ile uyumluluğudur.

DVWA:Çok savunmasız olan bir PHP/MySQL web uygulamasıdır. Temel amacı, güvenlik profesyonellerinin becerilerini ve araçlarını yasal bir ortamda test etmeleri için bir araç olmaktır. Fakat en büyük kullanım gerekçesi en yüksek seviyede güvenlik önlemlerinden en düşüğe olmak üzere 4 seviyede olmasıdır. Bu sayede güvenlik seviyesine göre logların nasıl değişebileceğine dair analiz için toplanacak datalar daha stabil olacaktır. DWVA'nın "https://hub.docker.com/r/vulnerables/web-dvwa" üzerindeki docker imajı ile çalışılmıştır. DVWA'a açıldığın ilk açılan sayfa ve konfigürasyon değişiminin mümkünlüğü Şekil 3.21. üzerindeki görsel üzerinden anlaşılabilmektedir.



Şekil 3.21. DVWA Kurulmuş Anasayfası

Azure VM: Cloud tabanlı olarak Windows veya Linux işletim sistemi tabanlı için sanal makineyi oluşturulabilmektedir. Erişim ve benzeri konularda stabil olduğu için tercih edilmiştir. SSH ile erişilip deney ortamları kurulacaktır. Kurulacak sanal makine özelikleri Şekil 3.22. üzerinde görüldüğü gibi 1 CPU ve 3.5 GB RAM şeklindedir.



Şekil 3.22. Oluşturulan Azure VM Linux Sunucu Özellikleri ve Sürümü

Docker: Docker, geliştiricilerin, sistem yöneticilerinin uygulamalarını Linux gibi ana bilgisayar işletim sisteminde çalıştırmak için bir sanal alanda kolayca dağıtmalarına olanak tanıyan bir araçtır. Docker'ın ana avantajı, kullanıcıların bir uygulamayı tüm bağımlılıklarıyla birlikte yazılım geliştirme için standart bir birime paketlemesine izin vermesidir.

Sanal makinelerin aksine, kapsayıcılar yüksek bir ek yüke sahip değildir ve bu nedenle temel sistem ve kaynakların daha verimli kullanılmasını sağlar. Kullanılmasının temel gerekçesiyse loglarının stabil olarak alınabilmesi ve izole olmasından dolayı logların analizi sonucu elde edilecek sonuçların daha güvenilir olmasıdır. Bununla beraber Linux ile ve DVWA ile uyumlu olması da tercih için ön plana çıkarmıştır. Şekil 3.23. ve Şekil 3.24. üzerinde görülen konfigürasyonlar her deney için sanal makinada 0'dan çalıştırılmıştır.

```
sudo su
sudo apt --assume-yes install apt-transport-https ca-certificates curl software-properties-common
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
sudo apt update
sudo apt --assume-yes install docker-ce
systemctl start docker
systemctl enable docker
systemctl status docker
```

Şekil 3.23. Linux Makine Üzerindeki Temel Kurulumlar

```
docker rm $(docker ps -q -a) -f
docker rmi $(docker images -q -a) -f
docker run --rm -it -p 80:80 vulnerables/web-dvwa &
docker ps
```

Şekil 3.24. DVWA Deployment Komutları

Yük ve Zafiyet Tarama Yöntemleri

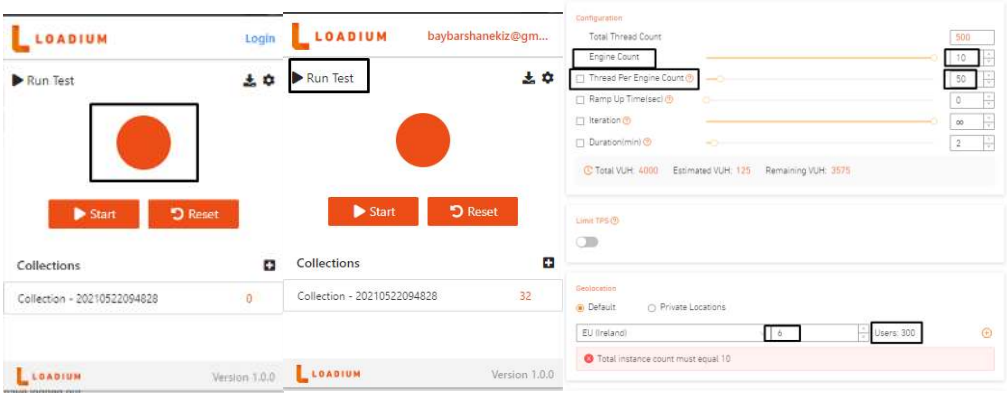
Owasp Zap ve Loadium için tool tabanlı ne gibi kaynak davranış değişikliğinin olduğunun gözlenmesi için farklı toollar denenmiştir. Tablo 3.4. üzerindeki dağılım yapısıyla aynı toollarla farklı zafiyetler ve farklı zafiyetlerle aynı toollar denendiği zaman sunucu kaynaklarının zafiyet tabanlı ve yük yapısı tabanlı değişimlerinin boyutlarını analiz edebilmek amaçlanmaktadır.

Tablo 3.4. Uygulanacak Test Senaryolarının Yük Yüzdeleri

Yük Senaryo Tipi	Toplam Oranı				
	100%	80%	20%	16%	4%
Loadium Direk Yük	X	X			
Loadium Scripted Owasp	X	X	X	X	X
OWASP ZAP	X	X	X	X	X

Loadium: Bir performans test platformudur. Performans Testi, uygulamanın tanımlanmış kullanıcı yükü altında iyi çalışmasını garanti eden bir tür yazılım testidir. Performans testinin temel amacı hataları bulmak değil, performans darboğazlarını ortadan kaldırmaktır. Sistemin kalite özelliklerini değerlendirir. Loadium'un kullanılış amacı normal kullanıcıların davranışlarını kayıt edip bu davranışı istenildiği yük ile sistemde test edebilmektedir.

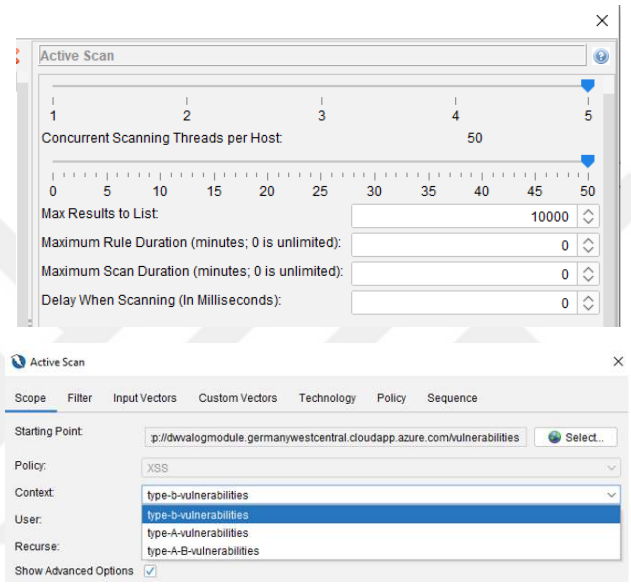
Bu yük sonucu alınacak olan yükler analiz edilecektir. Loadium aynı zamanda manuel olarak zafiyetin sömürüsünün manuel olarak yapılmasına yönelik yük testi de gerçekleştirmek üzere kullanılacaktır. Yük testinin amacı, giderek artan sayıda kullanıcı altında uygulama performansının performansını değerlendirmektir. Test edilen uygulama, yük veya artan sayıda kullanıcı ile yüklenir ve gereksinimlerin karşılandığını doğrulamak için sonuçlar ölçülür. Fakat logları analiz etmek için loadium'un yük testi özelliğinden Şekil 3.25. üzerinde görüldüğü gibi faydalanılacaktır.



Şekil 3.25. Loadium Kullanıcı Davranış Kaydı ve Kaydın Yük Olarak Ayarlanması- 300 Kullanıcı İçin Yapılan Ayarlara Yönelik Örnektir.

Owasp Zap: OWASP ZAP, web uygulamaları için açık kaynaklı bir güvenlik tarayıcısıdır. Hem acemi uygulama güvenliği uygulayıcıları hem de profesyonel sızma test uzmanları tarafından kullanılması amaçlanmıştır. OWASP'ın en aktif projelerinden biridir ve amiral gemisi statüsü verilmiştir.

Direk olarak Owasp tarafından tasarlanması ve Owasp top 10 üzerinden istenen kategoriye yönelik saldırı taramasının Şekil 3.26. üzerinde görüldüğü gibi yapılabilmesi ön plana çıkarmıştır. Ayrıca tarama yükünün ayarlanabilmesi log analizi için önem taşıdığı için tercih edilmiştir.



Şekil 3.26. Owasp Zap Tarama için Yük Seçimi ve Zafiyet Tip Profili Seçimi

Tablo 3.5. Kullanıcı Sayısı Bazında Yük Tipleri Tablosu

Yük Senaryo Tipi	Toplam Kullanıcı Sayısı				
	300	240	60	48	12
Loadium Direk Yük	X	X			
Loadium Scripted Owasp	X	X	X	X	X
OWASP ZAP	X	X	X	X	X

Tablo 3.5. üzerindeki yük dağılımlarının her biri için zafiyetlerin DVWA üzerinde 4 üzerinden 4 seviyede olduğu çok kolay ve 4 üzerinde 2 seviyede olduğu çok zor için 2 farklı güvenlik seviyesi için gerçekleştirilerek toplam 16 kere yapılacaktır. Ayrıca toplam yük yapı dağılımı Tablo 3.6. üzerinde görüldüğü gibi 8 adet olacaktır. Bu durum 2 farklı ve 1 hibrit yapıdaki zafiyet profilleri için toplam 3 kere uygulanarak 48 deneme yapılacaktır. Çok zor ve kolay güvenlik seviyesindeki 2 normal trafik deneyiyle beraber toplam 50 adet deney yapılacaktır. DVWA üzerinde a ve b tip zafiyetler için istekte bulunulacak sayfalar:

B: http:// domain.azure.com/vulnerabilities/xss_d/

B: http:// domain.azure.com/vulnerabilities/xss_r/

B: http:// domain cloudapp.azure.com/vulnerabilities/xss_s/

A: http:// domain.azure.com/vulnerabilities/sqli/

A: http:// domain.azure.com/vulnerabilities/sqli_blind/

Tablo 3.6. Yük Yapı Tiplerinin Dağılım Tablosu

Numara	Yük Yapısı				
	Yük Yapı Numarası	Normal Kullanıcı Yükü	Zafiyet Yükü		
			Zafiyet Tarama Oranı	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı
1	1	300	0	0	0
2	2	0	300	0	300
3	3	0	300	300	0
4	4	0	300	240	60
5	5	0	300	60	240
7	6	240	60	0	60
6	7	240	60	60	0
8	8	240	60	48	12
9	9	240	60	12	48

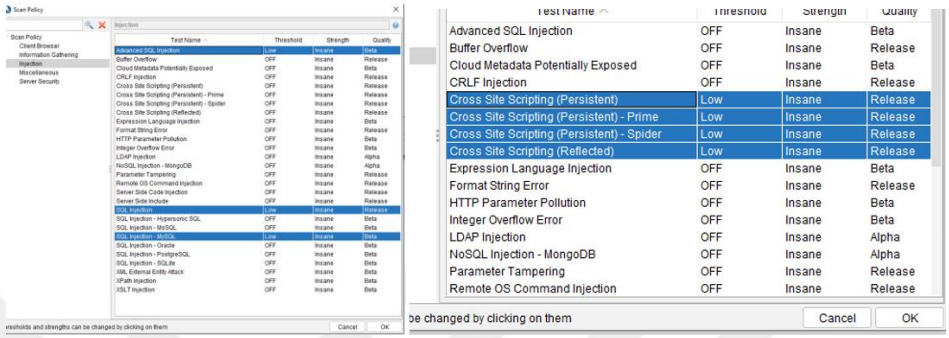
Zafiyet Tipleri

XSS : Siteler Arası Komut Dosyası Çalıştırma (XSS) saldırıları, kötü amaçlı komut dosyalarının normalde zararsız ve güvenilir web sitelerine enjekte edildiği bir tür enjeksiyondur. XSS saldırıları, bir saldırgan, genellikle tarayıcı tarafı komut dosyası biçiminde kötü amaçlı kodu farklı bir son kullanıcıya göndermek için bir web uygulaması kullandığında meydana gelir. Bu saldırıların başarılı olmasına izin veren kusurlar oldukça yaygındır ve bir web uygulamasının, doğrulamadan veya kodlamadan ürettiği çıktı içinde bir kullanıcıdan gelen girdiyi kullandığı her yerde ortaya çıkar.

Tablo 3.7. Kullanılacak Zafiyet Tipleri

Zafiyet Tarama	
İsim	**Kod**
Sql Injection	A
XSS	B
Sql Injection+XSS	A+B

Injection: İstemciden uygulamaya giriş verileri aracılığıyla bir SQL sorgusunun eklenmesinden veya "enjekte edilmesinden" oluşur. Başarılı bir SQL enjeksiyon istismarı, Veritabanı üzerinden hassas verileri okuyabilir, Veritabanı Şekil 3.28. üzerindeki örnek üzerinde görülen ekran aracılığıyla verilerini değiştirebilir. Ekle/Güncelle/Sil gibi komutların yetkisiz olarak Veritabanı üzerinde çalışması yoluyla SQL enjeksiyonu gerçekleşmiş olur.



Şekil 3.27. OWASP ZAP Üzerinde Injection ve XSS Profillerinin Seçimi

Bu iki benzer kaynak davranışları gösterebilecek zafiyetler birbirinden ayrılıyorsa daha az benzeşen zafiyet tipleri daha kolay ayrılmaktadır. Bu sebeple Injection ve XSS yapı olarak benzer kaynak kullanım davranışını göstereceği için bu iki zafiyet tipinin sömürüsü deney için ortaya çıkan sonuçların daha güvenilir olmasını sağlayacaktır. Deneylerde Owasp Zap üzerinden Şekil 3.27. görülen konfigürasyonlarla zafiyet sömürüsü yapmak için seçilecek olanlar Tablo 3.8. üzerinde görülmektedir. Zafiyetin varlığı durumunda ve olmaması durumunda sistemin ne gibi bir davranış gösterdiğinin anlaşılması için zorluk seviyeleri her bir deney için kolay ve çok zor olan seviyelerde deney yapılarak analiz edilecektir.

Tablo 3.8. XSS ve SQL Injection Zafiyetleri için Seçilecek Olan Owasp ZAP Zafiyetleri

Cross Site Scripting(XSS)	**Sql Injection**
Cross Site Scripting (Persistent)	Advanced SQL Injection
Cross Site Scripting (Persistent) - Prime	SQL Injection
Cross Site Scripting (Persistent) - Spider	SQL Injection - MySQL
Cross Site Scripting (Reflected)	

Şekil 3.28. DVWA Üzerindeki SQL Injection Alanı

Gerçekleştirilen Deney Yapıları

Gerçekleşecek 50 adet deneyin yapısı Tablo 3.9 üzerinde görülmektedir. Tablo 3.9. üzerinde deney yük dağılımları, zafiyet dağılımları, deneylerin zorluk seviyeleri ve deney numaraları görülmektedir. Loadium ve Owasp zap üzerinde yapılacak olan taramaların çıktıları ise Şekil 3.29. ve Şekil 3.30 üzerinde görülmektedir.

Tablo 3.9. Yapılacak Olan Tüm Testlerin Zorluk Seviyesi, Kullanıcı Sayısı, Kullanıcı Dağılımı, Yük Dağılımı, Yük Tipi ve Zafiyet Tipi Tablosu

Numara	dwva Zafiyet Zorluk	Normal Kullanıcı Yükü	Zafiyet Yükü			
			Zafiyet Tarama Oranı	Tip	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı
1	**Kolay**	300	0	-	0	0
2	**Kolay**	0	300	A	300	0
3	**Kolay**	0	300	A	0	300
4	**Kolay**	0	300	A	240	60
5	**Kolay**	0	300	A	60	240
6	**Kolay**	240	60	A	60	0
7	**Kolay**	240	60	A	0	60
8	**Kolay**	240	60	A	48	12
9	**Kolay**	240	60	A	12	48
10	**Kolay**	0	300	B	300	0
11	**Kolay**	0	300	B	0	300
12	**Kolay**	0	300	B	240	60
13	**Kolay**	0	300	B	60	240
14	**Kolay**	240	60	B	60	0
15	**Kolay**	240	60	B	0	60
16	**Kolay**	240	60	B	48	12
17	**Kolay**	240	60	B	12	48
18	**Kolay**	0	300	A+B	300	0
19	**Kolay**	0	300	A+B	0	300
20	**Kolay**	0	300	A+B	240	60
21	**Kolay**	0	300	A+B	60	240
22	**Kolay**	240	60	A+B	60	0
23	**Kolay**	240	60	A+B	0	60
24	**Kolay**	240	60	A+B	48	12
25	**Kolay**	240	60	A+B	12	48
26	**Çok Zor**	300	0	-	0	0
27	**Çok Zor**	0	300	A	300	0
28	**Çok Zor**	0	300	A	0	300
29	**Çok Zor**	0	300	A	240	60
30	**Çok Zor**	0	300	A	60	240
31	**Çok Zor**	240	60	A	60	0
32	**Çok Zor**	240	60	A	0	60
33	**Çok Zor**	240	60	A	48	12
34	**Çok Zor**	240	60	A	12	48
35	**Çok Zor**	0	300	B	300	0
36	**Çok Zor**	0	300	B	0	300
37	**Çok Zor**	0	300	B	240	60
38	**Çok Zor**	0	300	B	60	240
39	**Çok Zor**	240	60	B	60	0
40	**Çok Zor**	240	60	B	0	60
41	**Çok Zor**	240	60	B	48	12
42	**Çok Zor**	240	60	B	12	48
43	**Çok Zor**	0	300	A+B	300	0
44	**Çok Zor**	0	300	A+B	0	300
45	**Çok Zor**	0	300	A+B	240	60
46	**Çok Zor**	0	300	A+B	60	240
47	**Çok Zor**	240	60	A+B	60	0
48	**Çok Zor**	240	60	A+B	0	60
49	**Çok Zor**	240	60	A+B	48	12
50	**Çok Zor**	240	60	A+B	12	48

Şekil 3.29. Tablo 3.9. Göre 1 Nolu Yapının Loadium Çıktısı

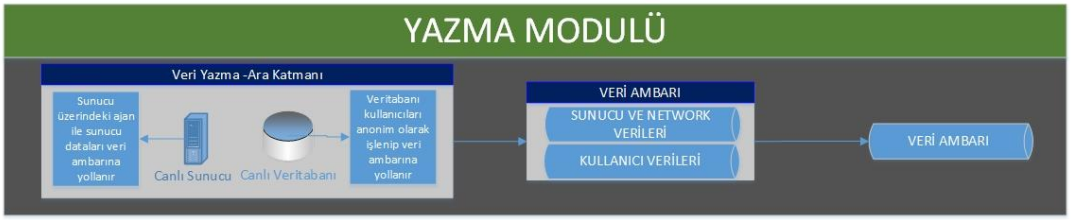
Analyser			00:00.099	1	
Plugin					
SQL Injection	Insane		00:49.626	1062	0
SQL Injection - MySQL	Insane		00:15.139	330	0
Advanced SQL Injection	Insane		01:01.925	185	0
Totals			02:06.809	1581	0

Şekil 3.30. OWASP ZAP Sql Injection Tarama Ekranı

3.3. Simülasyon Ortam Verilerine Uygulanan İşlemler

3.3.1. Veri Oluşturma ve Toplama

Şekil 3.31. üzerinde önerilen modüler akışta bulunan Yazma Modülü tarafından oluşturulan Veri Ambarına ait verilerin oluşturulma sürecidir. Bu modülün daha detaylanmış hali olan Şekil 3.31 akışının simülasyon üzerinden toplanabilmesi için 10 adet adım Tablo 3.9. üzerindeki her bir deney için uygulanmıştır. Bu yolla Veri Ambarına yönelik veriler elde edilebilmiştir.

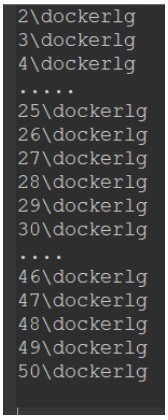


Şekil 3.31. Önerilen Sistem İçin Yazma Modülü

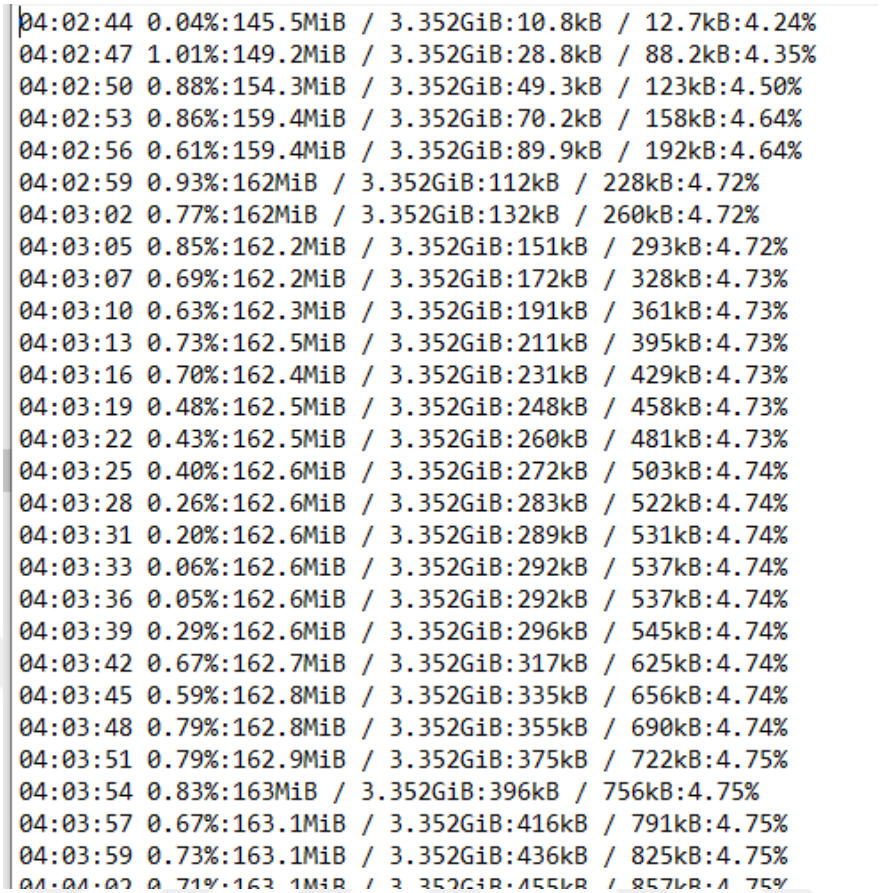
Simüle veri ambar verilerinin oluşturulma adımları:

I. Sanal Sunucunun Oluşturulması
II. Sunucu Alt Yapı Kurulumların Yapılması
III. Docker Image'ların Deployment Sağlanması
IV. Analiz için toplanacak logları toplayacak komutun çalıştırılması
V. Toplamda 300 adet kullanıcının 2 dk boyunca yüklenmesi(Loadium Normal Kullanıcılar/Loadium Sömürü Kullanıcısı/Owasp Zap)

IV. adım olan log toplama komutu: " watch -t -n 1 'echo $(date -d "+3 hour" +%T) $(docker stats --no-stream --format "{{.CPUPerc}}:{{.MemUsage}}:{{.NetIO}}:{{.MemPerc}}" <container_id>) >> dockerlog' " Şekil 3.32. üzerindeki dockerlg dosyalarının her bir deney için kullanılmasını sağlamaktadır.



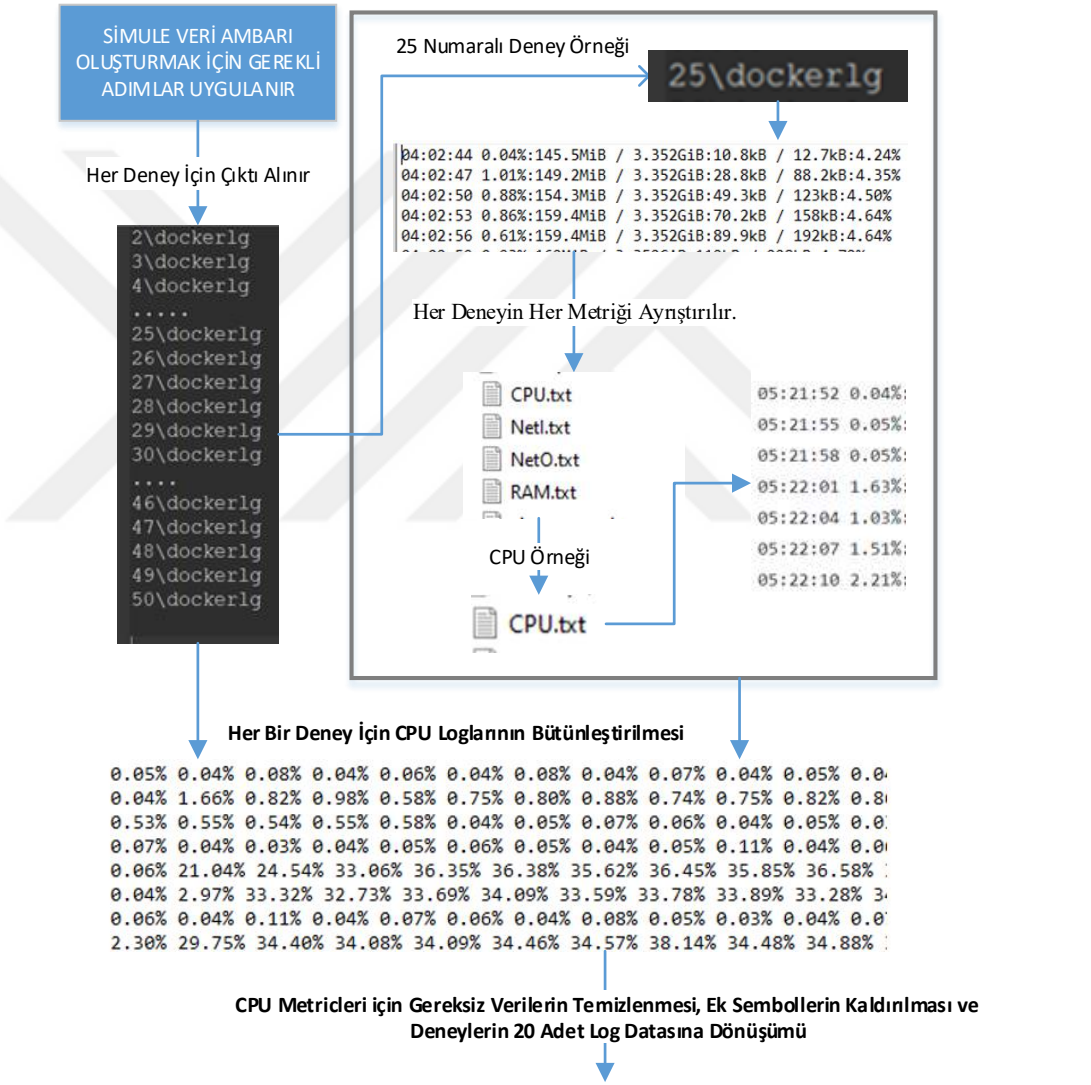
Şekil 3.32. Deney numaralarına göre Docker loglarının yazıldığı dosyalar



Şekil 3.33. Docker loglarının içeriği

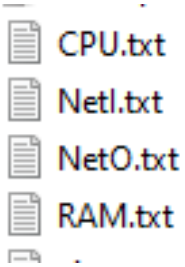
3.3.2. Veri Temizleme ve Bütünleştirme

Her kaynak tipinin deney anına göre metrik verileri deney dışı zamandaki metriklerden tek tek zaman damgası göz önünde bulundurularak Şekil 3.33 şeklinde görülen loğlar her bir deney için temizlenildi. Bu temizleme işleminin ardından CPU, RAM, NETI, NETO metriklerinin her bir deney için Şekil 3.35. görüldüğü gibi ayrıştırma işlemi yapılmıştır. Bu ayrıştırma işlemini Şekil 3.36. üzerinde CPU örneği için görülen Python script'leri ile yapılmıştır. Genel Veri Temizleme ve Bütünleştirme sürecinin akışı Şekil 3.34. üzerinde özetlenmiştir.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0,04	7,26	35,89	35,56	35,39	35,72	35,66	35,91	36,12	36,26	35,83	35,43	33,65	35,8	36,11	36,54	36,65	35,96	36,04	35,78
2	0,04	1,66	0,82	0,98	0,58	0,75	0,8	0,88	0,74	0,75	0,82	0,86	0,71	0,45	0,36	0,05	0,06	1,19	0,82	0,87
3	0,03	4,08	22,98	17,95	28,75	35,71	36,23	35,63	35,91	36,14	36,93	36,73	37,22	37,94	36,32	35,97	37,54	36,05	35,84	37,58
4	15,21	17,38	18,4	17,94	17,05	17,47	16,33	18,48	17,04	16,82	18,66	19,75	16,39	16,17	16,94	18,54	15,35	21,58	16,32	16,8
5	0,06	21,04	24,54	33,06	36,35	36,38	35,62	36,45	35,85	36,58	36,69	35,78	36,58	36,72	37,2	36,43	37,23	36,39	37,11	36,57
6	0,04	2,97	33,32	32,73	33,69	34,09	33,59	33,78	33,89	33,28	34,71	33,92	33,9	33,36	33,64	33,52	34,22	33,84	33,72	33,53
7	0,04	2,69	28,68	36,75	36,26	36,71	38,66	36,3	36,22	40,7	41,45	42,9	48,38	39	37,44	38,84	47,39	38,52	36,39	36,32

Şekil 3.34. Veri Oluşturma ve Toplama Sürecinin CPU Metrik Örneğine Göre Deneylere Uygulama Özeti



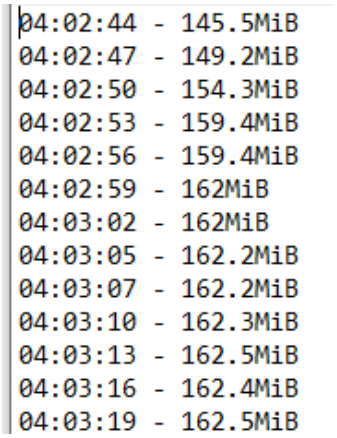
Şekil 3.35. Her Deneyin Python Script ile Ayrıştırılmış Metrikleri

```
def CPUfunc(numb):
  path='%s/dockerlg/dockerlog' %(numb)
  f = open(path, "r")
  allIncpu=""
  cpuList = ["Toyota"]
  cpuList.append("1")
  for x in f:
    cpuList.append(getCPULineInMetric(x))
  return cpuList

def getCPULineInMetric(x):
  try:
    allVar=x.split(' ')
    cputake=allVar[1].split(':')
    CPU=cputake[0]
  except:
    CPU=0
  return CPU

def makePure(array):
  leng=len(array)
  pureList = ["Toyota"]
  pureList.append("1")
  for pureNumb in range(1,leng):
    if(array[pureNumb]!=array[pureNumb-1]):
      pureList.append(array[pureNumb-1])
  return pureList

def str2FileCPU(filename, aarray):
  f = open(filename, "a")
  makePureCPU=makePure(aarray)
  for x in makePureCPU:
    print(x)
    f.write(x)
    f.write(" ")
  f.write("\n")
  f.close()

str2FileCPU("CPU.txt",CPUfunc(1))
```

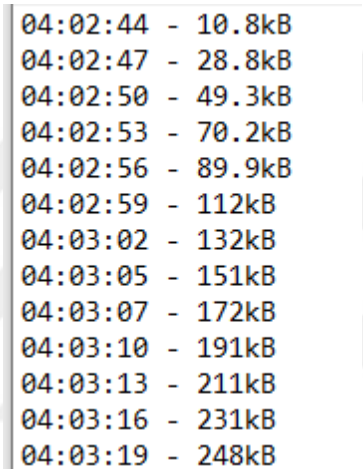
Şekil 3.36. Her Deneyi Metriklerine Göre Ayrıştıran Python Script'inin 1 Numaralı Deneyin CPU Metriğini Ayırma Örneği

```
05:21:52 0.04%
05:21:55 0.05%
05:21:58 0.05%
05:22:01 1.63%
05:22:04 1.03%
05:22:07 1.51%
05:22:10 2.21%
```

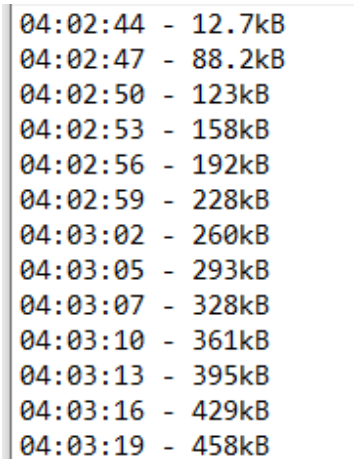
Şekil 3.37. CPU Metriklerinin İçerisi



Şekil 3.38. RAM Metriklerinin İçerisi



Şekil 3.39. NETI Metriklerinin İçerisi



Şekil 3.40. NETO Metriklerinin İçerisi

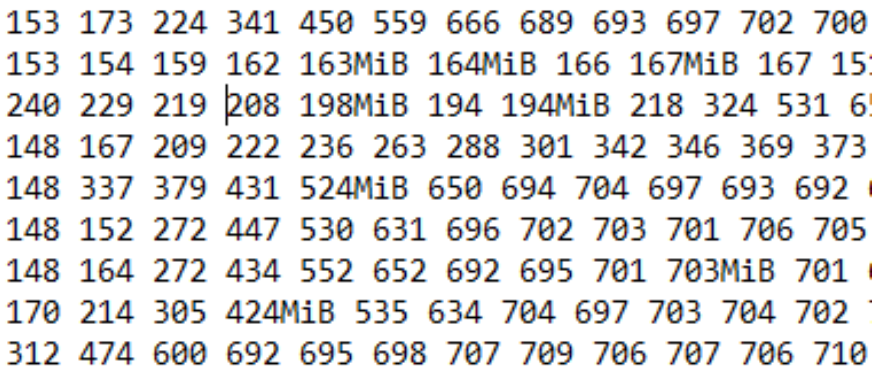
Uygulanan işlemlerin ardından elde edilen log verileri her kaynaktan yapı olarak Şekil 3.37. Şekil 3.38. , Şekil 3.39. ve Şekil 3.40. üzerinde görüldüğü gibi elde edilmiştir. Ardından ise Şekil 3.41. üzerindeki Python komutlarıyla her deney metriğe göre sıralı olarak tek bir dosyada Şekil 3.42. , Şekil 3.43. , Şekil 3.44. ve Şekil 3.45. üzerinde görüldüğü gibi toplanmıştır.

```
def CPUfunc(numb):
  path='%s/dockerlg/dockerlog' %(numb)
  f = open(path, "r")
  allIncpu=""
  cpuList = ["Toyota"]
  cpuList.append("1")
  for x in f:
    cpuList.append(getCPULineInMetric(x))
  return cpuList

def getCPULineInMetric(x):
  allVar=x.split(' ')
  cputake=allVar[1].split(':')
  CPU=cputake[0]
  return CPU

def makePure(array):
  leng=len(array)
  pureRamList = ["Toyota"]
  pureRamList.append("1")
  for pureNumb in range(1,leng):
    if(array[pureNumb]!=array[pureNumb-1]):
      pureRamList.append(array[pureNumb-1])
  return pureRamList

def makePureStr(arr):
  allResult=""
  for arrFull in arr:
    allResult='%s %s' %(allResult,arrFull)
  return allResult

def str2File(filename, aarray):
  f = open(filename, "a")
  for x in aarray:
    f.write(x)
  f.write(" ")
  f.write("\n")
  f.close()

str2File("CPU.txt",CPUfunc(2))
```

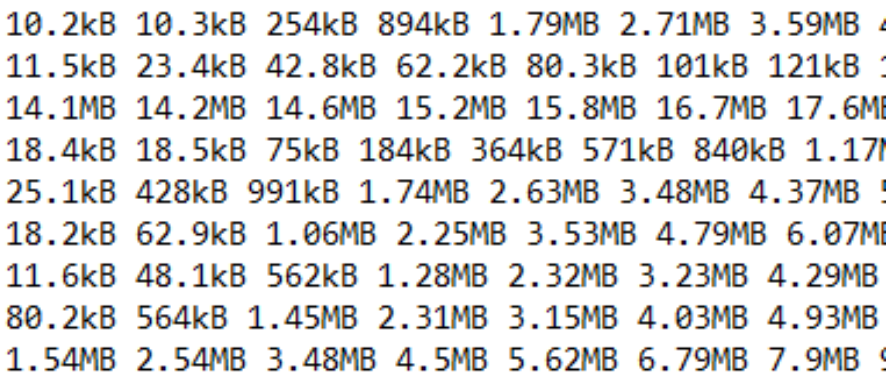
Şekil 3.41. Tüm Deneylerin CPU Metriklerini Sıralı Olarak Toplayan Python Script Örneği

```
0.05% 0.04% 0.08% 0.04% 0.06% 0.04% 0.08% 0.04% 0.07% 0.04% 0.05% 0.0
0.04% 1.66% 0.82% 0.98% 0.58% 0.75% 0.80% 0.88% 0.74% 0.75% 0.82% 0.8
0.53% 0.55% 0.54% 0.55% 0.58% 0.04% 0.05% 0.07% 0.06% 0.04% 0.05% 0.0
0.07% 0.04% 0.03% 0.04% 0.05% 0.06% 0.05% 0.04% 0.05% 0.11% 0.04% 0.0
0.06% 21.04% 24.54% 33.06% 36.35% 36.38% 35.62% 36.45% 35.85% 36.58% 
0.04% 2.97% 33.32% 32.73% 33.69% 34.09% 33.59% 33.78% 33.89% 33.28% 3
0.06% 0.04% 0.11% 0.04% 0.07% 0.06% 0.04% 0.08% 0.05% 0.03% 0.04% 0.0
2.30% 29.75% 34.40% 34.08% 34.09% 34.46% 34.57% 38.14% 34.48% 34.88% 
36.11% 36.77% 35.90% 35.83% 36.08% 36.31% 36.26% 35.93% 36.60% 36.45%
7% 3.10% 3.39% 2.57% 3.80% 3.09% 3.18% 4.38% 4.30% 3.42% 4.07% 33.89% 
0.04% 1.01% 0.88% 0.86% 0.61% 0.93% 0.77% 0.85% 0.69% 0.63% 0.73% 0.7
0.04% 0.05% 0.04% 0.08% 0.05% 0.04% 0.08% 0.04% 0.10% 0.08% 0.04% 0.0
0.07% 0.04% 1.26% 0.06% 0.05% 0.04% 0.06% 0.09% 0.04% 0.03% 0.06% 0.0
0.07% 0.04% 1.19% 0.83% 1.64% 7.40% 5.07% 4.99% 7.86% 6.47% 7.90% 6.5
0.04% 0.66% 24.17% 35.52% 35.74% 35.91% 36.56% 35.19% 36.03% 36.60% 3
0.04% 0.05% 0.04% 0.05% 0.04% 0.05% 0.04% 0.06% 0.03% 0.04% 0.06% 0.0
12.95% 36.47% 35.89% 36.29% 36.53% 36.25% 36.34% 36.45% 36.50% 36.31%
9.26% 36.30% 37.69% 37.89% 37.49% 38.38% 38.65% 38.99% 38.16% 38.21% 
0.04% 1.86% 0.73% 1.01% 0.82% 0.93% 0.78% 1.87% 1.38% 1.60% 1.03% 1.2
0.04% 0.09% 0.04% 0.06% 0.10% 0.04% 0.08% 0.07% 0.04% 0.06% 0.04% 0.0
```

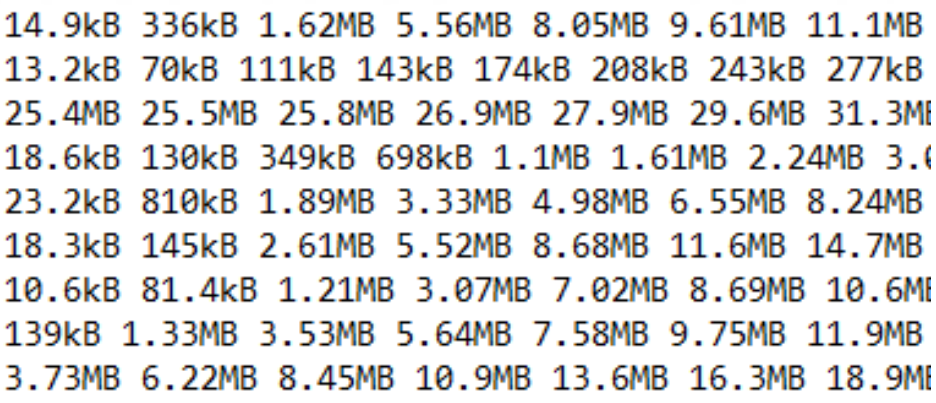
Şekil 3.42. CPU Metriği için Deneylerin Bütünleşmiş Hallerinin Bulunduğu Dosyanın İçeriği



Şekil 3.43. RAM Metriği için Deneylerin Bütünleşmiş Hallerinin Bulunduğu Dosyanın İçeriği



Şekil 3.44. NETI Metriği için Deneylerin Bütünleşmiş Hallerinin Bulunduğu Dosyanın İçeriği



Şekil 3.45. NETO Metriği için Deneylerin Bütünleşmiş Hallerinin Bulunduğu Dosyanın İçeriği

Bu süreçlerin ardından 3 saniyede bir alınan loglar 6 saniyede bir olacak şekilde 2 dakika içinde 20 adet loga dönüşecek şekilde bütünleştirilir. Bu sayede 2 dakikalık sürede 20 adet log kaydı olacaktır. Veri işleme açısından kolaylık sağlamaya yönelik bu işlem yapılmıştır. T anındaki 2 ardışık metriğin ortalaması alınmıştır. Veriler %, MB, Kb gibi sembollerden ayrıştırılır ve tarama yapılan aralıktaki 20 log kaydı dışındaki loglar haricindeki diğer loglar silinir. 50 adet deneylerden ham olarak 2 dakika boyunca elde edilen veriler 4 üzerinden Tablo 3.10. , Tablo 3.11. , Tablo 3.12. ve Tablo 3.13. üzerinde gösterilmiştir.

Tablo 3.10. Veri Temizleme ve Bütünleştirme İşleminin Ardından CPU Logları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0,04	7,26	35,89	35,56	35,39	35,72	35,66	35,91	36,12	36,26	35,83	35,43	33,65	35,8	36,11	36,54	36,65	35,96	36,04	35,78
2	0,04	1,66	0,82	0,98	0,58	0,75	0,8	0,88	0,74	0,75	0,82	0,86	0,71	0,45	0,36	0,05	0,06	1,19	0,82	0,87
3	0,03	4,08	22,98	17,95	28,75	35,71	36,23	35,63	35,91	36,14	36,93	36,73	37,22	37,94	36,32	35,97	37,54	36,05	35,84	37,58
4	15,21	17,38	18,4	17,94	17,05	17,47	16,33	18,48	17,04	16,82	18,66	19,75	16,39	16,17	16,94	18,54	15,35	21,58	16,32	16,8
5	0,06	21,04	24,54	33,06	36,35	36,38	35,62	36,45	35,85	36,58	36,69	35,78	36,58	36,72	37,2	36,43	37,23	36,39	37,11	36,57
6	0,04	2,97	33,32	32,73	33,69	34,09	33,59	33,78	33,89	33,28	34,71	33,92	33,9	33,36	33,64	33,52	34,22	33,84	33,72	33,53
7	0,04	2,69	28,68	36,75	36,26	36,71	38,66	36,3	36,22	40,7	41,45	42,9	48,38	39	37,44	38,84	47,39	38,52	36,39	36,32
8	2,3	29,75	34,4	34,08	34,09	34,46	34,57	38,14	34,48	34,88	34,33	33,89	41,75	34,99	34,58	35,14	34,35	34,48	35,51	34,56
9	36,11	36,77	35,9	35,83	36,08	36,31	36,26	35,93	36,6	36,45	35,91	36,13	35,25	36,49	36,66	36,52	35,99	36,11	36,39	36,64
10	0,04	1,01	0,88	0,86	0,61	0,93	0,77	0,85	0,69	0,63	0,73	0,7	0,48	0,43	0,4	0,26	0,2	0,06	0,05	0,29
11	2,12	17,14	16,58	1,4	3,89	4,26	14,21	8,77	10,64	17,61	21,14	13,1	7,32	8,79	10,17	9,65	11,46	29,28	21,28	17,05
12	0,07	0,04	1,26	0,06	0,05	0,04	0,06	0,09	0,04	0,03	0,06	0,04	0,08	0,04	0,06	0,08	0,04	0,06	0,07	0,04
13	0,07	0,04	1,19	0,83	1,64	7,4	5,07	4,99	7,86	6,47	7,9	6,52	7,51	8,34	7,16	6,62	7,33	7,94	6,49	4,38
14	24,17	35,52	35,74	35,91	36,56	35,19	36,03	36,6	35,83	35,95	35,62	36,02	35,89	35,95	35,7	36,19	36,46	36,92	35,8	36,29
15	17,43	34,6	34,42	34,74	34,95	36,24	35,07	35,02	36,81	34,91	35,31	35,89	34,82	35,05	35,11	35,8	35,2	35,71	35,76	36,26
16	12,95	36,47	35,89	36,29	36,53	36,25	36,34	36,45	36,5	36,31	36,13	34,82	36,83	36,59	36,73	37,27	36,27	36,74	36,33	36,25
17	9,26	36,3	37,69	37,89	37,49	38,38	38,65	38,99	38,16	38,21	38,8	38,79	38,28	37,92	38,52	38,53	38,52	39,24	39,54	38,62
18	0,04	1,86	0,73	1,01	0,82	0,93	0,78	1,87	1,38	1,6	1,03	1,23	0,97	0,7	0,83	0,04	0,05	0,04	0,81	0,76
19	30,06	33,53	33	33,29	33,18	32,86	33,71	32,98	33,5	33,25	33,46	33,15	33,26	32,91	33,38	33,08	33,4	32,86	33,07	33,37
20	0,96	3,4	4,49	4,22	6,64	5,35	5,91	4,79	3,57	5,18	8,08	8,65	7,09	11,01	12,35	7,8	10,01	10,35	8,33	13,01
21	9,4	7,26	10,47	12,35	18,63	17,6	22,51	12,97	10,92	10,45	15,39	16,24	13,64	11,72	19,31	16,44	15,66	12,93	15,82	11,88
22	17,81	36,81	37,21	36,72	37,42	37,36	37,73	37,3	37,86	37,75	37,58	37,69	37,46	37,18	37,98	35,03	37,67	37,17	37,22	37,56
23	9,97	24,22	32,52	34,3	35,97	35,89	35,31	36,7	35,73	36,65	36,56	34,38	36,98	36,42	36,73	36,61	35,85	36,04	36,66	35,94
24	15,09	31,1	36,24	36,28	37,16	38,26	37,64	36,87	36,35	36,07	35,73	36,95	35,79	36,7	36,89	35,91	35,97	36,62	37,22	36,65
25	36,59	36,19	37,1	36,88	36,77	37,35	37,69	37,11	37,69	37,68	37,11	37	37,65	37,33	37,51	40,04	37,81	36,44	37,48	37,29
26	0,05	7,67	14,72	35,48	35,96	35,81	36,13	36,11	35,7	35,79	36,26	35,83	35,49	35,99	35,86	36,5	35,84	36,86	36,5	36,42
27	0,04	0,64	0,52	0,46	0,48	0,59	0,6	1,75	1,7	1,72	0,59	0,51	0,55	0,47	0,52	1,5	1,49	1,11	0,59	0,52
28	0,08	3,53	14,46	29,06	36,23	35,97	35,21	37,18	36,82	37,17	38,61	37,49	36,76	37,22	37,62	37,77	37,09	37,36	37,38	37,23
29	0,56	1,44	1,95	2,4	2,87	3,25	3,6	4,44	5,32	6,56	5,91	3,74	3,84	3,06	4,24	6,17	4,02	6,04	4,04	6,48
30	0,48	1,62	2,53	3,56	0,72	2,7	5,32	6	5,99	5,23	5,49	4,69	9,03	6,65	13,24	7,12	12,14	9,09	9,23	11,8
31	0,04	5,35	32,98	35,66	36,21	35,3	35,81	35,23	36,08	36,05	36,08	35,78	35,75	34,94	36,1	36,16	36,06	36,18	35,63	36,44
32	0,04	0,06	0,04	0,08	0,04	0,07	0,04	0,05	0,04	0,05	0,04	0,05	0,04	0,08	0,04	0,05	0,09	0,05	0,07	0,04
33	0,04	38,55	38,42	38,11	37,82	38,52	37,43	38,11	38,33	37,48	37,57	8,24	6,39	5,2	5,59	5,42	4,81	6,4	3,74	4,2
34	36,37	36,67	36,05	38,21	45,91	35,43	40,59	40,77	35,69	36,85	35,98	36,02	36,01	36,76	35,73	36,38	35,84	36,6	35,85	36
35	0,79	0,82	0,74	0,56	0,63	0,68	0,65	0,91	0,67	0,75	0,85	0,4	0,62	0,43	0,27	0,29	0,04	0,07	0,04	0,07
36	1,16	13,16	14,37	10,41	8,34	5,02	4,99	11,7	10,71	13,72	6,72	22,68	33,13	14,29	10,16	4,77	6,89	11,14	10,2	9,41
37	0,09	0,31	0,89	0,83	0,71	0,97	0,9	0,8	0,97	0,81	0,77	0,87	0,47	0,44	0,42	0,25	0,28	0,05	0,06	0,08
38	0,05	0,36	0,74	0,97	0,77	0,9	0,93	0,79	0,84	0,91	0,83	0,92	0,81	0,67	0,56	0,5	0,28	0,23	0,04	0,14
39	37,61	37,76	38,45	38,28	38,43	38,4	38,38	37,55	37,39	36,99	38,06	38,35	38,87	38,24	39,87	38,92	38,85	39,11	38,02	37,92
40	2,42	20,66	35,19	36,5	35,56	34,78	35,52	35,49	35,83	35,01	34,51	36,56	35,59	36,13	37,17	35,86	35,52	35,64	35,72	35,22
41	0,06	3,74	36,34	36,33	36	36,21	36,01	36,46	36,34	36,19	36,12	36,4	36,44	36,4	35,6	36,16	35,52	36,56	36,76	36,93
42	39,12	35,97	38,33	38,17	39,44	39,11	38,22	39,12	39,18	39,2	38,39	39,7	40,15	40,29	39,39	40,29	38,73	39,91	39,85	38,82
43	0,04	0,06	0,88	0,73	0,94	0,55	0,76	0,86	0,83	0,71	0,87	0,73	0,6	0,43	0,39	0,47	0,29	0,28	0,08	0,41
44	0,06	20,8	35,36	35,7	35,75	35,73	36,24	35,85	36,6	36,23	36,52	36,98	36,8	36,79	36,51	36,94	36,42	41,17	36,68	36,87
45	0,58	2,55	4,15	7,69	8,09	4,62	5,77	9,75	8,71	8	6,83	5,84	10,42	8,67	10,18	5,98	7,65	8,31	6,74	9,71
46	0,05	7,27	10,49	9,96	22,14	36,28	36,06	37,2	36,07	36,68	37,26	36,44	36,74	37,53	36,97	37,18	36,82	36,57	35,34	39,48
47	7,45	29,84	36,21	36,1	36,47	36,3	36,15	35,83	36,2	36,01	35,66	36,71	36,96	34,82	36,93	36,18	36,11	36,42	36,46	37,56
48	36,62	36,52	35,84	36,39	36,52	37,05	36,57	37,46	36,25	36,32	36,54	36,77	36,1	36,32	36,25	36,37	36,48	37,1	36,45	36,51
49	0,03	37,03	36,38	37,32	37,23	36,92	37,17	37,73	37,55	37,02	38,05	37,63	37,59	37,02	37,62	38,21	38,16	38,49	37,33	36,99
50	9,26	36,3	37,69	37,89	37,49	38,38	38,65	38,99	38,16	38,21	38,8	38,79	38,28	37,92	38,52	38,53	38,52	39,24	39,54	38,62

Tablo 3.11. Veri Temizleme ve Bütünleştirme İşleminin Ardından RAM Logları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	689	693	697	702	700	706	703	706	705	702	708	711	707	701	705	703	705	702	706	703
2	153	154	159	162	163	164	166	167	167	151	156	159	162	162	163	163	166	167	168	173
3	654	683	686	691	686	685	684	683	686	685	684	686	688	684	687	685	688	689	684	687
4	148	167	209	222	236	263	288	301	342	346	369	373	374	374	375	378	379	378	379	379
5	694	704	697	693	692	694	693	693	692	693	690	689	692	690	692	688	686	684	683	684
6	696	702	703	701	706	705	706	707	706	704	703	706	708	705	704	708	713	708	713	708
7	692	695	701	703	701	696	698	702	704	701	703	701	705	706	703	700	706	707	706	709
8	704	697	703	704	702	709	705	706	710	709	706	705	707	708	705	709	706	712	712	710
9	707	709	706	707	706	710	706	705	707	706	708	705	704	707	708	712	707	705	706	712
10	145	149	154	159	162	162	163	163	164	165	165	167	167	168	169	170	170	171	171	172
11	658	659	660	663	664	666	668	669	671	672	673	672	667	674	675	673	661	647	633	620
12	239	253	269	312	302	305	340	353	344	337	347	348	338	331	339	344	347	348	344	341
13	483	495	588	589	628	631	625	650	664	654	644	645	648	642	632	622	609	595	585	572
14	695	701	698	696	699	697	699	700	704	700	701	694	697	696	701	700	701	703	706	707
15	686	695	696	700	700	696	694	692	696	693	700	699	698	698	696	697	700	698	696	694
16	698	704	701	706	701	705	707	701	708	704	707	705	707	706	708	707	708	703	706	703
17	682	695	698	700	703	705	704	706	703	705	707	711	705	703	704	703	702	700	704	702
18	145	148	153	163	166	168	169	169	170	170	171	171	172	171	172	172	172	172	173	173
19	694	701	694	692	696	699	698	697	698	701	697	698	695	700	695	694	698	700	698	697
20	151	152	157	159	179	192	197	200	223	224	232	248	259	297	294	318	315	308	320	361
21	520	514	539	663	667	668	671	672	674	674	675	676	675	677	676	679	680	677	665	655
22	704	700	705	711	706	704	705	709	708	710	707	701	705	711	708	707	709	703	713	707
23	677	695	694	695	697	699	700	701	704	704	704	702	703	700	702	706	704	707	702	699
24	704	694	699	694	702	698	703	700	701	708	706	705	704	703	706	707	709	706	707	703
25	687	698	705	703	700	699	702	701	706	704	705	707	706	709	700	707	709	710	709	708
26	694	693	698	701	695	705	700	703	704	701	700	703	700	704	702	704	703	706	702	703
27	148	149	152	154	157	160	162	163	163	164	159	179	192	197	200	205	207	209	215	219
28	681	685	688	687	689	690	687	688	690	688	686	689	690	685	693	690	689	690	686	687
29	198	241	255	276	295	319	339	341	338	342	363	357	351	347	357	354	354	351	354	360
30	319	322	334	375	376	424	432	513	538	617	635	626	637	665	665	666	666	667	668	670
31	694	700	698	699	701	698	702	704	701	700	704	700	697	698	700	698	703	702	699	703
32	684	697	699	701	705	702	705	701	702	705	703	705	699	698	701	703	706	702	705	703
33	699	704	702	703	702	706	703	688	680	670	660	650	640	630	620	609	599	589	578	568
34	686	692	690	691	690	692	694	699	694	695	694	697	698	699	697	701	705	704	701	707
35	154	157	159	162	163	162	163	163	164	164	164	163	164	165	166	167	167	168	169	169
36	488	659	661	661	662	663	668	670	666	674	675	676	676	670	673	674	675	673	667	653
37	148	152	155	160	162	163	163	164	180	199	218	256	271	280	284	277	267	257	247	236
38	145	152	154	159	162	163	163	163	163	164	166	167	181	193	220	241	256	278	286	276
39	631	704	700	703	707	704	705	709	707	706	703	702	706	709	704	708	702	704	705	702
40	701	702	705	703	702	704	702	704	706	705	709	704	707	705	704	705	714	707	703	683
41	692	706	704	701	700	703	707	705	699	702	703	705	701	702	702	701	699	702	701	705
42	516	686	694	695	696	699	698	696	698	699	697	698	694	698	696	695	698	697	696	697
43	150	151	152	159	162	163	162	163	164	166	168	169	169	170	170	171	171	172	171	172
44	695	700	701	700	696	694	697	699	693	702	697	698	691	690	690	695	699	696	698	692
45	145	149	154	159	162	164	192	224	265	266	290	307	333	337	330	331	328	321	325	328
46	690	696	698	693	695	696	697	695	695	697	699	699	701	702	697	700	702	703	699	700
47	706	702	706	704	702	698	699	710	711	710	711	710	712	708	704	712	708	706	703	702
48	690	699	698	699	697	696	694	695	700	696	695	693	698	696	695	700	696	693	701	700
49	693	697	698	699	701	709	706	704	701	704	703	707	704	702	701	706	697	702	703	698
50	700	703	705	704	706	703	705	707	711	705	703	704	703	702	700	704	702	698	709	706

Tablo 3.12. Veri Temizleme ve Bütünleştirme İşleminin Ardından NETI Logları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	102	103	254	894	1790	2710	3590	4480	5490	6420	7430	8390	9370	1030	1130	1230	1330	1440	1540	1640
2	393	414	434	456	476	498	518	531	543	555	564	569	575	579	580	587	609	631	651	674
3	1410	1420	1460	1520	1580	1670	1760	1860	1950	2050	2140	2260	2360	2450	2550	2640	2740	2830	2930	3030
4	184	185	75	184	364	571	840	1170	1580	1990	2470	2990	3520	4050	460	5160	570	6260	6820	7370
5	251	428	991	1740	2630	3480	4370	5290	6240	7240	8180	9150	1010	1110	1210	130	140	150	1590	1690
6	182	629	1060	2250	3530	4790	6070	7370	8710	1010	1140	1270	1410	1550	1680	1820	1950	2080	2220	2350
7	116	481	562	1280	2320	3230	4290	5330	6350	7390	850	9820	110	1270	1370	150	1640	1780	1880	1990
8	802	564	1450	2310	3150	4030	4930	5940	690	7820	8790	9730	1090	1180	1280	1380	1480	1580	1680	1780
9	1540	2540	3480	450	5620	6790	790	910	1030	1140	1250	1370	1480	1590	1710	1820	1940	2050	2170	2280
10	108	288	493	702	899	112	132	151	172	191	211	231	248	260	272	283	289	292	296	317
11	118	536	473	742	1010	1150	1330	1670	2160	2480	320	4050	4660	5020	5350	5750	6140	6550	7410	8160
12	155	31	345	347	348	478	68	952	134	181	236	295	356	462	548	633	735	818	881	957
13	177	179	272	50	949	195	334	467	633	833	1030	1230	150	1780	2020	2240	2460	2690	2910	3080
14	5310	6370	7470	8550	9630	1080	1190	1290	140	1510	1620	1730	1830	1940	2050	2150	2260	2370	2480	2590
15	3050	3980	4880	590	6910	7970	9030	1010	1120	1220	1320	1420	1530	1630	1730	1830	1940	2050	2160	2260
16	7740	8890	1010	1120	1240	1350	1460	1580	1690	1810	1930	2040	2150	2270	2390	250	2610	2730	2840	2960
17	8310	9550	1080	120	1330	1460	1590	1720	1840	1970	210	2230	2350	2480	2610	2740	2860	2990	3120	3250
18	179	227	275	320	359	395	422	445	450	469	492	516	539	563	585	609	631	654	671	690
19	590	6990	8120	9210	1030	1150	1260	1360	1480	1590	170	1810	1920	2030	2140	2250	2360	2470	2580	270
20	354	476	583	701	837	936	1070	1260	1410	1630	1850	210	2270	250	2810	3020	3240	3550	3790	4010
21	1450	1650	190	2130	2390	2640	2910	3390	3880	4370	5020	5470	5920	6440	7070	7670	8260	8780	9360	9910
22	7230	8310	9420	1050	1160	1270	1380	1490	1590	1710	1820	1930	2040	2150	2260	2370	2490	2590	270	2820
23	1130	1930	2730	3490	4250	5030	5790	660	7390	8160	8970	9790	1060	1140	1220	130	1370	1450	1520	1590
24	1470	2150	3140	3990	4850	5710	6690	7670	8670	9660	1060	1160	1260	1360	1460	1560	1660	1760	1860	1960
25	3310	4290	5340	6430	7570	8690	9720	1080	1190	130	1410	1530	1640	1750	1860	1970	2080	2190	230	2410
26	1830	270	3620	4490	5470	6420	7420	8420	9380	1040	1140	1240	1340	1440	1540	1640	1740	1840	1960	2050
27	110	138	164	192	220	249	278	306	333	362	388	413	438	463	488	511	537	562	588	620
28	1970	2960	3980	4970	5930	6930	7920	8930	9930	110	120	1310	1410	1510	1610	1710	1810	1910	2010	2110
29	991	124	151	177	203	229	270	323	392	457	577	668	826	944	110	1190	1320	1420	1560	1730
30	5950	6260	650	680	7080	7370	7640	7960	820	8470	8690	8920	9170	9340	9540	9720	990	1010	1020	1030
31	2440	330	4160	5120	610	7080	8080	9020	9980	110	1190	1290	140	150	160	170	180	190	200	210
32	238	605	1470	2280	3070	3880	4850	5780	6770	7710	8650	9620	1060	1160	1270	1370	1460	1560	1660	1760
33	3110	3220	3330	3440	3550	3660	3770	3880	3990	4060	4080	4090	410	4110	4120	4140	4150	4160	4170	4180
34	1790	1890	1980	2080	2170	2270	2370	2470	2570	2670	2770	2870	2960	3070	3160	3260	3360	3460	3560	3650
35	246	260	272	284	293	299	300	318	338	359	380	401	422	445	466	487	507	526	542	554
36	1830	2250	2510	3010	3780	4320	4790	5070	5310	5630	5970	630	6650	7260	8060	8580	8930	9190	9470	9810
37	204	223	243	257	270	282	292	298	299	300	323	342	363	382	402	422	445	470	498	529
38	206	228	248	261	273	286	293	298	299	300	323	345	366	385	407	427	446	465	487	516
39	1430	1540	1650	1770	1890	200	2120	2240	2360	2470	2590	270	2810	2920	3040	3150	3260	3370	3480	360
40	7810	9150	1040	1170	1310	1450	1580	1710	1850	1980	2110	2250	2380	2510	2650	2780	2910	3040	3170	330
41	6280	7050	7830	8640	950	1030	110	1180	1260	1340	1420	150	1580	1660	1740	1820	190	1970	2050	2130
42	1960	2120	230	2470	2640	2810	2980	3150	3320	3490	3660	3830	3990	400	4010	4020	4040	4050	4060	4080
43	219	237	254	266	279	289	295	296	304	327	349	371	393	414	437	459	482	506	529	551
44	160	2750	3860	4880	6030	7190	8360	9550	1080	1190	1310	1420	1550	1670	1790	1910	2020	2140	2250	2370
45	200	225	246	263	295	371	531	693	879	1110	1390	1620	1950	220	2470	2780	3060	3330	3640	3940
46	6180	7150	810	9060	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250
47	8870	9810	1080	1180	1280	1370	1480	1570	1670	1770	1870	1970	2070	2160	2260	2360	2460	2560	2660	2760
48	5320	6260	7280	8250	9260	1030	1120	1220	1320	1430	1520	1620	1720	1820	1920	2020	2120	2220	2310	2420
49	110	1220	1330	1440	1550	1660	1780	1890	2010	2120	2230	2340	2460	2570	2680	280	2910	3020	3140	3250
50	120	1330	1460	1590	1720	1840	1970	210	2230	2350	2480	2610	2740	2860	2990	3120	3250	3380	3510	3650

Tablo 3.13. Veri Temizleme ve Bütünleştirme İşleminin Ardından NETO Logları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	3630	390	4130	4340	4540	4730	4920	5110	530	550	5710	5920	6150	6390	6640	6880	710	7350	7570	7790
2	651	685	723	828	897	954	1010	1080	1150	120	1230	1280	1330	105	139	175	212	243	277	311
3	5270	5450	5640	5820	5990	6160	6340	6510	6690	6860	7040	7230	740	7580	7760	7940	8120	8290	8470	8750
4	1610	2240	3040	3830	4740	5710	6710	7720	8750	9830	1090	1190	130	1410	1520	1620	1730	1840	1940	2050
5	2980	3160	3340	3520	370	3890	4070	4210	4320	4410	4480	4550	4630	4730	4820	490	4960	5020	510	5160
6	5060	5360	5660	5950	6250	6560	6890	7210	7510	7780	8070	8080	8090	810	8180	8460	8750	9050	9190	9220
7	4160	4360	4550	4760	4970	5210	5450	5670	5950	6220	6460	6710	6920	7140	7350	7550	7760	7970	8180	8460
8	3790	4020	4260	450	4710	490	510	5320	5530	5750	5960	6180	6410	6640	6840	7060	7270	7480	7690	7830
9	4410	4660	490	5140	5380	5630	5870	6120	6370	6620	6870	7150	7390	7610	7620	7640	7650	7660	7680	770
10	531	537	545	625	656	690	722	756	791	825	857	891	925	960	987	1010	1030	1050	1060	1070
11	1020	1090	1250	140	1490	1540	160	1680	1750	1830	2050	2080	2230	2440	2620	2890	3080	3240	3410	3580
12	1530	1640	1780	1940	2080	2230	2440	2620	2890	3080	3240	3410	3580	3730	3970	4170	440	4620	4790	4950
13	4430	4850	5250	5610	5920	6160	6430	6630	6830	7050	720	7380	7550	7750	7920	8070	8240	840	8680	890
14	3460	3690	3940	4190	4430	4670	490	5120	5350	5580	5850	6110	6360	6610	6880	7110	7350	7510	410	8020
15	280	3010	3230	3450	3690	3920	4130	4370	4620	4860	5080	530	5510	5720	5940	6170	640	6640	6860	7090
16	4120	440	4650	4910	5160	5410	5650	590	6150	640	6670	6950	7220	7490	7740	800	8010	8020	8030	8040
17	4460	4750	5040	5310	5590	5870	6160	6430	6710	700	730	7580	7870	8170	8460	8750	9050	9190	9220	9240
18	927	966	1010	1040	1090	1120	1160	120	1240	1270	130	1330	1350	1370	1440	1490	1530	1570	1610	1650
19	2980	3170	3370	3570	3770	3970	4160	4350	4560	4760	4960	5150	5350	5550	5760	5950	6110	6280	6640	6860
20	2980	3390	3850	4160	4590	5170	5560	5960	6540	6990	7380	7810	8270	8740	9160	9660	1010	1060	1120	1160
21	80	920	1010	1090	1180	130	1410	1510	1610	1720	1820	1930	2020	2130	2230	2340	2450	2550	2660	2770
22	3930	4180	4460	4710	4960	520	5440	570	5930	6160	6420	6680	6950	720	7460	7690	7940	8170	8390	8630
23	1720	1880	2030	2190	2350	2520	2690	2860	3020	3180	3360	3540	3720	390	4090	4250	4420	4580	4750	4930
24	3670	3890	4110	4340	4580	4810	5050	5280	550	570	5910	6120	6350	6580	680	7040	7260	7480	7710	7930
25	4590	4830	5070	5320	5550	5780	6010	6250	6480	6710	6950	7190	7430	7680	7920	8170	8390	8640	8860	9060
26	310	3340	3590	3890	4130	4380	4580	4770	4970	5150	5330	5540	5770	600	6240	6520	6770	7020	7250	7470
27	819	854	890	925	967	1020	1070	1130	1170	1220	1270	1310	1360	1410	1460	1570	1690	1750	1830	1890
28	2520	2710	2890	3080	3270	3450	3630	3810	400	4170	4360	4540	4720	4910	5110	530	5480	5670	5860	6060
29	1540	1740	2010	2180	240	2590	2850	3160	3450	370	390	4130	4410	470	5020	5280	5530	5780	6030	6260
30	3650	4110	4470	4970	5380	5830	6260	6690	7150	7590	8030	8530	9010	9510	9930	1040	1080	1140	1190	1240
31	320	3420	3650	3880	4120	4340	4570	4790	4990	5210	5410	5620	5840	6080	6310	6530	6740	6970	7190	7410
32	2360	2560	2770	2970	3170	3360	3580	3780	400	4220	4430	4660	4870	5090	530	5490	5690	590	610	630
33	9350	9370	940	9420	9440	9470	9490	9520	9540	9070	9090	9110	9140	9160	9190	9210	9230	9250	9210	9210
34	5250	5460	5670	5890	6120	6340	6550	6760	6960	7160	7360	7580	7790	7990	820	8390	860	8820	9020	9160
35	544	618	652	689	724	758	793	833	868	903	937	969	999	1020	1040	1060	1070	1150	1190	1220
36	8430	8820	9340	9920	1050	110	1210	1380	1460	1510	1550	160	1660	1730	1770	1860	200	1750	1830	2050
37	531	540	543	544	629	662	696	729	760	794	834	877	924	980	1040	110	1150	1220	1270	1310
38	533	542	543	545	629	665	701	732	770	806	837	869	904	954	1010	1050	1110	1170	1220	1250
39	4910	5180	5430	5680	5930	6180	6430	6680	6960	7210	7480	7720	7960	8210	8480	8720	8970	9220	9410	9420
40	380	410	4410	4690	4980	5250	5540	5820	610	6390	6670	6970	7270	7560	7870	8170	8460	8750	9050	9190
41	250	2660	2830	3010	3180	3360	3550	3730	3910	4080	4260	4420	460	4760	4930	5090	5260	5420	5590	5750
42	6690	7060	7450	7830	8210	8580	8940	8970	900	9020	9040	9070	9090	9110	9140	9160	9190	9210	9230	9250
43	537	540	557	608	656	701	746	791	839	884	923	956	989	1020	1060	1090	1130	1180	1230	1270
44	1650	1870	2080	2290	2490	2730	2950	3160	3370	3570	3790	400	4210	4440	4650	4870	5090	530	5520	5740
45	969	1270	1610	2050	2570	30	3610	4070	4570	5120	5640	6110	6680	7210	7750	8310	8830	9440	990	1040
46	220	2390	2580	2760	2950	3140	3320	350	3680	3870	4060	4250	4430	460	4790	4970	5150	5320	550	5680
47	3380	360	3820	4060	4290	4510	4730	4930	5140	5370	5570	580	6020	6260	6490	6720	6940	7150	7370	760
48	2850	3060	3260	3460	3670	3890	410	4310	4520	4740	4950	5180	5390	560	580	6010	6220	6450	6670	6880
49	3860	4110	4350	4580	4820	5090	5350	5580	5830	6060	630	6540	6780	7020	7270	7520	7770	8010	8280	8540
50	4460	4750	5040	5310	5590	5870	6160	6430	6710	700	730	7580	7870	8170	8460	8750	9050	9190	9220	9240

3.3.3. Z-Skor ile Veri Seçme

Z-Skor

Verilerin arasında anormallik gösteren değerler Z-Skor ile tespit edilebilmektedir. Anormalliği tespit etmek için Z-Skor ideal bir göstergedir. Herhangi bir metrik tipi normal değerlerine göre farklılık gösteriyor ise, bu farklılığın düzeyine göre Z-Skor değeri de yüksektir. Bu sayede değerlerin Z-Skorlarının yükseklik seviyesine göre, anormal durumlar tespit edilebilmektedir.

$$z_i = \frac{x_i - \bar{x}}{\sigma} \tag{3.2}$$

Veri seçme işlemleri için de Şekil 3.20 ile gösterilen süreç modelinde Z-Skor'un Önerilen mimaride oynadığı rol görülmektedir. Mevcut simülasyon datalarının yük tipi, zafiyet tipi, güvenlik seviyesi gibi kriterler kapsamında ne düzeyde farklılaştığı Z-Skor ile analiz edilecektir. Farklılaşma düzeylerinin de ne olduğuna yönelik analiz Z-Skor aracılığıyla sağlanacaktır. Z-Skor dönüşüm denklemi 3.2. üzerinde gösterildiği şekilde uygulanmaktadır.

Yük Yapısına Göre Z-Skor Çıktılarının Güvenlik Düzeyine Yönelik Analiz Örnekleri

Yük dağılımının güvenlik düzeyinin değişmesi durumunda aynı tip zafiyetlerde farklılık gösterme düzeyine yönelik analiz için çalışmalar yapılmıştır. Bu çalışmalarda amaç, zafiyet tipi aynı bile olsa yük dağılımının ne düzeyde etki gösterdiğini ve Z-skor'un bu farkı anlamada ne düzeyde faydalı olduğuna dair analizin yapılabilmesini sağlayacak verilerin elde edilmesidir.

Tablo 3.14. Yük Yapıları

Yük Yapısı				
Yük Yapı Numarası	**Normal Kullanıcı Yükü**	**Zafiyet Yükü**		
		Zafiyet Tarama Oranı	**OWASP Zap Tarama Oranı**	**Scripted Zafiyet Tarama Oranı**
1	300	0	0	0
2	0	300	0	300
3	0	300	300	0
4	0	300	240	60
5	0	300	60	240
6	240	60	0	60
7	240	60	60	0
8	240	60	48	12
9	240	60	12	48

Örnek olarak 2 numaraları yük tipi için CPU logları ile oluşturulan Şekil 3.46 ve Şekil 3.47. kıyaslandığında Z-Skor'un gerekliliğine dair gözlem mümkün olabilmektedir. Bu gözlem sonucunda Z-Skor'un yük tipine ve kaynak tipine göre güvenlik düzeyinin değişmesi durumunda normal metrik değerlerine kıyasla z-skor değerlerinin zafiyetin gerçekleşmesi durumunda ne kadar farklılık gösterdiği gözlemlenebilecektir.

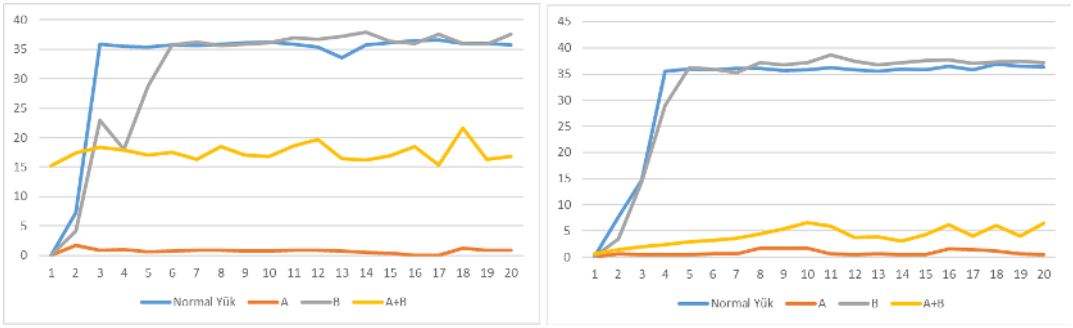
Şekil 3.46 üzerindeki yüksek ve düşük güvenlik düzeylerine dair dataların benzeştiği gözlemlenirken, Şekil 3.47. üzerinde Z-Skordan kullanımından dolayı anormal veriler normal verilere göre düşük güvenlik düzeyinde daha ciddi bir ayrışma göstermektedir. Bu gözlem tüm yük tipleri ve kaynak tipleri ile yapılarak Z-Skor'un gerekliliği analiz edilecektir.

Tablo 3.15. CPU için 2 Numaralı Yük Tipinin Düşük Güvenlik Düzeyindeki Log Değerleri

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Normal Yük	0,04	7,26	35,89	35,56	35,39	35,72	35,66	35,91	36,12	36,26	35,83	35,43	33,65	35,8	36,11	36,54	36,65	35,96	36,04	35,78
A	0,04	1,66	0,82	0,98	0,58	0,75	0,8	0,88	0,74	0,75	0,82	0,86	0,71	0,45	0,36	0,05	0,06	1,19	0,82	0,87
B	0,03	4,08	22,98	17,95	28,75	35,71	36,23	35,63	35,91	36,14	36,93	36,73	37,22	37,94	36,32	35,97	37,54	36,05	35,84	37,58
A+B	15,21	17,38	18,4	17,94	17,05	17,47	16,33	18,48	17,04	16,82	18,66	19,75	16,39	16,17	16,94	18,54	15,35	21,58	16,32	16,8

Tablo 3.16. CPU için 2 Numaralı Yük Tipinin Yüksek Güvenlik Düzeyindeki Logları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Normal Yük	0,05	7,67	14,72	35,48	35,96	35,81	36,13	36,11	35,7	35,79	36,26	35,83	35,49	35,99	35,86	36,5	35,84	36,86	36,5	36,42
A	0,04	0,64	0,52	0,46	0,48	0,59	0,6	1,75	1,7	1,72	0,59	0,51	0,55	0,47	0,52	1,5	1,49	1,11	0,59	0,52
B	0,08	3,53	14,46	29,06	36,23	35,97	35,21	37,18	36,82	37,17	38,61	37,49	36,76	37,22	37,62	37,77	37,09	37,36	37,38	37,23
A+B	0,56	1,44	1,95	2,4	2,87	3,25	3,6	4,44	5,32	6,56	5,91	3,74	3,84	3,06	4,24	6,17	4,02	6,04	4,04	6,48



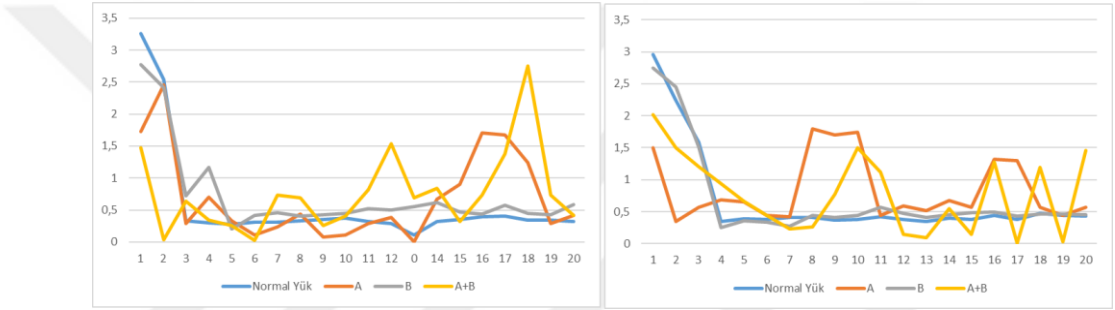
Şekil 3.46. Tablo 3.15.(Sol) ve Tablo 3.16.(Sağ) Verileri Dikkate Alınarak 2 Numaralı Yük Dağılımının CPU Üzerindeki Hareketlerinin Düşük(Sol) ve Yüksek(Sağ) Güvenlik Düzeyindeki Grafiği

Tablo 3.17. CPU için 2 Numaralı Yük Tipinin Düşük Güvenlik Düzeyindeki Log Değerlerinin Z-Skorları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Normal Yük	3,260	2,537	0,332	0,299	0,282	0,315	0,309	0,334	0,355	0,369	0,326	0,286	0,107	0,323	0,354	0,397	0,408	0,339	0,347	0,321
A	1,730	2,456	0,286	0,699	0,335	0,105	0,234	0,441	0,079	0,105	0,286	0,389	0,001	0,671	0,903	1,704	1,678	1,242	0,286	0,415
B	2,769	2,408	0,722	1,171	0,208	0,413	0,460	0,406	0,431	0,452	0,522	0,504	0,548	0,612	0,468	0,437	0,577	0,444	0,425	0,580
A+B	1,471	0,034	0,642	0,337	0,252	0,026	0,729	0,695	0,259	0,405	0,814	1,536	0,689	0,835	0,325	0,734	1,378	2,747	0,736	0,418

Tablo 3.18. CPU için 2 Numaralı Yük Tipinin Yüksek Güvenlik Düzeyindeki Log Değerlerinin Z-Skorları

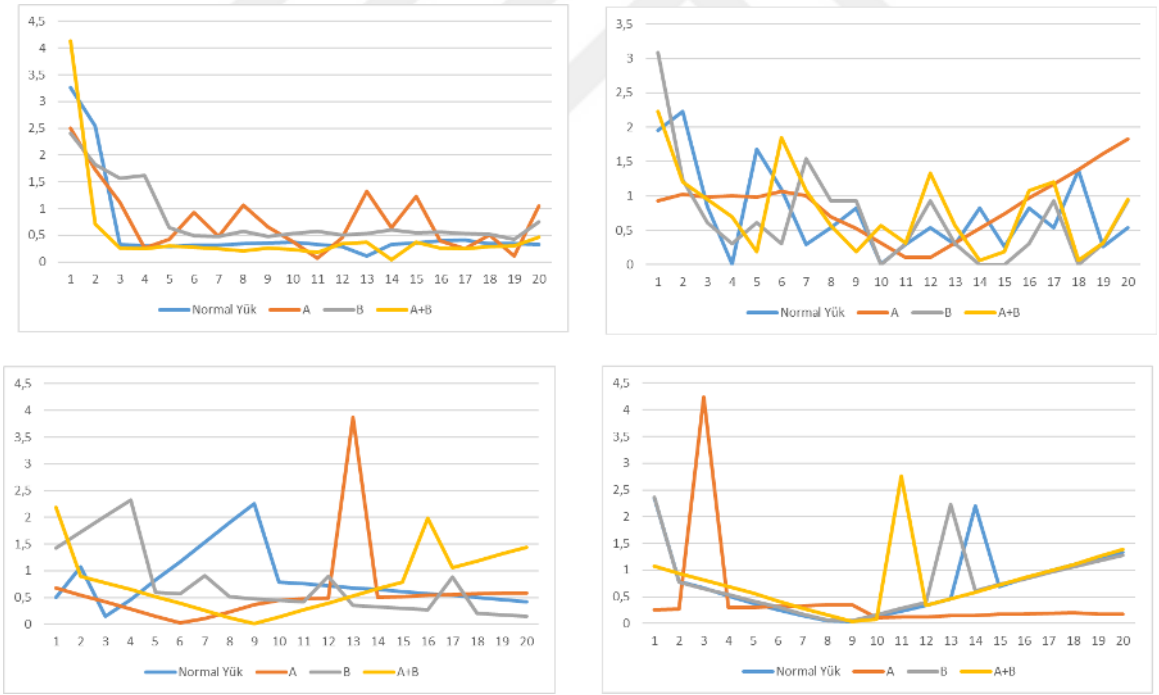
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Normal Yük	2,953	2,243	1,587	0,348	0,392	0,378	0,408	0,406	0,368	0,377	0,420	0,380	0,349	0,395	0,383	0,443	0,381	0,476	0,443	0,435
A	1,501	0,343	0,574	0,690	0,651	0,439	0,420	1,800	1,703	1,742	0,439	0,594	0,516	0,671	0,574	1,317	1,298	0,565	0,439	0,574
B	2,748	2,451	1,510	0,254	0,363	0,341	0,275	0,445	0,414	0,444	0,568	0,471	0,409	0,448	0,483	0,495	0,437	0,460	0,462	0,449
A+B	2,014	1,498	1,199	0,935	0,660	0,437	0,232	0,260	0,775	1,502	1,121	0,150	0,092	0,549	0,143	1,274	0,014	1,197	0,025	1,455



Şekil 3.47. Tablo 3.17. (Sol) ve Tablo 3.18.(Sağ) Verileri Dikkate Alınarak 2 Numaralı Yük Dağılımının CPU Üzerindeki Hareketlerinin Düşük(Sol) ve Yüksek(Sağ) Güvenlik Düzeyindeki Değerlerinin Z-Skorunun Grafiği

8 Numaralı yük yapısı üzerinden Şekil 3.48. ve Şekil 3.49. kıyaslandığında Z-Skor'un, güvenlik düzeyinin değişmesi durumunda güvenlik düzeyinin etkisine dair gözlem yapılabilecektir. Normal yük ile diğer zafiyet sömürülerinin ne düzeyde güvenlik düzeyine bağımlı olarak benzeştiği gözlemlenebilmektedir. Burada her yük yapısı için bu durum gözlenecektir ve bulgularda açıklanacaktır.

Şekil 3.48. Kaynakların 8 Numaralı Yük Yapı Numarasına Göre Düşük Güvenlik Düzeyi için Z-Skorlarının Grafikleri (Sol Üst-CPU, Sağ Üst RAM, Sol Alt-NETI, Sağ Alt-NETO)



Şekil 3.49. Kaynakların 8 Numaralı Yük Yapı Numarasına Göre Yüksek Güvenlik Düzeyi için Z-Skorlarının Grafikleri (Sol Üst-CPU, Sağ Üst RAM, Sol Alt-NETI, Sağ Alt-NETO)

Zafiyet Tipine Göre Z-Skor Çıktılarının Güvenlik Düzeyine Yönelik Analiz Örnekleri

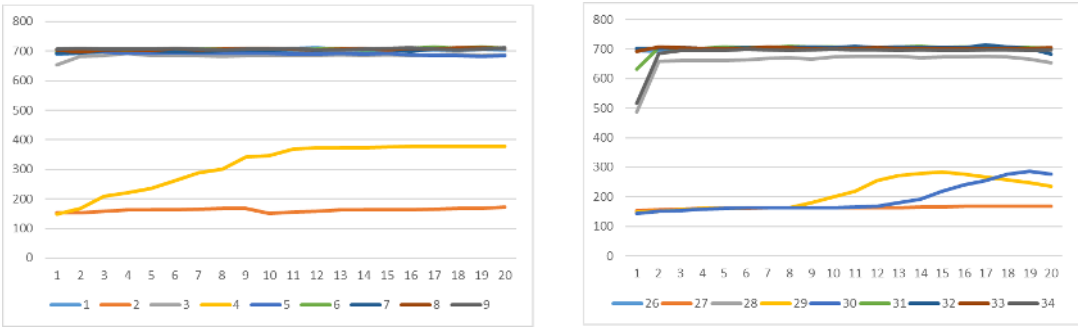
Farklı zafiyetlerin davranışlarının yük tipinden bağımsız olarak davranışının anormalliğinin Z-Skor'a göre tespit edilebilirliğine yönelik analiz için çalışmalar gerçekleştirilmiştir. Tablo 3.7 üzerindeki zafiyet tipleri için güvenlik düzeyi kapsamındaki kıyası yapılarak Tablo 3.14. üzerindeki yük yapıları bağımsız tutularak Z-Skor'a göre zafiyetlerin ne düzeyde farklılık gösterdiğine yönelik analiz yapılacaktır.

Tablo 3.19. A Zafiyetinin Sömürülmesi Durumunda Düşük Güvenlik Düzeyi İçin RAM Kaynağından Toplanan Log Verileri

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	689	693	697	702	700	706	703	706	705	702	708	711	707	701	705	703	705	702	706	703
2	153	154	159	162	163	164	166	167	167	151	156	159	162	162	163	163	166	167	168	173
3	654	683	686	691	686	685	684	683	686	685	684	686	688	684	687	685	688	689	684	687
4	148	167	209	222	236	263	288	301	342	346	369	373	374	374	375	378	379	378	379	379
5	694	704	697	693	692	694	693	693	692	693	690	689	692	690	692	688	686	684	683	684
6	696	702	703	701	706	705	706	707	706	704	703	706	708	705	704	708	713	708	713	708
7	692	695	701	703	701	696	698	702	704	701	703	701	705	706	703	700	706	707	706	709
8	704	697	703	704	702	709	705	706	710	709	706	705	707	708	705	709	706	712	712	710
9	707	709	706	707	706	710	706	705	707	706	708	705	704	707	708	712	707	705	706	712

Tablo 3.20. A Zafiyetinin Sömürülmesi Durumunda Yüksek Güvenlik Düzeyi İçin RAM Kaynağından Toplanan Log Verileri

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
26	694	693	698	701	695	705	700	703	704	701	700	703	700	704	702	704	703	706	702	703
27	154	157	159	162	163	162	163	163	164	164	164	163	164	165	166	167	167	168	169	169
28	488	659	661	661	662	663	668	670	666	674	675	676	676	670	673	674	675	673	667	653
29	148	152	155	160	162	163	163	164	180	199	218	256	271	280	284	277	267	257	247	236
30	145	152	154	159	162	163	163	163	163	164	166	167	181	193	220	241	256	278	286	276
31	631	704	700	703	707	704	705	709	707	706	703	702	706	709	704	708	702	704	705	702
32	701	702	705	703	702	704	702	704	706	705	709	704	707	705	704	705	714	707	703	683
33	692	706	704	701	700	703	707	705	699	702	703	705	701	702	702	701	699	702	701	705
34	516	686	694	695	696	699	698	696	698	699	697	698	694	698	696	695	698	697	696	697



Şekil 3.50. Tablo 3.19 (Sol) ve Tablo 3.20 (Sağ) Verileri Dikkate Alınarak A Kodlu Zafiyetin RAM Üzerindeki Hareketlerinin Düşük(Sol) ve Yüksek(Sağ) Güvenlik Düzeyindeki Grafiği

Şekil 3.50. üzerinde görülmektedir ki örnek olarak kullanılan RAM verileri A zafiyet tipi için log dataları zorluğa göre gözlemlendiğinde ciddi bir benzeşme göstermektedir. Bu sebeple yük yapısına göre anormalliğin yük yapısına göre Z-Skor kullanılarak yapılması gerekmektedir. Şekil 3.50. gözlendiğinde RAM metriklerinin zafiyetin varlığı durumundaki anormalliğin algılanması için ne düzeyde anlamlı verilere sahip olduğu incelenebilecektir.

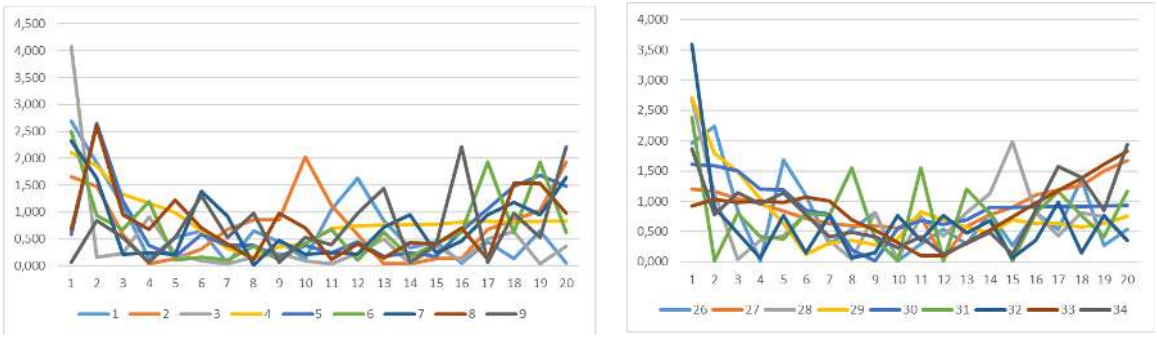
Mevcut örnekte görülmektedir ki Şekil 3.51. RAM örneği ile Şekil 3.47. üzerindeki CPU örneğini basitçe yan yana koyduğumuzda A zafiyetini gözlemek için RAM yetersiz görülse de, 2 numaralı yük tipi için A, B ve A+B zafiyetlerinin var olması durumunda CPU yeterli olarak yorumlanabilmektedir. Bu çalışmalar sonucunda yük tipinin değişimi ve zafiyet tipinin değişimine göre CPU, RAM, NETI, NETO'nun hangisinin anlamlı veri taşıdığı yorumunu yapabilmenin mümkünlüğü analiz edilecektir.

Tablo 3.21. A Zafiyetinin Sömürülmesi Durumunda Düşük Güvenlik Düzeyi İçin RAM Kaynağından Toplanan Log Verilerinin Z-Skorları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2,697	1,910	1,122	0,138	0,532	0,650	0,059	0,650	0,453	0,138	1,043	1,634	0,847	0,335	0,453	0,059	0,453	0,138	0,650	0,059
2	1,660	1,481	0,583	0,045	0,135	0,314	0,673	0,853	0,853	2,019	1,122	0,583	0,045	0,045	0,135	0,135	0,673	0,853	1,032	1,930
3	4,081	0,169	0,236	0,911	0,236	0,101	0,034	0,169	0,236	0,101	0,034	0,236	0,506	0,034	0,371	0,101	0,506	0,641	0,034	0,371
4	2,104	1,863	1,331	1,166	0,989	0,646	0,330	0,165	0,355	0,406	0,697	0,748	0,760	0,760	0,773	0,811	0,824	0,811	0,824	0,824
5	0,589	2,657	1,209	0,382	0,176	0,589	0,382	0,382	0,176	0,382	0,238	0,444	0,176	0,238	0,176	0,651	1,065	1,478	1,685	1,478
6	2,497	0,936	0,676	1,197	0,104	0,156	0,104	0,364	0,104	0,416	0,676	0,104	0,624	0,156	0,416	0,624	1,925	0,624	1,925	0,624
7	2,328	1,626	0,222	0,246	0,222	1,392	0,924	0,012	0,480	0,222	0,246	0,222	0,714	0,948	0,246	0,456	0,948	1,182	0,948	1,650
8	0,677	2,611	0,953	0,677	1,229	0,704	0,401	0,124	0,981	0,704	0,124	0,401	0,152	0,428	0,401	0,704	0,124	1,533	1,533	0,981
9	0,069	0,847	0,527	0,069	0,527	1,306	0,527	0,985	0,069	0,527	0,389	0,985	1,443	0,069	0,389	2,222	0,069	0,985	0,527	2,222

Tablo 3.22. A Zafiyetinin Sömürülmesi Durumunda Yüksek Güvenlik Düzeyi İçin RAM Kaynağından Toplanan Log Verilerinin Z-Skorları

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
26	1,956	2,233	0,846	0,014	1,678	1,096	0,291	0,541	0,818	0,014	0,291	0,541	0,291	0,818	0,264	0,818	0,541	1,373	0,264	0,541
27	1,200	1,159	1,038	0,957	0,836	0,715	0,634	0,594	0,594	0,553	0,755	0,053	0,578	0,780	0,901	1,103	1,184	1,264	1,507	1,668
28	2,682	1,127	0,039	0,350	0,427	0,816	0,350	0,039	0,816	0,039	0,738	0,427	0,816	1,127	1,982	0,816	0,427	0,816	0,738	0,350
29	2,712	1,791	1,491	1,041	0,634	0,120	0,308	0,351	0,287	0,373	0,823	0,694	0,566	0,480	0,694	0,630	0,630	0,566	0,630	0,758
30	1,604	1,583	1,496	1,199	1,192	0,845	0,787	0,201	0,020	0,552	0,682	0,617	0,696	0,899	0,899	0,906	0,906	0,913	0,921	0,935
31	2,378	0,020	0,806	0,413	0,373	0,806	0,767	1,553	0,373	0,020	1,553	0,020	1,199	0,806	0,020	0,806	1,160	0,767	0,413	1,160
32	3,594	0,893	0,478	0,062	0,769	0,145	0,769	0,062	0,145	0,769	0,353	0,769	0,478	0,686	0,062	0,353	0,976	0,145	0,769	0,353
33	0,923	1,028	0,986	1,007	0,986	1,070	1,007	0,693	0,525	0,315	0,105	0,105	0,315	0,525	0,735	0,965	1,175	1,385	1,616	1,826
34	1,864	0,778	1,140	0,959	1,140	0,778	0,416	0,489	0,416	0,235	0,416	0,127	0,308	0,489	0,127	0,851	1,574	1,393	0,851	1,936

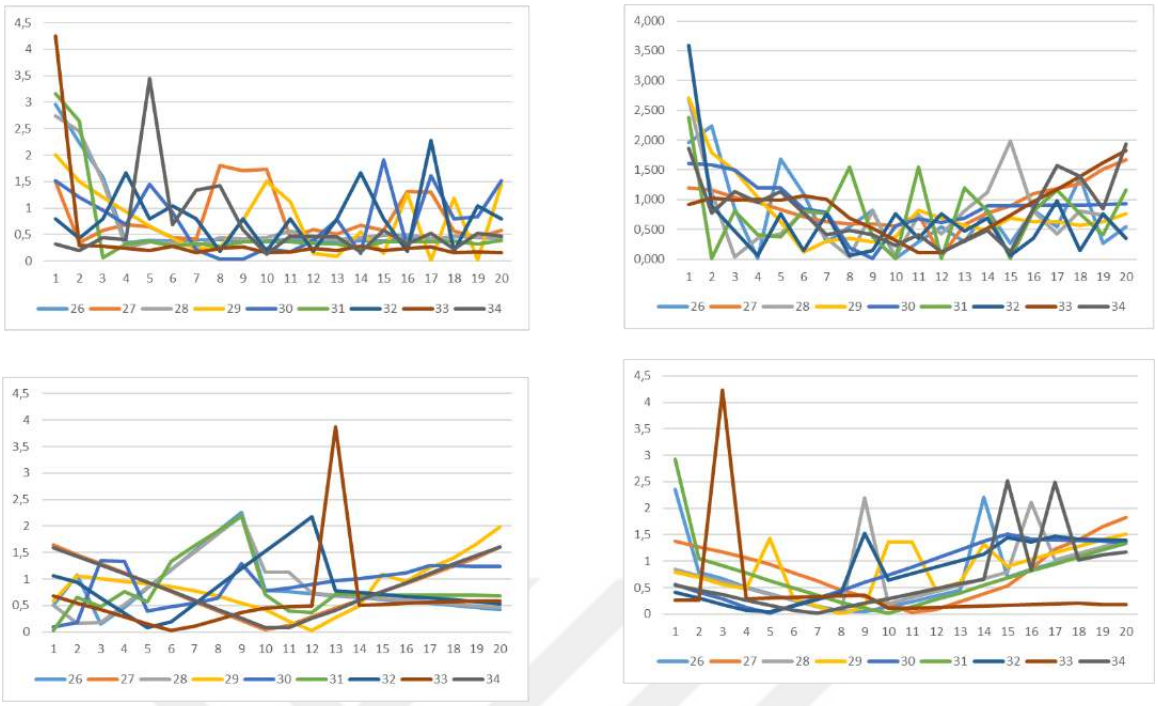


Şekil 3.51. Tablo 3.21. (Sol) ve Tablo 3.22. (Sağ) Verileri Dikkate Alınarak A Kodlu Zafiyetin RAM Üzerindeki Hareketlerinin Düşük(Sol) ve Yüksek(Sağ) Güvenlik Düzeyindeki Değerlerinin Z-Skorunun Grafiği

Şekil 3.52 ve Şekil 3.53 üzerinde A Zafiyet Tipi için farklı kaynaklar güvenlik düzeyine göre Z-Skor değerleri baz alınarak kıyaslanıldığında düşük güvenlik düzeyinde ayrışmanın hangi kaynak tipinde olduğu gözlenebilmektedir. Bu sayede her zafiyet tipi için oluşan anormal durumlarda hangi kaynağın bunu algılamak için uygun olduğu tespit edilebilecektir. Bu analizin yapılabilmesi için gerekli çalışmalar sağlanabilmiştir.

Bu tablolar kıyaslanarak örnek üzerinde görülmektedir ki zafiyetin var olması durumunda NETO kaynağı A zafiyetinin var olması durumunda diğer kaynaklara göre daha gözlemlenebilir davranış göstermektedir. Bu gözlem diğer zafiyet tipleri için, diğer kaynaklara yönelikte yapılarak Z-Skor'un doğru metriği seçmek konusundaki faydası analiz edilebilecektir.

Şekil 3.52. Kaynakların A Zafiyet Tipine Göre Göre Düşük Güvenlik Düzeyi için Z-Skorlarının Grafikleri (Sol Üst-CPU, Sağ Üst RAM, Sol Alt-NETI, Sağ Alt-NETO)



Şekil 3.53. Kaynakların A Zafiyet Tipine Göre Göre Yüksek Güvenlik Düzeyi için Z-Skorlarının Grafikleri (Sol Üst-CPU, Sağ Üst RAM, Sol Alt-NETI, Sağ Alt-NETO)

3.3.4. Z-Normalleştirme İşleminin Uygulanması

Z-Normalleştirme

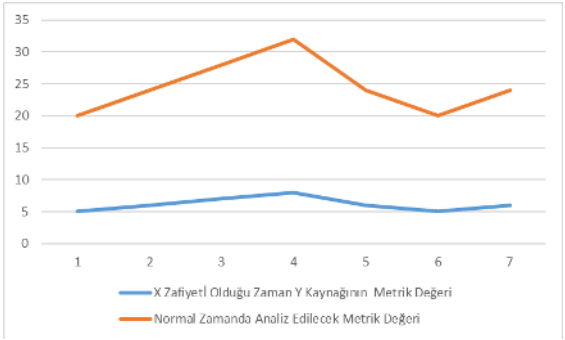
Bir zafiyetin sömürülmesi durumunda kaynağın davranışının sayısal olarak aynı olması sistemde bulunan kullanıcı miktarı, altyapı, sistem teknolojisi, internet bağlantı hızı gibi değişkenlerden dolayı değişkenlik gösterebilmektedir. Bu değişkenlik olsa dahi önemli olan belirli bir örüntünün oluşması durumunda bu örüntünün zafiyete özgü olduğunun anlaşılmasıdır.

$$z = \frac{\boldsymbol{deger - maksimum(x)}}{\boldsymbol{maksimum(x) - minimum(x)}} \qquad (\mathbf{3.3.})$$

Tablo 3.23. üzerinde Y kaynağından alınan örnek verilerle zaman akışına göre değişim zafiyetin olduğu zaman ve normal zamanda analiz edilecek değerleriyle verilmiştir. Önemli olan Şekil 3.54. görüldüğü gibi örüntü olarak benzerlik oranıdır. Bu sebeple veriler belirli bir aralıkta dönüştürülerek analiz edilmelidir. Bu dönüşüm işlemi için Z-Normalleştirme metodu 3.3. denklemi kullanarak uygulanacaktır. 0-10 aralığında ve 0-100 aralığında dönüşüm işlemi sağlanacaktır. Dönüşüm aralığının sonuca olan etkisi de analiz edilecektir.

Tablo 3.23. Herhangi Bir Kaynaktan Alınan Örnek Bir Metrik Değerleri

t	X Zafiyeti Olduğu Zaman Y Kaynağının Metrik Değeri	Normal Zamanda Analiz Edilecek Metrik Değeri
1	5	20
2	6	24
3	7	28
4	8	32
5	6	24
6	5	20
7	6	24



Şekil 3.54. Y Kaynağının Davranış Benzerliğinin Frekans Olarak Gösterim Örneği

Z-Normalleştirmenin 0-10 ve 0-100 Aralığında Uygulanması

0-10 ile 0-100 arasında Z-Normalleştirme tüm kaynakların metrik değerlerine uygulanacaktır, ardından ise bu değerlerin diğer deneylerin deneyleriyle korelasyonları analiz edilecektir. Tablo 3.24. üzerinde 1, 2, 3 ve 4 numaralı deneylerin güvenlik düzeyi düşük ve yüksek olmak üzere metrik değerleri görülmektedir. 3.4. denklemi kullanıldığında 0-10 aralığında bir dönüşüm gerçekleştirilmektedir. 0-10 dönüşümü sonucu Tablo 3.25. değerleri ortaya çıkmaktadır. 0-100 aralığı için ise 3.5. denklemi kullanılmıştır.

Tablo 3.26. üzerinde 0-100 aralığındaki dönüşüm sonucunda ortaya çıkan değerler görülmektedir. Genel olarak Tablo 3.25. ve Tablo 3.26. kıyaslandığında 0-100 aralığı daha faydalı olarak görülmektedir. Fakat bunu anlamak için her iki durumda da korelasyon durumu "Ne gibi bir sonuç verecek ve sonuçları etkileyecek bir farklılık oluşacak mı?" sorularının cevaplarına yönelik analiz yapılmalıdır.

Tablo 3.24. CPU'nun Metrik Örnekleri

Zaman\ Deney No	1	2	3	4	26	27	28	29
1	0,04	0,04	0,03	15,21	0,05	0,04	0,08	0,56
2	7,26	1,66	4,08	17,38	7,67	0,64	3,53	1,44
3	35,89	0,82	22,98	18,4	14,72	0,52	14,46	1,95
4	35,56	0,98	17,95	17,94	35,48	0,46	29,06	2,4
5	35,39	0,58	28,75	17,05	35,96	0,48	36,23	2,87
6	35,72	0,75	35,71	17,47	35,81	0,59	35,97	3,25
7	35,66	0,8	36,23	16,33	36,13	0,6	35,21	3,6
8	35,91	0,88	35,63	18,48	36,11	1,75	37,18	4,44
9	36,12	0,74	35,91	17,04	35,7	1,7	36,82	5,32
10	36,26	0,75	36,14	16,82	35,79	1,72	37,17	6,56
11	35,83	0,82	36,93	18,66	36,26	0,59	38,61	5,91
12	35,43	0,86	36,73	19,75	35,83	0,51	37,49	3,74
13	33,65	0,71	37,22	16,39	35,49	0,55	36,76	3,84
14	35,8	0,45	37,94	16,17	35,99	0,47	37,22	3,06
15	36,11	0,36	36,32	16,94	35,86	0,52	37,62	4,24
16	36,54	0,05	35,97	18,54	36,5	1,5	37,77	6,17
17	36,65	0,06	37,54	15,35	35,84	1,49	37,09	4,02
18	35,96	1,19	36,05	21,58	36,86	1,11	37,36	6,04
19	36,04	0,82	35,84	16,32	36,5	0,59	37,38	4,04
20	35,78	0,87	37,58	16,8	36,42	0,52	37,23	6,48

$$z = \frac{deger - maksimum(x)}{maksimum(x) - minimum(x)} * 10 \qquad (3.4.)$$

Tablo 3.25. CPU Metriklerinin Bir Kısmına Uygulanan 0-10 Aralığında Z-Normalleştirme Değerleri

Zaman\ Deney No	1	2	3	4	26	27	28	29
1	0	0	0	0	0	0	0	0
2	2	10	1	3	2	4	1	1
3	10	5	6	5	4	3	4	2
4	10	6	5	4	10	2	8	3
5	10	3	8	3	10	3	9	4
6	10	4	9	4	10	3	9	4
7	10	5	10	2	10	3	9	5
8	10	5	9	5	10	10	10	6
9	10	4	9	3	10	10	10	8
10	10	4	10	3	10	10	10	10
11	10	5	10	5	10	3	10	9
12	10	5	10	7	10	3	10	5
13	9	4	10	2	10	3	10	5
14	10	3	10	2	10	3	10	4
15	10	2	10	3	10	3	10	6
16	10	0	9	5	10	9	10	9
17	10	0	10	0	10	8	10	6
18	10	7	10	10	10	6	10	9
19	10	5	9	2	10	3	10	6
20	10	5	10	2	10	3	10	10

$$z = \frac{deger - maksimum(x)}{maksimum(x) - minimum(x)} * 100 \qquad (3.5.)$$

Tablo 3.26. CPU Metriklerinin Bir Kısmına Uygulanan 0-100 Aralığında Z-Normalleştirme Değerleri

Zaman\ Deney No	1	2	3	4	26	27	28	29
1	0	0	0	0	0	0	0	0
2	20	100	11	80	0	14	0	0
3	98	68	60	85	0	5	0	0
4	97	82	47	83	96	26	75	37
5	97	47	76	78	98	0	94	4
6	97	62	94	80	37	1	59	4
7	97	66	95	75	37	1	58	4
8	98	73	94	85	37	2	49	5
9	99	41	95	78	36	2	39	6
10	99	42	95	77	37	2	39	8
11	98	46	97	86	37	1	40	7
12	97	48	95	91	36	1	39	4
13	92	39	96	75	36	1	38	5
14	98	24	98	74	36	0	38	4
15	99	19	94	78	36	1	39	5
16	100	1	93	86	37	2	39	7
17	100	1	97	70	36	1	38	4
18	99	67	93	100	37	1	38	7
19	98	46	93	97	37	1	38	4
20	97	49	97	100	36	1	38	7

3.3.5. Nokta Çift Serili Korelasyon İşleminin Uygulanması

Nokta Çift Serili Korelasyon İşleminin Uygulanması

Nokta çift serili korelasyon kullanımı 2 farklı serinin arasındaki bağlantının ne düzeyde olabileceğinin tespiti için kullanılmaktadır. Önerilen model için önerilmesinin ana gerekçesi ise 2 farklı serinin bir birleriyle aynı sayıda veriye sahip olmaması durumunda dahi korelasyonun hesaplanabilmesidir. Bununla beraber, bu 2 farklı seriyi birbirinden ayıran temel faktörün serideki verilere ait bir ikili(binary) tipteki değeri içermesidir. Nokta çift serili korelasyon işlemi 3.8. denklemi üzerinde gösterildiği şekildedir.

$$\overline{x}_{z=}\frac{\sum_0^k z}{N_k} \qquad (3.6.)$$

$$ss=\sqrt{\frac{\sum x^2-\frac{(\sum x^2)}{N}}{N-1}} \qquad (3.7.)$$

$$r=\frac{\overline{x_p}-\overline{x_q}}{ss}\sqrt{p.q} \qquad (3.8.)$$

Yapılacak işlemleri örnek üzerinden açıklamak gerekirse A zafiyetinin CPU üzerinden tespit edilebileceğine dair Z-Skor değerleri elde edilmesi varsayımıyla Tablo 3.27. üzerinde bulunan değerleri ele alalım. Bu değerlerin CPU Metriklerini Tablo 3.28. üzerinde sıraladığımızda şunu söyleyebilmeliyiz ki, A zafiyetinin sömürülmesi durumunda ortaya çıkan 3 numaralı deney değerleri ile sömürülmemesi durumunda ortaya çıkan 1 ve 26 numaralı deney değerlerinin benzerlik anlamında birbirinden ayrışması gerekmektedir.

Tablo 3.29. üzerinde nokta çift serili korelasyon denemesi yapıldığında CPU metriklerinin benzerliği yani korelasyonu kıyaslandığında 1 ve 3 numaralı deneylerin 1 ve 26 numaralı deneylere kıyasla daha az benzeşmesi yani daha yüksek bir korelasyon değerine sahip olması gerekmektedir.

Tablo 3.27. A Zafiyetinin Var Olduğu ve Olmadığı Deney Örnekleri

Numara	dwva Zafiyet Zorluk	Normal Kullanıcı Yükü	Zafiyet Yükü			
			Zafiyet Tarama Oranı	Tip	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı
1	Kolay	300	0	-	0	0
3	Kolay	0	300	A	0	300
26	Çok Zor	300	0	-	0	0

Tablo 3.28. A Zafiyetinin Var Olduğu ve Olmadığı 1, 3 ve 26 Deneylerindeki CPU Metrik Değerleri

Zaman	Deney No		
	1	3	26
1	0,04	0,03	0,05
2	7,26	4,08	7,67
3	35,89	22,98	14,72
4	35,56	17,95	35,48
5	35,39	28,75	35,96
6	35,72	35,71	35,81
7	35,66	36,23	36,13
8	35,91	35,63	36,11
9	36,12	35,91	35,7
10	36,26	36,14	35,79
11	35,83	36,93	36,26
12	35,43	36,73	35,83
13	33,65	37,22	35,49
14	35,8	37,94	35,99
15	36,11	36,32	35,86
16	36,54	35,97	36,5
17	36,65	37,54	35,84
18	35,96	36,05	36,86
19	36,04	35,84	36,5
20	35,78	37,58	36,42

Tablo 3.29. CPU Metriklerinin Bağlı Olduğu Zafiyet Varlık Değeri Olan İkili Değerin Gösterimi (A Zafiyetinin Olması ve Olmaması Durumunda)

1 ve 3 No'lu Deneylerin Korelasyonu İçin Değerler		1 ve 26 No'lu Deneylerin Korelasyonu İçin Değerler	
CPU Metriği	A Zafiyeti Sömürülüyor Mu?	CPU Metriği	A Zafiyeti Sömürülüyor Mu?
0,04	Hayır	0,04	Hayır
7,26	Hayır	7,26	Hayır
35,89	Hayır	35,89	Hayır
35,56	Hayır	35,56	Hayır
35,39	Hayır	35,39	Hayır
35,72	Hayır	35,72	Hayır
35,66	Hayır	35,66	Hayır
35,91	Hayır	35,91	Hayır
36,12	Hayır	36,12	Hayır
36,26	Hayır	36,26	Hayır
35,83	Hayır	35,83	Hayır
35,43	Hayır	35,43	Hayır
33,65	Hayır	33,65	Hayır
35,8	Hayır	35,8	Hayır
36,11	Hayır	36,11	Hayır
36,54	Hayır	36,54	Hayır
36,65	Hayır	36,65	Hayır
35,96	Hayır	35,96	Hayır
36,04	Hayır	36,04	Hayır
35,78	Hayır	35,78	Hayır
0,03	Evet	0,05	Hayır
4,08	Evet	7,67	Hayır
22,98	Evet	14,72	Hayır
17,95	Evet	35,48	Hayır
28,75	Evet	35,96	Hayır
35,71	Evet	35,81	Hayır
36,23	Evet	36,13	Hayır
35,63	Evet	36,11	Hayır
35,91	Evet	35,7	Hayır
36,14	Evet	35,79	Hayır
36,93	Evet	36,26	Hayır
36,73	Evet	35,83	Hayır
37,22	Evet	35,49	Hayır
37,94	Evet	35,99	Hayır
36,32	Evet	35,86	Hayır
35,97	Evet	36,5	Hayır
37,54	Evet	35,84	Hayır
36,05	Evet	36,86	Hayır
35,84	Evet	36,5	Hayır
37,58	Evet	36,42	Hayır

Tablo 3.30. A Zafiyetinin CPU Üzerinde Var Olması ve Olmaması Durumunda Korelasyon Değerleri

Z-Normalleştirme Aralığı	1 ve 3 No'lu Deneyler için Korelasyon	1 ve 26 No'lu Deneyler için Korelasyon
10	0,369908517	0,354101492
100	0,759444278	0,704119279

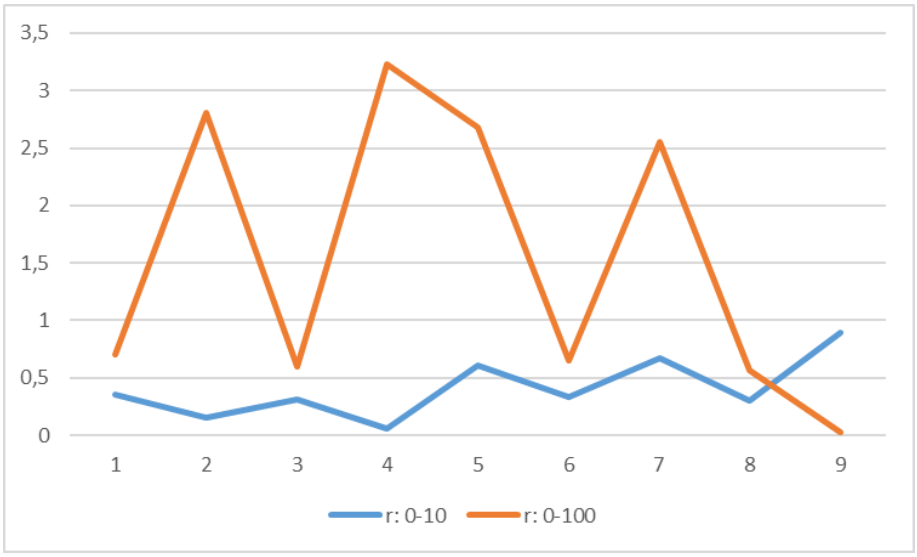
Tablo 3.30. üzerinde görülmektedir ki Z Normalleştirme 0-10 aralığında da olsa 0-100 aralığında da olsa 1 ve 3 numaralı deneylerin benzerlik göstergesi olan korelasyon, 1 ve 26 numaralı deneylerin benzerliğini gösteren korelasyona kıyasla daha yüksektir. Bu da CPU metriğinin A'nın var olması durumunda olmamasına kıyasla daha karakteristik davrandığını göstermektedir. Bu göstergeyi tüm kaynaklar ve zafiyetler için kıyaslayarak detaylı analiz sağlanmıştır.

Korelasyon Eşleştirmelerinin Z-Normalleştirmeye Göre Yapılması

Şekil 3.55 örneği göstermektedir ki korelasyonlar kendi içinde tutarlı olsa da başta seçilen aralıklandırım hep uygulanmalıdır. 0-10 aralıklandırımına göre zafiyet profili oluşturulursa ve anlık olarak gelen veri ambarı verileri 0-100 aralığına göre işlenirse yanlış olacağı ön görülen sonuçların alınması muhtemeldir. Bu durum tüm kaynaklara göre analiz edilip, elde edilen sonuçlar paylaşılacaktır. Temel olarak yükselme ve düşüşlerin göz ardı edilebilecek düzeyde farklılık göstermesi durumunda 0-10 ve 0-100 dönüşümleri farklılık göstermeyecektir. CPU için örnek korelasyon sonuçları Tablo 3.31. üzerinde görülmektedir. Diğer kaynaklar içinde aynı sistemler Tablo 3.31. üzerinde görülen sonuçlar çıkarılıp analiz edilecektir.

Tablo 3.31. CPU Metrik Değerleri Kullanılarak, Deneylerin Kıyaslanması Durumunda 0-10 ve 0-100 Z-Normalleştirme Değerlerinin Korelasyonlarının Sonuçları

Kıyaslama No	Deney No	Deney No	r: 0-10	r: 0-100
1	1	26	0,354101	0,704119
2	2	27	0,157445	2,809548
3	3	28	0,316189	0,599853
4	4	29	0,057965	3,233955
5	5	30	0,612509	2,675887
6	6	31	0,336526	0,654583
7	7	32	0,676433	2,553643
8	8	33	0,302334	0,561651
9	9	34	0,894662	0,021783



Şekil 3.55. CPU Metriklerine Göre Kıyaslama Numarasıyla 0-10 ve 0-100 Aralığındaki Z-Normalleştirme Korelasyonlarının Frekansları

Korelasyon Matrisinin Elde Edilmesi

Her bir deneyin, tüm kaynaklar üzerinden 0-10 ve 1-100 aralığındaki dönüşüm değerleriyle, diğer deneylerden elde edilen dönüşümlerin değerlerinin korelasyonu kıyaslanarak 50-50'lik matris elde edilecektir. Bu matrise göre örnek olarak 1 ile 20 numaralı deneylerin ne düzeyde benzeştiği her metrik için 0-10 ve 0-100 aralığında dönüştürülerek korelasyon hesaplaması yapılıp görülebilecektir. Genel olarak tüm deneylerin CPU, RAM, NETI ve NETO metriklerinin 0-10 ve 0-100 aralığındaki korelasyonlarını içeren 8 adet korelasyon matrisi elde edilmiştir. Tablo 3.32 ve Tablo 3.33. bu korelasyon matrislerinin örnekleridir.

Tablo 3.32. CPU Metriğinin 0-10 Aralığındaki Değerlerinin Her Deney İçin Korelasyon Matrisi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	0	75	14	73	1	1	33	28	43	63	69	76	40	4	6	7	4	66	21	59	65	6	1	8	78	4	67	9	51	63	2	73	42	81	54	76	45	45	66	1	0	36	53	13	44	27	3	67	8	4
2	75	0	69	6	78	74	63	73	57	26	14	9	46	78	78	80	80	12	73	35	24	80	77	77	11	72	2	70	42	22	74	13	31	33	36	3	41	46	27	77	74	64	34	75	55	62	79	20	81	80
3	14	69	0	66	16	13	19	11	30	54	61	70	28	12	10	22	19	58	5	49	57	21	14	8	73	9	60	4	40	54	12	67	33	77	44	70	35	34	57	14	14	21	43	3	31	13	18	59	22	19
4	73	6	66	0	76	72	59	70	53	21	8	14	43	76	76	79	78	7	71	30	18	79	75	75	16	70	6	67	38	17	72	17	28	37	32	9	38	43	21	76	72	61	30	73	51	59	77	14	79	78
5	1	78	16	76	0	2	36	31	46	66	72	79	42	6	8	7	3	68	24	62	69	5	2	10	81	6	69	11	54	66	3	75	44	83	57	79	48	48	69	2	1	39	56	15	47	29	2	70	8	3
6	1	74	13	72	2	0	31	26	41	62	68	75	38	3	5	8	5	65	19	58	64	7	0	7	77	3	66	8	49	61	1	72	41	80	53	75	44	44	64	0	1	34	52	12	42	25	4	66	9	5
7	33	63	19	59	36	31	0	11	13	44	53	65	14	33	31	41	39	50	17	37	47	41	34	29	68	28	53	22	26	44	30	61	21	73	31	64	21	19	47	34	32	2	31	24	14	4	38	50	42	39
8	28	73	11	70	31	26	11	0	25	56	64	75	24	28	26	38	35	59	8	49	59	37	29	23	77	22	61	16	38	55	25	69	29	81	43	74	31	30	60	29	27	14	42	17	27	5	34	61	39	35
9	43	57	30	53	46	41	13	25	0	35	46	60	3	45	43	52	50	42	30	27	39	51	45	41	63	38	47	33	15	36	41	56	13	70	21	59	11	8	38	45	42	11	21	37	2	16	49	42	52	50
10	63	26	54	21	66	62	44	56	35	0	12	32	27	66	65	70	69	11	58	9	3	69	65	64	35	59	21	56	19	3	61	32	13	49	13	29	21	25	1	65	62	45	12	61	33	45	68	7	70	69
11	69	14	61	8	72	68	53	64	46	12	0	21	37	72	71	75	74	0	65	22	10	74	71	70	23	66	12	63	30	9	67	23	22	41	24	17	31	36	12	71	68	54	23	68	44	53	73	6	75	74
12	76	9	70	14	79	75	65	75	60	32	21	0	50	79	79	81	81	19	75	40	30	81	78	79	1	74	5	71	47	28	75	5	35	25	41	6	46	50	33	79	75	67	40	77	58	64	80	27	81	81
13	40	46	28	43	42	38	14	24	3	27	37	50	0	40	39	46	45	35	28	20	30	46	41	37	52	36	40	31	10	28	38	49	9	61	15	48	7	4	29	41	39	13	16	33	2	17	44	33	47	45
14	4	78	12	76	6	3	33	28	45	66	72	79	40	0	2	13	9	68	20	62	68	12	3	5	82	1	68	7	52	65	2	75	42	84	56	79	46	46	69	3	4	37	55	10	45	26	8	70	14	9
15	6	78	10	76	8	5	31	26	43	65	71	79	39	2	0	15	12	67	18	61	68	14	6	2	82	1	68	5	51	64	4	75	41	84	55	79	45	45	69	6	6	35	54	8	44	24	10	69	16	12
16	7	80	22	79	7	8	41	38	52	70	75	81	46	13	15	0	3	71	31	66	72	1	9	17	83	11	71	17	58	69	9	77	47	85	61	81	52	52	73	9	7	45	60	22	52	35	4	73	1	3
17	4	80	19	78	3	5	39	35	50	69	74	81	45	9	12	3	0	70	28	65	71	2	5	14	83	9	70	14	56	68	6	76	46	85	59	81	50	51	72	5	4	43	58	19	50	32	1	72	5	0
18	66	12	58	7	68	65	50	59	42	11	0	19	35	68	67	71	70	0	61	20	9	71	67	67	21	63	11	60	28	8	64	21	21	38	23	15	29	33	11	68	65	50	21	64	40	50	70	5	72	70
19	21	73	5	71	24	19	17	8	30	58	65	75	28	20	18	31	28	61	0	52	61	30	22	15	78	15	62	10	41	57	19	70	32	81	45	74	35	34	61	22	20	20	45	9	31	11	27	63	32	28
20	59	35	49	30	62	58	37	49	27	9	22	40	20	62	61	66	65	20	52	0	13	66	61	60	43	55	29	51	11	12	57	39	7	56	4	37	13	17	11	61	58	37	3	56	25	38	64	16	66	65
21	65	24	57	18	69	64	47	59	39	3	10	30	30	68	68	72	71	9	61	13	0	72	68	67	33	62	19	59	22	0	64	31	16	49	16	27	24	28	2	68	64	48	15	64	36	47	70	4	72	71
22	6	80	21	79	5	7	41	37	51	69	74	81	46	12	14	1	2	71	30	66	72	0	8	16	83	10	71	16	57	68	8	77	47	85	60	81	51	52	72	8	6	45	59	21	51	34	3	73	2	2
23	1	77	14	75	2	0	34	29	45	65	71	78	41	3	6	9	5	67	22	61	68	8	0	8	81	4	68	9	52	65	1	74	43	83	56	78	46	46	68	0	1	37	55	13	45	27	4	69	10	5
24	8	77	8	75	10	7	29	23	41	64	70	79	37	5	2	17	14	67	15	60	67	16	8	0	81	3	67	3	50	63	6	74	40	83	54	78	44	43	68	8	8	33	53	6	42	22	12	68	18	14
25	78	11	73	16	81	77	68	77	63	35	23	1	52	82	82	83	83	21	78	43	33	83	81	81	0	76	6	73	49	30	77	5	37	25	44	7	48	53	36	81	77	70	42	79	61	66	82	29	84	83
26	4	72	9	70	6	3	28	22	38	59	66	74	36	1	1	11	9	63	15	55	62	10	4	3	76	0	64	5	46	59	3	70	39	79	50	73	41	41	62	4	4	30	49	8	39	22	8	63	12	9
27	67	2	60	6	69	66	53	61	47	21	12	5	40	68	68	71	70	11	62	29	19	71	68	67	6	64	0	62	35	19	65	9	28	23	31	1	36	39	21	68	66	53	29	65	45	53	70	16	71	70
28	9	70	4	67	11	8	22	16	33	56	63	71	31	7	5	17	14	60	10	51	59	16	9	3	73	5	62	0	42	56	8	68	35	77	46	71	37	37	59	9	9	25	46	2	34	17	13	60	18	14
29	51	42	40	38	54	49	26	38	15	19	30	47	10	52	51	58	56	28	41	11	22	57	52	50	49	46	35	42	0	21	49	45	1	60	6	44	3	7	21	53	50	26	7	46	13	28	55	26	58	56
30	63	22	54	17	66	61	44	55	36	3	9	28	28	65	64	69	68	8	57	12	0	68	65	63	30	59	19	56	21	0	61	29	15	46	15	25	22	27	2	65	62	45	14	61	34	45	67	4	69	68
31	2	74	12	72	3	1	30	25	41	61	67	75	38	2	4	9	6	64	19	57	64	8	1	6	77	3	65	8	49	61	0	72	41	80	52	75	43	43	64	1	2	33	51	11	41	25	5	65	10	6
32	73	13	67	17	75	72	61	69	56	32	23	5	49	75	75	77	76	21	70	39	31	77	74	74	5	70	9	68	45	29	72	0	36	16	40	10	45	49	33	74	72	62	39	72	55	61	76	27	77	76
3	42	31	33	28	44	41	21	29	13	13	22	35	9	42	41	47	46	21	32	7	16	47	43	40	37	39	28	35	1	15	41	36	0	48	3	33	4	6	15	43	42	21	4	37	12	24	45	18	48	46
34	81	33	77	37	83	80	73	81	70	49	41	25	61	84	84	85	85	38	81	56	49	85	83	83	25	79	23	77	60	46	80	16	48	0	56	30	58	62	52	83	80	75	54	82	69	72	84	46	85	85
35	54	36	44	32	57	53	31	43	21	13	24	41	15	56	55	61	59	23	45	4	16	60	56	54	44	50	31	46	6	15	52	40	3	56	0	39	9	12	15	56	53	31	1	50	19	33	59	19	61	59
36	76	3	70	9	79	75	64	74	59	29	17	6	48	79	79	81	81	15	74	37	27	81	78	78	7	73	1	71	44	25	75	10	33	30	39	0	43	48	30	78	75	66	37	77	57	63	80	23	81	81
37	45	41	35	38	48	44	21	31	11	21	31	46	7	46	45	52	50	29	35	13	24	51	46	44	48	41	36	37	3	22	43	45	4	58	9	43	0	3	23	47	45	20	9	40	9	23	49	27	52	50
38	45	46	34	43	48	44	19	30	8	25	36	50	4	46	45	52	51	33	34	17	28	52	46	43	53	41	39	37	7	27	43	49	6	62	12	48	3	0	27	47	44	18	13	40	6	22	50	31	53	51
39	66	27	57	21	69	64	47	60	38	1	12	33	29	69	69	73	72	11	61	11	2	72	68	68	36	62	21	59	21	2	64	33	15	52	15	30	23	27	0	68	65	48	13	65	36	47	71	6	73	72
40	1	77	14	76	2	0	34	29	45	65	71	79	41	3	6	9	5	68	22	61	68	8	0	8	81	4	68	9	53	65	1	74	43	83	56	78	47	47	68	0	1	38	55	13	46	27	4	69	10	5
41	0	74	14	72	1	1	32	27	42	62	68	75	39	4	6	7	4	65	20	58	64	6	1	8	77	4	66	9	50	62	2	72	42	80	53	75	45	44	65	1	0	35	52	13	43	26	3	66	8	4
42	36	64	21	61	39	34	2	14	11	45	54	67	13	37	35	45	43	50	20	37	48	45	37	33	70	30	53	25	26	45	33	62	21	75	31	66	20	18	48	38	35	0	31	28	13	7	42	51	46	43
43	53	34	43	30	56	52	31	42	21	12	23	40	16	55	54	60	58	21	45	3	15	59	55	53	42	49	29	46	7	14	51	39	4	54	1	37	9	13	13	55	52	31	0	49	19	33	58	18	60	58
44	13	75	3	73	15	12	24	17	37	61	68	77	33	10	8	22	19	64	9	56	64	21	13	6	79	8	65	2	46	61	11	72	37	82	50	77	40	40	65	13	13	28	49	0	38	18	18	66	23	19
45	44	55	31	51	47	42	14	27	2	33	44	58	2	45	44	52	50	40	31	25	36	51	45	42	61	39	45	34	13	34	41	55	12	69	19	57	9	6	36	46	43	13	19	38	0	18	49	39	53	50
46	27	62	13	59	29	25	4	5	16	45	53	64	17	26	24	35	32	50	11	38	47	34	27	22	66	22	53	17	28	45	25	61	24	72	33	63	23	22	47	27	26	7	33	18	18	0	31	50	35	32
47	3	79	18	77	2	4	38	34	49	68	73	80	44	8	10	4	1	70	27	64	70	3	4	12	82	8	70	13	55	67	5	76	45	84	59	80	49	50	71	4	3	42	58	18	49	31	0	71	6	1
48	67	20	59	14	70	66	50	61	42	7	6	27	33	70	69	73	72	5	63	16	4	73	69	68	29	63	16	60	26	4	65	27	18	46	19	23	27	31	6	69	66	51	18	66	39	50	71	0	73	72
49	8	81	22	79	8	9	42	39	52	70	75	81	47	14	16	1	5	72	32	66	72	2	10	18	84	12	71	18	58	69	10	77	48	85	61	81	52	53	73	10	8	46	60	23	53	35	6	73	0	5
50	4	80	19	78	3	5	39	35	50	69	74	81	45	9	12	3	0	70	28	65	71	2	5	14	83	9	70	14	56	68	6	76	46	85	59	81	50	51	72	5	4	43	58	19	50	32	1	72	5	0

Tablo 3.33. NetI Metriğinin 0-100 Aralığındaki Değerlerinin Her Deney İçin Korelasyon Matrisi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	0	35	20	1	2	4	10	6	10	13	12	21	9	1	7	10	8	40	9	13	28	7	5	6	2	1	30	4	6	31	10	2	71	29	21	36	18	21	41	9	5	57	19	1	4	29	23	10	44	45
2	35	0	14	34	36	41	45	40	46	49	23	13	25	35	42	46	45	8	44	22	7	43	30	39	34	34	4	32	28	3	42	33	56	5	13	3	19	15	11	46	39	34	17	36	30	57	54	43	13	14
3	20	14	0	20	22	25	30	26	30	33	8	1	11	20	27	30	29	21	29	7	7	28	16	25	19	19	10	17	14	14	28	18	60	9	1	16	3	0	23	30	25	42	2	20	16	44	40	29	25	26
4	1	34	20	0	1	4	9	5	9	12	12	21	9	2	6	9	7	40	8	13	28	6	5	5	2	2	30	5	7	31	9	3	70	29	21	35	18	21	41	8	4	56	19	2	4	27	21	9	44	44
5	2	36	22	1	0	2	7	4	7	11	14	23	11	3	5	7	5	41	7	14	29	5	7	4	4	4	31	6	8	32	8	4	71	30	23	37	20	23	42	7	3	57	21	3	6	26	20	8	45	46
6	4	41	25	4	2	0	6	2	6	10	17	27	14	6	3	6	3	46	5	17	34	3	9	2	6	6	35	9	11	35	7	7	76	34	26	41	24	27	47	5	1	62	24	6	8	26	20	6	50	50
7	10	45	30	9	7	6	0	3	1	3	21	31	18	11	3	1	3	49	1	22	38	3	14	3	12	12	39	14	16	39	2	12	77	38	30	45	29	31	50	1	4	64	29	11	13	21	14	1	53	53
8	6	40	26	5	4	2	3	0	3	6	18	27	15	8	1	3	1	45	2	18	33	0	11	0	8	8	35	10	12	35	4	9	74	34	27	41	24	27	46	2	1	60	25	8	10	23	16	4	49	49
9	10	46	30	9	7	6	1	3	0	4	22	32	19	11	2	0	2	51	0	23	39	2	15	3	12	12	40	14	16	39	2	12	79	39	31	46	30	32	51	1	4	65	30	11	13	23	15	1	55	55
10	13	49	33	12	11	10	3	6	4	0	25	35	22	15	6	4	7	53	4	26	42	7	18	7	16	15	43	18	20	42	1	16	80	42	34	49	33	36	54	5	8	67	33	15	16	19	11	2	57	57
11	12	23	8	12	14	17	21	18	22	25	0	9	2	11	19	22	20	29	21	1	16	19	7	17	11	11	18	8	6	21	21	10	65	17	9	24	6	9	31	21	17	48	7	11	8	38	33	21	33	34
12	21	13	1	21	23	27	31	27	32	35	9	0	12	21	29	32	30	21	30	8	6	29	17	26	20	20	9	18	15	13	30	19	61	8	0	15	4	1	23	32	26	42	3	21	17	46	42	30	25	26
13	9	25	11	9	11	14	18	15	19	22	2	12	0	9	16	19	17	31	18	3	18	16	5	14	8	8	20	5	3	23	18	7	65	19	12	26	8	11	32	18	14	49	9	8	5	35	30	18	35	35
14	1	35	20	2	3	6	11	8	11	15	11	21	9	0	9	11	9	41	10	12	28	8	4	7	1	1	30	3	6	31	12	1	73	29	21	36	18	21	42	11	6	58	19	0	3	30	25	11	45	46
15	7	42	27	6	5	3	3	1	2	6	19	29	16	9	0	2	0	47	2	20	35	0	12	1	9	9	37	12	13	37	4	10	75	36	28	42	26	29	48	1	2	62	26	9	11	23	16	3	51	51
16	10	46	30	9	7	6	1	3	0	4	22	32	19	11	2	0	2	51	0	23	39	2	15	3	12	12	41	14	16	40	2	12	79	39	31	46	30	33	52	1	4	66	30	12	13	23	15	1	55	55
17	8	45	29	7	5	3	3	1	2	7	20	30	17	9	0	2	0	50	2	21	37	0	13	1	10	9	39	12	14	38	4	10	78	38	30	45	28	31	50	2	2	65	28	9	11	25	18	3	54	54
18	40	8	21	40	41	46	49	45	51	53	29	21	31	41	47	51	50	0	49	28	15	48	36	44	40	40	12	38	34	4	46	39	49	13	20	5	26	23	3	51	45	26	24	41	35	60	58	47	4	6
19	9	44	29	8	7	5	1	2	0	4	21	30	18	10	2	0	2	49	0	21	37	2	14	3	11	11	39	13	15	38	2	11	77	38	30	44	28	31	50	0	3	63	28	11	12	22	15	2	53	53
20	13	22	7	13	14	17	22	18	23	26	1	8	3	12	20	23	21	28	21	0	15	20	8	18	11	11	17	9	6	20	22	11	64	16	8	23	5	8	30	22	18	47	6	12	9	38	34	22	32	33
21	28	7	7	28	29	34	38	33	39	42	16	6	18	28	35	39	37	15	37	15	0	36	23	33	27	27	3	25	21	8	36	26	59	2	6	10	11	8	17	39	33	38	9	28	23	51	48	36	20	21
22	7	43	28	6	5	3	3	0	2	7	19	29	16	8	0	2	0	48	2	20	36	0	12	1	9	9	37	12	13	37	4	10	76	36	28	43	27	29	49	2	2	63	27	9	11	24	17	3	52	52
23	5	30	16	5	7	9	14	11	15	18	7	17	5	4	12	15	13	36	14	8	23	12	0	10	3	3	26	1	2	27	15	2	69	24	17	31	14	16	38	14	10	54	14	4	1	32	27	14	40	41
24	6	39	25	5	4	2	3	0	3	7	17	26	14	7	1	3	1	44	3	18	33	1	10	0	8	7	34	10	12	35	5	8	73	33	26	40	24	26	45	2	1	59	24	7	9	23	16	4	48	48
25	2	34	19	2	4	6	12	8	12	16	11	20	8	1	9	12	10	40	11	11	27	9	3	8	0	0	29	3	5	30	12	1	72	28	20	35	17	20	41	11	7	57	18	1	2	31	25	12	44	45
26	1	34	19	2	4	6	12	8	12	15	11	20	8	1	9	12	9	40	11	11	27	9	3	7	0	0	29	3	5	30	12	1	72	28	20	35	17	20	41	11	7	57	18	0	2	30	25	12	44	44
27	30	4	10	30	31	35	39	35	40	43	18	9	20	30	37	41	39	12	39	17	3	37	26	34	29	29	0	27	24	6	37	28	56	1	9	7	14	11	14	40	34	36	12	30	25	52	49	38	16	17
28	4	32	17	5	6	9	14	10	14	18	8	18	5	3	12	14	12	38	13	9	25	12	1	10	3	3	27	0	2	28	14	2	71	26	18	33	15	18	39	14	9	55	15	3	0	32	27	14	42	42
29	6	28	14	7	8	11	16	12	16	20	6	15	3	6	13	16	14	34	15	6	21	13	2	12	5	5	24	2	0	26	16	4	68	22	15	29	12	14	36	15	11	52	13	5	2	33	28	16	38	39
30	31	3	14	31	32	35	39	35	39	42	21	13	23	31	37	40	38	4	38	20	8	37	27	35	30	30	6	28	26	0	37	29	44	7	13	0	17	15	7	39	35	26	16	31	27	50	47	38	8	9
31	10	42	28	9	8	7	2	4	2	1	21	30	18	12	4	2	4	46	2	22	36	4	15	5	12	12	37	14	16	37	0	12	73	36	29	42	27	30	47	3	5	60	28	12	13	17	11	1	50	50
32	2	33	18	3	4	7	12	9	12	16	10	19	7	1	10	12	10	39	11	11	26	10	2	8	1	1	28	2	4	29	12	0	71	27	19	34	16	19	40	12	8	56	17	1	2	31	25	12	43	43
33	71	56	60	70	71	76	77	74	79	80	65	61	65	73	75	79	78	49	77	64	59	76	69	73	72	72	56	71	68	44	73	71	0	56	59	52	65	63	45	79	73	24	63	73	67	81	81	75	47	45
34	29	5	9	29	30	34	38	34	39	42	17	8	19	29	36	39	38	13	38	16	2	36	24	33	28	28	1	26	22	7	36	27	56	0	8	8	12	9	15	39	33	36	11	29	24	51	48	37	17	18
35	21	13	1	21	23	26	30	27	31	34	9	0	12	21	28	31	30	20	30	8	6	28	17	26	20	20	9	18	15	13	29	19	59	8	0	15	4	1	22	31	26	41	3	21	17	45	41	30	24	25
36	36	3	16	35	37	41	45	41	46	49	24	15	26	36	42	46	45	5	44	23	10	43	31	40	35	35	7	33	29	0	42	34	52	8	15	0	20	17	8	46	40	30	18	36	31	56	54	43	10	11
37	18	19	3	18	20	24	29	24	30	33	6	4	8	18	26	30	28	26	28	5	11	27	14	24	17	17	14	15	12	17	27	16	65	12	4	20	0	3	28	29	24	47	1	18	14	45	41	28	30	31
38	21	15	0	21	23	27	31	27	32	36	9	1	11	21	29	33	31	23	31	8	8	29	16	26	20	20	11	18	14	15	30	19	63	9	1	17	3	0	25	32	26	45	2	21	16	47	43	30	27	28
39	41	11	23	41	42	47	50	46	51	54	31	23	32	42	48	52	50	3	50	30	17	49	38	45	41	41	14	39	36	7	47	40	45	15	22	8	28	25	0	51	45	23	26	42	37	60	58	48	1	3
40	9	46	30	8	7	5	1	2	1	5	21	32	18	11	1	1	2	51	0	22	39	2	14	2	11	11	40	14	15	39	3	12	79	39	31	46	29	32	51	0	3	65	29	11	12	23	16	2	55	55
41	5	39	25	4	3	1	4	1	4	8	17	26	14	6	2	4	2	45	3	18	33	2	10	1	7	7	34	9	11	35	5	8	73	33	26	40	24	26	45	3	0	60	24	7	9	24	17	5	48	49
42	57	34	42	56	57	62	64	60	65	67	48	42	49	58	62	66	65	26	63	47	38	63	54	59	57	57	36	55	52	26	60	56	24	36	41	30	47	45	23	65	60	0	45	58	53	71	70	62	23	21
43	19	17	2	19	21	24	29	25	30	33	7	3	9	19	26	30	28	24	28	6	9	27	14	24	18	18	12	15	13	16	28	17	63	11	3	18	1	2	26	29	24	45	0	19	15	44	40	28	28	29
44	1	36	20	2	3	6	11	8	11	15	11	21	8	0	9	12	9	41	11	12	28	9	4	7	1	0	30	3	5	31	12	1	73	29	21	36	18	21	42	11	7	58	19	0	3	31	25	11	45	46
45	4	30	16	4	6	8	13	10	13	16	8	17	5	3	11	13	11	35	12	9	23	11	1	9	2	2	25	0	2	27	13	2	67	24	17	31	14	16	37	12	9	53	15	3	0	30	25	13	39	40
46	29	57	44	27	26	26	21	23	23	19	38	46	35	30	23	23	25	60	22	38	51	24	32	23	31	30	52	32	33	50	17	31	81	51	45	56	45	47	60	23	24	71	44	31	30	0	8	19	63	63
47	23	54	40	21	20	20	14	16	15	11	33	42	30	25	16	15	18	58	15	34	48	17	27	16	25	25	49	27	28	47	11	25	81	48	41	54	41	43	58	16	17	70	40	25	25	8	0	12	61	62
48	10	43	29	9	8	6	1	4	1	2	21	30	18	11	3	1	3	47	2	22	36	3	14	4	12	12	38	14	16	38	1	12	75	37	30	43	28	30	48	2	5	62	28	11	13	19	12	0	51	51
49	44	13	25	44	45	50	53	49	55	57	33	25	35	45	51	55	54	4	53	32	20	52	40	48	44	44	16	42	38	8	50	43	47	17	24	10	30	27	1	55	48	23	28	45	39	63	61	51	0	1
50	45	14	26	44	46	50	53	49	55	57	34	26	35	46	51	55	54	6	53	33	21	52	41	48	45	44	17	42	39	9	50	43	45	18	25	11	31	28	3	55	49	21	29	46	40	63	62	51	1	0

Korelasyon Matrisindeki Eşleştirme Sıralamalarının En Karakteristik Olanlarının Alınması

Deneylerin arasındaki korelasyonun eşleştirildiği matris değerlerinin sıralama işlemi gerçekleştirilmiştir. Bu işleme göre 2 deney arasındaki korelasyon değeri en yüksekten düşüğe doğru sıralanmıştır. Bu durum Z-Normalleştirme işleminin 0-10 ve 0-100 aralığında gerçekleştiği tüm kaynak verilerine ayrı olarak uygulanarak sıralanmıştır.

Yapılan işlemin RAM için 0-10 Z-Normalleştirmesi ve NetO için 0-100 normalleşmesi durumunda deneylerin birbiriyle olan korelasyonlarının en yüksekten düşüğe sıralandığı örnek Tablo 3.34 üzerinde görülmektedir. Eğer korelasyon değeri yüksekliği 2 deney arasındaki benzerliğin ne düzeyde az olduğunun göstergesidir.

Tablo 3.34 üzerindeki örnekte anlaşılmaktadır ki RAM metrikleri 0-10 aralığına dönüştürüldüğünde en birbiriyle benzerliği olmayan 2 deney 5-42 numaralı deneylerdir. Ardından 5-36, 5-39, 9-42, 9-36, 9-39 takip etmektedir. Buradan görülmektedir ki 5, 9, 36 ve 39 numaralı deneyler ciddi anlamda RAM için karakteristik bir yapıdadır.

Önerilen modelin Şekil 3.17 üzerinde anlatıldığı gibi her karakteristiğin belirleyicisi olan kaynak seçimi zafiyet tipine göre değişken olmalıdır durumunun geçerliliği görülmektedir. Aynı durumda Tablo 3.34 üzerindeki 0-100 aralığı için NetO incelendiğinde 4, 33, 42 ve 36 numaralı deneylerin sürekli ilk sıralarda bulunmasından dolayı karakteristiğinin NetO üzerinden elde edilmesinin mantıklı olacağı sonucu çıkarılıyor.

Tablo 3.34 üzerindeki örnekler için NetO ve RAM metrikleri 36 numaralı deneyin tekrar gerçekleşmesi durumunda tespit etmek için ideal durmaktadır. Genel olarak Tablo 3.34 üzerindeki yapıların tüm Z-Normalleştirme aralıklarının her kaynağa yönelik analiz edilmesi gerekmektedir. Ardından genel olarak diğer deneylerle farklılaşması oranında deneyin karakteristiğine dair puanlama yapılmalıdır. Hangi kaynağın hangi zafiyetin tespiti için uygun olduğuna yönelik süreç yaklaşımı bu şekilde olacaktır.

Tablo 3.34: Deneylerin RAM ve NETO Metriklerin Z-Normalleştirme Sonucunda Korelasyon Matris Değerlerinin Büyükten Küçüğe Sıralamanmasına Yönelik Örnek

RAM Metriğinin 0-10 Aralığında Dönüşümü Sonucu Korelasyonlar			NETO Metriğinin 0-100 Aralığında Dönüşümü Sonucu Korelasyonlar		
1. Deney No	2. Deney No	Korelasyon	1. Deney No	2. Deney No	Korelasyon
5	42	77	4	33	75
5	36	76	33	36	75
5	39	75	4	42	72
9	42	74	36	42	72
9	36	72	27	33	68
9	39	72	33	38	66
38	42	72	19	33	64
42	50	72	27	42	64
14	42	71	33	45	64
36	38	71	3	36	63
36	50	71	6	36	63
3	5	70	33	43	63
38	39	70	2	33	62
39	50	70	7	33	62
14	36	69	20	33	62
14	39	69	38	42	62
20	42	69	3	4	61
34	42	68	4	6	61
2	42	67	9	36	61
20	36	67	11	33	61

3.3.6. Nokta Çift Serili Korelasyon İşleminden Alınan Çıktıların İşlenmesi

En Karakteristik Deneylerin DKP Puanlama Yapılarak Tespiti

Tablo 3.35 üzerinde deneylerin kaynak metriklerine Z-Normalleştirmesinin ardından elde edilecek tablonun örneği görülmektedir. Bu tablo üzerinde deneyin karakteristiğini analiz edebilmek için puanlama işlemine ihtiyaç duyulmaktadır.

Yapılan her deneyin diğer 49 deney ile korelasyonun matris yapısı tablo 3.33 ve tablo 3.32 örneklerinde görülmektedir. Korelasyon matrisleri x ve y deneyi için tablo 3.35 üzerindeki gibi her kaynağın 0-10 ve 0-100 Z-Normalleştirme değerleri için alınacaktır ve sıralanacaktır. Bu aşamanın ardından x ve y yada y ve x şeklinde tekrarlı yapıların çıkartılmasının ardından toplamda 1225 tane korelasyon değeri her metriğin normalleştirme aralığı için bulunmaktadır. Tablo 3.35 üzerindeki gibi sıralama işleminin ardından deneylerin puanlaması yapılmaktadır.

Tablo 3.35: 2 Deneyin Z-Normalleştirmeye Göre Korelasyon Matrisine Göre Sıralama Tablosu Örneği

1. Deney No	2. Deney No	Z Index'e Göre Korelasyon
33	36	76
4	33	75
36	42	73
4	42	71
27	33	68
33	38	67
33	45	66
3	36	64
6	36	64
7	33	64

Bu puanlama işleminin amacı deneyin karakteristiğini tespit etmek için yapılacaktır. Deney ne kadar üst sıralarda bulunuyorsa o kadar karakteristik niteliği yüksek olduğu için en korelasyon sıralamasında üstte bulunmasıyla orantılı olarak puanı yüksek olacaktır.

$$\mathrm{DKP} = (25 \times \mathrm{DKL10BS}) + (10 \times \mathrm{DKL50BS}) + (5 \times \mathrm{DKL100BS}) + (1 \times \mathrm{DKL250BS}) \qquad (3.9)$$

Bu puanlama işlemi için tablo 3.36 örneği üzerinde görüldüğü gibi deneyin ilk 10'da, ilk 50'de, ilk 100'de ve ilk 250 bulunma sayısı kadar puanı orantısal olarak yükselmektedir. Bu puan denklem 3.9 ile hesaplanmaktadır.

Burada DKP hesaplanırken bir deneyin korelasyon sıralamasında bulunma durumuna göre en değerli olanların yani en üstte olanların yüksek ağırlığa sahip olması sağlanmıştır. Bu noktada DKP için kullanılan 4 sıralama tipinden ilk 10'da bulunma ve ilk 250'de bulunma aynı ağırlığa sahip olarak görülmesinden dolayı '25 x DKL10BS' ve '1 x DKL250BS' şeklinde ağırlıklandırıldı. Orta düzeyde bulunma ile "en tepede bulunma(ilk 10 için)" ve "ilk 250'de bulunma" durumu kıyaslandığında ilk 50 ve ilk 100 daha değerli olarak görüldüğü için tamamında bulunma durumu kıyaslandığında 2 katı olan 500 değerini elde edecek şekilde '10 x DKL50BS' ve '5 x DKL100BS' ağırlıklandırılmıştır. DKP puanına göre belirli senaryolar ele alınırsa:

a. İlk 250'da bulunma durumunda DKP: 1500
b. Sadece ilk 10'da bulunma durumunda DKP: 410
c. Sadece son 240'de bulunma durumunda DKP: 1090
d. Sadece ilk 50'de bulunma durumunda DKP: 1050
e. Sadece son 200'de bulunma durumunda DKP: 450
f. Sadece ilk 100'de bulunma durumunda DKP: 1350
g. Sadece son 150'da bulunma durumunda DKP: 150

Bu noktada ortalamanın ağırlıklandırmada orta düzeyde bulunanları göz önünde bulundurarak yüksek sıradakilere ağırlık verdiğini c>b ile d>e ve f>g kıyaslamalarına bakılarak anlaşılabilmektedir. Bununla beraber ortalarda bulunma oranı ilk 10 ve tüm Ardından ise puana göre deneyler Tablo 3.37 örneği üzerinde görüldüğü gibi sıralanmaktadır.

Tablo 3.37 için "33,42,36 ve 4 numaralı deneyin benzeri bir örüntünün tespiti için kullanılan Z-Normalleştirme ve kaynak metriği ciddi anlamda idealdir." anlamı çıkarılmaktadır.

Tablo 3.36: Korelasyon Matris Değerlerinin Bulunduğu Sıraya Göre Deney Numarası için Oluşan DKP Puanlama Tablo Örneği

Deney Numarası	İlk 10 için	İlk 50 için	İlk 100 için	İlk 250 İçin	DKP Puanlama
1	0	1	2	4	24
2	0	2	2	8	38
3	1	2	4	21	86
4	2	5	12	30	190
5	0	0	2	7	17
6	1	2	4	22	87
7	1	2	2	7	62
8	0	0	1	5	10
9	0	2	2	20	50
10	0	1	2	4	24
11	0	1	2	6	26
12	0	1	2	7	27
13	0	0	2	5	15
14	0	0	2	4	14
15	0	0	2	4	14
16	0	0	0	4	4
17	0	0	2	7	17
18	0	1	2	13	33
19	0	2	2	5	35
20	0	2	2	7	37
21	0	1	2	4	24
22	0	0	1	4	9
23	0	1	3	4	29
24	0	0	1	4	9
25	0	2	2	6	36
26	0	0	2	5	15
27	1	2	4	16	81
28	0	0	3	4	19
29	0	2	2	7	37
30	0	2	2	7	37
31	0	1	3	13	38
32	0	1	2	5	25
33	6	26	38	48	648
34	0	1	2	9	29
35	0	2	2	6	36
36	4	9	22	37	337
37	0	1	3	4	29
38	1	2	4	13	78
39	0	1	2	6	26
40	0	0	2	4	14
41	0	0	2	6	16
42	2	14	28	45	375
43	0	2	2	8	38
44	0	1	2	6	26
45	1	2	2	12	67
46	0	1	2	5	25
47	0	0	2	4	14
48	0	1	2	4	24
49	0	1	3	7	32
50	0	0	2	7	17

Tablo 3.37: Deneylerin DKP Puanına Göre Sıralandığı Tablo Örneği

dwva Zafiyet Zorluk	Normal Kullanıcı Yükü	Zafiyet Tarama Oranı	Tip	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı	Deney No	DKP Puanlama
Çok Zor	240	60	A	48	12	33	648
Çok Zor	240	60	B	12	48	42	375
Çok Zor	0	300	B	0	300	36	337
Kolay	0	300	A	240	60	4	190
Kolay	240	60	A	60	0	6	87
Kolay	0	300	A	0	300	3	86
Çok Zor	0	300	A	300	0	27	81
Çok Zor	0	300	B	60	240	38	78
Çok Zor	0	300	A+B	240	60	45	67
Kolay	240	60	A	0	60	7	62

Deney Karakteristiğine Göre DKP Puanlamalarının Sıralanması

Metriklerin zafiyet tipinin, zorluğunun, z-normalleştirme tipinin, yük tipinin ve kaynak metrik tipinin karakteristiği anlamak için her 3.37 tablosunun her deneyin 4 metriğinin 0-10 ve 0-100 z-normalleştirmesine göre kıyaslanması anlam çıkartmak için önem taşımaktadır.

Tablo 3.38 üzerinde bu kıyaslama işlemi gerçekleştirilebilmektedir. Örnek olarak Z-Normalleştirme aralığının CPU için yapılması durumunda deney sıralamalarına tablo 3.37 üzerinde ilk 10 için bakıldığında 34, 25, 36, 49,16 ve 17 numaralı deneyler 0-10 ve 0-100 aralığındaki Z-Normalleştirme için farklılık göstermemektedir. 10 deneyin 6 tanesi CPU için Z-Normalleştirmeden bağımsız olarak ilk 10'da aynıyken, RAM için 10 deneyin 10'u aynıdır. NetI için 10 deneyin 8'i NetO için ise 10 deneyin 9'u aynıdır. Bu durumda Z-Normalleştirme RAM, NETI ve NetO için gereksiz olarak görülse de CPU için ciddi olmamakla beraber sonuçların farklılaştığını göstermektedir.

Tablo 3.38: Her Deney için Tüm Kaynaklara Uygulanan Z-Skorların DKP Puanlarına Göre Sıralaması

	Deney Numaraları							
	CPU		RAM		NETI		NETO	
SIRALAMA	**z10**	**z100**	**z10**	**z100**	**z10**	**z100**	**z10**	**z100**
1	**34**	34	**42**	42	**33**	33	**33**	33
2	**25**	35	**36**	36	**42**	42	**42**	42
3	12	25	**39**	5	**46**	46	**36**	36
4	2	3	**5**	39	**47**	47	**4**	4
5	**36**	49	**9**	9	**49**	49	**6**	27
6	**49**	13	**38**	38	**10**	10	**3**	3
7	**16**	17	**3**	3	7	50	**27**	38
8	**17**	50	**14**	50	40	16	**38**	45
9	22	16	**50**	14	**9**	9	**45**	6
10	50	36	**32**	32	**16**	40	7	19
11	47	26	20	20	50	7	9	9
12	5	15	27	27	6	17	2	31
13	14	23	34	2	17	19	31	2
14	15	47	2	34	19	6	43	43
15	4	18	22	22	18	39	20	12
16	40	5	49	23	15	22	29	7
17	23	6	23	6	39	18	30	11
18	24	40	44	49	2	15	25	20
19	32	24	6	21	22	48	35	29
20	1	37	28	18	31	2	19	30
21	44	22	21	11	36	8	18	35
22	6	32	11	44	41	31	49	25
23	31	44	18	7	48	36	23	32
24	41	2	29	29	8	24	34	18
25	8	12	15	25	24	41	37	23
26	19	9	25	28	4	1	12	39
27	11	14	37	37	5	4	11	44
28	26	41	7	40	14	5	39	37
29	48	7	19	12	44	14	44	46
30	18	20	17	15	1	25	32	10
31	21	27	40	17	25	26	46	48
32	27	33	46	46	32	44	1	49
33	28	31	12	48	23	32	10	26
34	39	11	1	1	26	23	21	41
35	3	4	8	8	28	28	48	28
36	10	19	48	19	29	45	28	34
37	30	42	24	26	37	37	5	5
38	42	48	31	24	38	38	17	17
39	7	28	13	31	43	43	50	50
40	9	29	10	41	45	11	41	1
41	45	39	35	13	11	13	13	14
42	46	8	41	4	13	20	26	15
43	13	21	16	35	20	29	14	21
44	20	30	4	10	27	27	15	8
45	29	46	26	16	34	3	40	13
46	33	10	43	43	21	12	47	16
47	35	45	45	45	3	21	8	22
48	37	1	47	47	12	34	22	24
49	38	38	33	33	35	35	24	40
50	43	43	30	30	30	30	16	47

4. BULGULAR VE TARTIŞMA

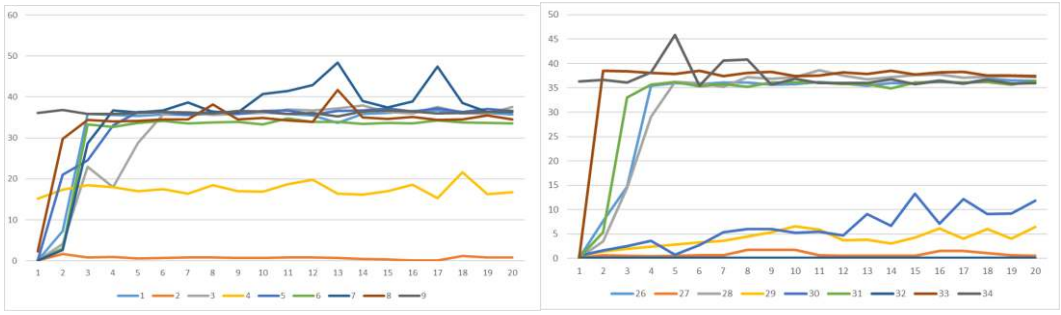
4.1. Anormal Durum Tanılama

4.1.1. Zafiyet Tipine Göre Anormal Durum Değişimi

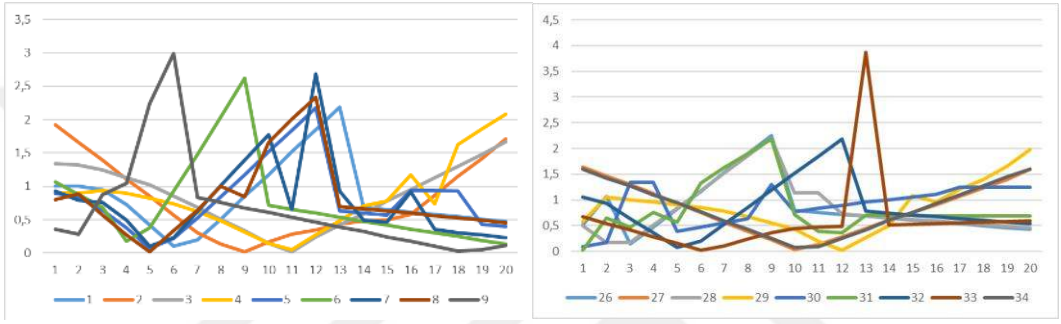
Şekil 4.1 ve Şekil 4.2 üzerinden CPU örneği ile A zafiyeti için Z-Skor'un ne gibi farklılık gösterdiği anormal durum tespiti için analiz edilmiştir. Aynı tip yaklaşımla tüm kaynaklar analiz edilmiştir. Anlaşılmaktadır ki, bir zafiyet için saldırı şekli farklı olsa dahi aynı zafiyetin sömürüsü durumundaki sistem davranışı farklı zafiyet sömürülerine kıyasla benzer şekildedir. Bu sebeple zafiyetin var olmasından bağımsız olarak aynı zafiyetlerin birbirine benzeşme durumunu olumludur, her zafiyet tipinin sömürü durumu sistemde farklı bir davranış yaratmaktadır.

Bununla beraber normal trafiğe verilen sistem tepkisi ile zafiyet sömürüsüyle orantılı ayrışma görülmektedir. Bu sebeple sistemin normal trafiğinin içerisinden zafiyet sömürü tespitinin mümkünlüğü de olumludur. Zafiyetin yük kaynak dağılımı değişse dahi karakteristik olarak benzeşmektedir. Bununla beraber normal kullanıcı hareketlerine zafiyet sömürü yükünün karakteristik olarak yakınlaşması ancak güvenlik seviyesi arttırıldığında görülmektedir. Yani zafiyet kapatıldıysa ve bu zafiyete dair sömürü yapılıyor ise, zafiyetin kapatılmaması durumunda zafiyet sömürüsüne sistemin verdiği tepkiler kıyaslanmıştır. Bu kıyas sonucunda zafiyetin var olması durumunda sistem davranışı normal trafiğe olan davranıştan daha karakteristik bir davranış göstermektedir. Bu durum zafiyetin var olması durumundaki sömürülerin daha kolay olarak tespitini sağlamaktadır.

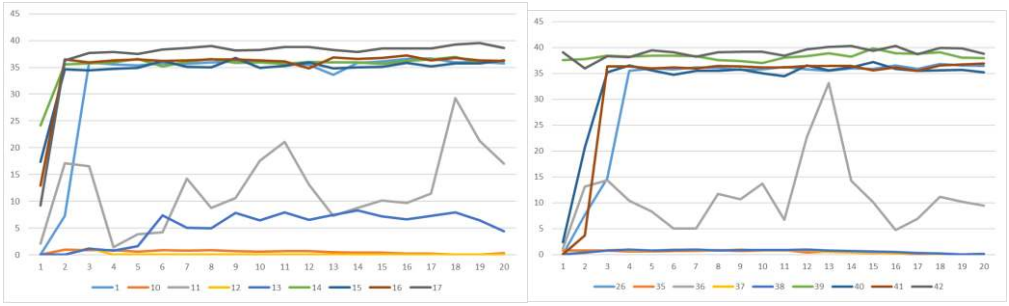
Zafiyetin var olmaması durumda sistem davranışı üzerinde çok ciddi farklılık görülmektedir. Genel anlamda kapatılmamış zafiyetin sömürülmesi durumunda normal kullanıcı yüküne kıyasla sistem davranışları arasındaki entropi seviyesi daha yüksektir. Bu entropi aynı kaynak yükü dağılımıyla zafiyetin kapatılması durumunda daha düşüktür. Bu durum grafikteki zikzaklarla da gözlenebilmektedir. Aynı durum Şekil 4.3 ve Şekil 4.4. üzerindeki CPU tepkisinin B zafiyetinin taranmasına dair verilen örnekten anlaşılmaktadır ki zafiyet tipinden bağımsız olarak zafiyetin var olması durumunda anormal davranış görülmektedir.



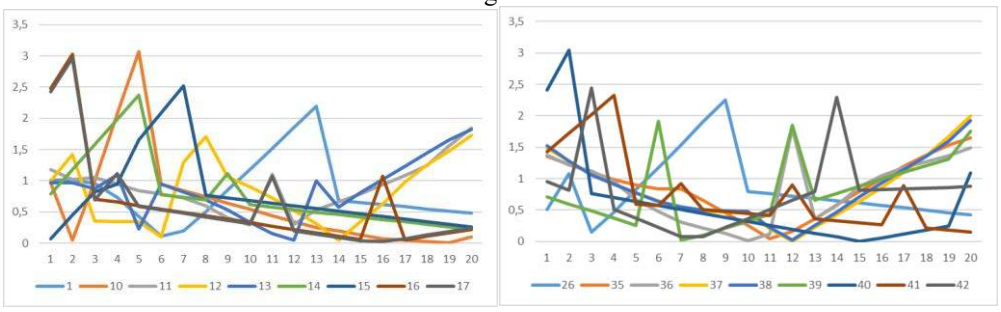
Şekil 4.1. CPU için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki A Zafiyeti için Yük Yoğunluklarının Dağılımı



Şekil 4.2. CPU için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki A Zafiyeti için Yük Yoğunluklarının Z-Skor Dağılımı



Şekil 4.3. CPU için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki B Zafiyeti için Yük Yoğunluklarının Dağılımı



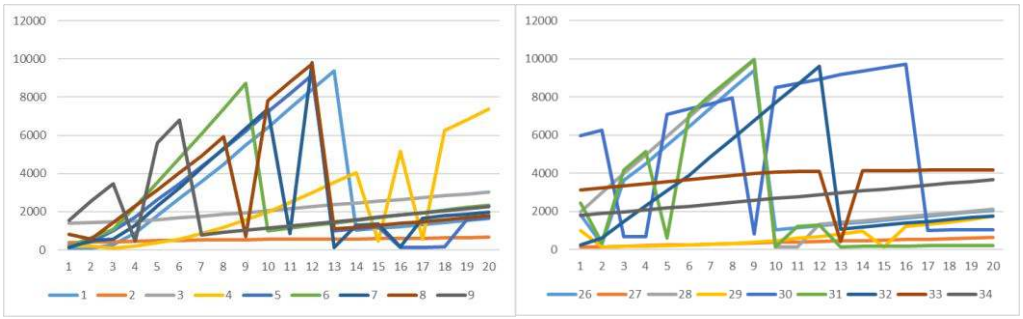
Şekil 4.4. CPU için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki B Zafiyeti için Yük Yoğunluklarının Z-Skor Dağılımı

Şekil 4.5, Şekil 4.6 , Şekil 4.7 ve Şekil 4.8 üzerinden görülmektedir ki Deney No 3 ve 28, Deney No 8 ve 33, Deney No 4 ve 8 için zafiyet kapatılınca ciddi bir farklılık görülmektedir. Zafiyetin kapatıldığında sömürüsüne yönelik tarama durumunda sistemler arasında ciddi farklılık oluşmuştur. Zafiyetin sömürülmediği karakteristikler olan 1 ve 26da zafiyetin kapatılma seviyesi değiştirilse dahi değişmemiştir. Bu gösterge ise zafiyetin var olması durumunda normal kullanıcı hareketlerinin zararlı kullanıcılara göre ayrışabileceğini göstermektedir. Fakat bu ayrışma durumu zafiyetin var olması durumunda çok daha yüksektir.

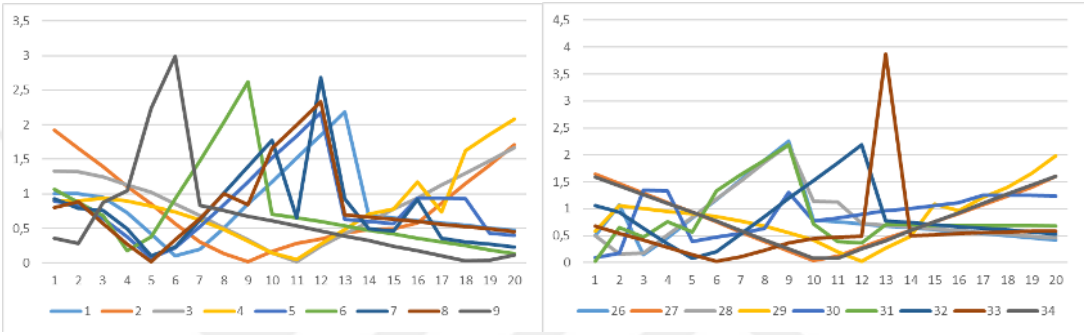
Görülmektedir ki normal kullanıcıların hareketleri ile zafiyetlerin varlığı arasında bağımlılık kesin göz ardı edilebilecek düzeyde azdır. Bu durum zafiyeti aynı yük deseniyle sömüren kullanıcıların hareketleriyle zafiyetin varlığı arasındaki bağlantıyla kıyaslanırsa göz ardı edilemeyecek düzeyde farklıdır. Bu durumda çalışmalarda kullanıcı trafiğinin rastsal olarak cüz'i bir bölümü açıkların bilinçli olarak bırakıldığı bir honeypot sunucuya yönlendirilmesi yoluyla hangi zafiyet türlerinin sömürülmek istendiği tespit edilebilmektedir.

Bu sayede sonarcloud gibi sistemlerin quality gate'leri tasarlanabilmektedir. Zafiyetlerin farklılıkları da aynı desen tipinde nasıl ayrıştığı karakteristiğinin algılanması için önemlidir. Bunun içinde zafiyet karakteristiği grafiklerine bakılmalıdır. Burada her bir zafiyetin normal sisteme kıyasla ne kadar ayrıştığı korelasyon analizleriyle, her bir yük dağılımı gerçekten oransal olarak gözle görülür düzeyde zafiyet -sömürü kaynağına göre mi yoksa sömürü aracına göre mi değiştiği analiz edilerek araştırılmalıdır.

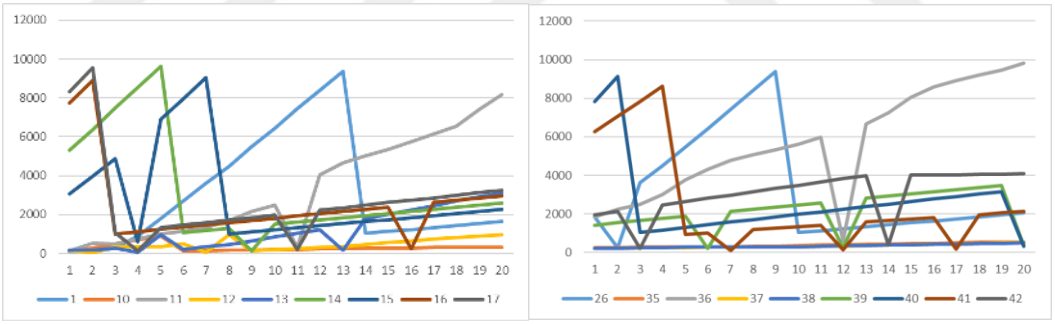
Farklı zorluk düzeyleri bir biriyle kıyaslanarak entropi kıyaslamayası yapılmalıdır. Ayrıca aynı zafiyet düzeyi ve yük dağılımına göre hangi kaynak kullanılarak daha öncelikli analiz yöntemi oluşturulabileceği z-skor kullanılarak yapılabilmekte olduğu görülmektedir.



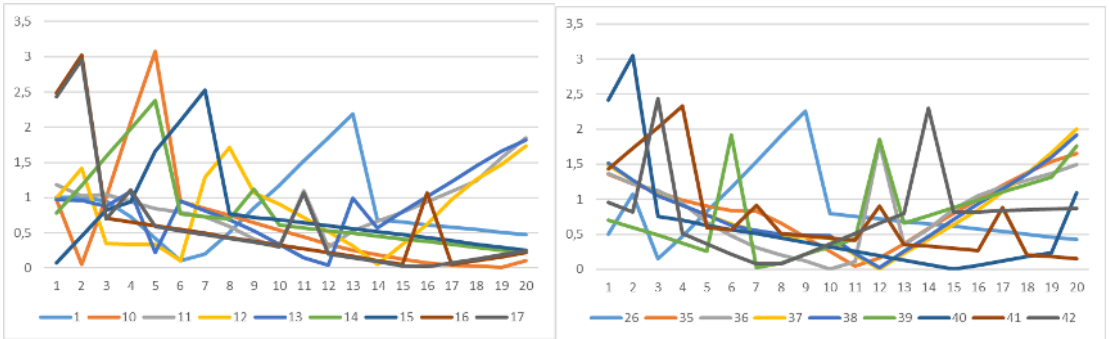
Şekil 4.5. NETI için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki A Zafiyeti için Yük Yoğunluklarının Dağılımı



Şekil 4.6. NETI için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki A Zafiyeti için Yük Yoğunluklarının Z-Skor Dağılımı



Şekil 4.7. NETI için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki B Zafiyeti için Yük Yoğunluklarının Dağılımı



Şekil 4.8. NETI için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki B Zafiyeti için Yük Yoğunluklarının Z-Skor Dağılımı

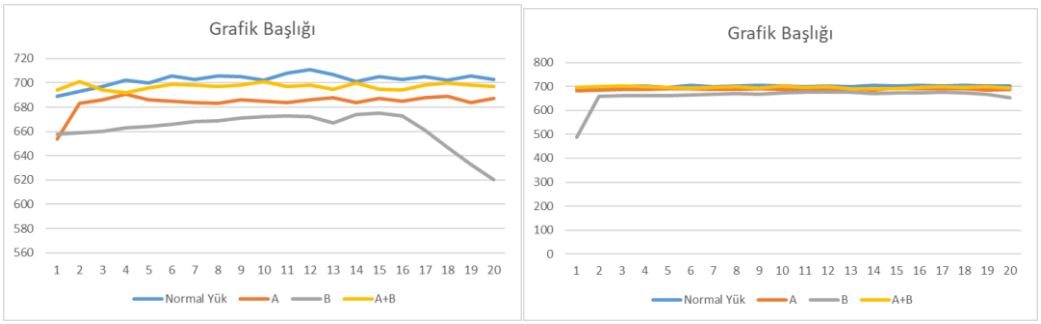
4.1.2. Yük Tipine Göre Anormal Durum Değişimi

Bulgular örnek olarak herhangi bir kaynak incelendiğinde ciddi değişiklik göstermemekle beraber örnek RAM için Şekil 4.9, Şekil 4.10, Şekil 4.11 ve Şekil 4.12 kaynağı için incelenirse görülebilmektedir. Ayrıca bu kapsamda diğer kaynak tiplerinden NETI için de zafiyetin ortadan kalkması durumunda zafiyet yükü ile normal yükün aynılaşma çok kritik değere sahiptir. Normal trafiğin yapısının entropisi ile mevcut trafiğin entropisine bakılmalıdır. Ayrıca normal trafiğin ortaya çıkardığı z-skor mevcut trafik datalarına uygulanılarak farklılık gösterilmelidir.

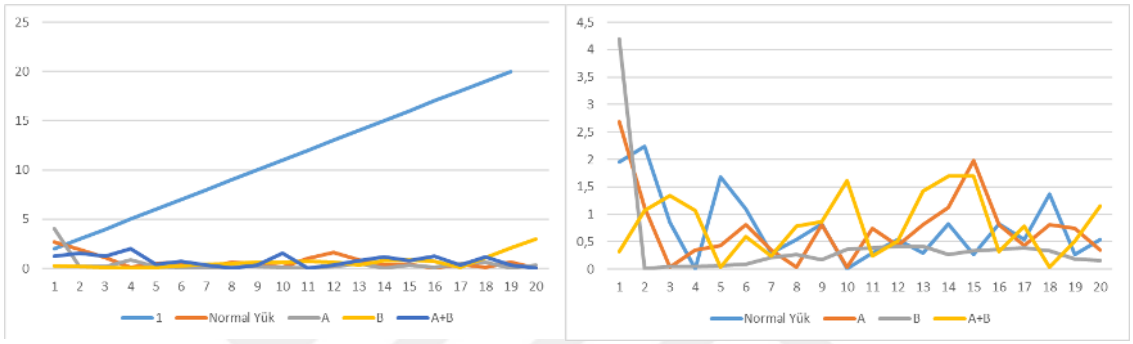
Zafiyetin kapatılması durumunda zafiyetli olan yani mevcut kabul edilen trafiğin 10 saniyede bir alınan tek datasının z-skoruna bakılır. Owasp zap taraması burada görülmektedir ki, NetI taraması için verimli sonuç vermemektedir. Normale göre zafiyet taramaları ayrışmaktadır. Fakat zafiyetin ne olduğuna dair tekillik tabanlı olarak ayrıştırma yapılarak hangi zafiyetin tarandığını anlamak mümkün değildir.

Genel yük tiplerinin davranışlarına bakıldığında görülmektedir normal kullanıcı yükü için, zafiyetlerin bulunması durumundaki sistem davranışı ile zafiyetlerin kapatılması durumundaki sistem davranışı ciddi bir benzerlik göstermektedir. Bu önerilen sistem için olumlu bir durumdur çünkü zafiyetin sömürülmesi durumunda normal kullanıcı trafiğinde zafiyet sömürüsünün olduğuna dair sistem davranış farklılığının tespitini mümkün kılmaktadır. Bununla zafiyetin kapatılması ve kapatılmaması durumunda zafiyetli yük trafiklerinin oluşturduğu sistem davranışı normal yük davranışıyla örnek olarak Şekil 4.9 ve Şekil 4.10 üzerindeki 2-Zor dışında benzeşmemektedir.

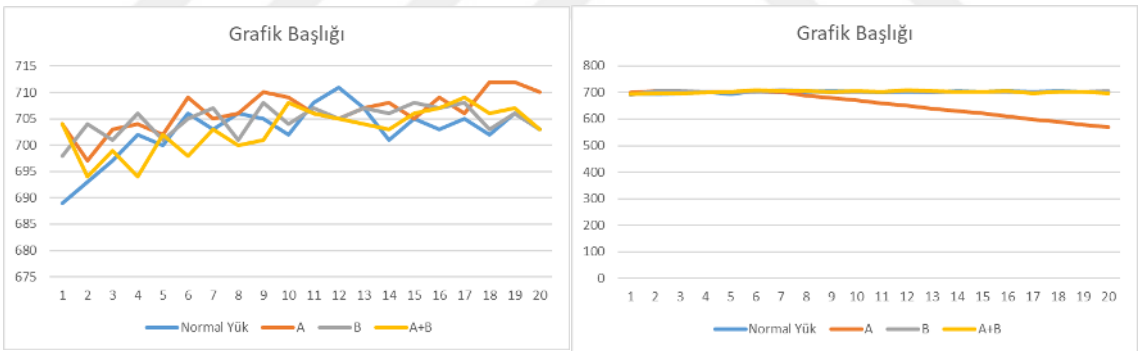
Bu durumda zafiyet tarama yöntemi olarak 2 ve 3 yani scripted ve owasp zap ile tarama yöntemleri NetI olarak ayrışmaktadır. Fakat zafiyet türleri benzeşmektedir. Yani NetI ile sistemdeki kötü niyetli kullanıcıların sistem üzerinde oluşturduğu normalden farklı davranış yoluyla anormal durum tespit edilebilir. Bununla beraber nasıl zafiyet taraması yapıldığı da tespit edilebilir. Ancak hangi zafiyete yönelik tarama yapıldığı tespit edilemez.



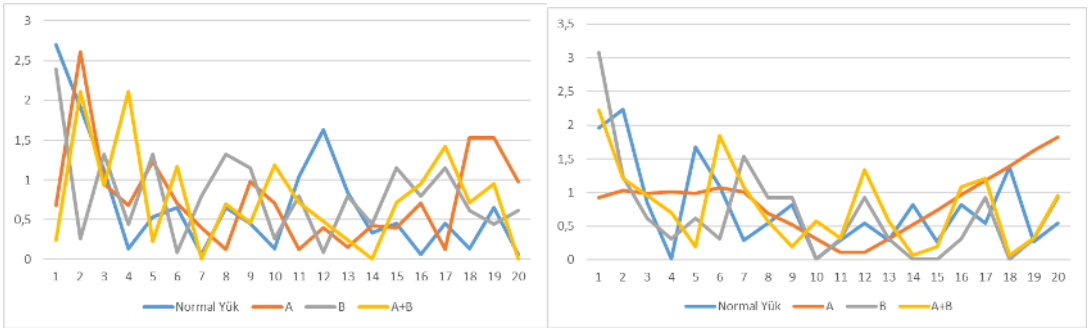
Şekil 4.9. RAM için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki 2 Numaralı Yük Tipi için Zafiyet Davranışlarının Dağılımı



Şekil 4.10. RAM için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki 2 Numaralı Yük Tipi için Zafiyet Davranışlarının Z-Skor Dağılımı



Şekil 4.11. RAM için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki 8 Numaralı Yük Tipi için Zafiyet Davranışlarının Dağılımı



Şekil 4.12. RAM için DWVA Kolay(Sol) ve Zor(Sağ) Seviyesindeki 8 Numaralı Yük Tipi için Zafiyet Davranışlarının Z-Skor Dağılımı

4.2. Nokta Çift Serili Korelasyon Yönteminin Zafiyet Karakteristik Tespitine Yönelik Başarımı

4.2.1. Zafiyet Tipine Göre Zafiyet Karakteristik Değişimi

Tablo 3.38 üzerinde korelasyon matrisinin sıralanması yoluyla elde edilen DKP'lere göre sıralaması her bir kaynağa göre analiz edilmiştir. Elde edilen analiz Tablo 3.9. üzerindeki deney numaralarına göre zafiyetlerin bir birinde ne düzeyde ayrıştığına dair bulgular bulunmaktadır. Bu noktada Tablo 3.38 incelendiğinde deney numaralarının Zafiyet tiplerine göre analizi yapılmalıdır.

CPU örneği için Tablo 4.1. üzerinde olduğu gibi bu sıralamaların Z-Normalleştirmeden bağımsız olarak toplamda her bir kaynakta ilk 10 deneyinin incelenmesi ve bu ilk 10'da hangi zafiyet tiplerinin bulunma sayısı Tablo 4.2 üzerinde verilmiştir. Bu noktada A, B ve A+B tipi zafiyetlerin karakteristik olarak üst noktalarda olma oranlarına yönelik incelemelerde bulunulmuştur.

Tablo 4.1. CPU Üzerinde Alınan Metriklerin 0-10 VE 0-100 Aralığında Normalleştirildiği Deney Numaralarının Göre DKP'ye Göre Sıralaması

Z-Normalleşme							
0 - 10 Aralığı				0 - 100 Aralığı			
dwva Zafiyet Zorluk	**Tip**	**Deney No**	DKP	**dwva Zafiyet Zorluk**	**Tip**	**Deney No**	DKP
Çok Zor	A	34	552	**Çok Zor**	A	34	378
Kolay	A+B	25	269	**Çok Zor**	B	35	179
Kolay	B	12	153	**Kolay**	A+B	25	164
Kolay	A	2	152	**Kolay**	A	3	156
Çok Zor	B	36	152	**Çok Zor**	A+B	49	155
Çok Zor	A+B	49	150	**Kolay**	B	13	128
Kolay	B	16	135	**Kolay**	B	17	125
Kolay	B	17	125	**Çok Zor**	A+B	50	125
Kolay	A+B	22	125	**Kolay**	B	16	119
Çok Zor	A+B	50	125	**Çok Zor**	B	36	117

Tablo 4.2. Zafiyet Tiplerinin Her Bir Kaynak İçin Z-Normalleştirmeden Bağımsız Olarak DKP'ye Göre Sıralandığında İlk 10 Üzerinde Bulunan Deneylerin Zafiyet Tiplerine Göre Sayıları

Kaynak Tipi	A Zafiyeti Miktarı	B Zafiyeti Miktarı	A+B Zafiyeti Miktarı
CPU	4	9	7
RAM	8	10	2
NETI	5	8	7
NETO	11	6	3
Toplam	28	33	19

Tablo 4.2 üzerinde görülmektedir ki kaynaklara göre zafiyetler karakteristik olarak farklı yüktedir. Her bir zafiyet tipinin daha karakteristik olduğu kaynak tipleri bulunmaktadır. Bununla beraber kaynak tipinden bağımsız olarak her bir zafiyet tipinin kaynaktan bağımsız olarak DKP'ye göre ilk 10'da bulunma sayısı incelenmelidir.

Tablo 4.2 üzerinde her zafiyetin toplam olarak kaynaklarda bulunma sayılarına bakıldığında A zafiyeti toplam 28, B zafiyeti toplam 33 ve A+B zafiyeti toplam 19 kere bulunmaktadır. Bu rakamlar şunu göstermektedir ki zafiyetten bağımsız olarak tek bir zafiyet tipini sömürmek üzere sisteme bir saldırı yapılması durumunda bu durum karakteristik olarak ayrışabilmektedir.

Eğer A veya B zafiyeti ilk DKP'ye göre 10'da bulunma sayısında ciddi bir farklılık olsaydı bu durumun deney koşulları ve kullanılan zafiyet tipi materyalinden kaynaklandığı söylenebilirdi. Mevcut durumda bu sebeple zafiyetten bağımsız olarak tek bir zafiyet tipiyle saldırılma durumunun karakteristik farklılığının gözlendiği söylenebilmektedir.

Ayrıca eğer birçok zafiyet aynı anda taranırsa Tablo 4.2 üzerinde A+B zafiyetinin kaynaklardan bağımsız olarak toplam DKP'ye göre ilk 10'da bulunma oranına bakılabilmektedir. Bu noktada A zafiyetinin ve B zafiyetinin direk olarak sömürülmesiyle, birlikte sömürülmesi gözle görülür bir farklılık A+B'nin toplamda 19 kere bulunması üzerinden anlaşılmaktadır.

4.2.2. Yük Tipine Göre Zafiyet Karakteristik Değişimi

Tablo 4.3. üzerindeki CPU örneği tüm kaynaklar için Tablo 4.4.'e dönüştürülerek yük numaralarının kaynak bağımlı olarak toplam değerleri alınmıştır. Daha sonra daha detaylı yüklerin yapılarına dair analiz sağlayabilmek için yük içerikleriyle beraber olacak şekilde Tablo 4.5. analiz edilmiştir.

Tablo 4.3. RAM Örneği İçin DKP Sıralamasının Z-Normalleşme Ayrımına Göre Yük Numaralarının Sıralaması

Z-Norm:10					Z-Norm:100				
dwva Zafiyet Zorluk	Normal Kullanıcı Yükü	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı	Yük Yapı Numarası	dwva Zafiyet Zorluk	Normal Kullanıcı Yükü	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı	Yük Yapı Numarası
Çok Zor	240	12	48	9	Çok Zor	240	12	48	9
Çok Zor	0	0	300	3	Çok Zor	0	0	300	3
Çok Zor	240	60	0	6	Kolay	0	60	240	5
Kolay	0	60	240	5	Çok Zor	240	60	0	6
Kolay	240	12	48	9	Kolay	240	12	48	9
Çok Zor	0	60	240	5	Çok Zor	0	60	240	5
Kolay	0	0	300	3	Kolay	0	0	300	3
Kolay	240	60	0	6	Çok Zor	240	12	48	9
Çok Zor	240	12	48	9	Kolay	240	60	0	6
Çok Zor	240	0	60	7	Çok Zor	240	0	60	7

Tablo 4.4. Z-Normalleştirmeden Bağımsız Olarak Deney Yük Yapı Numaralarının DKP Sıralamasına Göre İlk 10'da Her Kaynak İçin Bulunma Sayılarının Toplamları

Deney Yapı Numarası	CPU	RAM	NETI	NETO	Toplam
2	2	0	2	2	6
3	3	4	0	5	12
4	1	0	0	4	5
5	1	4	2	2	9
6	1	4	2	2	9
7	0	2	3	1	6
8	4	0	6	2	12
9	8	6	5	2	21

Tablo 4.5. Deney Yük Yapı Numaralarının Detaylı Olarak Dağılımı ve DKP Üzerinde İlk 10'da Kaynak ve Z-Normalleştirmeden Bağımsız Olarak Bulunma Toplamları

Deney Yük Numarası	Normal Kullanıcı Yükü	OWASP Zap Tarama Oranı	Scripted Zafiyet Tarama Oranı	Toplam
2	0	300	0	6
3	0	0	300	12
4	0	240	60	5
5	0	60	240	9
6	240	60	0	9
7	240	0	60	6
8	240	48	12	12
9	240	12	48	21

Tablo 4.5 üzerinden anlaşılmaktadır ki Deney Yük Numarası 3, 8 ve 9 numaralı deneyler daha karakteristiktir. Özellikle 8 ve 9 numaralı yük tipleri incelendiğinde anlaşılmaktadır ki aynı anda 2 farklı yük tipi kullanılırsa sistem daha karakteristik bir davranış göstermektedir. Ayrıca 9 ve 3 numaralı yük numaralarının ortak noktası scripted olarak zafiyetin sömürülmesidir. Buradan bu zafiyet yük tipine göre daha karakteristik bir yapı oluşabilmesinin mümkün olabileceği anlaşılmaktadır. Hangi yöntem ile sistem zafiyetlerinin taranıldığını tespit etmenin mümkün olduğu görülebilmektedir.

Tablo 4.5 üzerinde ayrıca anlaşılmaktadır ki direk olarak normal trafiğin olmadığı bir sistem taramasının yapılması durumunda bu karakteristik olarak anlaşılır olma oranını, normal trafiğin olduğu bir sisteme kıyasla düşürmektedir.

4.2.3. Z-Normalleştirme Dönüşümüne Göre Zafiyet ve Yüke Göre Karakteristik Değişimi

Zafiyete yönelik Z-Normalleştirme aralığının sonuçlar üzerinde ne düzeyde farklılık gösterdiğini anlamak için Tablo 4.6 incelendiğinde anlaşılmaktadır. Tablo 4.6.ya göre Kaynaktan bağımsız olarak 0-10 ve 0-100 aralığına dönüştürme sırasıyla incelenmelidir. İnceleme sağlandığında şu görülmektedir ki sırasıyla A zafiyeti 15 ve 13, B zafiyeti 16 ve 17, A+B zafiyeti 9 ve 10 olarak görülmektedir. Net olarak görülmektedir ki zafiyet tiplerinin karakteristik yapıları Z-normalleştirme aralığına bağımlı olarak sonuç vermemektedir.

Tablo 4.6. Zafiyetlerin DKP Puanlarının İlk 10'da Bulunma Sayısına Göre Z-Normalleştirme ve Kaynak Tipine Göre Sayıları

	Z-Normalleşme					
	0 - 10 Aralığı			0 - 100 Aralığı		
Kaynak Tipi	A Sayısı	B Sayısı	A+B Sayısı	A Sayısı	B Sayısı	A+B Sayısı
CPU	2	4	4	2	5	3
RAM	4	5	1	4	5	1
NETI	3	4	3	2	4	4
NET O	6	3	1	5	3	2
Toplam	15	16	9	13	17	10

Tablo 4.6 üzerinden zafiyete yönelik Z-Normalleştirme aralığının sonuçlar aynı şekilde yük tipinin ne gibi farklılık gösterdiğini analiz etmek içi aynı yapıda yapılan analizler göstermektedir ki yük yapısı Z-Normalleştirme aralığı da bağımsız olarak sonuç vermektedir.

5. SONUÇ

Yapılan deneylere yönelik analizlerde görülmektedir ki zafiyetin tipine göre hangi kaynağın önem taşıyacak bir niteliğe sahip olduğu değişkenlik göstermektedir. Bazı zafiyet tiplerinde anormal olarak kabul edilen değerler CPU RAM ve NETI gibi 3 adet metrik tipiyken bazıları için 2, bazen ise 4 adet metriğin kullanılması gerekebilmektedir.

Deneyler yoluyla Şekil 3.17 üzerindeki Ön Süreç ve Sistem İyileştirme süreçlerinin "Anormal Durum Tanılama" modülünün gerekli olduğu anlaşılmaktadır. Bununla beraber zafiyet tiplerinin sömürülebilir olması durumundaki sistem davranışının, sömürünün olmadığı durumlarla kıyaslaması yapıldığında aynı olmadığı görülmüştür. Bu sebeple sistemin güvenlik düzeyinin değişimi davranış değişimini de beraberinde getirdiği görülmektedir.

Yük yapılarının farklılaşması durumunda zafiyet yapısından bağımsız olarak sistem davranış farklılığının tespiti, gürültünün yüksek olmasından dolayı zafiyet sömürü tahminlem başarımı düşürecektir. Bu sebeple Şekil 3.14 üzerindeki gibi Honeypot'un da bulunduğu bir topolojinin mantıklı olduğu görülmektedir. Fakat anlık olarak sömürülerin davranışları değişmese dahi deneylerde görülmüştür ki, yük dağılımı ve zafiyetlerin giderilmesi gibi teknik gerekçeler de davranış farklılığının oluştuğunu göstermektedir. Bu sebeple geçmişte dataları da hesaplamaya katarak zafiyet sömürü yoğunluğunun Şekil 3.20 üzerinde gösterildiği gibi yapılmasının şekil 3.15 üzerinde önerilen sistem için en stabil ve güvenilir olarak sağlayacaktır.

Veri Ambarı verilerinin dönüşüm ihtiyacı bulunmaktadır. Fakat dönüşüm aralıkları log tipine göre değişmektedir. Bazen ise farklılık göstermemektedir. Bu sebeple analizi yapılacak logların z-normalleştirme dönüşümlerinin özel olarak analizi sistemin daha sağlıklı çalışması için önem taşımaktadır.

Önerilen sistemin sürekli olarak güncel veri vermesi performansı arttırmak üzere gereksiz güvenlik önlemlerinin kaldırılmasını sağlayacaktır. Bu yolla emniyet ve performans artışı sağlanabilecektir. Sistemin öğrenme süreçleri yapay yapay sinir ağları gibi daha kompleks algoritmalarla sağlanabilmektedir. Ayrıca nginx, apache, iss gibi sistemlerin erişim hata logları veya kullanıcıların direk olarak oturum süreleri, istekleri gibi dataların da analiz edilmesi yoluyla 3.15 üzerinde önerilen sistemin başarımı arttırılabilmektedir.

Sistem puanlama ve raporlama sürecini daha detaylı olarak yapabilir, bu kapsamda DevSecOps süreçlerindeki güvenlik parametrelerini dinamik değiştirebilir. Bu tip bir durumda önerilen yarı-otomatik sistemden insan faktörünün etkisi azaltılıp tam-otomatik bir sistem kullanılabilmektedir.

KAYNAKLAR

[1] Seeking the Relation between Performance and Security in Modern Systems: Metrics and Measures: Fujdiak, R., Mlynek, P., Blazek, P., Barabas, M., & Mrnustik, P. (2018, July). Seeking the relation between performance and security in modern systems: metrics and measures. In 2018 41st International Conference on Telecommunications and Signal Processing (TSP) (pp. 1-5). IEEE.

[2] Fan, W. C. (2020). Interactivity performance benchmark for Windows and Mac OS (Doctoral dissertation, UTAR).

[3] Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2017, October). Energy efficiency across programming languages: how do energy, time, and memory relate?. In Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering (pp. 256-267).

[4] Islam, R., Rokon, M. O. F., Darki, A., & Faloutsos, M. (2020). HackerScope: The Dynamics of a Massive Hacker Online Ecosystem. arXiv preprint arXiv:2011.07222.

[5] https://www.slideserve.com/hera/owasp-top-10-2010 [Accessed: 07.06.2021]

[6] Kumar, N., Zadgaonkar, A. S., & Shukla, A. (2013). Evolving a new software development life cycle model SDLC-2013 with client satisfaction. International Journal of Soft Computing and Engineering (IJSCE), 3(1), 2231-2307.

[7] Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum(Vol. 18). PTR Upper Saddle River^ eNJ NJ: Prentice Hall.

[8] https://kubernetes.io/docs/concepts/overview/components/#node-components [Accessed: 07.06.2021]

[9] https://www.ic3.gov/Home/AnnualReports (The Internet Crime Complain Center Annual Reports. 2020,2019,2018.2017,2016)

[10] https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf [Accessed: 07.06.2021]

[11] https://cwe.mitre.org/data/definitions/284.html [Accessed: 07.06.2021]

[12] https://owasp.org/www-committee-compliance/[Accessed: 07.06.2021]

[13] https://owasp.org/about/[Accessed: 07.06.2021]

[14] Viitasuo, E. (2020). Adding security testing in DevOps software development with continuous integration and continuous delivery practices.

[15] Deshlahre, R., & Tiwari, N. (2020). A Review on Benchmarking: Comparing the Static Analysis Tools (SATs) in Web Security. Social Networking and Computational Intelligence, 327-337.

[16] Ahmed, A. M. A. A. (2019). DevSecOps: Enabling Security by Design in Rapid Software Development.

[17] Heilmann, J. (2020). Application Security Review Criteria for DevSecOps Processes.

[18] Bravo, S., & Mauricio, D. Agile method for detecting DDoS attacks in the application layer based on user's dynamism.

[19] Rowanhill, J. (2016). Class server toolset: Design and implementation.

[20] Renita, J., & Elizabeth, N. E. (2017, March). Network's server monitoring and analysis using Nagios. In 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 1904-1909). IEEE.

[21] Tso, F. P., Jouet, S., & Pezaros, D. P. (2016). Network and server resource management strategies for data centre infrastructures: A survey. Computer Networks, 106, 209-225.

[22] Liu, Y., Gunasekaran, R., Ma, X., & Vazhkudai, S. S. (2016, November). Server-side log data analytics for I/O workload characterization and coordination on large shared storage systems. In SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 819-829). IEEE.

[23] Maimon, D., Babko-Malaya, O., Cathey, R., & Hinton, S. (2017, November). Re-Thinking Online Offenders' SKRAM: Individual Traits and Situational Motivations as Additional Risk Factors for Predicting Cyber Attacks. In 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (pp. 232-238). IEEE.

[24] Yu, S. (2013). Behavioral evidence analysis on Facebook: A test of cyber-profiling. Defendologija, 16(33), 19-30.

[25] Booth, H., & Turner, C. (2016). Vulnerability Description Ontology (VDO): a Framework for Characterizing Vulnerabilities.

[26] Gao, H., Zhu, J., Liu, L., Xu, J., Wu, Y., & Liu, A. (2019, May). Detecting SQL Injection Attacks Using Grammar Pattern Recognition and Access Behavior Mining. In 2019 IEEE International Conference on Energy Internet (ICEI) (pp. 493-498). IEEE.

[27] Gawron, M., Cheng, F., & Meinel, C. (2017, April). PVD: Passive vulnerability detection. In 2017 8th International Conference on Information and Communication Systems (ICICS) (pp. 322-327). IEEE.

[28] Xiao, Z., Zhou, Z., Yang, W., & Deng, C. (2017, May). An approach for SQL injection detection based on behavior and response analysis. In 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN) (pp. 1437-1442). IEEE.

[29] Matin, I. M. M., & Rahardjo, B. (2019, November). Malware detection using honeypot and machine learning. In 2019 7th International Conference on Cyber and IT Service Management (CITSM) (Vol. 7, pp. 1-4). IEEE.

[30] Asselin, E., Aguilar-Melchor, C., & Jakllari, G. (2016, October). Anomaly detection for web server log reduction: A simple yet efficient crawling based approach. In 2016 IEEE Conference on Communications and Network Security (CNS) (pp. 586-590). IEEE.

[31] Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., ... & Zhang, D. (2019, August). Robust log-based anomaly detection on unstable log data. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 807-817).

[32] Liu, Z., Qin, T., Guan, X., Jiang, H., & Wang, C. (2018). An integrated method for anomaly detection from massive system logs. IEEE Access, 6, 30602-30611.

[33] Khan, S., & Motwani, D. (2017, June). Implementation of IDS for web application attack using evolutionary algorithm. In 2017 International Conference on Intelligent Computing and Control (I2C2) (pp. 1-5). IEEE.

[34] Khairkar, A. D., Kshirsagar, D. D., & Kumar, S. (2013, April). Ontology for detection of web attacks. In 2013 International Conference on Communication Systems and Network Technologies (pp. 612-615). IEEE.

[35] Fang, X., Xu, M., Xu, S., & Zhao, P. (2019). A deep learning framework for predicting cyber attacks rates. EURASIP Journal on Information Security, 2019(1), 1-11.

[36] Lu, T., & Chen, J. (2017, September). Research of penetration testing technology in docker environment. In 2017 5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017) (pp. 1354-1359). Atlantis Press.

[37] Rafter, R., & Smyth, B. (2001). Passive profiling from server logs in an online recruitment environment. In Workshop on Intelligent Techniques for Web Personalization at the the 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, August, 2001.

[38] Sekharan, S. S., & Kandasamy, K. (2017, March). Profiling SIEM tools and correlation engines for security analytics. In 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 717-721). IEEE.

[39] Zakaria, W. Z. A., & Kiah, M. L. M. (2012, April). A review on artificial intelligence techniques for developing intelligent honeypot. In 2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT) (Vol. 2, pp. 696-701). IEEE.

[40] Antunes, N., & Vieira, M. (2015, June). On the metrics for benchmarking vulnerability detection tools. In 2015 45th Annual IEEE/IFIP international conference on dependable systems and networks (pp. 505-516). IEEE.

[41] Tunde-Onadele, O., He, J., Dai, T., & Gu, X. (2019, June). A study on container vulnerability exploit detection. In 2019 IEEE International Conference on Cloud Engineering (IC2E) (pp. 121-127). IEEE.

[42] Juvonen, A., Sipola, T., & Hämäläinen, T. (2015). Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Computer Networks, 91, 46-56.

[43] Ramaki, A. A., Khosravi-Farmad, M., & Bafghi, A. G. (2015, September). Real time alert correlation and prediction using Bayesian networks. In 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC) (pp. 98-103). IEEE.

[44] Subasic, P., Holtan, H. M., Cabacungan, R. S., Shukla, M., Mehta, R. A., Chang, J. L., ... & Qu, Y. (2013). Systems and methods for online user profiling and segmentation.

[45] Bhuvaneswari, M. S., & Muneeswaran, K. (2020). User Community Detection From Web Server Log Using Between User Similarity Metric. International Journal of Computational Intelligence Systems.

[46] Albladi, S. M., & Weir, G. R. (2020). Predicting individuals' vulnerability to social engineering in social networks. Cybersecurity, 3(1), 1-19.

[47] Çokluk, G. (2019). Examining the predictive roles of cyber victimization, gender, revenge, and empathy on cyber bullying perpetration.

[48] Fabra, J., Álvarez, P., & Ezpeleta, J. (2020). Log-based session profiling and online behavioral prediction in E–Commerce websites. IEEE Access, 8, 171834-171850.

[49] Singh, A., & Sharma, A. (2019). A multi-agent framework for context-aware dynamic user profiling for web personalization. Software Engineering, 1-16.

[50] Singh, K., Singh, P., & Kumar, K. (2018). Fuzzy-based User Behavior Characterization to Detect HTTP-GET Flood Attacks. International Journal of Intelligent Systems and Applications, 10(4), 29.

[51] Najafabadi, M. M., Khoshgoftaar, T. M., Calvert, C., & Kemp, C. (2017, August). User behavior anomaly detection for application layer ddos attacks. In 2017 IEEE International Conference on Information Reuse and Integration (IRI) (pp. 154-161). IEEE.

[52] Anitha, V., & Isakki, P. (2016, January). A survey on predicting user behavior based on web server log files in a web usage mining. In 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) (pp. 1-4). IEEE.

[53] Grace, L. K., Maheswari, V., & Nagamalai, D. (2011). Analysis of web logs and web user in web mining. arXiv preprint arXiv:1101.5668.

[54] Goseva-Popstojanova, K., Anastasovski, G., & Pantev, R. (2012, July). Classification of malicious Web sessions. In 2012 21st International Conference on Computer Communications and Networks (ICCCN) (pp. 1-9). IEEE.

[55] Antunes, N., & Vieira, M. (2011, July). Enhancing penetration testing with attack signatures and interface monitoring for the detection of injection vulnerabilities in web services. In 2011 IEEE International Conference on Services Computing (pp. 104-111). IEEE.

[56] Hevner, A., and Chatterjee, S. Design Science Research in Information Systems. Springer US, Boston, MA, 2010, pp. 9–22.

[57] "IEEE Standard Glossary of Software Engineering Terminology," in IEEE Std 610.12-1990 , vol., no., pp.1-84, 31 Dec. 1990, doi: 10.1109/IEEESTD.1990.101064.

ÖZGEÇMİŞ

Baybarshan Ekiz

[redacted]