T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GÜVENLİ BULUT BİLİŞİM İÇİN SALDIRI TESPİT SİSTEMİ KULLANIMI

Fatma Didem ÖĞRETMEN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman

Prof. Dr. Ahmet SERTBAŞ

II. Danışman

Yrd. Doç. Dr. Muhammed Ali AYDIN

Temmuz, 2015

İSTANBUL

Bu çalışma 10/07/2015 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi:

İmza
Prof. Dr. Ahmet SERTBAŞ (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

İmza
Doç. Dr. Hakan DOĞAN
İstanbul Üniversitesi
Mühendislik Fakültesi

İmza
Yrd. Doç. Dr. Derya YILTAŞ KAPLAN
İstanbul Üniversitesi
Mühendislik Fakültesi

İmza
Yrd. Doç. Dr. Zeynep GÜRKAŞ AYDIN
İstanbul Üniversitesi
Mühendislik Fakültesi

İmza
Yrd. Doç. Dr. Akhan AKBULUT
İstanbul Kültür Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Bu çalışma İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yapılan "Güvenli Bulut Bilişim İçin Saldırı Tespit Sistemi Kullanımı" adlı yüksek lisans tez çalışmasını içermektedir.

Hayatımın her anında yanımda olarak bana sundukları maddi ve manevi desteklerinden ve sonsuz sevgilerinden ötürü minnettar olduğum değerli aileme en içten dileklerimle teşekkür ediyorum.

Tez çalışmam boyunca gösterdiği her türlü destek ve yardımlardan dolayı danışmanım ve değerli hocam Sn. Prof. Dr. Ahmet SERTBAŞ'a; çalışmamın her aşamasında engin bilgi ve anlayışıyla bana rehber olan, karşılaştığım bütün sorunlarda ilgiyle ve sabırla bana yol gösterip yardımcı olan ikinci danışmanımve değerli hocam Sn. Yrd.Doç.Dr. Muhammed Ali AYDIN'a sonsuz teşekkürlerimi sunarım.

Ayrıca yüksek lisans eğitimim boyunca bana sağlamış olduğu 2210 yurt içi yüksek lisans burs desteğinden dolayı TÜBİTAK'a özellikle teşekkür ediyorum.

Temmuz, 2015 Fatma Didem ÖĞRETMEN

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
BT	**:** Bilişim Teknolojileri
CPU	**:** Central Processing Unit
DMA	**:** Direct Memory Access
DoS	**:** Denial of Service
HIDS	**:** Host-Based Intrusion Detection System
HPC	**:** High Performance Computing
HTC	**:** High Throughput Computing
IAAS	**:** Infrastructure as a Service
IDS	**:** Intrusion Detection System
IOMMU	**:** Input/Output Memory Management Unit
KMD	**:** Kernel Monitor Daemon
NIDS	**:** Network-Based Intrusion Detection System
NIST	**:** National Institute of Standards and Technology
OS	**:** Operating System
OSSEC	**:** Open Source Security
PAAS	**:** Platform as a Service
PC	**:** Personal Computer
RAM	**:** Random Access Memory
RSS	**:** Really Simple Syndication
SAAS	**:** Software as a Service
SLA	**:** Service Level Agreement
SOA	**:** Service-oriented Architecture
SQL	**:** Structured Query Language
TPM	**:** Trusted Platform Module
VM	**:** Virtual Machine
VMM	**:** Virtual Machine Monitor
XML	**:** Extensible Markup Language

ÖZET

YÜKSEK LİSANS TEZİ

GÜVENLİ BULUT BİLİŞİM İÇİN SALDIRI TESPİT SİSTEMİ KULLANIMI

Fatma Didem ÖĞRETMEN

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ahmet SERTBAŞ

II. Danışman : Yrd. Doç. Dr. Muhammed Ali AYDIN

Bulut bilişim, bilişim kaynaklarından (sunucu, veri depolama, ağ, uygulama ve servisler, vb.) oluşan ortak bir havuza, uygun koşullarda ve istendiğinde hızlı bir şekilde zaman ve mekana bağlı olmaksızın erişim imkânıtanıyan bir modeldir. Teknolojinin hızla gelişmesiyle, bu hıza ayak uydurabilmek için tamamen internet tabanlı olarak geliştirilen bu modelin dünyada olduğu gibi ülkemizde de kullanımı giderek yaygınlaşmaktadır.

Bulut bilişim gelişmekte olan bir teknoloji olup,servis odaklı mimari (Service Oriented Architecture - SOA) ve web servisler, sanallaştırma (virtualization) ve grid bilişim gibi mevcut bir dizi teknolojinin bir araya getirilip farklı yollarla işletilmesiyle ile hayata geçebilmesi mümkün olmuştur. Özellikle sanallaştırma teknolojisi, bulut bilişim altyapısının temel taşlarından biri niteliğindedir.

Bulut bilişimin kullanıcılarına sunduğu başta ekonomik fayda olmak üzere birçok faydanın yanı sıra, bu yeni modelle yeni güvenlik sorunları da ortaya çıkmıştır.Bu yeni konumlandırma modelinin özellikleri geleneksel mimarilerden büyük ölçüde farklılık göstermektedir, bundan dolayı bilişim teknolojilerinde kullanılangeleneksel güvenlik mekanizmalarının etkinliği ve verimliliği bulut bilişim açısından yeniden değerlendirilmektedir. Bulut bilişimin maruz kaldığı güvenlik tehditlerine, açıklıklarına, yetersizliklerine karşı çeşitli güvenlik mekanizmaları geliştirilmesi üzerine çalışmalar yapılmaktadır. Bu tez çalışmasının temeli de bulut bilişimde güvenliğin sağlanması konusuna dayanmaktadır.

"*Güvenli Bulut Bilişim İçin Saldırı Tespit Sistemi Kullanılması*" olarak adlandırılan bu tez çalışmasında, bulut bilişimde sanallaştırma ve sanal makine güvenliğinin sağlanması için Sunucu-Tabanlı Saldırı Tespit Sistemi (Host Based Intrusion Detection System – HIDS) kullanılması ve böylelikle güvenli bulut bilişim sağlanması amaçlanmıştır. Çalışma kapsamında farklı güvenlik yaklaşımlarına dayanan beş ayrı güvenlik mekanizması oluştulmuştur. Bunlar; Sadece Ana Makinede OSSEC HIDS Kullanım Mekanizması, AdjointVM Koruma Mekanizması, İyileştirilmiş AdjointVM Koruma Mekanizması, Döngüsel Zincir VM Koruma Mekanizması ve Mesh VM Koruma Mekanizmasıdır.Oluşturulan test ortamında bu mekanizmaların bulut bilişim için önemli bir konu olan donanım kullanım verimliliği açısından karşılaştırmaları yapılmış ve elde edilen bulgular değerlendirilmiştir. Sonuç olarak bulut bilişimde güvenlik seviyesinin arttırılmasının donanım kullanım maliyetlerini de arttırdığı gözlenmiştir. Elde edilecek güvenlik kazanımlarının donanım maliyetlerine göre daha önemli olduğu göz önüne alındığında, hem güvenlik hem de maliyet açısından en uygunu olarak güvenli bulut bilişim içinDöngüsel Zincir VM Koruma Mekanizmasının kullanımının tercih edilmesi sonucuna varılmıştır.

Temmuz 2015, 129 sayfa.

Anahtar kelimeler: Bulut Bilişim, Güvenlik, Sanallaştırma, Saldırı Tespit Sistemi.

SUMMARY

M.Sc. THESIS

USING INTRUSION DETECTION SYSTEM FOR SECURE CLOUD COMPUTING

Fatma Didem ÖĞRETMEN

İstanbul University

Institute ofGraduate Studies inScience and Engineering

Department of Computer Engineering

Supervisor : Prof. Dr. Ahmet SERTBAŞ

Co-Supervisor : Asst. Prof. Dr. Muhammed Ali AYDIN

Cloud computing is a model that allows rapidly access to a shared pool of computing resources (servers, storage, network, applications and services, etc.) without time-dependent and location-dependent in appropriate conditions when requested. With the rapid development of technology, this model that is developed entirely based on internet to keep up the development speed is increasingly being used in our country as well as over the world.

Cloud computing is an emerging technology that has been possible to implement with bringing and operating togetherin different waysanumber of existing technologies such as service-oriented architecture (SOA) and web services, virtualization and grid computing. Especially virtualization technology is one of the cornerstones of cloud computing infrastructure.

Cloud computing provides number of benefits out of which economic benefit is main benefit. With the lots of benefits, this new model has emerged as new security problems. The features of this new deployment model shows greatly differ from traditional architecture, therefore efficiency and effectiveness of traditional security mechanisms used in computing technologies is re-evaluated in terms of cloud computing. There are many studies on improving a variety of security mechanisms for security threats, vulnerabilities and insufficiencies in cloud computing. The basis of this thesis is based on the issue of ensuring security of cloud computing.

This thesis study which is called "*Using Intrusion Detection System for Secure Cloud Computing*" is aimstouse host-based intrusion detection system (HIDS) for enabling security of virtualization and virtual machine in cloud computing and thus providing secure cloud computing. In this study, five different security mechanisms based on various security approaches are implemented. These mechanisms are the mechanism of using OSSEC HIDS on the domain only, AdjointVM Protection Mechanism, Improved AdjointVM Protection Mechanism,Circular Chain VM Protection Mechanism and Mesh VM Protection Mechanism. All of these mechanisms are tested and comparised in terms of hardware utilization efficiency that is an important issue for cloud computing, and the obtained results are evaluated. In conclusion, it is observed that increasing the level of security of cloud computing also increase hardware utilization costs. Given the security gains that will be achieved is more important than hardware utilization costs, the choice of using Circular Chain VM Protection Mechanism for secure cloud computing as the most appropriate mechanism in terms of both security and cost has been concluded.

July 2015,129 pages.

Keywords: Cloud Computing, Securtiy, Virtualization, Intrusion Detection System.

1. GİRİŞ

Bilişim teknolojileri, bilgi çağı olarak nitelendirilen modern çağın getirdiği yenilikler ve kullanıcıların hızla değişen ihtiyaçları nedeniyle kendi içinde bir değişim ve gelişim göstermektedir. Ofis-bağımsız kullanıcıların artması, ofislerin dinamik bir yapıya kavuşması, daha az kaynak ile daha fazla servis verme gerekliliği ve ekonomik nedenlerden dolayı kurumlarda az, ancak etkili kaynak kullanımı önem kazanmış ve tüm iş gruplarının kendi içlerinde yeniden yapılanması gerekliliği ortaya çıkmıştır. Bilişim sektöründeortaya çıkan bu ihtiyaçlara yönelik olarak fiziksel sistemlerinin sanal ortamlara taşınması ve bir fiziksel sistemin üzerinde birçok sanal sistem kullandırılması çözümleri geliştirilmiştir.Zamanla kullanıcıların uygulamalarını mekan, zaman ve platformdan bağımsız olarak kullanabilme yönündeki artan talepleri doğrultusunda "bulut bilişim (cloud computing)" kavramı ortaya çıkmış, sanallaştırmanın gelişmesi ve bilgi teknolojilerinde hızla yer edinmesinden sonra, sanallaştırma alt yapısının üzerine yeni bir bilişim teknolojisi olarak yapılandırılmıştır.

Bu tez çalışmasında detaylı olarakanlatılacak olan bulut bilişim, ölçeklenebilir bir ağ üzerinde veri ve işlem paylaşımına odaklı bir servis modelidir. Bu ağ üzerinde son kullanıcı cihazları, veri merkezleri ve web servisleri gibi bileşenler yer alır. Bulut bilişim altyapısındaki temel taşlardan biri sanallaştırma teknolojisidir.Böyleliklehem donanım hem de yazılım taraflı sanallaştırılmasayesinde kullanıcılara bir çok farklı platform üzerinde çalışabilen servislerin sağlanması mümkün olabilmektedir. Bulut bilişim teknolojisinin oluşmasındaki esas fikir, mevcut altyapıyı uygun servisler haline getirerek, bu servislere zaman ve mekâna bağlı olmaksızın internet üzerinden erişimi sağlamaktır. Ölçeklendirilebilen, istenildiği zaman kullanılabilen ve kullanıldığı kadar ücretlendirilen kaynak paylaşımı, kolay kullanım, yüksek servis kalitesi gibi bilişim teknolojilerinin tümünü içerisinde barındıran bir bilişim servisi olarak kullanıcıya sunulmuştur.

Bulut bilişim, kullanıcılara sunduğu ekonomik fayda başta olmak üzere birçok fayda sağlamaktadır. Bu faydalarının yanında bazı zorlukları da içermektedir. Güvenlik, bulut

bilişimde çözülmesi gereken en önemli zorluktur. Bulut bilişimde güvenlik sorunu kritik bir sorundur. Bu sorun nedeniyle kullanıcılar bulutları kullanmaya tereddüt etmektedirler.

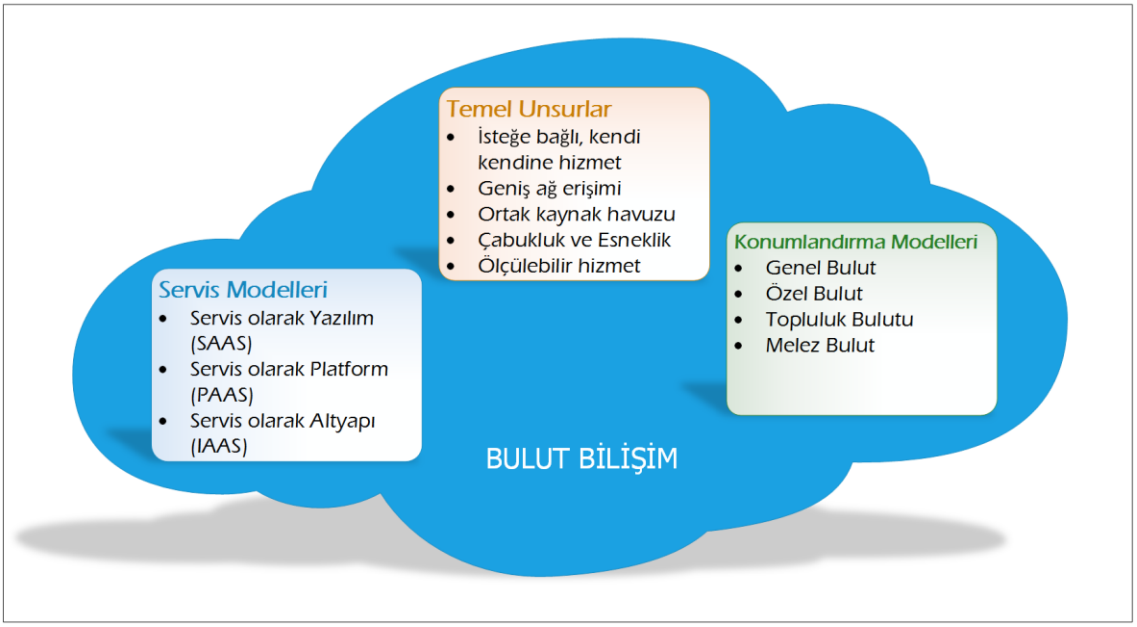
Bulut bilişimdoğası gereği dağıtık servise dayalı mimarisi, çoklu-kullanıcılar ve çoklu-domain altyapıları nedeniyle, tehditlere ve savunmasızlıklarakarşı dayanıksız olarak görülmektedir. Güvenlik sorunları, servisleri barındıran bulut sağlayıcılarını daha fazla ilgilendirmektedir. Antivirüs yazılımları, güvenlik duvarları, bekçi sistemleri ve saldırı tespit/engelleme sistemleri gibi mekanizmaları kullanılarak güvenlik sorunlarının üstesinden gelinmeye çalışılmaktadır.

Bu tez çalışmasında bulut bilişimde kritik konu olan güvenlik sorunlarının giderilmesi üzerine bulut bilişim alt yapısının dayandığı sanallaştırma yapısında sunucu-tabanlı saldırı tespit sistemi kullanımında çeşitli güvenlik yaklaşımlarına dayananbeş farklı modelin uygulanması ile oluşturulan güvenlik mekanizmaları gerçekleştirilmiş,bu mekanizmalardan elde edilen bulgularındonanım kullanım miktarlarına göre verimlilik açısından karşılaştırılması yapılmış ve sonuçlar tartışılmıştır.

2. GENEL KISIMLAR

2.1. BULUT BİLİŞİM

Bulut bilişim, kullanıcılara veriye daha az maliyetle ve daha hızlı bir şekilde ulaşma imkanı sağlayan, veri ve uygulamaları muhafaza etmek, işlemek ve kullanmak için internet ve merkezi uzak sunucuları kullanan servis tabanlı bir teknolojidir. Yeni nesil bilişim konularından olan bulut bilişimin literatürde bir çok tanımı olmasıyla birlikte, ilgili kaynaklarda sıklıkla atıf yapılan ve en çok benimsen Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology - NIST) [1] tarafından yapılan tanıma göre bulut bilişim, yapılandırılabilir bilişim kaynaklarından (bilgisayar ağları, sunucular, veri depolama, uygulamalar ve hizmetler vb.) oluşan ortak bir havuza, uygun koşullarda ve isteğe bağlı olarak her zaman, her yerden erişime imkân veren bir modeldir. Söz konusu kaynaklar asgari düzeyde yönetimsel çaba ve hizmet alıcı-hizmet sağlayıcı etkileşimi gerektirecek kolaylıkta tedarik edilebilmekte ve elden çıkarılabilmektedir [1].



Şekil 2.1: Bulut bilişimingenel özellikleri.

Bulut Bilişim kavramına tanım olarak Cliff [2] 'in yayınında da bulut bilişim şu şekilde anlatılmaktadır: "Bulut bilişim hem internet üzerinden hizmetler olarak sunulan

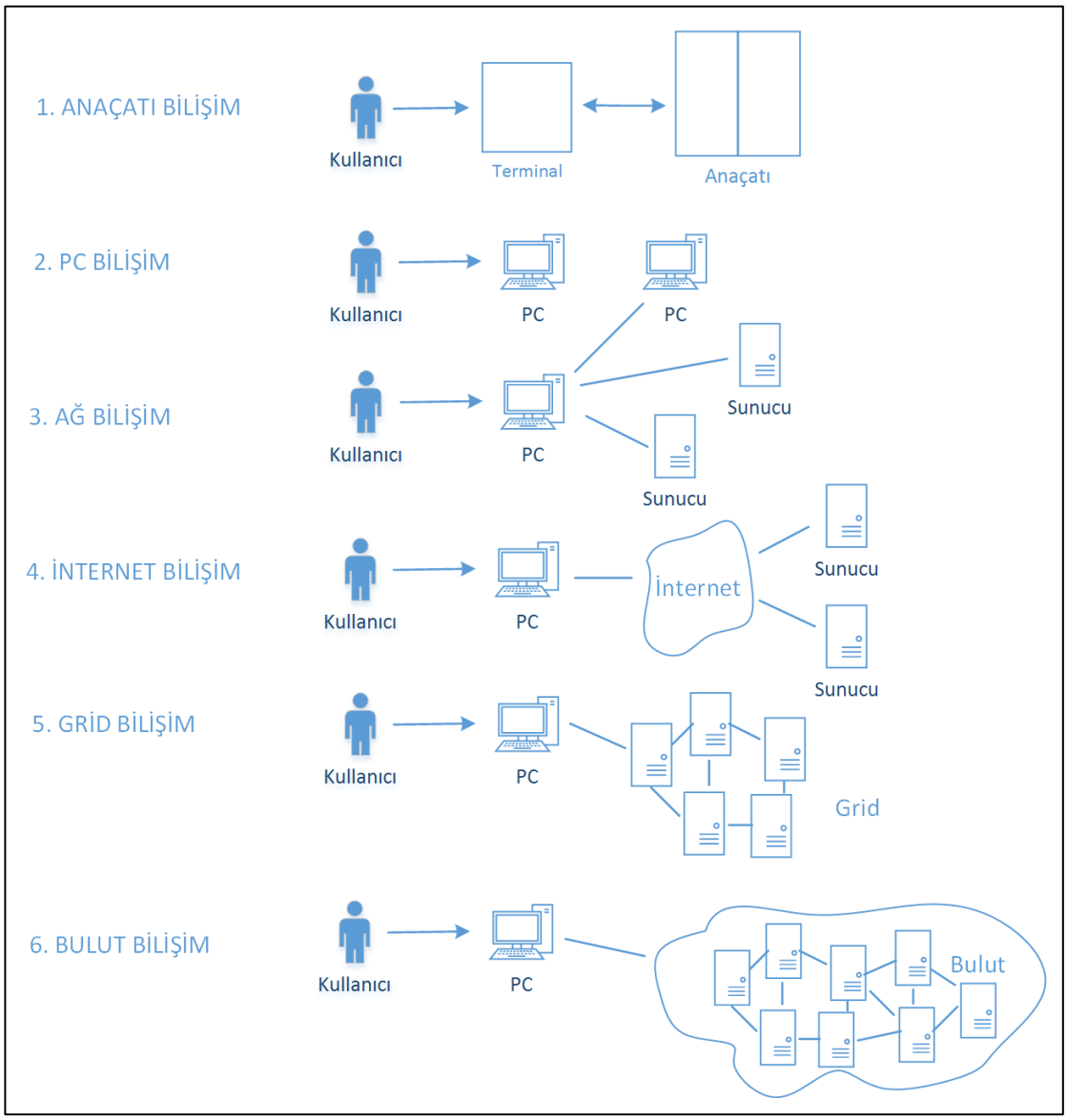
uygulamalara, hem de bu hizmetleri sağlayan veri merkezlerindeki donanım ve sistem yazılımlarına karşılık gelmektedir. Hizmetlerin kendisine uzun süreden beri servis olarak yazılım adı verilmiştir, bu yüzden bu terimin kullanılmasını tercih etmekteyiz. Veri merkezi donanım ve yazılımını ise bulut olarak nitelemekteyiz."

Bulut bilişim, "kullandığın kadar öde (pay per use)" yaklaşımına dayalı olarak işlemci ve depolama alanı gibi bilişim kaynaklarının gerektiği zamangerektiği kadarının kullanım bazlı olarak kiralanması prensibine dayanmaktadır. Altyapı ile uygulamaların birbirinden bağımsız olması ve verilerin tek bir merkezden kontrolü sayesinde her yerden veriye erişimin mümkün olduğu, gerektiğinde hızlı bir şekilde kapasite ölçeklendirmesi yapılabildiği, kaynak kullanımının kolaylıkla kontrol edilebildiği ve rapor ve log kayıtlarının alınabildiği bir altyapı sunmaktadır. Örnek vermek gerekirse elektrik şebekelerinden sağlanan elektriğin kullanılması gibi bilişim kaynakları da ihtiyaca göre istenildiği an kullanılmakta, kullanılmadığı zamanlarda isekaynaklar için herhangi bir ücret ödenmemektedir. Bulut bilişim sayesinde, kurumların kendi bünyelerinde bilişim kaynaklarına sahip olmaları yerine bu kaynakların kurum dışında barındırılması ve ihtiyaç olduğu anda kiralanabilmesi mümkün olmaktadır.

Bulut sağlayıcıları tarafından sağlanan servislerden bireysel kullanıcılar, küçük ya da orta ölçekli işletmeler, kamu ve özelkurumveya kuruluşlara kadar birçok farklı profilden kullanıcılar yararlanabilmektedir. Bulut servisleri internet tabanlıdır, böylelikle internete bağlanabilen herhangi bir cihaz vasıtasıyla sağlanan servislere erişim yapılabilmektedir. Bulut bilişim; kurumlar için verimerkezi, yazılım, donanım ve enerji maliyetlerinin düşürülerek hizmet sunulması, kullanıcılar için ise veriyePC, dizüstü ve tablet bilgisayarları ve akıllı telefonları her yerden hızlı ve kesintisiz ulaşabilmek anlamına gelmektedir.

Bilişim teknolojilerinin geçmişine bakıldığında büyük ebatlı bilgisayarlar olarakilk başlarda Anaçatı (Mainframe) bilgisayarlar kullanılmıştır. Eski bir teknoloji olan Anaçatı Hesaplama (mainframe computing) düşünüldüğünde, bulut bilişim alanındaki birçok yeniliğe öncülük ettiği söylenebilmektedir.Yedekli çalışma yeteneğine sahip olan bu büyük tekil sistemler, yüksek hesaplama hızı ile yüksek performans, güvenlik ve erişilebilirlik sağlıyorlardı. Ayrıca, anaçatı sistemler, sanallaştırma teknolojisinin de temelini oluşturmaktadır. 1960'lı yıllarda kullanılan anaçatı bilgisayarlardan, bulut

bilişime kadar gelişim süreci, Furht ve Escalante [3]'den uyarlanan Şekil 2.2'de gösterilmektedir.



Şekil 2.2: Bulut bilişimin gelişimi.

2.1.1. Bulut Bilişimin Sunduğu İmkanlar ve Temel Özellikleri

Bulut Bilişim; bir çok servis, platform ve altyapıyı bünyesinde barındırmaktadır.Bulut yapısında dinamik olarak sanallaştırılmış bilgisayarlar, birbiriyle ilişkili esnek uygulama ve servisler, hesaplama, depolama ve ağ kaynakları yer almaktadır. NIST tarafından verilen bulut bilişim tanımından yola çıkarak bir bulut bilişim platformu, temel olarak aşağıdaki karakteristik özellikleri taşıması gerekir [1]:

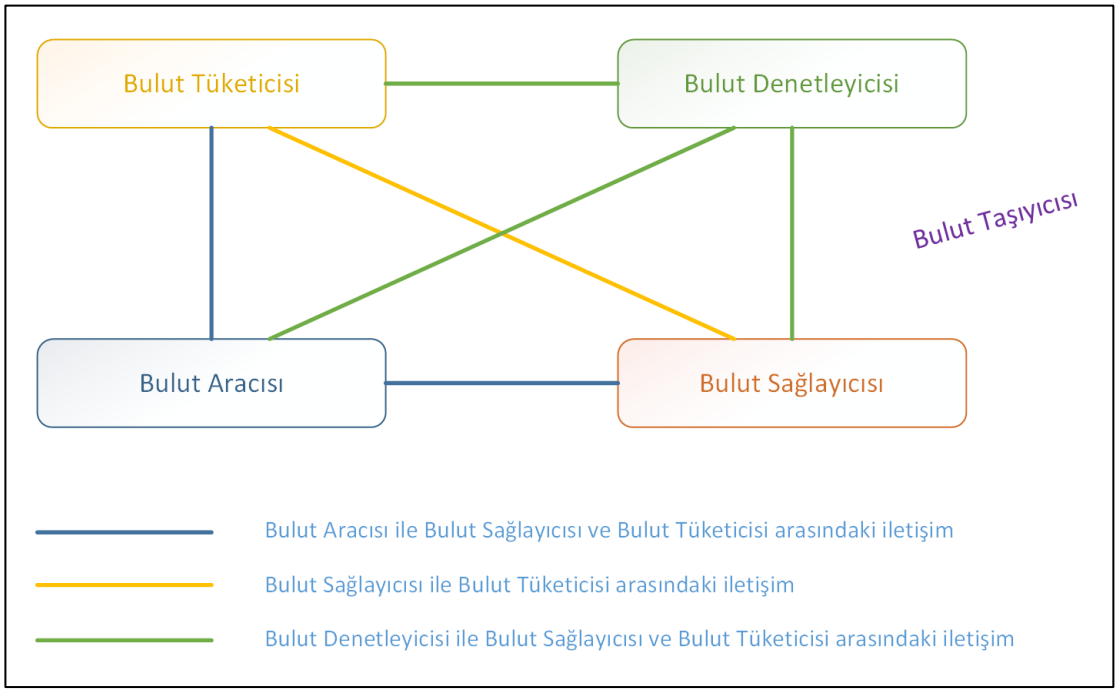
- **İhtiyaca göre kaynakların belirlenebilmesi:**Sunucu işlem zamanı ve depolama alanı gibi bilişim kapasiteleri ihtiyaç duyulduğu anda kullanıcıtarafından otomatik bir şekilde arttırılıp azaltılabilinir. Böylecekullanıcıların uygulamaları için ihtiyaç duyulan altyapının ne kadar güçlü olacağını, ne kadar kaynak kullanacağını önceden kesin bir şekilde analiz etmelerine gerek yoktur.
- **Geniş ağ erişimi:** Bulut bilişim kaynakları kullanıcılara, ağ üzerinden dizüstü bilgisayarlar, tabletler ve mobil telefonlar gibi farklı platformlardan erişilebilecek şekilde sunulur.
- **Ölçülebilir servis:** Bulut bilişimde kaynak kullanım miktarları, servisin türüne uygun bir soyutlama katmanında (depolama alanı, işlemci gücü, bant genişliği ve aktif kullanıcı hesabı sayısı gibi) bir ölçümleme mekanizması yardımıyla otomatik bir şekilde kontrol altında tutularak optimize edebilir. Hem kullanıcılar hem de bulut sağlayıcısı açısından kaynak kullanım miktarları şeffaf bir şekilde kontrol edilebilir, izlenebilir ve raporlanabilir.
- **Kaynak havuzu oluşturma (Çoklu-Kiralama):** Farklı yapıdaki bilişim kaynaklarından oluşturulan paylaşılabilir bir kaynak havuzundaki fiziksel ve sanal kaynaklar dinamik olarak çoklu kiralama (multi-tenancy) mimarisi ile bulut sağlayıcısı tarafından çok sayıda kullanıcıya sunulur. Çoklu kiralama mimarisi, çok sayıda kullanıcı tarafından aynı altyapının kullanılabilmesine olanak tanıyan bir mimaridir. Kullanıcının kendisine sağlanan kaynakların tam olarak nerede olduğuna ilişkin bilgisi ve kontrolü olmamakla beraber ülke, bölge veya veri merkezi bazında seçim yapabilmektedir.
- **Anlık esneklik:** Kullanıcı talebi veya sistem durumuyla orantılı olarak ihtiyaç duyulan kaynaklar için bilişim kapasitesianında esnek bir şekilde artırılabilir veya kapasiteazaltılabilir.

Ayrıca, Höfer ve Karagiannis [4] tarafından bulut bilişimin geleneksel bilişim yaklaşımıyla karşılaştırıldığında aşağıdaki özelliklerin bulunduğu belirtilmiştir:

- Uygulama ve altyapıların servis olarak sunulması.
- Paylaşımlı ve çok kullanıcılı olması.
- Kaynaklar için "kullandıkça öde" yaklaşımını benimsemesi.
- Kalite ve hizmet sürekliliği garantisi sunması.
- Esnek ve ölçeklenebilir bir yapı sunması.

2.1.2. Bulut Bilişim Aktörleri

NIST tarafından tanımlanan bulut bilişim mimarisinde beş temel aktör tanımlanmaktadır [5].Bunlar; bulut tüketicisi, bulut sağlayıcısı, bulut denetleyicisi, bulut aracısı ve bulut taşıyıcısıdır. Her aktör (bir kişi veya bir organizasyon) bulut bilişim içerisindeki bir işlem veya sürece katılan veya bir görevi gerçekleştiren bir varlıktır. Şekil 2.3'de NIST tanımına göre aktörler arasındaki etkileşim gösterilmektedir.



Şekil 2.3: Bulut bilişim aktörleri.

2.1.2.1. Bulut Tüketicisi (Cloud Consumer)

Bulut tüketicisi, bulut bilişim hizmetinin desteklediği nihai paydaşdır. Bulut tüketicisi, bulut sağlayıcısı ile iş ilişkisi kuran veya bulut sağlayıcısı tarafından sağlanan bulut servisini kullanan bir şahıs veya organizasyonu temsil etmektedir. Bulut sağlayıcısının sunduğu servisler arasından ihtiyaçlarına göre tercih ettiği uygun servisi talep eder, sağlayıcı ile servis sözleşmesini yapar ve servisi kullanır. Kullandığı bu servisler, servis sağlayıcı ile yapmış olduğu sözleşme dâhilinde ücretlendirilmektedir. Servis ihtiyaçlarına bağlı olarak, kullanım senaryolarının farklılıkları Tablo 2.1'de gösterilmiştir [5].

Tablo 2.1: Bulut Tüketicisi ve Bulut Sağlayıcısı.

Tür	Tüketici Faaliyetleri	Sağlayıcı Faaliyetleri
SAAS	İş süreç operasyonları için uygulamaları/servisleri kullanır.	Bulut altyapısına yazılım uygulamasının kurulumunu, yönetimini, bakım ve desteğini yapar.
PAAS	Bir bulut ortamında barındırılan uygulamaları geliştirir, test eder, konumlandırır ve yönetir.	Platform tüketicileri için bulut altyapısı ve ara katmanını hazırlar ve yönetir; platform tüketicilerine geliştirme, konumlandırma ve yönetim araçları sunar.
IAAS	BT altyapı operasyonları için servisleri oluşturur, kurulumunu yapar, yönetir ve izler.	IAAS tüketicileri için fiziksel işlemleri, depolama, ağ kaynaklarını, barındırma ortamını ve bulut altyapısını hazırlar ve yönetir.

2.1.2.2.Bulut Sağlayıcısı (Cloud Provider)

Bulut sağlayıcısı, bir şahıs, organizasyon veya tüketicilerin kullandığı servisleri geliştiren bir varlık olabilmektedir. İhtiyaç duyulan yazılım/platform/altyapı servislerini yöneterek, servislerin sürekliliği için gerekli olan teknik altyapıyı oluşturmakta, güvenlik ve gizlilik başta olmak üzere belirlenen servis seviyelerini karşılamaktadır. Sağlayıcılar Tablo 2.1.'de belirtildiği gibi, farklı servis modelleri için farklı görevler üstlenebilmektedir.

2.1.2.3.Bulut Denetleyicisi (Cloud Auditor)

Bulut denetleyicisi, bağımsız bir şekilde bulut servislerine, bilgi sistemi işlemleriyle performans ve güvenlik uygulamalarına bağlanabilen üçüncü parti kişi veya organizasyonlar olup, servis sağlayıcısını güvenlik kontrolleri, gizlilik uygulamaları ve performans gibi ölçütler kapsamında denetlemektedir.

2.1.2.4.Bulut Aracısı (Cloud Broker)

Bulut bilişim geliştikçe bulut servislerinin entegrasyonu bulut tüketicilerinin yönetimi için çok karmaşık hale gelebilmektedir. Tüketici bulut servislerini, doğrudan bulut sağlayıcısıyla iletişime geçerek değil, bir bulut aracısının yardımıyla kullanabilmektedir. Bulut aracısı, bulut servislerinin tüketici ile bulut sağlayıcısı arasındaki ilişkilerini düzenleyen, performans ve kullanılabilirliği yöneten bir varlıktır.

2.1.2.5.Bulut Taşıyıcısı (Cloud Carrier)

Bulut taşıyıcısı, bulut servislerinin bulut sağlayıcılar ile bulut tüketicileri arasındaki bağlantıları ve aktarımlarını gerçekleştirmekte olup, tüketicilerin servislere ağ,

haberleşme altyapıları ve diğer cihazlar üzerinden erişimlerini sağlamaktadır. Tüketiciler bulut servislerine masaüstü veya dizüstü bilgisayarlar ile mobil cihazlar gibi ağ erişim aygıtları üzerinden erişebilmektedir. Bulut servislerinin dağıtımı normal olarak ağ ve haberleşme altyapısı veya yüksek kapasiteli depolama imkânlarının kullanıldığı fiziksel iletişim ortamlarına sahip iletişim ajanları aracılığıyla gerçekleştirilebilmektedir.

Bulut sağlayıcısının, bulut taşıyıcısıyla servislerin istikrarlı bir şekilde iletimine yönelik olarak SLA hazırlaması gerekmekte, bulut taşıyıcıları da bulut sağlayıcısına ayrılmış bir altyapı ile bulut sağlayıcısı ve bulut tüketicisi arasında şifreli bağlantıları gerçekleyebilmelidir.

2.1.3. Bulut Bilişim Servis Modelleri

Bulut Bilişimde, farklı servis modelleriyle son kullanıcılar, uygulama geliştiriciler ya da sistem yöneticileri gibi farklı kullanıcılara hitap eden farklı servisler sunulmaktadır. Bu servisleri SaaS, PaaS ve IaaS olmak üzere üç temel servis sunum modelinden oluşmaktadır [1]. Şekil 2.4'tebulut bilişim servis modellerini içeren fiziki altyapıdan uygulamalara kadar bulut yığınının katmanlı yapısını göstermektedir.

Şekil 2.4: Bulut servis modelleri.

2.1.3.1. Servis olarak Yazılım (Software as a Service - SaaS)

Servis olarak Yazılım (SaaS), sunucu üzerinde bulundurulan yazılım uygulamalarının bulut sağlayıcısı tarafından birden fazla kişi veya kuruluşun kullanımına sunulmasıdır. Bu servis modeli sayesindekullanıcılaraynı kullanım şekliyle lisanslama ve yazılım maliyetlerinin azalması sayesinde daha düşük maliyetle kullandıkları kadar ücret ödeyerek bulut bilişim servislerine sahip olmaktadırlar. Kullanıcı yazılımınkurulumu, lisansı ve bakımı gibi sorunlarıyla uğraşmamakta ve böylelikle bu işler için gerekli maliyet ve kaybedilen zaman da kendiliğinden ortadan kalkmaktadır. Kullanılan uygulamalara internete bağlı herhangi bir bilgisayar veya mobil cihazlaerişebilmekte ve kullanılabilmektedir. Böylece uygulamanın güncelleştirilmesi, bakımı ve erişilebilirliği (high-availability) kesintisiz olarak sağlanabilmektedir. Kurum bünyesindeki veri merkezi yerine bu servisi sunan bulut sağlayıcılarının veri merkezinde bulunan uygulamalar ile daha güvenli ve daha hızlı bir yapı oluşturulabilmektedir.

2.1.3.2. Servis olarak Platform (Platform as a Service - PaaS)

Servis olarak Platform (PaaS) modeli, bulut sağlayıcısı tarafından kullanıcılarauygulama geliştirme – çalıştırma platformu ile gerekli tamamlayıcı servislerin ve teknolojik altyapının sunulmasıdır. Kullanıcınınplatform altyapısını oluşturan bileşenler üzerinde kendi kurduğu uygulamalar dışında herhangi bir yönetim ve kontrololanağı yoktur. Kullanıcılar, uygulama geliştirmek için gerekli işletim sistemi, platform ve araçları bu servisi sunan bulut servis sağlayıcılardan alabilmektedir. Bu sayede kullanıcılar ihtiyaç duyulan işletim sistemi, uygulama ve donanımların temin edilmesi için uğraşmadan doğrudan uygulama geliştirmeye başlayabilmektedirler.

2.1.3.3. Servis olarak Altyapı (Infrastructure as a Service - IaaS)

Servis olarak Altyapı (IaaS) modeli, ihtiyaç duyulandonanım, ağ ekipmanları ve depolama birimleri gibi temel kaynaklarınkullanıcı tarafından yapılandırılabilmesi ve bu altyapı üzerindegerekli olan işletim sistemi, araç ve uygulamaların kurabilabilmesidir. Kullanıcının alt yapı üzerinde doğrudan kontrol ve yönetimi bulunmamakta,sisteme işletim sistemi seviyesinde tam bir hâkimiyet ve kontrolü bulunmaktadır. IaaS sağlayıcıları fiziksel ve sanal mimariler yoluylaservislerini sunmakta,sanallaştırma teknolojisi ile kendi bünyelerindeki fiziksel sunucuların üzerlerinde sanal mimari koşturmaktadırlar. Bu sayede birbirinden izole edilmiş sanal sunucu makineleri

oluşturularak Iaas sağlayıcıları sahip oldukları fiziksel sunuculardan çok daha fazla sayıda sanal sunucuyu kullanıcılar için servis olarak verebilmektedir.

IaaS modelinde en büyük avantaj olarak özellikle büyük ölçekli kurum veya kuruluşların donanım yatırımı yapmalarına gerek olmaması görülebilir. Ölçeğe göre ücretlendirme ve esneklik sağlanması, yeni teknolojiler için sürekli mevcut altyapıya yönelik yatırım yapma zorunluluğun da ortadan kalkması, kurumlar için elde edilecek büyük kazanımlardır. IaaS modeli ile yönetimin basitleştirilmiş olması ve dinamik bir yapıda faaliyet göstermesi de bu kazanımları fazlasıyla arttırmaktadır [6].

2.1.4. Bulut Bilişim Konumlandırma Modelleri

2.1.4.1. Özel Bulut (Private Cloud)

Bir kurumun kendi bünyesindeki veri merkezleri üzerine kurulan bulut yapısıdır. Bu modelde kurumlar bulut alt yapısını kendi bünyelerindebarındırmakta ve kendi servislerini sunmaktadırlar. Özel bulut sayesinde kurumlar, kendi bilişim altyapılarını esneklikle ve daha az maliyetleişletebilmekte, kendi güvenlik politikalarını yönetebilmektedir. Veri gizliliği ve güvenliğininen üst seviyede tutulmasını gerektiren durumlarda tercih edilen geleneksel bir modeldir. Özel bulut, bir güvenlik duvarının ardında, herkese açık olmaksızın sadece kurum içi kullanıcılarıservis vermektedir.

2.1.4.2. Topluluk Bulutu (Community Cloud)

Birden fazla kurum veya organizasyon tarafından ortak bir alanda bulut servislerinin paylaşılmasıdır. Bu kurum ve kuruluşlar bir topluluk oluşturup, internet ortamından farklı olarak kendi veri merkezlerini ortak paylaşıma açarlar ve var olan servislerini birlikte kullanırlar. Böylece kaynaklarını daha az maliyetli bir çözüm ile paylaştırmaktadır.

Topluluk bulutları genellikle birbirleriyle ilişkilikurum veya kuruluşlar tarafından güvenlik ve yasal gereksinimler sebebiyle kullanılmaktadır. Bu sayede birbirlerinin kaynaklarına erişme ve bunlar üzerinde veri alışverişi yapabilme imkanı sağlamaktadırlar [6].

2.1.4.3. Genel Bulut (Public Cloud)

Standart olarak kabul edilen bulut bilişim modelidir. Uygulama ve altyapıservislerinin üçünü parti bir bulut sağlayıcısı tarafından kullanıcılara sunulmasıdır. Bütün servislerin birden çokbireysel kullanıcı, kurum veya kuruluşlar arasında internet üzerinden paylaştırılmasını ele almaktadır. Donanım, uygulama ve altyapı hizmetlerinin tamamı bulut servis sağlayıcıları tarafından sunulduğu için kurulumu kolay ve masrafsızdır. İhtiyaça göre ölçeklenebilir. Kullanılam miktara göre ücretlendirme yapılmaktadır.

2.1.4.4. Melez Bulut (Hybrid Cloud)

Bulut bilişim altyapısı diğer konumlandırma modellerinden en az ikisinin bir arada kullanılmasıyla oluşturulan yapılardır.İhtiyaç duyulan görevler, güvenlik gereksinimleri ile politikalar ve teknolojik imkânlar dâhilinde yük dağılımı gibi veri ve uygulama taşınabilirliğini gerçekleştirebilecek şekilde teşkil edilmektedir Bu bulut yapılarının en önemli sorunu, farklı mimarilerde farklı servislerin birlikte çalışabilirliğinin sağlanması, karmaşıklığın gizlenmesidir.Böyle yapılarda yüksek boyuttaki veri transferi ve senkronizasyon sıkıntıları ile karşılaşılabilmektedir [6].

2.1.5. Bulut Bilişimin Yararları

Bulut bilişim, beraberinde bazı uyarılarla birlikte kullanıcılara birçok yarar sağlamaktadır. Herhangi bir fiziksel sistemde olduğu gibi bulut hesaplama da belirlenen fiziksel sınır parametreleri dahilinde çalışmalıdır. Bulut bilişim büyük miktarlarda hesaplama gücü ve depolama olanağı sunmaktadır, ancak bu miktarlar sınırsız değildir. Bu nedenle bulut kullanıcıları bulut sağlayıcısı tarafından sunulan kaynak kullanım seçeneklerinden uygun olanı tercih edip uygulamalarını sığdırmak zorunda kalabilir.

Bulut hesaplama kaynakları talep üzerine büyütülüp küçültülebilmekte ve ölçülen kullanım miktarına göre ücretlendirilmektedir. ve aşağı talep üzerine ve ölçülü kullanım esasına göre ödenebilir. Bu yetenek, bulut tüketicileri için muazzam avantajlar sağlamaktadır.

Bulut Bilişim teknolojileri sayesinde uygulama geliştirme için gerekli maliyetler oldukça azalmaktadır. Sunucu, veri depolama, güvenlik donanımları gibi sahip olma maliyeti yüksek unsurların bulut sağlayıcılar tarafından sunulması ile ihtiyaca yönelik donanımlar çok hızlı bir şekilde temin edilebilmektedir. Bulut sağlayıcıları

tarafındanyazılımların ve işletim sisteminin sunulması ile, uygulama geliştiricilerin yalnızca gerçekleştirilecek projelere odaklanması sağlanabilmekte, böylelikle maliyet avantajı ve verimlilik kazanılmaktadır. Bulut sağlayıcılar tarafından yeni teknolojilerin takip edilip bulut sistemine sürekli adapte edilmesi, bulut kullanıcıları açısından oldukça yararlıdır.

Bulut bilişimin sunduğu yararlar aşağıda sıralanmaktadır.

2.1.5.1. Esneklik ve Dayanıklılık

Bulut bilişimin en önemli yararı esnekliktir. Farklı özelliklerde oluşturulan sistemlerin bir arada uyumlu bir şekilde çalışabilmesi ve ihtiyaca göre tasarlanması bulut bilişimin ayırıcı özelliklerindendir. Bulut bilişimin esneklik açısından sunduğu faydalar şunlardır:

- Sunucuların güncellenmesi endişelerden sıyrılması
- Yazılım güncellemelerinin kolaylıkla yapılabilmesi,
- Yeni servis ve teknolojilerin otomatik provizyonu
- İhtiyaç duyulan ölçüde kaynakların arttırılması
- Bakım detayları yerine yeniliğe odaklanma yeteneği
- Donanım bağımsızlığı

Dayanıklılık, çoklu yedekli kaynakların ve konumların mevcudiyeti ile elde edilir. Otonom hesaplama daha gelişmiş hale geldikçe, öz-yönetim ve kendi kendini iyileştirme mekanizmaları bulut kaynaklarının güvenilirlik ve sağlamlığında arttırım sağlayabilmektedir. Ayrıca, olağanüstü durum kurtarma ve iş sürekliliği planlaması bulut bilişim platformlarınındoğasında vardır.

2.1.5.2. Düşük Maliyet

Bulut bilişim, genel olarak maliyet tasarrufu için bir temel oluşturmaktadır. Kurum vekuruluşların kendi veri merkezlerini oluşturmadanihtiyaç duydukları servisleribulut sağlayıcılarından almaları, bilgisayar alt yapısı için büyük yatırımlar yapılmadan gerekli kaynakları kullanılabilmeleri sayesinde donanım, bakım, enerji ve personel gibi konularda maliyet azalmaktadır. Böylece donanım ve yazılım maliyetleri daha azyönetilebilir, ölçeklenebilir faaliyet giderleri ile değiştirilir.

2.1.5.3. Merkezi Veri Depolama

Bulut bilişim, kurumsal, yerel bilgi işlem merkezlerindeki mevcut veri depolama kaynaklarından daha büyük miktarlarda depolama sunmaktadır. Buna ek olarak, kullanılan bulut depolama kaynakları ilgili işletme maliyeti ayarlamaları ile istenilen miktarlarda azaltılabilir veya arttırılabilir.Bu şekilde depolama altyapılarının merkezileştirilmesi kurum ve kuruluşlara depolama ile ilgiliyazılım, barındırma ve eğitimli personel maliyet verimliliği sunmaktadır. Ayrıca, daha merkezi bir sistemde verilerin koruması ve izlenmesi, bir organizasyonun coğrafik olarak farklı yerlere dağıtılmışbirden çok birimindeki bilgisayar platformlarında tutulmasından çok daha kolaydır.

2.1.5.4. İndirgenmiş Dağıtım Süresi ve Güncelleme

Rekabetçi bir ortamda yeni yaklaşımların hızlı değerlendirilmesi ve geliştirilmesi kritik öneme sahiptir. Bu değişim ve gelişime ayak uydurarak bulut, kısa süre içinde güçlü hesaplama ve büyük miktarlarda depolama sağlayabilmekte ve böylelikle donanım, yazılım, personelde ilk yatırım maliyeti gerektirmeden gelişebilmektedir.Bu hızlı provizyon nispeten küçük bir ücret karşılığında gerçekleştirilmektedir ve bulut tüketicileri bulut sağlayıcı tarafından alınan ileri teknolojilere sürekli erişim sağlamaktadır.

2.1.5.5. Ölçeklenebilirlik

Bulut bilişimin alt yapısını oluşturan sanallaştırma teknolojileri, kullanıcılardan gelen taleplerinin istenilen ölçüde karşılanabilmesine olanak sağlamaktadır. Veri merkezlerinde kullanılan donanımlar kolaylıklagenişletilebilmekte, yeni donanımların yapıya kolayca adapte edilebilmektedir.

2.1.5.6. Cihaz, Konum ve Platformdan Bağımsızlık

Kullanıcıların, kullandıkları cihaz ve platformdan bağımsız olarak internet bağlantısının mevcut olduğu her konumda istedikleri zaman bulut servislerine erişmeleri mümkündür.

2.1.5.6. Performans

Bulut bilişimin kullanıcılarına sunduğu en önemli avantajlarından biri performanstır. Bulut bilişimin kolayca genişleyebilir altyapısı sayesinde tek bir bilgisayar ile beş yüz saatte tamamlanabilecek bir işlem, eğer çok sayıda bilgisayarın aynı anda çalışmasına

izin verecek bir işlem ise, beş yüz adet bilgisayarın tam kapasite ile çalıştırılmasıyla yaklaşık bir saatte tamamlanabilmektedir [7].

Bulut bilişimin önemli kullanım alanlarından biri yoğun işlemci gücü gerektiren uygulamalardır. Sanallaştırma yardımıyla bulut bilişim altyapısının çok sayıda işlemi aynı anda yüksek performans ile yürütebilmesi sağlanmaktadır.

2.1.6. Bulut Bilişimin Zorlukları

Bulut bilişimin bireysel ya da kurumsal olabilen kullanıcıları aşağıda bazı örnekleri verilen zorluklarla karşılaşılabilmektedir.

2.1.6.1. Mevcut Uygulamaların Bulut Altyapısına Taşınmasındaki Zorluklar

Bulut bilişim sağlayıcılarına göre farklılık göstermekle birlikte, bulut bilişim altyapısına aktarılan uygulamalarda önemli miktarda kodun yeniden yazılması gerektiği ifade edilmektedir [8]. Geleneksel mimariye göre geliştirilmiş bir yazılım dağıtık mimariye aktarılırken büyük oranda değişiklik yapılması gerekmektedir. Bununla beraber, grid bilişim gibi dağıtık bir mimari için geliştirilmiş bir yazılımın bulut bilişim altyapısına taşınmasının nispeten daha kolay olacaktır.

Bulut sağlayıcıları, programlama dillerindeki kimi fonksiyonel özellikleri güvenlik gerekçesiyle devre dışı bırakabilmektedir. Devre dışı bırakılmış bu özellikleri kullanan yazılımların, ilgili kısımlarının değiştirilmesi gerekecektir. Ayrıca programlama dillerindeki bu farklılıkları anlama ve dikkate alma sorumluluğu servis alıcısına bırakılmaktadır [9].

2.1.6.2. Güvenlik ve Gizlilik

Bulut bilişimi kullanan kurum ve kuruluşlar hala güvenlik konusunda endişe taşımaktadır. Bilgi ve kritik bilişim teknolojileri (BT) kaynaklarının kendi güvenlik duvarının dışında kalması, saldırılara karşı açıklık konusunda bulut tüketicilerini kaygılandırmaktadır. Kötü niyetli bulut sağlayıcıları veya sistem yöneticileri tarafından bu bilgilere erişildiğinde verilerin kötü amaçlarla kullanılması veya hiç kullanılamaması söz konusu olabilir. Bulut bilişimde güvenlik konusu Bölüm 2.3'te detaylı olarak anlatılmıştır.

2.1.6.3. Kontrol

Bulut bilişim sağlayıcılarının bulut platformları üzerinde tüm kontrole sahip olması kurum veya kuruluşların bazı BT birimlerini endişelendirmektedir. Bulut bilişim sağlayıcıları genellikle özel şirketler ve şirketlerin kendi iş uygulamaları için platformları tasarlamazlar.

2.1.6.4. Bantgenişliği Maliyetleri

Bulut bilişim ile kullanıcılar donanım ve yazılım maliyetlerinden tasarruf ederken bazen daha yüksek ağ bant genişliği ücretleriyle karşılaşabilmektedir. Bant genişliği maliyet verileri yoğun olmayan küçük internet tabanlı uygulamalar için düşük olabilir, ancak önemli ölçüde veri-yoğun uygulamalar için artmaktadır.

2.1.6.5. Güvenilirlik

Bulut bilişim altyapılarında sistem işleyişindeki aksaklıklar ya da veri kaybının oluşma olasılığıyedekli çalışan sistemler sayesinde asgari düzeye indirilmekte ve böylece devamlılık sağlanmaktadır. Ancak yine de her zaman tam güvenilirlilik sunmayabilir. Birkaç saatlik bulut servis kesintisin meydana geldiği yaşanmış vakalar bulunmaktadır [3].

2.2. BULUT BİLİŞİM TEKNOLOJİLERİ

Bulut bilişim, altyapsında mevcut bir dizi teknolojinin bir araya getirilip farklı yollarla işletilmesiyle oluşturulmuş yeni bir bilişim modelidir. Bulut bilişim sanallaştırma, yarar-bazlı ücretlendirme (utility –based pricing) gibi aslında yeni olmayan mevcut teknolojileri, günümüz bilgi teknolojileri talebinin teknolojik ve ekonomik gereksinimlerini karşılamak için güçlendirmektedir.

Bulut bilişim, altyapısında ilişkili olduğu bazı kavram ve teknolojilerle karıştırılabilmektedir. Bulut bilişimin dayandığı kavram ve teknolojiler bu bölümde kısaca açıklanmış ve bulut bilişimin farklı yönlerinden bahsedilmiştir.

2.2.1. Servis Odaklı Mimari

Servis odaklı mimari (Service-oriented Architecture (SOA)), uygulama bileşenlerinin bir iletişim protokolü aracılığıyla tipik bir ağ üzerinden diğer bileşenlere servis sağladığı bilgisayar yazılım tasarımında bir mimari modeldir. SOA esasen birbirleriyle iletişim kuran servisler topluluğudur. Servis, iyi tanımlanmış kendi kendine yeten bir işlevdir ve

diğer servislerin bağlamına ya da durumuna bağlı değildir. Düşük bağımlı, dağıtık ve programlı erişilebilen, yeniden kullanılabilir yazılım bileşenidir.Web Servis teknolojisi iseservis odaklı mimarilerin en olası bağlantı teknolojisidir. Bir web servis, XML tabanlı protokoller ve internet standartları vasıtasıyla erişilen servistir.

SOA, uygulamaya ait iş katmanlarının, farklı servisler üzerinden kullanılmasına olanak sağlamaktadır.SOA altyapısı, farklı yazılım uygulamalarının birbirleriyle veri alışverişinde bulunmalarını sağlar. Servis-yönelim ilkeleri herhangi bir satıcı, ürün ya da teknolojiden bağımsızdır. Servis yönelimi, servislerinprogramlama dilleri, işletim sistemleri ve gerekli diğer teknolojik bileşenlerle gevşek bir bağ oluşturmasını amaçlamaktadır.[10].

SOA, sistemin çok sayıda kullanıcıya sorunsuz bir şekilde servis verebilecek şekilde bilişim altyapısında tasarlanır. Sistemi oluşturan sunuculardan herhangi birinde sorun yaşanması durumunda sistem tarafından otomatik olarak yedek sunucular devreye alınır. Uygulama yazılımları için gerekli olan hafıza, sunucular için ortak hafıza biriminden sağlanır.

Bulut bilişim, prensipte hem mimari altyapı olarak, hem de servisin sunuş şekli açısından SOA ile benzer nitelikler taşımaktadır. GerçekleştirmedeSOA'yı içine alan daha geniş bir kapsama sahiptir. Bir SOA'nın bulut bilişim yaklaşımına uygun olabilmesi için kullanılan altyapı ve bu altyapı üzerinde çalışan uygulamaların bulut bilişim esnekliğine sahip olması gerekir.Yani, servis sağlayıcısının az sayıda kullanıcıya servis verdiği altyapıyı, altyapıda değişiklik yapmadan kolay ve hızlı bir şekilde daha çok kullanıcıya servis verebilmek için kullanılabilmesi ve gerektiğinde altyapının çok kısa zamanda yeni donanımlarla genişletilebilmesidir. Ayrıca çoklu kiralama mimarisine uygun olarak tasarlanmış olmasıdır.Çoklu kiralama mimarisine sahip olmayan bir altyapının, bulut bilişimin en önemli avantajlarından olan esneklik ve kaynaklardan tasarruf konularında beklenen faydayı sağlayamayacağı görülmektedir. [11].

2.2.2. İş Akışları (Workflows)

Servis odaklı faaliyetlerinbütünleşik bir görünümünün elde edilmesi iş akışıkavramı ile sağlanmaktadır.Bir iş akışı, gevşek ve sıkı bağlı (ve genellikle asenkron) işlem bileşenleri arasında bağlantı kuran veri akışlarını temsil eden yönlü bir grafiktir.

İş akışı destekli bir BT sistemi, bilgi-destekli problem çözmede ortaya çıkan bir dizi yapısal faaliyetler ve hesaplamaları temsil eder. İş akışları veritabanı ve bilgi sistemleri alanında araştırma-geliştirme topluluklarının oldukça dikkatini çekmiş, önemli araştırma alanlarından biri haline gelmiştir [12].

Bulut bilişimde iş akışı, bulut tarafından sağlanan servis tabanlı aktivitelerin bütünleşik bir görünümünü temsil etmektedir.

2.2.3. Web 2.0

Web 2.0, kullanıcılar arasında yenilik, bilgi paylaşımı ve işbirliğini geliştirmek için web teknolojisi ve web tasarım kullanımı anlamına gelen bir kavramdır.İnternet kullanıcıları tarafından paylaşarak ve ortaklaşa oluşturulan toplumsal iletişim siteleri, vikiler, iletişim araçları gibi ikinci nesil internet hizmetlerini içeren sistemi tanımlar.Web 2.0 katılımlı, interaktif ve paylaşımlı, yaygın olarak AJAX, widget, RSS, SOA, XML, Web Servis gibi teknojiler kullanılır. Öte yandan, Mashup tek bir entegre depolama aracına birden fazla kaynaktan gelen verileri birleştiren bir web uygulamasıdır. Her iki teknoloji bulut bilişimin özellikle servis olarak yazılım (SAAS) modelinde sunulan servisleriçin temel oluşturmaktadır.

2.2.4. Hizmet Bilişimi (Utility Computing)

Hizmet bilişimi (utility computing) yazılım, donanım, ağ bant genişliği gibi bilişim kaynaklarının ihtiyaç duyulduğunda kiralanması, tıpkı elektrik şebekelerinden sağlanan elektrik hizmeti gibi kiralanan kaynakların kullanım miktarlarının ölçülerek ücretlendirilmesi esasına dayanır. Bulut bilişim, hizmet bilişimin bir gerçeklemesi olarak algılanabilir. Ekonomik nedenlerden dolayı tamamen hizmet-bazlı bir fiyatlandırma planı benimsenir.Talep üzerine kaynak sağlama ve hizmet-bazlı ücretlendirme ile, bulut sağlayıcıları kaynak kullanımını en üst düzeye çıkarabilir ve işletme maliyetlerini en aza indirebilirir.

Bulut bilişimin hizmet bilişiminden en önemli farkı, belirli bir ağla sınırlı olmayıp, sunulan hizmete internet üzerinden erişimin mümkün olmasıdır. Hizmet bilişimi belli bir teknolojiden ziyade bir iş modeli olduğu söylenebilir. Bulut bilişim, hizmet bilişiminden daha geniş bir kavram olup sunulan hizmetin altyapısına ilişkin önemli ölçütler taşımaktadır.

2.2.5. Grid Bilişim (Grid Computing)

Grid Bilişim (Grid Computing), farklı yönetim birimleri tarafından sahip olunan bilgisayar kaynaklarının ortak bir amaç doğrultusunda bir veya birden fazla mantıksal havuzlar olarak organize edildiği bir platform olarak tanımlanabilir. Grid bilişim,heterojen, birbirine gevşek bağlı ve farklı coğrafi konumlarda bulunan birden çok altyapının beraber kullanıldığıdağıtık bir yapıdır. En basit şekliyle grid bilişim, gevşek yapıdaki ağlarla birbirine bağlı bilgisayarların yüksek bilişim gücü gerektiren işlemler için beraberce kullanıldığı süper sanal bilgisayar olarak nitelenebilir [13].

Grid bilişim, 1990'lı yılların başlarından itibaren bilgisayar bilimlerinde araştırma alanı olarak karşımıza çıkmaktadır. Grid bilişim projeleri ile elde edilen teknolojik gelişmeler, özellikle ağ erişimi, kaynak havuzu ve ölçeklenebilirlik ve esneklik gibi ortak özellikleri açısından bulut bilişim platformları ve mekanizmalarının çeşitli yönlerini etkilemiştir. Grid bilişim, akademik çalışmalardaki gereksinimler doğrultusunda ortaya çıkmış ve sonrasında iş alanlarındaki ihtiyaçları çözme amacıyla geliştirilmeye devam edilmiştir. Özellikle iş dünyasına yönelik grid bilişim sistemleri, bulut bilişimle büyük oranda örtüşmektedir.

Grid bilişim, bilişim kaynaklarının gerek duyulduğu zaman talep üzerine kullanılması ve sunucuların yerel olarak konumlandırılmalarının gerekli olmaması gibi özelliklere sahip olması yönünden bulut bilişimle benzerlik göstermektedir.Ancak günümüzde bulut bilişim kadar geniş kullanım alanı bulunmamaktadır. Grid bilişimde donanım arızası nedeniyle sistemin kesintiye uğrama olasılığı daha fazladır.Sebebi, sanallaştırma ile sunulan altyapı sağlamlığından yeteri kadar faydalanamamasıdır [14]. Grid bilişime göre bulut bilişim daha etkin dağıtık bir mimariye sahiptir. Bu açıdan bakıldığındagrid bilişimin teknoloji gelişim sürecinde bulut bilişimin bir önceki evresi olduğu söylenebilir [8].

Teknik altyapı açısından bulut bilişim altyapısı çok sayıda düşük kapasiteli sunucuyla oluşturulurken grid bilişim,HPC (High Performance Computing) mimariden HTC(High Throughput Computing) mimariye kadar geniş bir aralığa hizmet verebilmek için yüksek performansa sahip güçlü sunucuların bir araya gelmesiyle oluşturulmaktadır. Bu nedenle bulut bilişimde başlangıç altyapı maliyetleri düşük seviyede tutularak altyapı ihtiyacına göre ihtiyaç duyulan kaynaklarla kapasite kolaylıkla arttırılabilmektedir [15].

2.2.6. Sanallaştırma (Virtualization)

Sanallaştırma, bulut bilişimin gelişiminde önemli bir teknoloji olanağıdır. Sanallaştırma, donanım ve işletim sistemi arasında yer alan ve üzerinde uygulamaların çalıştırıldığı bir yazılım soyutlama katmanıdır. Genel anlamda bilgisayar kaynaklarının kullanıcılardan soyutlanması olarak tanımlanabilir. Soyutlamanın gerçekleştirilmesi kaynakların paylaşıması veya birleştirilmesiyle yapılmaktadır. Bu soyutlama katmanı Sanal Makine Denetleyicisi (Virtual Machine Monitor - VMM) veya hipervizör (hypervisor) olarak adlandırılmakta ve temelde bilgisayar sisteminin fiziksel kaynaklarını işletim sisteminden gizlemektedir. Donanım kaynaklarının işletim sistemi yerine doğrudan VMM tarafından kontrol edilmesi sayesinde birden fazla (çoğunlukla farklı türde) işletim sisteminin aynı donanım üzerinde paralel olarak çalıştırılması mümkün olmaktadır.

Sanallaştırma, 1960'lı yıllarda IBM şirketininanaçatı (mainframe) sistemlerinde "Zaman Paylaşımı" fikrini ortaya çıkardığı ve büyük bir anaçatı bilgisayarı birkaç mantıksal örneğine ayırması amacıyla geliştirdiği günden bu yana bilişim dünyasında yer almaktadır [16]. İlk çıktığı günden bu yana sanallaştırma kavramı önemli ölçüde olgunlaşmış ve hafıza, depolama, işlemciler, yazılım, ağ servisleri gibi bilişim teknolojilerinin tüm yönlerine uygulanmıştır. Temelinde fiziksel bilgisayar sistemlerinin sanal kopyalarının oluşturulması yatmaktadır. Sanallaştırma teknolojisi, barındırdığı alt kavram ve teknolojiler sayesinde işlemci (CPU), hafıza (RAM), depolama ve ağ bağdaştırıcıları gibi mevcut fiziksel kaynakların mantıksal olarak bölümlere ayrılmasını ve her mantıksal bölümün farklı bir bilgisayar sistemi gibi davranmasınısağlamaktadır.Doğrudan donanım platformuna erişimi bulanan, VMM veya hypervisor adındaki ana yazılım katmanı ile bir bilgisayar ortamını simule eden sanal bir makine (virtual machine – VM) oluşturulur ve bu sanal makine içerisinde

"misafir (guest)" yazılımlar çalıştırabilir. Bu misafir yazılım genellikle bir işletim sistemidir ve tıpkı fiziksel bir bilgisayar üzerindeymiş gibi çalışmaktadır.

Sanallaştırma teknolojisinin sunduğu iki önemli kavram vardır:

- **Kaynak Paylaşımı** – Tüm kaynakların çalışan programlar için atandığı sanallaştırılmamış ortamın aksine sanallaştırılmış ortamda hostun arka planındaki hafıza, disk ve ağ cihazları gibi fiziksel kaynaklar sanal makineler tarafından paylaşılır.
- **İzolasyon** – Aynı fiziksel donanım üzerinde çalışan sanal makineler arasında izolasyon sağlanır. Bir sanal makinede çalışan programlar başka bir sanal makine üzerinde çalışan programları göremez.

Sanallaştırma, fiziksel sınırlamaların ortadan kalkmasının sağlanması, tek bir merkezden birden çok sunucunun yönetilebilmesi ile yönetim yükünün en azaindirgenmesi, alt yapı maliyetlerinin büyük ölçüdeazaltılması, fiziksel sunuculara oranla yeni sunucuların kullanıma alınması işleminin oldukça kısa zaman alması, aynı makine üzerinde birbirinden farklı birden fazla işletim sisteminin yürütülebilmesi gibi birçok fayda sağlamaktadır ve bu sayede bilişim teknolojilerinde oldukça yaygın kullanım alanına sahip bir teknolojidir.

Genel olarak sanallaştırma mimarileri iki ana gruba ayrılmaktadır:

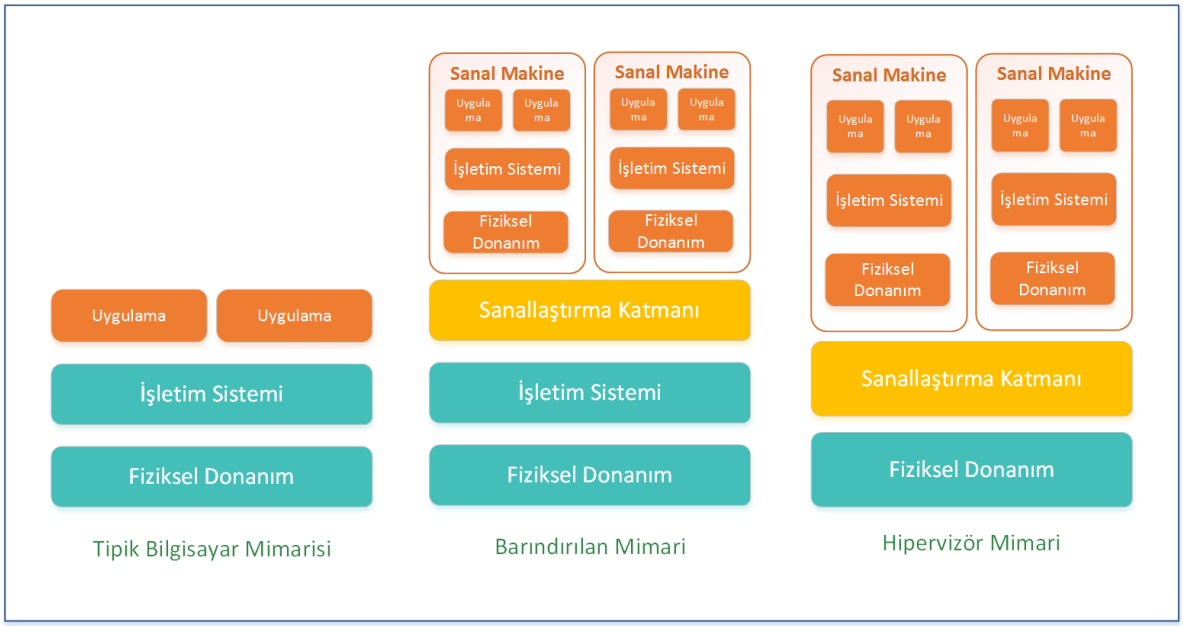
- **Barındırılan Mimari (Hosted Architecture):** Bu yaklaşımda, bir veya daha fazla VM'yi çalıştırmak için sanallaştırma platformu host OS içerisine yüklenmiştir.
- **Hipervizör Mimari (Hypervisor Architecture):** Bu yaklaşımda, sanal makine soyutlamasını sağlayan sanallaştırma katmanı donanımın üstünde yer almaktadır.

Şekil 2.5'te bu sanallaştırma mimarileri ve tipik bir bilgisayar mimarisi gösterilmektedir.

Bir sanallaştırma ortamında aşağıda yer alan bileşenler bulunmaktadır:

- **Hipervizör (Hypervisor)** – Sanallaştırmayı sağlayan yazılım katmanıdır. Sanal Makine Denetleyicisi (Virtual Machine Monitor – VMM) olarak da bilinen hipervizör, misafir sanal makinelerin üzerinde işletileceği sanal ortamın oluşturulmasından sorumludur. Misafir sistemleri denetler ve kaynakların misafir sanal makinelere gerektiği şekilde tahsis edilmesini sağlar. Tip-1 ve Tip-2 olmak üzere iki çeşit hipervizör bulunur:

  1. Tip-1 Hipervizör : Native veya bare-metal hipervizör olarak bilinen bu hipervizör türü en düşük seviyeli hipervizördür, doğrudan host donanımı üzerinde çalışır.



Şekil 2.5: Tipik bir bilgisayar mimarisi ile sanallaştırma mimarilerinin karşılaştırılması.

Tüm disk, hafıza, işlemci gibi tüm kaynakların misafir sanal makinelerine tahsis edilmesinden ve sanal makinelerin kaynaklara erişiminin koordine edilmesinden ve yönetiminden sorumludur. Bazı hipervizörler, hipervizörün yönetim ve kontrol arayüzüne erişim sağlamak için Domain-0 veya Dom0 olarak adlandırılan ayrıcalıklı bir misafir sanal makineye sahiptir. Xen, Hyper-V ve VMWare ESX gibi sanallaştırma platformları bu tür hipervizörlere örnektir. Tip-1 hipervizör çoğunlukla sunucu sanallaştırma ortamlarında görülmektedir.

2. Tip-2 Hipervizör: Barındırılan (hosted) hipervizör olarak da bilinen tip-2 hipervizör mevcut host işletim sistemi üzerinde bir uygulama olarak çalışır.

Host işletim sistemi donanım ile arayüz oluşturulmasından sorumlu iken, tip-2 hipervizör her bir VM için gerekli fiziksel kaynakları öykünür ve host işletim sistemi açısından sadece bir uygulama olarak kabul edilir.

- **Misafir (Guest) veya Sanal Makine (Virtual Machine - VM)** – hipervizörün üstünde sanallaştırılan uygulama veya işletim sistemidir. Fiziksel makinenin sanallaştırılmış bir temsilidir. Her bir sanal makine (VM) işlemci, hafıza, ağ başdaştırıcısı, çıkarılabilir aygıtlar ve çevresel aygıtları taklit ederek ayrı bir bilgisayar gibi davranan kendine yeten bir operasyon ortamıdır. Aynı fiziksel makinede farklı işletim sistemli birkaç VM eş zamanlı olarak işletilebilmektedir, fakat her bir misafir işletim sistemi için hiperviör tarafından tek bir donanım sunumu vardır.

Bilişim sistemlerinde kullanılan sanallaştırma yöntemleri iseaşağıda açıklanmaktadır:

2.2.6.1. Tam Sanallaştırma (Full Virtualization)

Bu yaklaşımda, sanal makine denetleyicisi (VMM) ana (host) bilgisayarda işletim sisteminin üstünde genellikle kullanıcı alanında bir uygulama olarak çalışır.Bunun sonucu olarak sanal makinelerde (VMs) de uygulamalar ve misafir işletim sistemi (guest OS) VMM tarafından sağlanan sanal bir donanımın üstünde çalışır. Tam sanallaştırmada, misafir işletim sistemlerinin herhangi bir değişiklik yapmaksızın çalışması için izin veren sanal makine ortamının, altta yatan donanımın yeterli ölçüde temsilini sağladığı düşünülür. Bu tür kurulumlarda giriş/çıkış (I/O) aygıtları, VMM'deki fiziksel aygıtları taklit eden sanal makinelere tahsis edilir, sanal ortamdaki bu aygıtlarla etkileşim ya host OS sürücüsü ya da VM sürücüsü [17] tarafından gerçek fiziksel aygıtlara yönlendirilir.

Bu sanallaştırma yönteminin başlıca avantajı, kullanımının kolay olmasıdır. İşletim sistemine herhangi bir yazılım ürünü ekler gibi ortak kullanıcı tercih edilen sanallaştırma yazılımını yükleyebilir.

2.2.6.2. Donanım Sanallaştırma

Bu yaklaşım, yüksek sanal makine izolasyonu ve performansı sebebiyle piyasada yaygın olarak kullanılır. VMM, doğrudan donanım üzerinde çalışır ve misafir işletim sisteminin donanım kaynaklarına erişimini kontrol ve senkronize eder.

2.2.6.3. Yarı Sanallaştırma (Paravirtualizaton)

Tam sanallaştırmanın aksine bu yöntemde çalışan misafir işletim sistemi sanal ortamda işletilebilmek amacıyla modifiye edilmelidir. Yarı sanallaştırma, sunucu sanallaştırmanın bir alt kümesidir ve host donanımı ve modifiye misafir işletim sistemi arasında ince bir yazılım arayüzü sağlar. Bu teknolojinin dikkat çekici yanı, sanal makineler sanal bir ortamda çalıştırıldıklarının farkındadır. Yarı sanallaştırmanın temel özelliklerinden biri, sanal olmayan donanıma yakın bir performans elde edilmesini sağlayan VMM'in basit olmasıdır.Yarı sanallaştırılmış ortamda aygıt etkileşimi tam sanallaştırılmış ortamdakine oldukça benzer, bu ortamda sanal aygıtlar, hostun altında yatan fiziksel aygıt sürücülerine de güvenirler.

Velte ve ark. [18], yarı sanallaştırma konusunda bazı noktalara dikkat çekmişlerdir. İşletim sistemlerinin sanallaştırma altyapısında etkin çalışabilmesi için muhtemelen bazı değişikliklerin yapılmasını gerektirmesi, yarı sanallaştırmada esnekliğin azalmasına neden olmaktadır. Yeni işletim sisteminin bu tip sanallaştırmada kullanıbilir olması için muhtemelen biraz zamana ihtiyaç duyacağı anlamına gelmektedir. Ayrıca bu değiştirilmiş işletim sistemi, diğer sanallaştırılmış sistemler ve host OS üzerinde etkisi olan donanım altyapısı üzerinde daha fazla kontrole sahip olmasının güvenlik konusundaki etkisini arttırmaktadır.

2.2.6.4. İşletim Sistemi Seviyesi Sanallaştırma (OS-Layer Virtualization)

Bu yaklaşımda ayrı bir sanal makine denetleyicisi veya hipervisör katmanı bulunmasının yerine host işletim sisteminin kendisi, sanal makineler arasında donanım kaynaklarının paylaştırılması ve sunucuların birbirlerinden bağımsız olmalarının sağlanmasından sorumludur. Buradaki ayırt edici fark, sanal makinelerin aynı işletim sistemini çalıştırması gerektiğidir. İşletim sistemi seviyesi sanallaştırma, esneklik bakımından zayıf kalırken yerel hız performansı daha yüksektir. Ayrıca tek tip işletim sistemi kullanan mimarinin kontrolü, heterojen yapıdaki mimarilere göre daha kolaydır.

2.2.6.5. Uygulama Sanallaştırma (Application Virtualization)

Bu yöntem, uygulamaların gerçek anlamda istemciler üzerine kurulmadan çalıştırılabilmesine olanak verir. Kullanıcılar, bir sunucu uygulamasının, yerel olarak bilgisayar kaynaklarına tamamen yükleme karmaşıklığı olmaksızın uygulamayı kullanma yeteneğine sahiptirler. Böyle sanllaştırılmış uygulamalar sadece uygulamanın yürütülmesi için gerek duyulan kaynakları içeren küçük bir sanal ortamda çalıştırmak için tasarlanmıştır. Böylece uygulama sanallaştırmadaki her bir kullanıcı sanal olarak izole edilmiş bir uygulama ortamına sahip olur. Bu izole sanal ortam, uygulama ve host işletim sistemi arasında bir katman olarak davranmaktadır.

2.2.7. Sanallaştırma Platformları

2.2.7.1. Microsoft Hyper –V

Microsoft şirketininsunucu sanallaştırma konusundageliştirdiği Hyper-V x64 tabanlı işlemciye sahip sistemlerde çalışabilecek şekilde tasarlanmış hipervizör tabalı sanallaştırma platformudur.Hyper-V, sisteme yüklenebilmesi için herhangi bir işletim sistemine gerek duymaz.Doğrudan fiziksel donanımın üzerine kurulur ve tüm sanal işletim sistemleri bu yazılım üzerine kurulur.İşletim sistemi üzerinde çalışan Hyper-V rolü, Windows sunucularda yerleşik olan sanallaştırma teknolojisini kullanarak sanallaştırılmış bir bilgi işlem ortamı oluşturulmasına ve yönetilmesine olanak sağlamaktadır [19]. Hyper-V rolü için gerekli bileşenler olarak Windows hiper yöneticisi, Hyper-V Sanal Makine Yönetimi Hizmeti, sanallaştırma WMI sağlayıcısı, aynı zamanda sanal makine veri yolu (VMbus), sanallaştırma hizmet sağlayıcısı (VSP) ve sanal altyapı sürücüsü (VID) gibi diğer sanallaştırma bileşenleri yüklenmektedir. Donanım tarafında 320 mantıksal işlemci, 4 terabayt fiziksel bellek ve sanal bir makine üzerinde 1 TB bellek, 64 sanal işlemci ve bir kümede 4.000 sanal makine oluşturma ve yönetme kabiliyetine sahiptir.

2.2.7.2. VMWare

VMWare şirketi tarafından geliştirilen, sanal makine oluşturmak ve yönetmeyi sağlayan tip-1 hipervizördür. Bireysel kullanım için ücretsiz sunulmakta olan VMWare sanallaştırma platformu x64 tabanlı mimarilerde kullanılabilmektedir. 64 mantıksal işlem çekirdeği, 256 sanal CPU ve host başına 1 TB RAM miktarı ve yüksek birleştirme

(consodilation) oranı gibi özellikleri içermektedir [20]. VMWare ESX/ESXi/VSphere çözümleri kurumsal sanallaştırma ortamları için kullanılmaktadır.

2.2.7.3. VirtualBox

İlk olara Innotek şirketi tarafından geliştirilen bir ürün olan VirtualBox, 2008 yılında Sun şirketi, 2010 yılından bu yana da Oracle şirketi tarafından Oracle VM VirtualBox adıyla geliştirilmesine devam eden bir hipervizordur. Windows, Linux, Machintosh ve Solaris gibi işletim sistemlerinde çalışması, x86 ve x64 tabanlı sistemleri desteklemesi, kolay kullanımı, ayarlanabilir görüntü belleği, uzak masaüstü bağlantısı gibi yetenekleri ve özellikle açık kaynak yazılım olarak GNU General Public License Version 2(GPLv2) lisansına sahip olması nedeniyle kullanıcılar tarafından en çok tercih edilen sanallaştırma platformu olarak görülmektedir. [21,22]. VirtualBox, host altında çoklu misafir işletim sistemi çalıştırabilmektedir. Hostlar, bu misafirlerin her birini durdurabilmekte, yeniden devam ettirebilmekte ve yedekleme amacıyla her bir misafirin anlık görüntüsünü alabilmektedir.her sanal makine bağımsız olarak yapılandırılabilmekte ve yazılım benzetim modunda veya donanım destekli modda çalıştırılabilmektedir.

2.2.7.4. Xen

Xen, 2003 yılında Cambridge Üniversitesinde bir araştırma grubu tarafından geliştirilmiş yarı sanallaştırılmış tip-1 hipervizördür [23].Hem açık kaynak lisansı altında Xen topluluğu tarafından desteklenirken hem de 2007 yılında Citrix firması tarafından XenSource'un satın alınmasıyla Xen Server adıyla ürün olarak sunulmuştur.

İlk önceleri sadece modifiye edilmiş Linux misafir sanal makineleri çalıştırabilirken günümüzde qemu işlemci emulatörü [24] sayesinde binary yeniden yazımı veya Intel ve AMD'nin işlemcileriyle sunulan sanallaştırma yeteneklerinden donanım destekli mod kullanarak Microsoft Windows ve modifiye edilmemiş Linux veya BSD misafirleri çalıştırabilmektedir. Xen'in sanallaştırma ürünleri çoğunlukla anaçatı bilgisayarlarda ve sunucularda kullanılmaktadır. Qemu temelli harici aygıtların benzetimi, misafirler için giriş-çıkış sanallaştırmalarına olanak tanımaktadır. İş yükü birleştirmesi kazanımı için sanal makinelerin canlı göç ettirilmesi mümkün olabilmektedir.

Xen hipervizörü, sanallaştırma katmanının ince ve minimal olması düşüncesine dayanarak geliştirilmiştir. Bu tasarım fikrine dayanarak, asıl sanallaştırma işihipervizörün bir seviye üstüne devredilmesi gerekmektedtir. Bu nedenle sıradan VM'lerden daha fazla ayrıcalıklara sahip Domain 0 (Dom0) olarak adlandırılan özel bir VM yer almaktadır. Dom0, diğer VM'ler için aygıt benzetimini gerçekleştirme görevi için atanmıştır. Dom0, Xen hipervizör üzerinde çalışan ve yönetim araçlarını üzerinde bulunduran güvenli VM'dir. Dom0 hipervizörün boot aşamasında otomatik olarak çalışmaya başlamakta ve fiziksel donanıma doğrudan erişerek diskler, ağ bağdaştırıcısı gibi aygıtlarla iletişim kurmakta, diğer VM'ler için sanal aygıtların yönetilmesinden, karmaşık işlemlerin gerçekleştirilmesinden sorumlu olmaktadır.Ayrıca hipervizöre VM'lerin başlatılması ve durdurulması gibi komutların iletilmesini sağlayan bir platform görevi görmektedir. Xen terminolojisinde tüm VM'lere domain denilmektedir. Yönetimden sorumlu VM Dom0 iken, diğer tüm VM'ler de DomUs olarak adlandırılmaktadır.

Bulut bilişim ortamlarında ise sanallaştırma teknolojisi olarak Xen kullanıldığında, bir sunucunun büyük bulut altyapısının bir parçası olması için Dom0 ile altyapıya bağlantı sağlamakta, bir nevi kanca görevi görmektedir.

2.3. BULUT BİLİŞİMDE GÜVENLİK

Bulut bilişimin başta ekonomik fayda olmak üzere kullanıcılarına sunduğu faydalarının yanında bulut bilişimdeki en büyük sorun güvenliktir. Bu sorunlardan dolayı kullanıcılar bulutları kullanmada tereddüt etmişlerdir. D. Teneyuca'nın 2011' de yaptığı bir araştırmaya göre [26] bulut hesaplama kullanımının eksikliğinin %36 sı güvenlik endişelerindendir.

Bulut bilişimdeki güvenlik sorunlarına genel olarak iki açıdan yaklaşılmaktadır. Birincisi, bulut servis sağlayıcısının sağladığı servislerin güvenli olduğunu garanti edebilmesi ve müşterinin kimlik yönetimini başarabilmesi; ikincisi ise, müşterinin kullandığı servislerin yeteri kadar güvenli olduğundan emin olabilmesidir.

Bulut bilişimdeki güvenlik riskleri aşağıda verilmektedir.

2.3.1. Veri Koruması

Bulut bilişim, bulut tüketicileri ve sağlayıcıları için çeşitli veri koruma riskleri oluşturmaktadır. Hassas verilerin saldırılara maruz kalması ya da açık bırakılmasının yanındveri kaybıve veri kullanılamazlığı büyük riskler olarak sayılabilir. Bazı durumlarda, bulut tüketicisinin etkili bir şekilde (veri denetleyicisi rolünde) bulut sağlayıcısının veri işleme uygulamalarını kontrol etmesi zor olabilir bu nedenle veri yasal bir şekilde yönetildiğindenemin olunmalıdır. Bu sorun ilebirleşik bulut servisleri arasındakiveri transferlerinde daha fazla karşılaşılmaktadır.

Gizlilik (confidentiality), verinin belirli bir kullanıcıya ait olduğu anlamına gelir ve herhangi bir yetkisiz kullanım ile ilgili değildir. Sadece yetkilendirilmiş kullanıcı ve sistemlerin veriye erişebileceği anlamına gelir. Bulut sistemleri web doğasına sahiptir ve kaynak paylaşımı veri gizliliği riskini arttırmaktadır. Kullanıcı kimlik doğrulaması mutlaka yapılmalıdır.

Mahremiyet (privacy), kişisel bilginin açıklanması için kişinin arzusudur. Bulutlarda veri, dünyadaki herhangi yerde olabilen servis sağlayıcının sunucusunda depolanmıştır ve sunucunun yerleştirildiği ülkenin gizlilik yasaları ve gizlilik hakkını takip eder veya uyar. Gizlilik hakkı (mahremiyet) verinin silinmesi, fişlenmesi, veri çalıntısı, hatalı manipülasyonlar ve kullanıcı hesabının hacklenmesinden etkilenmiş olabilir. Bulut hesaplamada ya hesap hacklenerek ya da servis boyunca kötü niyetli üyelerle karşılaşır.

Bir tüketicinin, sağlayıcıdan servis alımını sonlandırdığında bulut kaynaklarındaki verilerinin silinmesi isteği, doğru bir şekilde verilerin temizlenmesiyle sonuçlanmayabilir. Yeterli veya zamanında verinin silinmesi mümkün olmayabilir. Verinin çok sayıda kopyası alınmış ve depolanmış olabilir fakat kullanılabilir durumda değildir. Yahut silinecek disk üzerinde diğer kullanıcıların verileri bulunuyor olabilir. Çoklu kiralama ve donanım kaynaklarının yeniden kullanımı durumu, kullanıcıya atanan özel kaynaklara göre daha fazla risk taşımaktadır.

2.3.2. Yönetim Yetersizliği

Yönetim, uygulama gelişimi için politikalar, prosedürler ve standartlar üzerine gözetim ve kontrol anlamına gelmektedir. Bulut bilişimdeki temel sorunlar risk yönetim uyumunu, etkili bilgi güvenliği yönetimini sürdürmek için organizasyonel yapı, süreç ve

kontrolleri tam olarak uygulanması ve kimlik doğrulama ile ilgilidir. Genel bulut dağıtımlarında tüketiciler mutlaka Genel bulutlarda tüketiciler, güvenliğini etkileyebilecek bir dizi konu üzerinde bulut sağlayıcısına kontrolünü devretmek zorundadır. Aynı zamanda, bulut servis seviyesi anlaşmaları (SLA) bulut sağlayıcısı adına bu tür özellikleri sağlamak için bir taahhüt oluşturduğundan güvenlik savunmalarında bazı boşlukları bırakabilmektedir [27].

Bulut bilişim servislerinin kullanımının tüketiciler ve bulut sağlayıcı organizasyonlar arasında yayılması göz önüne alındığında güvenlik yönlerinin sorumluluğu her iki tarafa da dağıtılmış olabilir. Sorumlulukların açıkça belirtilmesinde bir sorun varsa, savunma mekanizmasının hayati bir parçası korumasız bırakılmış demektir. Sorumlulukların tüketici ve sağlayıcı organizasyon arasında ayrılması kullanılan bulut bilişim modeline göre değişmektedir.

Algılama, raporlama ve güvenlik ihlallerine müteakip yönetim gibi konuları ele almaktan sorumlu olan sağlayıcılara güvenmek, tüketiciler için bir endişe kaynağıdır.

2.3.3. Yönetim Arayüzündeki Güvenlik Açığı

Genel bulut sağlayıcısının tüketici yönetim arayüzleri genellikle internet üzerinden erişilebilirdir. Bu arayüzler geleneksel barındırma sağlayıcılarınınkaynaklarında daha büyük kaynaklara erişim için aracılık yapar. Bu nedenle özellikle uzaktan erişim ve web tarayıcısı güvenlik riskleri ile birleştiğinde yüksek risk oluşturmaktadır.

2.3.4. Bulut Çalışanlarının Kötü Niyetli Davranışları

Bulut içerisinde verilen erişim izni ve yetkilerin kötü niyetli çalışanların kullanmasıyla bu eylemlerin neden olduğu hasarlar diğer risklere göre daha fazla olabilir. Bu durum, tüketici organizasyonunda, sağlayıcı kuruluşunda ya da her ikisi içinde de oluşabileceğinden bulut bilişim ortamı riski olarak genellenebilir. Bulut modeli, çoklu kiralama tabanlı bir modeldir ve bulut sağlayıcısının tekil yönetim alanı altındadır. Bulut çalışanlarının istihdam edilmesinde belirlenmiş standartlar olmayışı, üçüncü taraf bir satıcı kolaylıkla bulut verilerine kötü niyetli erişerek zarar vermesine veya başka organizasyonlara satmasına neden olabilir.

2.3.5. Kullanılabilirlik

Kullanılabilirlik, herhangi bir yerde herhangi bir anda kullanıcıların servisleri kullanabilmesini garantilemek anlamına gelmektedir. Altyapı, yazılım ve verinin kullanılabilirliği anlamındadır. SaaS, PaaS ve IaaS gibi tüm bulut bilişim sistemlerinde herhangi bir yerde herhangi bir anda kullanıcıların buluta erişebilmesine izin verilmelidir, bunu başarmak için, bulut servisler tüm zamanlarda kullanılabilir olmalıdır.

Servis kullanılamazlığı, tüketici sistemleri ve sağlayıcı servisleri arasındaki iletişim başarısızlıklarından, sağlayıcının veri merkezinde ekipman veya yazılım hataları gibi birçok faktörden kaynaklanabilir.

2.3.6. İzolasyon Başarısızlığı

Çoklu kiralama ve paylaşılan kaynaklar, genel bulut bilişim özelliklerini tanımlamaktadır. Bu risk kategorisi depolama, hafıza, yönlendirme kullanımlarını ayıran mekanizmalarındaki başarısızlığı kapsamaktadır.

2.3.7. Uyum ve Yasal Riskler

Bulut kaynaklarının coğrafik olarak bulundukları yerler sabit değildir. Kaynaklar, farklı faktörler ve sebeplerden dolayı fiziksel yerleşkeler arasında göç edebilir. Bu göç etmeden dolayı, birden çok yasal yargı kuralı altına geçebilir ve bu yargılar saldırı ve veri koruma gibi güvenlik meseleleri hakkında çelişkili kurallara sahip olabilir.

Göç kurallarını garantilemek için ülke yerleşim kısıtlamaları tanımlanmıştır ve SLA veya sözleşmede uygulanmış veya anılmıştır. SLA sadece servis sağlayıcısı ve müşteri arasındaki anlaşmadır. Sağlayıcılar müşterilere verilerinin güvenliğini, güvenilirliğini garantilemelidirler. Aralarında ortak anlayışa sahip olmalıdırlar. Kurallar sözleşmenin beklenmedik veya beklendik aktarımı ile ilgili SLA da ifade edilmiş olmalıdır. Yasal durumlar klasik saldırılar değildir, fakat güvenlik amaçlarını bozabilir.

2.4. BULUT BİLİŞİMDE GÜVENLİK MEKANİZMALARI

Bilişim dünyasında güvenlik tehditlerinin çeşitlerinin ve sayıısının hızla artmasıyla birlikte güvenlik mekanizmalarında da hızlı gelişim görülmektedir. Bilgisayarların güvenliğinin sağlanması,yetkisiz kullanıcıların sistemlere izinsiz erişip bilgileri ele geçirmelerini veya değiştirmelerini engellemek amacıyla ilk olarak kimlik doğrulama ve

erişim kontrolü gibi güvenlik mekanizmaları geliştirilmiştir. İnternet kullanımının yaygınlaşması ile birlikte bilişim sistemlerine karşı güvenlik tehditlerindeönemli artışlar ve saldırıtürlerinde genişlemeler olmuştur. Karşılaşılan tehditler ve saldırılar sebebiyle yeni mekanizmaların geliştirilmesi zorunluluğu ortaya çıkmıştır. Bilişim sistemlerinde güvenliğin sağlanması amacıyla güvenlik duvarları (firewall), güvenlik açığı tarayıcıları (vulnerability scanner) ve saldırı tespit sistemleri de kullanılmaktadır. Bu güvenlik mekanizmalarının hiçbirinin tek başına kullanılmasıgüvenlik açısındantam olarak yeterli görülmemektedir; çünkü her biri farklı açılardan güvenlik konularına odaklanmıştır. Sistemde güvenliğin sağlanması, bu mekanizmaların birbirini destekleyecek şekilde beraber kullanılmasını gerektirmektedir.

Bulut bilişim sistemleri de güvenliğin sağlanmasıamacıyla çeşitli yönetim modellerine odaklanmıştır [25]. Bu modellerde çoğunlukla Saldırı Tespit Sistemi, Güvenli Bilişim, Veri Şifreleme gibi mekanizmalar yer almaktadır.

2.4.1. Saldırı Tespit Sistemi (Intrusion Detection System – IDS)

Bir kaynağın veya veriningüvenilirliğini,bütünlüğünü, gizliliğini veya erişilebilirliğini engellemeye yönelik tüm eylemler **saldırı (intrusion)** olarak tanımlanmaktadır. Saldırı tespit sistemi (Intrusion Detection System – IDS), bir bilgisayar sistemi veya bilgisayar ağında meydana gelen olayların izlenmesini otomatik hale getiren, bu sistemlerde oluşan kötü niyetli faaliyetlerin ve bilgisayar güvenlik politikaları, kabul edilebilir kullanım politikaları veya standart güvenlik politikaları ihlallerinin analiz edilmesini ve yönetim birimine raporlanmasını sağlayan yazılım veya donanım sistemidir [28]. Kısaca IDS olarak ifade edilen saldırı tespit sistemi, bir tür alarm sistemi olarak düşünülebilir. Bir IDS, birkaç bileşenden oluşmaktadır [29]:

Algılayıcılar (Ajanlar):Güvenlik olaylarını oluşturur.

Monitör: Olayları ve uyarıları izlemek ve algılayıcıları kontrol etmek için kullanılır.

Merkezi Motor: Algılayıcılar tarafından günlüğe kaydedilen kayıtları bir veritabanında tutar ve bir güvenlik olayı alındığında uyarıları oluşturmak için bir kurallar sistemini kullanır.

Bir IDS, saldırı tespit edildiğinde karşılık olarak gerçek zamanlı olarak veya buna yakın cevap verebiliyorsa **aktif**, gelen saldırıyı kaydedip daha sonra incelemek için saklıyorsa **pasif** olarak nitelendirilir. Saldırıların tespit edilebilmesi tetikleme mekanizmaları ile gerçekleştirilir.Saldırı tespitinde sistem ve ağ kaynaklarının yanlış veyaolağan dışı kullanım bilgilerinin bilinmesi gerekir. Saldırı tespit sistemleri genel olarak Sunucu-Tabanlı IDS (Host-Based IDS)'ler ve Ağ-Tabanlı IDS (Network-Based IDS)'ler olmak üzere iki sınıfa ayrılmaktadır.

2.4.1.1. Sunucu-Tabanlı Saldırı TespitSistemi (Host-Based IDS - HIDS)

Sunucu-Tabanlı saldırı tespit sistemi (host-based IDS – HIDS) , asıl hedef sistemin dış etkileşimleri seyrek olan anaçatı bilgisayarların [30] olduğu, tasarlanan ilk saldırı tespit yazılım türüdür. HIDS, bağımsız bir bilgisayardan toplanan bilgilerle çalışır. HIDS, sadece bilgisayar sisteminden gelen ve giden paketleri izler ve şüpheli etkinlik 1 olarak tespit edilirse kullanıcı veya yönetici uyarır. Özellikle önemli sunucu sistemler üzerinde gizli ve kritik bilgileri korumak amacıyla kullanılmaktadır. HIDS'ler, belirli bir makinede meydana gelebilecek saldırıları önlemek üzere sunuculara ya da çalışma istasyonlarına yerleştirilmiş algılayıcılardan oluşmaktadır.HIDS sistem durumu izlemek için işletim sistemi denetim rotalarından ve sistem log kayıtlarından faydalanarak karar verebilmektedir. Hangi kaynakların hangi programlara eriştiğini algılayabilmektedir.

HIDS kullanımının sağladığı faydaların yanı sıra, bazı dezavantajları da vardır. Her bir sunucu için elde edilen bilginin yapılandırılması ve yönetilmesi gerekmektedir. Bu nedenle HIDS'in yönetimi zordur.HIDS'ler için bilgi kaynakları ve analiz motorlarının saldırıların hedefi olan sunucu bilgisayarda bulunması nedeniyleHIDS saldırılara maruz kalabilir ve saldırının bir parçası olarak devre dışı olabilir. Ayrıca, izleme yaparken bilgisayar kaynaklarını kullanmaktadır, bu nedenle izlenen sistemde bir performans maliyetine neden olmaktadır.

Diğer IDS'lerde olduğu gibi, HIDS'lerde de çeşitli tiplerde görevlendirilmiş varlıklar mevcuttur.

Algılayıcı veya ajan; sunucu, iş istasyonu ve uygulama hizmeti gibi bir sunucunun üzerine veya yakınına yerleştirilmektedir. Olay verisi kayıt altına alma hizmetine, olayları kaydetmek ve diğer muhtemel alakalı olaylarla bağlantı kurmak üzere

gönderilmektedir. HIDS ajanları sayısız sunucu tipine yerleşerek, buralarda çalışabilmektedir. HIDS algılayıcıları sunucuları, kullanıcı bilgisayarlarını ve uygulama sunucularını izleyebilmektedir.

Sunucu; genellikle, kullanıcıların web, e-posta veya FTP sunucularına veri göndermek veya almak üzere bağlandıkları, kullanıcılara belirtilen hizmetleri vermek üzere adanmış olarak çalışan bir bilgisayardır.

Kullanıcı makinesi; bir kullanıcının diğer makinelere bağlanmasını sağlayan masaüstü veya dizüstü bilgisayarı gibi bir iş istasyonudur.

Uygulama servisi; web hizmeti veya veritabanı uygulaması gibi sunucu üzerinde çalışan bir yazılımdır.

Her bir kullanıcı makinesi farklı işletim sistemi çalıştırmakta veya hizmet yürütmekte olduğu için, makinelere etki edecek saldırıların tipleri de söz konusu makinelere özgü olarak gerçekleşmektedir. HIDS algılayıcısı sadece ağ trafiğini değil, kullanıcı makinelerini de izlediği için ajan, kullanıcı makinelerinde yazılımın bir parçası olarak yer almaktadır. Mantıksal olarak ajan, varlık yani kullanıcı makinesi ve dış ağ arasında, bir ağ cihazı olmak yerine, trafiğin hizmet alabilmek için aşması gereken bir yazılım katmanı olarak görev yapmaktadır. HIDS, kullanıcıya özel olaylar; zararlı bir kodun çalıştırılması ve bellek taşması gibi kod analizlerini, bütünlük ve erişimi içerecek şekilde dosya sisteminin izlenmesini, kullanıcı kayıtları gözden geçirilirken meydana gelen kayıt analizlerini ve son olarak ağ ayarı yapılandırmalarındaki değişiklikleri kapsamaktadır.

2.4.1.2. Ağ-Tabanlı Saldırı Tespit Sistemi (Network-Based IDS –NIDS)

Ağ tabanlı saldırı tespit sistemleri (Network-Based IDS – NIDS), belirli bir sunucudan ziyade ağın kendisine odaklanır. Ağ üzerinden geçen trafiği veri kaynağı olarak görüntülemektedir. NIDS, ağ segmenti veya anahtarlama cihazı dinleyerek, bu ağ segmentine bağlı birden çok hostu etkileyen ağ trafiğini izleyebilmektedir. Genelde ağ kartı geçirgen (promiscuous) moddayken üzerinden geçen tüm trafiğin yakalaması sağlanır. Ağ paketleri yakalanarak ve analiz edilerek saldırılar algılanmaktadır.

NIDS'lerde temelde ağda dolaşan paketlerden,ağda belirli noktalarda yer alan algılayıcılar üzerinden geçen paketleriyle ilgilenilmektedir. Algılayıcıya gelen paket sistemdeki mevcut imzalarla karşılaştırılarak paketin analizi yapılır. Başlangıç düzeyindeki filtre hangi paketlerin kabul edilip hangilerinin atılacağını veya paketin saldırı tanıma modülüne gönderilip gönderilmeyeceğini belirlemektedir. Saldırı tespit edilirse cevap modülü saldırıya karşılık olarak alarm üretim mekanizmasını tetikler. Algılayıcı ile monitör arasındaki trafiğin şifrelenmesi veya algılayıcı ve monitörlerin ayrı bir ağa dahil edilmesi güvenlik açısından önemlidir. Bilgili ve deneyimli bir saldırgan için algılayıcı ve monitör arasındaki trafik (alarmlar, durum kayıtları, diğer paketler vb.) ağa saldırmak için çok önemli bilgiler içermektedir. Algılayıcı ve monitörlerinfarklı bir ağda yer almasıylaServis Reddi (Denial of Service - DoS) saldırılarından korunmak mümkün olmaktadır.

NIDS'ler araç ve yazılım olmak üzere iki bileşenden meydana gelmektedir. Araç işletim sistemi, uygulama ve ağ kartından oluşmaktadır. Yazılım ise, IDS yazılımı ve bazen de kullanıcı tarafından kullanılan OS'den oluşmaktadır. Sadece yazılım olarak NIDS araç tabanlı sistemler ile karşılaştırıldığında genellikle daha az maliyetli olmakla birlikte, donanımla uyumlu çalışabilmeleri için ayrıca yapılandırılmaları gerekmektedir. NIDS genellikle bir cihazın parçası olmamakta, ancak birçok ayrı donanımı bünyesinde bulunduran ayrı bir fiziksel cihaz olabilmektedir.

2.4.2. Güvenilir Bilişim (Trusted Computing)

Güvenilir bilişim (Trusted Computing – TC), Trusted Computing Group (TCG) tarafından geliştirilen ve desteklenen bir metodolojidir [30]. Güvenilir bilişim, donanım iyileştirmeleri ve buna bağlı yazılım değişiklikleri yoluyla bilgisayar güvenlik sorunlarını çözmek için teknoloji ve öneriler sunan geniş bir kavramdır. Bilişim sistemlerini oluşturan bileşenlerin her biri arasında gizlilik, bütünlük, erişebilirlik ve kurtarılabilirlik gereksinime dayanan bir "güven ilişkisi" planlar. Birçok büyük donanım üreticisi ve yazılım sağlayıcı firmalar TCG ile işbirliği yapmaktadır [30]. TCG, son kullanıcı haklarını ihlal etmeden, kötü niyetli kişiler tarafından yöneltilen tehditlere karşı bilgisayar kaynaklarının korunması için özelliklersunmaktadır.

2.4.2.1. Trusted Platform Module (TPM)

Yetkisiz değişikliklere ve tehditlere karşı bilgisayar kaynaklarını korumak için TC yaklaşımının ana parçası olan Trusted Platform Module (TPM) kullanılır. TPM, anakart üzerindeki tümleşik bir devredir ve sistem üzerinde çalışan yazılım tarafından iyi tanımlı olan veya olmayan komutların ve örneklerinbütünlüğünü kontrol eder [31].TPM, temelinde şifreleme anahtarlarından yararlanılarak oluşturulan, güvenlikle ilgili temel işlemleri sağlamak amacıyla tasarlanmıştır. TPM'de yer alan Platform Configuration Registers (PCRs)'daki işletim durum platformu hakkında bilgi depolar. Geçerli platformdan gelen platform isteklerini doğrular. TC uzaktan doğrulama (remote attestation), mühürlü depolama (sealed storage), güvenilir önyükleme (trusted boot) gibi teknolojileri içermektedir.

Kullanıcılar uzaktan doğrulama ile platform üzerinde çalışan yazılımdan sistem ile ilgili rapor alabilmektedir. Bu raporTPM tarafından imzalı ve onaylı PCRs yapılandırma değerlerinin bir listesidir [32]. TPM'e sahip bilgisayarlar, şifreleme anahtarları oluşturabilir ve bunları yalnızca TPM tarafından çözülebilecek şekilde şifreleyebilirler. Genellikle anahtarı "kaydırma" veya "bağlama" olarak bilinen bu işlem sayesinde anahtarın ortaya çıkarılması engellenebilir. Her TPM'nin depolama kök anahtarı adı verilen ve TPM'nin içinde depolanan bir ana "kaydırma" anahtarı vardır. TPM'de oluşturulan anahtarın özel kısmı asla başka bir bileşene, işleme, yazılıma veya kişiye gösterilmez.

3. MALZEME VE YÖNTEM

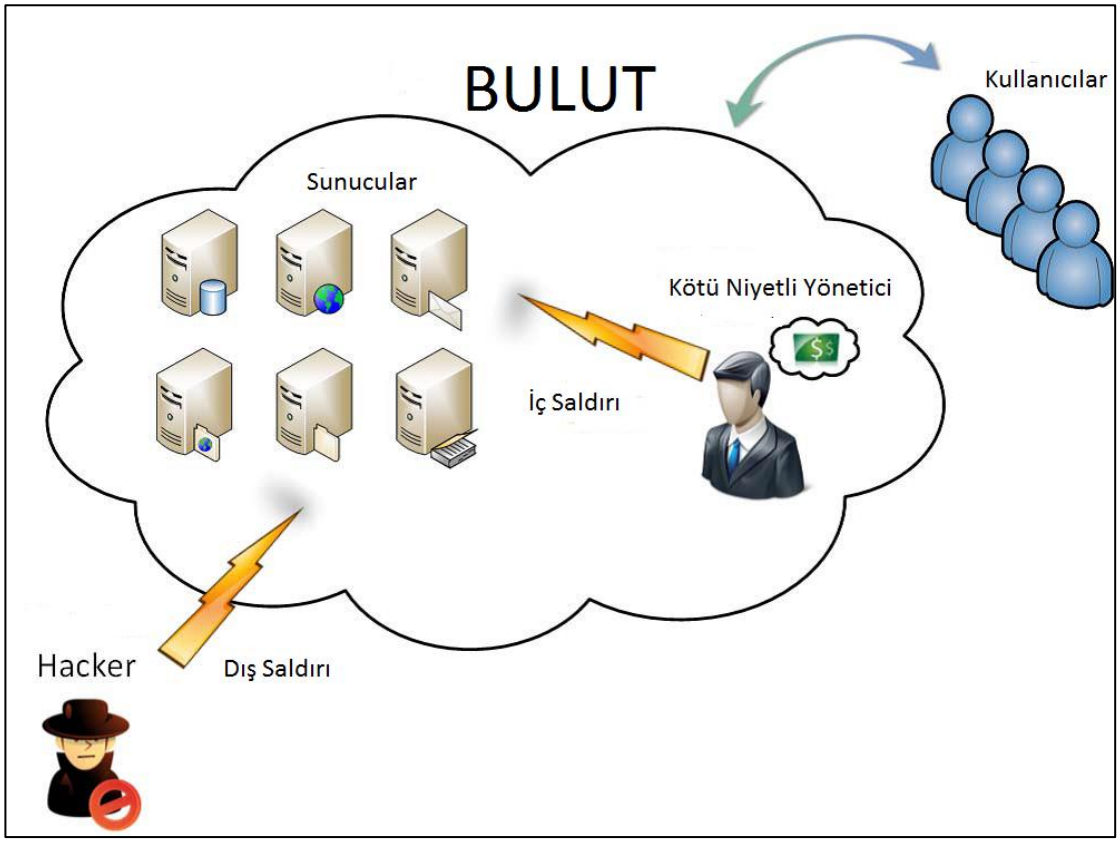
3.1. GENEL BİLGİLER

Bulut bilişim işletim sistemi, işlemci, depolama, ağ, uygulama gibi bilgisayar kaynaklarının ve servislerinin paylaşılmasını sağlayan dağıtık bir hesaplama yaklaşımıdır. Son yıllarda bulut bilişim paradigmasının oldukça kabul görmesi ve finans, kamu ve askeri alanlar gibi birçok faklı sektörde kullanımının yaygınlaşması sunduğu avantajlar sayesindedir. Forbes [33] dergisinde yayınlanan bir araştırmaya göre 2014 yılında pazarlama departmanlarının %47'si iki yıl içerisinde uygulamalarının %60'ını bulut plaftormlarına taşıyacaklardır. Veritabanı, e-posta ve iş uygulamalarının işletme ve barındırma yatırımlarının yarısı bulut bulut bilişim için yapılacaktır. Bunun yanında daha önce de belirtildiği gibi bulut bilişim teknolojisi kullanıcılarının güvenlik konusunda hala endişe taşımaları, araştırmacıları güvenlik konusunda çalışmaya yöneltmiştir.

Bulut bilişim altyapısında sanallaştırma önemli bir yer tutmaktadır. IAAS, PAAS ve SAAS modelleri açısından bulut bilişim gereksinimlerinin karşılanması için gerekli bilişim kaynaklarının gerçek kullanımından ziyade bu kaynakların sanal olarak oluşturulması sanallaştırma teknolojisi ile mümkün olmaktadır. Tüm özel ve kurumsal kullanımlarında kullanıcılar bulut bilişimden benzer beklentilere sahiptir ve bulut sağlayıcıları tarafından güvenlik ve güven sağlanması, tüm kullanımlar için öncelikli konuların başında gelmektedir.

Dağıtık yapısı nedeniyle bulut bilişim ortamları olası güvenlik açıklarını arayan saldırganlar/davetsiz misafirler için bir hedeftir. Çoğu çalışmalar göstermiştir ki, istemcilerin veri mahremiyetini, gizliliğini garantilenmesi için bulut bilişim sağlayıcılarına güvenmek zor bir konudur [34]. Bununla birlikte bulut sağlayıcısı veya yöneticisinin de her zaman güvenilir olacağının garantisi yoktur.

Bulut kullanıcıları Şekil 3.1'de gösterildiği gibi genellikle iki tür saldırılarla karşılaşır [35]:



Şekil 3.1: İç ve dış saldırılar [35].

Dış Saldırılar (External Attacks): Bulut sisteminde, tüm sanal platformu yöneten ve izleyen ayrıcalıklı domain (Dom0) işletim sistemi olması nedeniyle, bulut dışındaki bir saldırgan Dom0'a saldırabilir ve Dom0'ı sabote edebilir, zarar verebilir veya yönetebilir ve sanallaştırılmış ortamdakiVM'leri istismar edebilir. Bu tür saldırılara dış saldırılar denir. Bir VM istismar edildikten sonra, VM'deki her bilgiye erişebilir. Bir web servise yapılan kaba kuvvet (brute force) atakları, port tarama, herhangi bir UDP/TCP taşkın saldırıları (flooding attacks) ve SQL enjeksiyonları bu tür saldırılara örnek olarak verilebilir.

İç Saldırılar (Internal Attacks): Sağlayıcılar veya sistem yöneticileri her zaman güvenilir olmayabilirler, bulut kullanıcılarının gizli verilerini istismar edebilir ve çalabilirler. Bu tür saldırganlar bulut içerisindedir ve kullanıcıların gizli bilgilerini genellikle kötü niyetliolarak farklı amaçlar için kullanabilirler. Sistem yöneticileri kullanıcıların gizli bilgileri üzerinde işlem yaparken bu tür kötü niyetli saldırganlar gizli bilgilere ulaşabilirler. Bunun sonucunda kullanıcılar çok zor durumda kalabilirler.İzinsiz

olarak sanal disklerin sanal platform dışına bağlanması (mounting) ve rootkitler bu tür saldırılara örnek olarak verilebilir.

Sistem güvenliğini arttırmak için en popüler yöntem bulut bilişim ortamında sistemin sürekli izlenmesidir. Bunun için IDS kullanılması tercih edilen, iyi bir yöntemdir. IDS, bir sistemin hesaplama ve ağ kaynaklarını hedef alan zararlı faaliyetleri tanımlamaya çalışır. Tipik olarak imza veya anomali tabanlı izleme mekanizması veya her ikisinin aynı anda uygulanması ile çalışmaktadır. Literatürde bulut bilişim için birçok IDS mekanizması tanımlanmıştır. Bu tez çalışmasında güvenli bulut bilişim için farklı IDS kullanım modellerinin ele alınması amacıyla, öncelikletipik IDS kullanımına göre sistemde sadece üzerinde VM'lerin barındırıldığı ana makineye IDS sunucusu kurulmuş ve bu ana makine tarafından mevcut VM'ler izlenmiştir. Oluşturulan güvenlik mekanizmasının donanım verimliliği açısından sistem kaynak kullanım miktarları ölçülmüştür.Daha sonra bulut kullanıcılarını güvenilir olmayan bulut sağlayıcılarından ve bulut çalışanlardan korumak için bulut bilişimde kullanılabilecek IDS tabanlı bir hibrid yaklaşım sunan **AdjointVM** [32] ve AdjointVM yaklaşımıyla sunulan güvenlik mekanizmasının zayıf yönlerinin üstesinden gelmek amacıyla birtakım iyileştirmeler ekleyerek eksikliklerinin giderilmesini hedefleyen U. Oktay ve Ark. [35]'nın öneri olarak literatüre sunduğu**İyileştirilmiş AdjointVM Koruma Modeli** ve**Döngüsel Zincir VM Koruma Modeli** ele alınarak bulut bilişimde IDS kullanımının nasıl olması gerektiğinin ortaya konulması açısından test ortamında gerçekleştirilmiş ve bulgular karşılaştırılmıştır.

Döngüsel Zincir VM Koruma Modeli'nde gizlilik sorumluluğu bulut sağlayıcılarına ya da bulut yöneticilerine bırakılmamaktadır. Kullanıcılar kendi gizliliklerini kendileri yönetebilecek veya izleyebilecektir. Önerilen modelde, bir VM, diğer bir VM'yi koruma ve VM'nin güvenliğini sağlamaamacıyla izlemektedir. İzlenen VM de başka bir VM'yi izleyecek ve bir zincir halinde tüm VM'ler birbirini izleyecek, bu şekilde döngü halinde birden fazla VM korunmuş olacaktır. Döngüsel Zincir VM Koruma Modeli, AdjointVM Modelini iyileştirerek aşağıda verilen faydaları elde etmeyi hedeflemektedir [35]:

- AdjointVM'in yanlış-pozitif oranını arttırma,
- AdjointVM'in istismar direncini güçlendirme,

- Özellikle kötü niyetli sistem yöneticilerine, içi ve dış saldırılara karşı AdjointVM'i koruma,
- Esnek bir güvenlik politikası oluşturma,
- Sadece VM'inde barındırılan gizli verileri değil, VM'in kendi gizli verilerini de koruma,
- Gizli verileri korumak için çalışan VM'inlerin sayısını azaltmak, böylece sistem maliyetini düşürebilmek.

Ayrıca tez kapsamında yapılan çalışmaları bir adım daha ilerleterek bulut güvenliğinde önerilen Döngüsel Zincir VM Koruma Modeli'nin de olası eksikliklerine karşın ek iyileştirmeler yapılıp yapılamayacağının ortaya konması ve güvenlik modelin daha da geliştirilmesinin sağlanması hedeflenmiştir.U. Oktay ve Ark. [35]'nınönerdiği Döngüsel Zincir VM Korıma ModelindeVM'ler birbirini zincir halinde izlemektedir. İlk akla gelen nokta, bu takipte zincir şeklindeki yapıda VM'ler arasındaki iletişimin aksaklığa uğramasının zincir yapısında bir kopma meydana getirip getirmeyeceğidir. Bu sebeple yeni bir güvenlik mekanizmasının ortaya konulması ve sanal makinelerin zincir yapısı şeklinde değil de örgü (mesh) yapısı olarak birbirini izleyebilmesinin sağlanması amacıyla **Mesh VM Koruma Modeli**önerisinde bulunularak ilgili model oluşturulmuştur ve test ortamında incelenmiştir.

3.2. İLGİLİ ÇALIŞMALAR

3.2.1. AdjointVM Koruma Modeli

J.Kong [32], AdjointVM adında güvenilir olmayan bulut sağlayıcılarına yönelik önemli bir çalışma olarak görülen bir IDS mekanizması geliştirmiştir. Bu mekanizma, güvensiz sağlayıcılardan gelebilecek iç saldırılara karşı ve gerçek veya sanal ağ üzerinden gelebilecek dış saldırılara karşı IDS tabanlı sanal makine izleyicisi (Virtual Machine Monitor - VMM) tabanlıdır. Her iki saldırı türlerine karşı koruma sağlayan hibrid bir mimari geliştirilmiştir.

AdjointVM Koruma Modeli'nde sanallaştırma ortamı olarak açık kaynak kodlu bir platform olan Xen Hypervisor seçilmiştir. Xen platformunun başlıca üç bileşeni vardır: hipervizör, Dom0 ve DomU. Hipervizör, VM'ler için donanımı soyutlayan ve paylaştıran, VM'leri yöneten platformdur. Dom0, diğer VM'lerin yönetimi için

kullanılan ve sanal ortam araçlarını içeren ayrıcalıklı özelliklere sahip bir yönetim VM'sidir. DomU ise, sanal ortam yeteneklerini paylaşan ve kullanan tüm diğer VM'lerdir.

Sanal platform, TPM içeren bir donanım üzerinde çalışır. Sistemin uygulanması bir Secure Hypervisor Boot işlemine sahiptir, bu işlem güvenli önyükleme aracı olarak TrustedGrup [32] kullanır. Böylece sistem bir olası önyükleme sırası sağlayabilir.

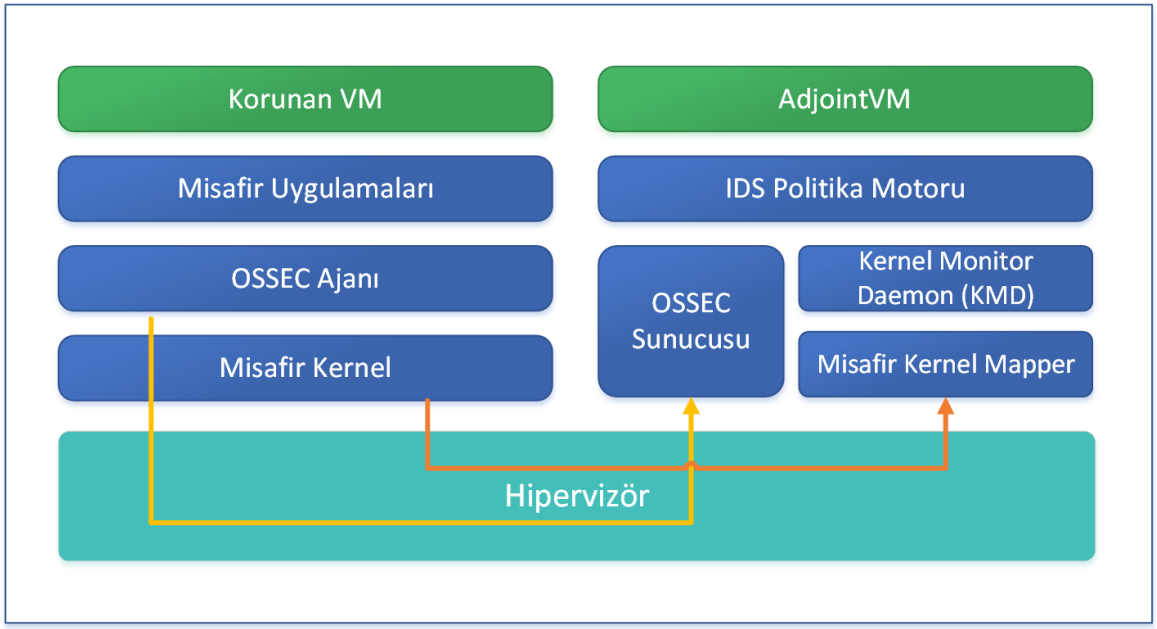
I/O Memory Management Units (IOMMU)'nun yardımı ile VM'lere atanmış bellek izole edilir [36]. Sistem çalışmadan önce IOMMU kullanılabilir durumda olması sağlanmalıdır ve hipervizör, IOMMU'nun mevcut olup olmadığını kontrol eder. Değilse önyükleme işlemi duracaktır.

AdjointVM modelinde her bir VM, AdjointVM denilen başka bir VM tarafından izlenmekte ve korunmaktadır. AdjointVM'in kurulması ile Xen bir VM'nin adres alanını diğer bir VM tarafından korunması için eşleştirilmesine yardımcı olur. Kullanıcı kesme noktalarını belirleyebilir, daha sonra bir olay izleme ve günlükleme daemon'u ile korunmuş VM'nin bellek alanını takip eder. Bir saldırı belirlendiğinde daemon, bunu kullanıcıya birkaç yolla raporlar.

VM'in güvenliğinin sağlanması amacıyla bu modelde hibrid bir IDS yapısı oluşturulmuş olup, sunucu tabanlı bir IDS olan Opearting System Security (OSSEC) HIDS ile işletim sistemi çekirdeğini izleyen Kernel Monitor Daemon (KMD) kullanılmıştır. Hibrid IDS modelinde kullanılan bir açık kaynak uygulama olan OSSEC, istemci – sunucu uygulamaları tarafından işletilir. OSSEC sunucusu AdjointVM üzerine yerleştirilir ve istemci korunmuş VM üzerindedir. OSSEC'in sunucu-istemci mimarisinden dolayı OSSEC sunucusu eşlenik olarak ortama eklenen VM (AdjointVM) üzerinde, istemci durumundaki OSSEC ajanları da asıl güvenliğinin sağlanması amaçlanan VM üzerinde teşkil edilmiştir. Saldırılara ait imzalar OSSEC sunucusu üzerinde bulunmakta ve ajanlar saldırı bilgilerini sunucu bünyesindeki saldırı imza veri tabanından almaktadır.

Ayrıca bu yaklaşım bir Kernel Monitor Daemon kullanır. KMD, korunmuş VM'in çekirdeğini eşleştirir ve izler ve eğer arkaplanda çalışan bir rootkit varsa tanımlar. KMD ise güvenliğinin sağlanması amaçlanan VM'in çekirdeğini haritalayarak, işletim

sisteminin arka planında kendini bir sistem servisi gibi gösteren gizli ve kötü niyetli bir yazılım olup olmadığını kontrol etmektedir. KMD üzerinde rutin ve olması gereken sistem servislerine ilişkin bilgiler bulunmakta, servis kimlik numaralarına ait özet değerler kaydedilmekte ve herhangi bir anormallik durumu işletim sistemi çekirdeğinin belirli zamanlarda denetlenerek mevcut bilgilerle karşılaştırılması sonucunda tespit edilmektedir. Güvenli VM başlangıç protokolünün kullanılmasıyla, güvenilmeyen bir ortamdaki bulut servis sağlayıcısı veya sistem yöneticisi tarafından kullanıcı VM'lerine ait sabit ya da sanal sürücülere istem dışı erişimin önüne geçilmiştir. AdjointVM Koruma Modeli'nin mimarisi Şekil 3.2'de verilmektedir.



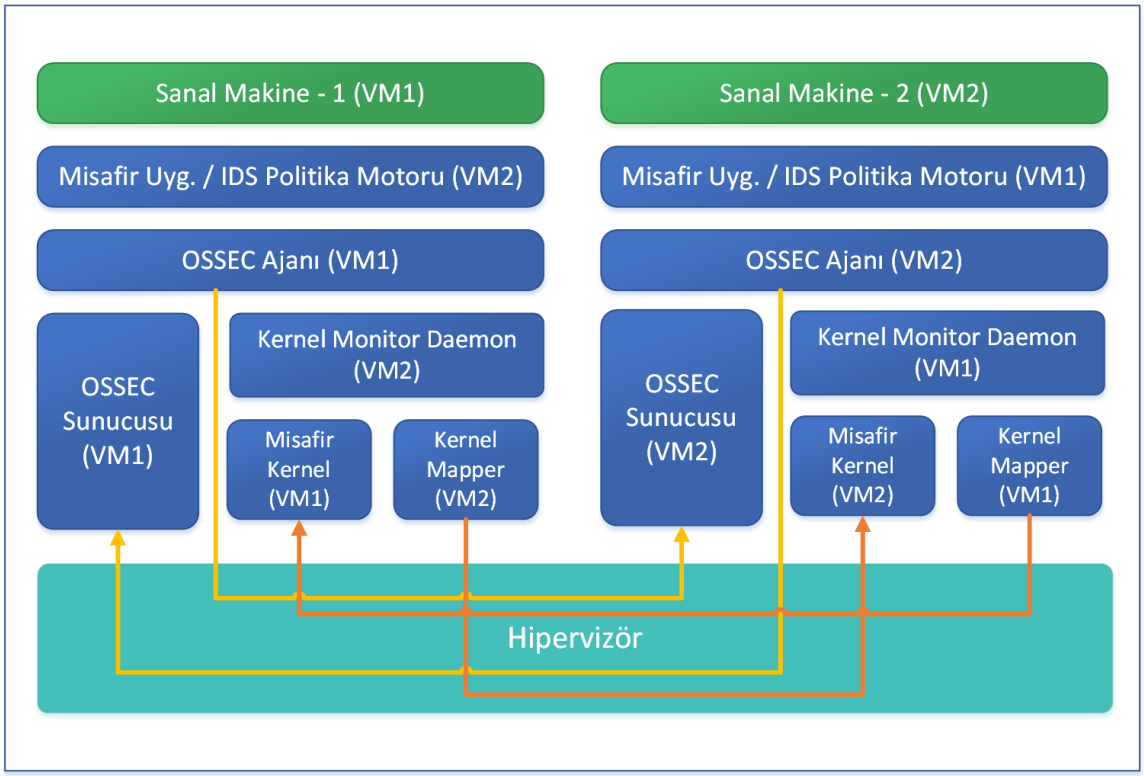
Şekil 3.2: AdjointVM Koruma Modeli mimarisi [35].

3.2.2. İyileştirilmiş AdjointVM Koruma Modeli

AdjointVM, bulut ortamı için iyi bir modeldir, fakat bazı güvenlik sorunları vardır. AdjointVM'yi hedef alan herhangi bir saldırı olursa ve servislerini manipule ederse sisteme doğru pozitif oranlı uyarı verememektedir,bu nedenle korunan VM iç ve dış saldırılara karşı savunmasız hale gelir. AdjointVM'de, bir VM çiftinde iki tür VM mevcuttur: koruyan VM ve korunan VM. Koruyan VM'nin görevi, korunan VM'nin güvenliğini sağlamaktır, ancak koruyan VM saldırılara karşı savunmasızdır.

AdjointVM modelindeki bu eksikliklere çözüm olarak U. Oktay ve Ark., AdjointVM Modeli'nin İyileştirilmesi [37] yaklaşımına göre yeni bir model önermişlerdir. Önerilen

modelde, AdjointVM modelinde korunan VM'nin güvenliği nasıl sağlanacak sorusuna cevap bulunmaktadır. İyileştirilmiş AdjointVM Koruma Modeli önerisinde, VM çiftlerindeki her iki VM de korunur ve diğer VM ile eşlenir. Hem her bir VM, diğer VM'inin hafızasını haritalalar, hem de her biri için bir OSSEC sunucusunun atandığı varsayılır. VM'lerden birine herhangi bir saldırı olursa, bu saldırı çiftler tarafından göz ardı edilmez. Şekil 3.3'te mimarisi verilen bu önerilen model sadece internet üzerinde sağlanan bulut servislerinin güvenilir olmayan bulut sağlayıcıları için değil, kurumsal firmaların kendi organisazyonlarında çalışan kötü niyetli BT çalışanları ve kendi intranetleri üzerinde tutulan bulut servisleri için de sunulmuştur.



Şekil 3.3: İyileştirilmiş AdjointVM Koruma Modeli mimarisi [35].

Önerilen sistemde başlıca hedef, başlangıcından sonlandırılmasına kadar sistemin tüm seviyelerinin izlenmesi ve ele alınmasıdır. Bulut sisteminin başlatılması ile, bir Güvenli Önyükleme Dizisi (Secure Boot Sequence) çalışır. Önyükleme dizisi hipervizörve VM önyüklemesini içerir. Hipervizör, TPM'e sahip bir anakart üzerinde çalışmalıdır. Bu kart ayrıca herhangi bir VM'nin, başka bir VM'nin hafıza alanına herhangi bir izin olmadan erişmemesi için Direct Memory Access (DMA) 'i kontrol etmek için IOMMU'yu desteklemelidir. Böylelikle herhangi olası bir hafıza istismarı engellenebilmektedir.

Sistem başlatılırken, TrustedGrub tarafından IOMMU'un bulunup bulunmadığı belirlemek için BIOS kontrol edilir. Eğer IOMMU özelliği kapalıysa, hipervizör önyükleme dizisi durabilir.

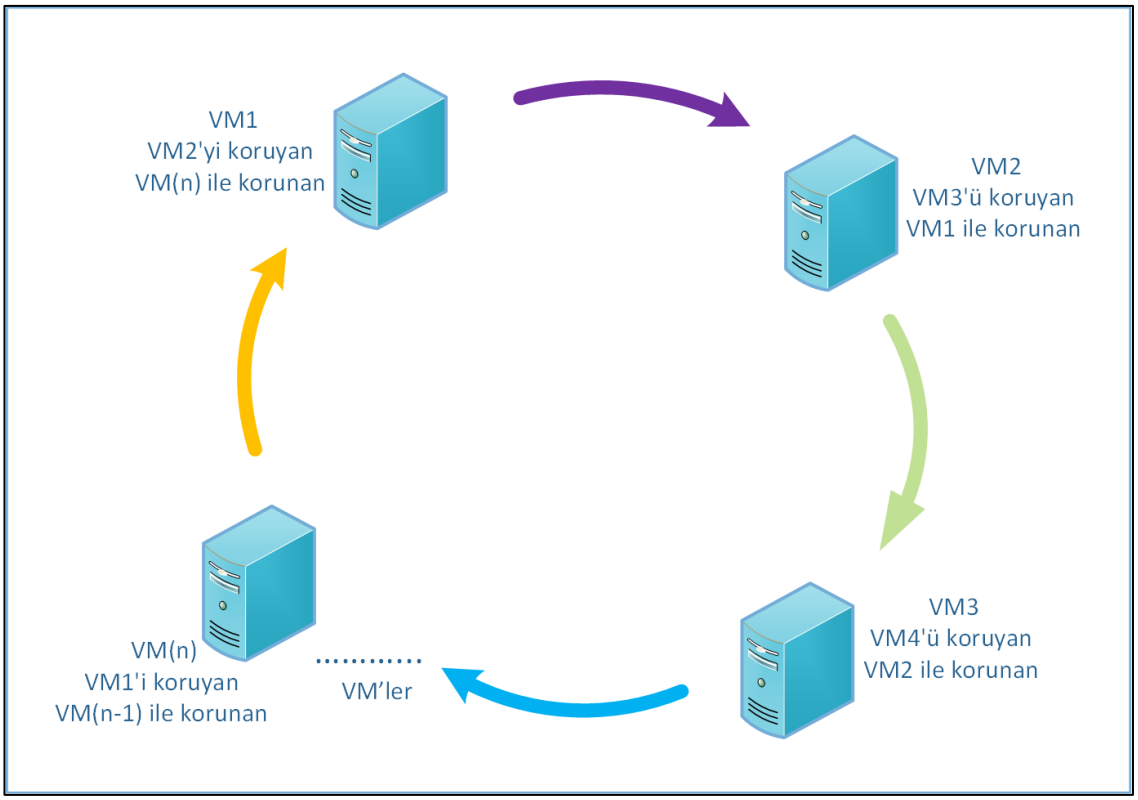
Güvenli VM için AdjointVM'in iyileştirilmesi yaklaşımında, iki VM yapılandırılmalıdır. Bir VM korunan VM, diğeri de onu koruyan VM dir. Korunan VM'de, kullanıcının gizli bilgileri depolanır ve koruyan VM'nde güvenlik mekanizmaları yer alır. Koruyan VM haritalama kabiliyetine sahiptir ve Korunan VM'in çekirdeğini herhangi KMD'ye karşı izler. Ayrıca kendi sunucu-tabanlı IDS'inin saldırı imzalarını kontrol eder. OSSEC HIDS, saldırı imzaları sunucusuna ve imza veritabanı ile iletişim kuran ve bu imzalar ile örnekleri analiz eden bir ajana sahiptir. OSSEC ajanı korunan VM üzerinde çalışır. Önerilen bu modelde, AdjointVM çiftindeki her iki VM de aynı anda hem koruyan hem de korunandır. VM'ler hem haritalama, hem de her birinin çekirdeğini herhangi bir rootkit'e karşı izleme yeteneğine sahiptir. Ayrıca, iki OSSEC sunucusu ve ajanı yer alır, birer sunucu ve ajan Korunan VM'de kurulur, diğerleri de Koruyan VM'de kurulur, böylece her iki VM de bir diğerini korur. Koruyan VM'ye herhangi bir saldırı olursa Korunan VM bunu engelleyebilir.

Geliştirilen sistem web tarama, sshd brute force, ftp scan, çoklu spam saldırıları, SQL enjeksiyonu gibi web saldırıları için ve knork ya da vlogger gibi çekirdek rootkitleri için dayanıklıdır [37]. Ayrıca host tapping girişimleri domainler arası şifreli kanallar ve IOMMU yardımı ile kovulabilir.Böylece Dom0, herhangi bir DomU'nun hafıza alanına erişimde Direct Memory Access (DMA) yararlılığını kullanamaz.

3.2.3. Döngüsel Zincir VM Koruma Modeli

U. Oktay ve Ark.'nın AdjointVM Koruma Modeli'nin iyileştirilmesine yönelik çalışmalar yapılması için getirdiği önerilerden biri de Döngüsel Zincir VM Koruma Modeli [35]'nin oluşturulmasıdır. Butez çalışması kapsamında bulut güvenliği için IDS kullanılması açısından bu modelin gerçekleştirilmesi ve test edilmesi sağlanmıştır. Döngüsel Zincir VM Koruma Modeli'nde bir VM, başka bir VM'yi, koruma altına almak ve güvenliğini sağlamak amacıylaizleyecektir. İzlenen VM de başka bir VM'yi izleyecek ve bu şekilde bir zincir halinde VM'ler birbirini izleyecek, böylelikletümVM'ler korunmuş olacaktır.

Önerilen modelde, AdjointVM Koruma Modeli'ndeki koruyan VM ve korunan VM yapısına ilave olarak bir VM diğer bir VM'yi korurken, üçüncü bir VM'nin de ilk VM'yi koruduğu bir zincir yapısı teklif edilmiştir. Böylece n adet VM'nin korunması gereken bir sistemde, AdjointVM modelinin aksine VM'lerin korunması için ilave herhangi bir VM'ye ihtiyaç duyulmadan n adet VM ile koruma planlanmıştır. Dinamik bir yaklaşımla güvenlik zincirine bir VM'nin eklenmesi ve çıkarılması ölçeklenebilirlik açısından mümkündür. Yeni bir VM (VMn) bulut sistemine eklenmek istediğinde, ilk olarak sistemdeki bir VM (VM1)'nin sorumluluğunu almak zorundadır. Daha sonra VM1'in koruyucusu olan VM2, kontrol ettiği VM'yi yeni VMn olarak değiştirecektir. Eğer zincirden (bulut sisteminden) bir VM çıkarılmak istenirse, koruyucu VM(VM1)'nin koruduğu VM (VM2), VMn sistem dışarısına çıkarıldıktan sonra değiştirilmelidir. Döngüsel Zincir VM koruma modeli Şekil 3.4'te gösterilmiştir



Şekil 3.4:Döngüsel Zincir VM Koruma Modeli [35].

Döngüsel Zincir VM Koruma Modeli'nde, her VM diğer bir VM'yi korur ve bir VM ile güvenli hale gelir. VM'lerin sayısının istemciler tarafından kontrol edilmesi gerekir ki bunlar kullanılacak VM sayısı olarak verilir. Bu şekilde sanal ortam harcaması ve VM'lerin güvenliğini sağlama çabası azaltılır. Bu döngüsel zincir kendi güvenliğini

sağlayabilir ve güvenlik için ek bir VM'ye ihtiyaç yoktur. Örnek olarak, AdjointVM yaklaşımında bir kullanıcının bulut içerisinde dört adet VM çalıştıracağı varsayılırsa, bu VM'lerin güvenliğini sağlamak amacıyla dört adet de AdjointVM çalıştırması gerekecektir. Bunun yerine geliştirilmiş olan zincir modeli kullanılırsa, zincir içerisindeki herbir VM diğer bir VM'nin güvenliğini sağlayacak ve böylelikle ilave bir VM'ye ihtiyaç olmadan, dört VMkendi aralarında güvenliği sağlamış olacaktır. AdjointVM Koruma Modeli ile genel olarak yapılan iyileştirmeden sonra önerilen modelin karşılaştırılması Tablo 3.1 [35]'de verilmiştir.

Tablo 3.1: AdjointVM ile İyileştirilmiş AdjointVM karşılaştırılması.

Özellikler	AdjointVM	İyileştirmeden Sonra
Zengin Bilgi	*Evet*	*Evet*
Düşük Yanlış-Pozitif Oranı	*Evet*	*Evet*
İstismar Direnci	*Evet*	*Evet*
Yapılandırılabilir Güvenlik Politikası	*Evet*	*Evet*
Saldırganlar İçin Görünmezlik	*Hayır*	*Hayır*
Kötü Niyetli Yönetici veya Sağlayıcıya Karşı Gizlilik	*Hayır, Bazı Sorunlar Var*	*Evet, Sorunların Üstesinden Gelir*
Ölçeklendirme	*Kolay Ölçeklenir*	*AdjointVM'e Göre Daha Zor Ölçeklenir*
Koruyan VM'in Güvenliğinin Sağlanması	*Hayır*	*Evet*
n Adet VM Güvenliği İçin Gereken VM sayısı	*2n*	*n*
Gider	*Azaltır*	*Daha Fazla Azaltır*

3.2.4. Mesh VM Koruma Modeli Önerisi

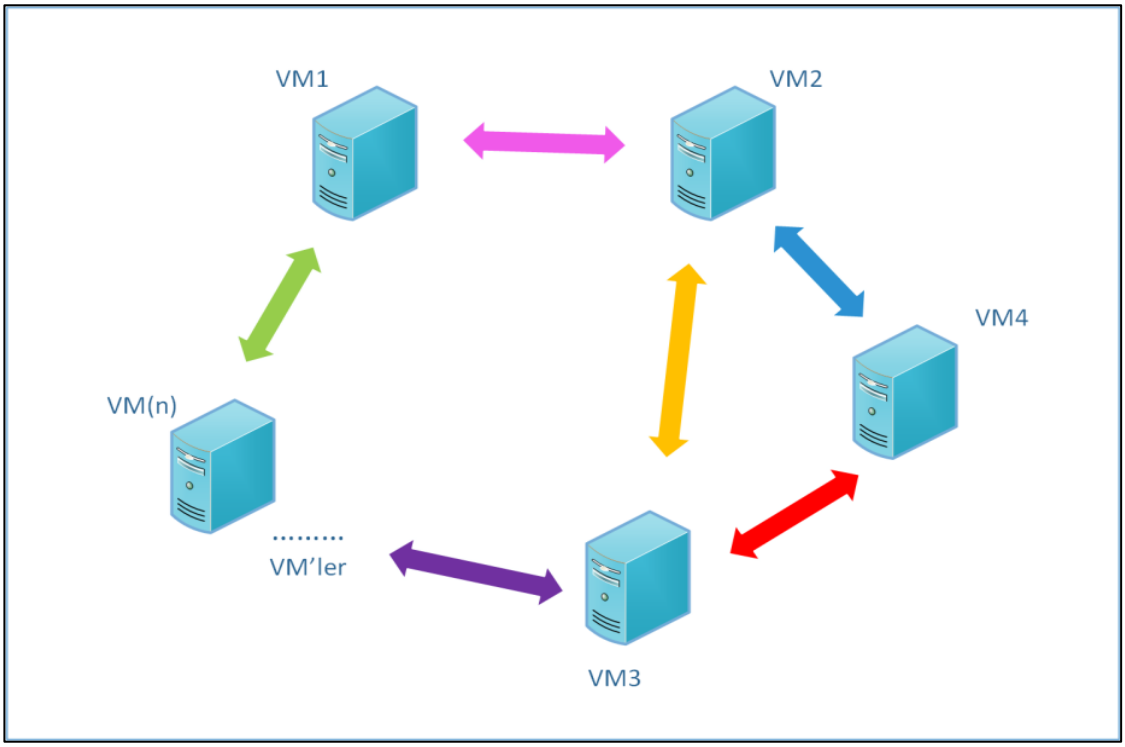
Mesh VM Koruma Modeli önerisinde güvenli bulut bilişim için gerçekleştirilen Döngüsel Zincir VM koruma modelinin de olası eksikliklerine karşın ek iyileştirmeler yapılıp yapılamayacağının ortaya konması ve modelin daha da geliştirilmesinin sağlanması amaçlanmıştır. Döngüsel Zincir VM Koruma Modeli'nde VM'ler birbirini zincir halinde izlemektedir. Eğer bu zincir koruma yapısındaVM'ler arasındaki iletişim

aksaklığa uğrarsa, herhangi bir VM çöker ya da devre dışı bırakılırsakoruma mekanizmasında açıklık meydana getirip getirmeyeceğinin garanti altına alınması gerekmektedir. Bu sebeple bu tez çalışmasıyla yeni bir mekanizma önerisi sunulmakta ve sanal makinelerin zincir yapısı şeklinde değil de mesh (örgü) yapısı olarak birbirini izleyebilmesinin sağlanması hedeflenmektedir. Mesh yapısı sayesinde ağ üzerindeki tüm sanal makineler birbirine bağlı olacak, gerektiğinde tüm sanal makinelerin birbirini izleyebilmesi mümkün olabilecektir.

Mesh, merkezi olmayan ve ağ üzerindeki her bir düğümün en az diğer iki düğüm ile bağlantılı olduğu bir ağ topolojisidir. Mesh topolojisinde bir düğüm, doğrudan başka bir düğüm ile konuşabilmektedir. Bu yapının en büyük avantajı merkezi olmaması, böylelikle tek bir hata noktasının bulunmamasıdır. İki düğüm arasında bağlantı kopması durumunda, düğümlerin bağlı olduğu başka düğümlerin var olması nedeniyle ağ iletişiminde sorun oluşturmadan bu aksaklık tolore edilebilmektedir. Böylelikle güvenilir bir bağlantı oluşturulmaktadır.

Mesh topolojinin bir diğer önemli avantajı esnekliğidir. Ağ yapısının büyümesi veya küçülmesi, düğümler arasında bağlantı sorunu oluşturmadan kolaylıkla gerçekleştirilebilmektedir. Mesh yapısı “self-confuring” özelliğine sahiptir, yani ağda otomatik olarak yeni bir düğüm mevcut yapıya eklenebilmektedir, ağ yönetimi açısından herhangi bir iyileştirme yapılmasına gerek yoktur.

Mesh topolojisine benzer bir yaklaşım sayesinde Döngüsel Zincir VM Modeli mesh yapısı olarak uyarlanmış ve bu koruma modeli ile oluşturulmak istenen güvenlik mekanizması seviyesinin en üste çıkarılması sağlanmıştır. Böylelikle sanal makineler arasındaki veri güvenliğinin sağlanmasında ve iletişimde olası bir aksama durumu tamamen ortadan kalkmış olacaktır. Yeni modelimiz olan Mesh VM Koruma Modeli'nde her bir VM, en az diğer iki VM'nin koruyanı ve en az iki VM tarafından korunan olacaktır. Her VM hem koruyan hem de korunan VM olarak davranacaktır. Örneğin Şekil 3.5’'te gösterildiği gibi VM1, VM2 ve VM(n) için koruyan olduğu gibi aynı zamanda VM2 ve VM(n) tarafından korunan olacaktır.



Şekil 3.5: Mesh VM Koruma Modeli.

Bu modelde ölçeklendirmede sisteme yeni bir VM eklendiği zaman yeni VM için mevcut VM'lerin donanım kullanım miktarlarına bakılarak en uygun VM koruyan VM'si olarak atanırken, kendisi de o VM için koruyan olacaktır. Yani karşılıklı olarak VM'ler birbirini korumuş olacaktır.

Ancak mesh yapısının güvenlik seviyesini yükselten avantajlarının yanında karmaşık yapısı nedeniyle oluşturulma maliyetinin yüksek olması dezavantaj olarak görülebilmektedir.Yapılacak olan deneysel incelemeler ile bu karmaşıklıkların üstesinden gelinmesi hedeflenmektedir.

3.3. GÜVENLİ BULUT BİLİŞİM İÇİN SALDIRI TESPİT SİSTEMİ KULLANIMI

3.3.1. Kullanılan Bileşenler

Güvenli bulut bilişim için saldırı tespit sistemi kullanımı çalışmasında ele alınan farklı yaklaşımlara dayalı AdjointVM Koruma Modeli, İyileştirilmiş AdjointVM Koruma Modeli ve Döngüsel Zincir VM Koruma Modeli ile Mesh VM Koruma Modeli'ninsunduğu güvenlik mekanizmalarınınIDS kullanımı açısından oluşturulması ve birbirleriyle kıyaslanması için gerekli gerçekleştirme ortamı hazırlanmıştır.

Belirtilen güvenlik mekanizmalarının saldırı tespit sistemlerini gerçekleştirebilmek için bulut bilişimin doğası gereği maliyet etkin bir çözüm olması ve sistem üzerine istenildiği şekilde müdahale edilebilmesi amacıyla kullanılan tüm bileşenlerin açık kaynak kodlu olması tercih edilmiştir.

Gerçekleştirme ortamında, referans model olarak ele alınan J. Kong'un yaptığı AdjointVM Koruma Modeli uygulamasının [32] gerçekleştirme ortamına uygunluk açısından Intel Virtualization Technology (Intel®VT) [36] ve hyper-therading özelliğine sahip Intel Core i7-2630QM 2.00 GHz CPU ve 4 GB DDR3 rastgele erişimli bellek (Random Access Memory – RAM) kapasiteli bir bilgisayar üzerinde sistemler inşa edilmiştir. Sistemde bahsi geçen referans modele uygunluk açısından hem ana makine hem de oluşturulan sanal makineler üzerinde 64 bit mimariye sahip Fedora 20 (Linux 3.19.5-100.fc20.x86_64) işletim sisteminin kullanılması tercih edilmiştir. Sanallaştırma platformu olarak referans modeldeki Xen Hypervisor seçilmiştir. Saldırı tespiti için hem referans modelinde belirtildiği üzere hem de yapılan literatür araştırmasında da etkinlik ve performans açısından tercih edilen açık kaynak kodlu sunucu tabanlı saldırı tespit sistemi (HIDS) olan OSSEC HIDS v2.8 kullanılmıştır.

3.3.1.1. Fedora İşletim Sistemi

Fedora işletim sistemi, tamamen açık kaynak kodlu ve özgür bir Linux dağıtımıdır. Dünyanın çeşitli bölgelerinde varlığını sürdüren özgür yazılım toplulukları arasında Fedora Projesi kapsamında hala geliştirilmekte ve yönetilmekte olup, RedHat tarafından destek görmektedir. 2004 yılında sona eren RedHat Linux dağıtımlarının devamı olarak bilinen Fedora, Linux dünyasında önemli bir kitleye sahip dağıtımdır. Aynı zamanda Fedora, RedHat Enterprise Linux ürününün test gereksinimlerini karşılamaktadır. Fedora, varsayılan olarak GNOME masaüstü kullanır, ayrıca KDE, Xfce ve LXDE masaüstü ortamlarını da seçenek olarak sunmaktadır.

3.3.1.2. Xen Hypervisor

Sanallaştırma platformu olarak Xen Hypervisor 4.3.4 versiyonu ana makineye kurularak sanallaştırma ortamı oluşturulmuştur. Xen, en popüler açık kaynak sanallaştırma mimarilerinden biridir. Sıradan işletim sistemi çekirdeğinden çok daha küçük kod tabanına sahip olması, güvenilir bilişime uygun olarak çalışması tercih sebeplerindendir.

3.3.1.3. OSSEC HIDS

OSSEC HIDS [39],sunucutabanlı bir saldırı tespit sistemidir. 2009 yılında Trend Micro tarafından alınan OSSEC HIDS, GPLv3 lisanslı açık kaynak kodlu bir uygulamadır.Temel çalışma prensibi "pro-aktif"tir. Sistemde yapılan faaliyetleri sürekli izler, belirli şartları sağlayan durumlar oluştuğunda harekete geçer. OSSEC HIDS, güvenliği sağlamak için sistem üzerinde birçok kontrol ve incelemelerde bulunur. Log analizi, dosya bütünlük kontrolü, politika izleme ve rootkit algılaması yapmaktadır.

Log analizi (log analysis); log dosyalarının incelenerek saldırı belirtisi olarak görülen kayıtlar için uyarı üretilmesidir.

Bütünlük kontrolü (integrity checking); sistemdeki dosyaların belirli zaman aralıklarıyla kontrol edilmesidir. Her kontrolde, varolan dosyaların toplamından MD5 (Message-Digest algorithm 5) gibi şifreleme algortimalarıyla belirli bir şifre elde edilir. Dosya boyutları ve oluşturulma/değişiklik tarihleri gibi bilgiler eklenerek kaydedilen bu şifre, her kontrolde bir önceki şifre bilgisiyle karşılaştırılarak dosyalarda herhagi bir değişiklik olup olmadığı incelenir.

Rootkit, sistemdeki bazı dosyaları ve işlemlerigizleyerek işletim sisteminin gerçeği görmesini engellemek için geliştirilmiş olan yazılımlardır. Örneğin "ps" komutunun işlevini değiştirmek üzere yazılan bir rootkit ile kullanıcıların, süreçlerin tümünü görüntülemesine engel olunacaktır. OSSEC, sistemde varolan rootkitleri tespit ederek kullanıcıları uyarmak amacıyla gerçek zamanlı ve yapılandırılabilir alarmlar üretir.

Aktif cevap (active response) ile tespit edilen saldırılara karşılık otomatik aksiyonlar alabilmektedir. Aktif cevap; belirli özel olaylar karşısında OSSEC HIDS'in otomatik olarak komut çalıştırmasıdır. Örneğin posta sunucusuna yapılan bir saldırı mevcut güvenlik duvarını geçmiş olsa bile OSSEC HIDS tarafından bu saldırıtespit edildiği zamangüvenlik duvarına komut gönderip saldırının oturumunun kapatılması istenecektir.

OSSEC HIDS içerisinde bir çok öntanımlı olarakkod çözümleyici (decoder) ve kural (rule)yer almaktadır.OSSEC HIDS Linux, MacOS, Solaris, HP-UX, AIX, Vmware ESX ve Windows platformlarında çalışabilmekte, agentless modu sayesindeyönlendirici (router), anahtarlama cihazı (switch) gibi ağ cihazlarını da izleyebilmektedir.

İstemci/Sunucu mimarisi ile merkezi yönetim sağlamaktadır. İletişim UDP 1514 portu üzerinden sağlanmakta ve simetrik anahtar Blowfish algoritması ile şifrelenmektedir [40].

OSSEC HIDS'in farklı platformlardaki kullanıcı bilgisayarları, web sunucuları, mail sunucuları veya WMware sunucuları üzerinde çalışabilmektedir. Ancak farklı formatlarda log üreten bu sunucuların OSSEC için alarm üretmesi için her log formatı ayrı kural ve çözümleyiciye ihtiyaç vardır. Buna göre elde edilen alarmlar ise daha sonra kritiklik seviyesine göre sınıflandırılmaktadır. OSSEC HIDS'in gelişmiş özelliklerine sahip olmasını sağlayan çözümleyici ve kurallardır. Bu ikisi yapılandırılarak elde edilen loglardan özgün alarm üretilebilmektedir.

Bütün OSSEC kuralları, OSSEC'in kurulu olduğu dizin altında *rules/* dizininde saklanmaktadır. Genellikle, ön tanımlı olarak yüklendiği yer olan */var/ossec/rules* altındadır. Bu dizin altında her bir kural için ayrı bir XML dosyası bulunur. Örnek olarak Apache http sunucu kuralları apache_rules.xml, CISCO PIX güvenlik duvarı ile ilgili kurallar ise pix_rules.xml dosyasında yer almaktadır. Bunlar gibi OSSEC HIDS ön tanımlı olarak 43 adet kural dosyası barındırmaktadır. Her bir kural dosyasında ilgili uygulama veya cihaz için tanımlanmış çok sayıda kural tanımı vardır. Bu kural setinde yer alan bazı dosyalar Tablo 3.2'de verilmiştir.

Kurallarla beraber ortak çalışan çözümleyiciler de decoders.xml dosyasında aynı şekilde ön tanımlı olarak */var/ossec/etc/decoders.xml* dosya yolunda yer almaktadır. Çözümleyiciler ham olay kayıtlarından veri ayıklamak amacıyla tasarlanmıştır. Böylece farklı kaynaklardan alınan olaylar ilişkilendirilmektedir.

OSSEC'de her kurala bir ID verilmiştir. Aynı kategorideki kurallar, belirli bir aralıkta yer alacak şekilde belirlenmiştir. Örnek kural ID aralıkları ve ilgili kategorileri Tablo 3.3'te verilmiştir.

Tablo 3.2:OSSEC HIDS örnek kural dosyaları.

Kural Adı	Tanımı
apache_rules.xml	Apache HTTP sunucu kuralları
attack_rules.xml	Yaygın saldırı kuralları
firewall_rules.xml	Yaygın firewall kuralları
ftpd_rules.xml	ftpd daemon için kurallar
ids_rules.xml	Yaygın IDS kuralları
imapd_rules.xml	imapd daemon için kurallar
local_rules.xml	OSSEC HIDS yerel, kullanıcı-tanımlı kurallar
mysql_rules.xml	MySQL veritabanı kuralları
ossec_rules.xml	OSSEC HIDS kuralları
policy_rules.xml	Politikaya özgü olay kuralları
postfi x_rules.xml	Postfix mail transfer ajanı kuralları
sshd_rules.xml	Secure Shell (SSH) ağ protokolü kuralları
syslog_rules.xml	Yaygın syslog kuralları
telnetd_rules.xml	telnetd daemon için kurallar
web_rules.xml	Yaygın web sunucusu kuralları

Tablo 3.3: OSSEC HIDS ön tanımlı kural ID aralıkları.

Kural ID Aralığı	Genel Kategori
00000–00999	OSSEC HIDS 'in kendi iç kuralları için ayrılmış
01000–01999	Genel syslog kuralları
02100–02299	Network File System (NFS) kuralalrı
02300–02499	xinetd kuralları
02500–02699	Erişim kontrolü kuralları
...	...
100000–119999	Kullanıcı tanımlı kurallar

Kullanıcı tarafından yazılacak yeni kurallar yerel kurallar (local rules) olarak tanımlanır ve */var/ossec/rules/local_rules.xml* dizini altındadır. Yerel kurallar için 100,000-119,999 kural ID aralığı belirlenmiştir. Kullanıcı tarafından belirlenen kurallar dışında OSSEC'in kendi kurallarında değişiklik yapılmaması gerekmektedir. Versiyon güncellemesi esnasında OSSEC tarafından belirlenen bütün kurallarda değişiklik yapılabilir ancak kullanıcı tarafından oluşturulan local_rules.xml dosyasında değişiklik yapılmaz.

```
<group name="syslog,sshd,">
 <rule id="123" level="2">
  <decoded_as>sshd</decoded_as>
  <description>Logging every decoded sshd message</description>
 </rule>
<rule id="124" level="7">
  <if_sid>123</if_sid>
  <match>^Failed password</match>
  <group>authentication_failure</group>
  <description>Failed SSHD password attempt</description>
 </rule>
 <rule id="125" level="3">
  <if_sid>123</if_sid>
  <match>^Accepted password</match>
  <group>authentication_success</group>
  <description>Successful SSHD password attempt</description>
 </rule>
</group>
```

Şekil 3.6:OSSEC HIDS örnek kural yazımı.

OSSEC kural yazımıyla ilgili Şekil 3.6'da tanımlanan üç tane kural görülmektedir. *<group name="syslog,sshd,">* başlangıç etiketiyle bu gruptaki kuralların syslog ve sshd ile ilgili kurallar olduğu belirtilir. 123 numaralı kuralda yer alan *<decoded_as>*etiketiyle, meydana gelen olaylardan sadece sshd olaylarının bu kurala tabi tutulması gerektiği belirtir. *<description>* etiketi de kuralla ilgili açıklama bilgisini içerir ve bu kuralla eşleşen olaylar için üretilen uyarılarda bu açıklama bulunur.

124 ve 125 numaralı kurallardaki<*if_sid*> etiketi, bu iki kuralın 123 numaralı kurala ait çocuk kural olduklarını ve 123 numaralı kural uygulandıktan sonra uygulanabileceklerini belirtir. <*match*>^*Failed password*</*match*> etiketiyle, meydana gelen sshd olaylarında Failed password kelime çiftinin geçip geçmediği kontrol edilerek eğer geçiyorsa <*description*> etiketinde belirtilen açıklamalı uyarının üretilmesi sağlanır. <*group*>*authentication_failure*</*group*> etiketi ile de uyarının hangi OSSEC HIDS grubuna ait olduğu belirtilir.

OSSEC HIDS, belirlenen kurallara göre logları değerlendirerek alarm üretir. Atomik ve birleşik (composite) olmak üzere iki kural çeşidi vardır. Atomik kurallar, herhangi bir korelasyon olmadan sadece bir olaya dayanır. Birleşik kural ise, farklı olayların bir araya gelmesiyle oluşan kurallardır. Aynı IP adresinden 10 kez yanlış parola girilmesi sonrasında kaba-kuvvet (brute-force) saldırısı olduğuna dair alarm üretilmesi buna örnek verilebilir.

OSSEC HIDS çeşitlı kurulum modları ve bileşenleri ile kullanıcıya sunulmaktadır. Bu modlar ve bileşenler ile ilgili bilgiler aşağıda verilmektedir:

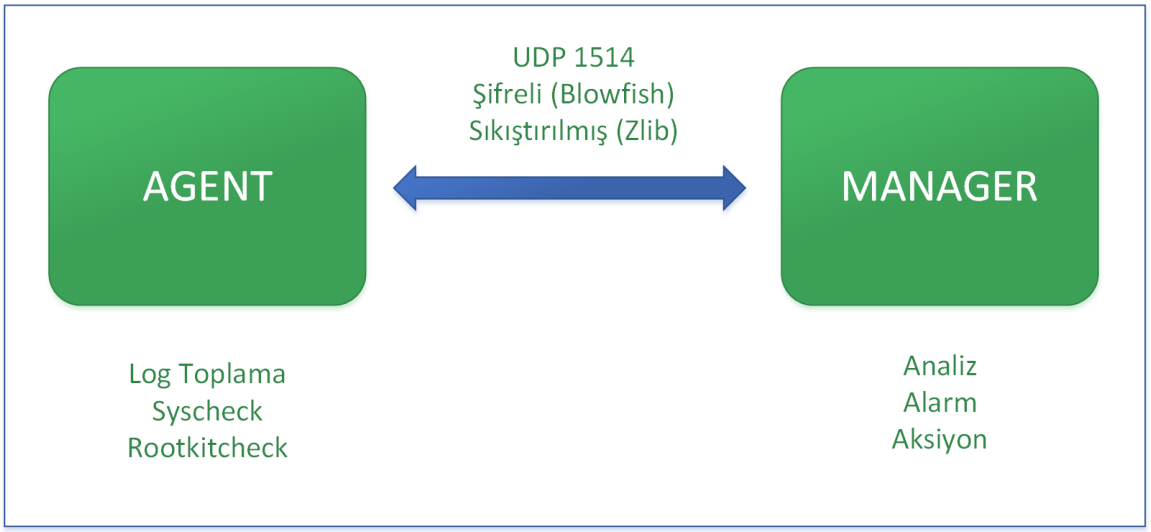
Manager: OSSEC'in temel bileşenidir. Tüm yapıyı izler ve uzak hostlardan aldığı bilgileri analiz eder. OSSEC Server olarak da adlandırılan Manager tüm sistemler için merkezi noktadır. Bu noktada dosya bütünlük kontrolü veritabanları, loglar, olaylar (events) ve sistem denetim (system auditing) girdileri yer almaktadır. Ayrıca, agent'ların kuralları ve kod çözümleyicileri Manager'da tutulmaktadır. Bu şekilde çok fazla sayıda sistemden oluşan yapılar merkezi olarak kolaylıkla yönetilebilmektedirler.

Agent: İzlenmesi istenen her sisteme kurulan küçük OSSEC programının adıdır. Ajan olarak da adlandırılmaktadır. Üzerinde çalıştığı sisteme ait bilgileri toplar ve analiz edilmesi için Manager'a gönderir. Bu uygulama çok küçük bir hafıza ve CPU footprint'esahip olduğu için sisteme ek yük bindirmez. Agent'lar kurulum sırasında oluşturulan düşük yetkili kullanıcı tarafından, chroot ortamda ana sistemden izole edilmiş bir şekilde çalıştırılmaktadır. Agent ile ilgili yapılandırma ayarlarının hemen hepsi Manager tarafında tutulabilmekte ve agent'in bulunduğu host üzerinde yapılandırmanın sadece bir kısmınun bulunması sağlanabilmektedir. Gerekli yapılandırma bilgileri Manager tarafından verilmektedir.Bu özelliklerin dışında

yerelherhangi bir yapılandırma dosyası değiştirilirse Manager, hemen alarm üretip sistem yöneticisini uyarmaktadır.

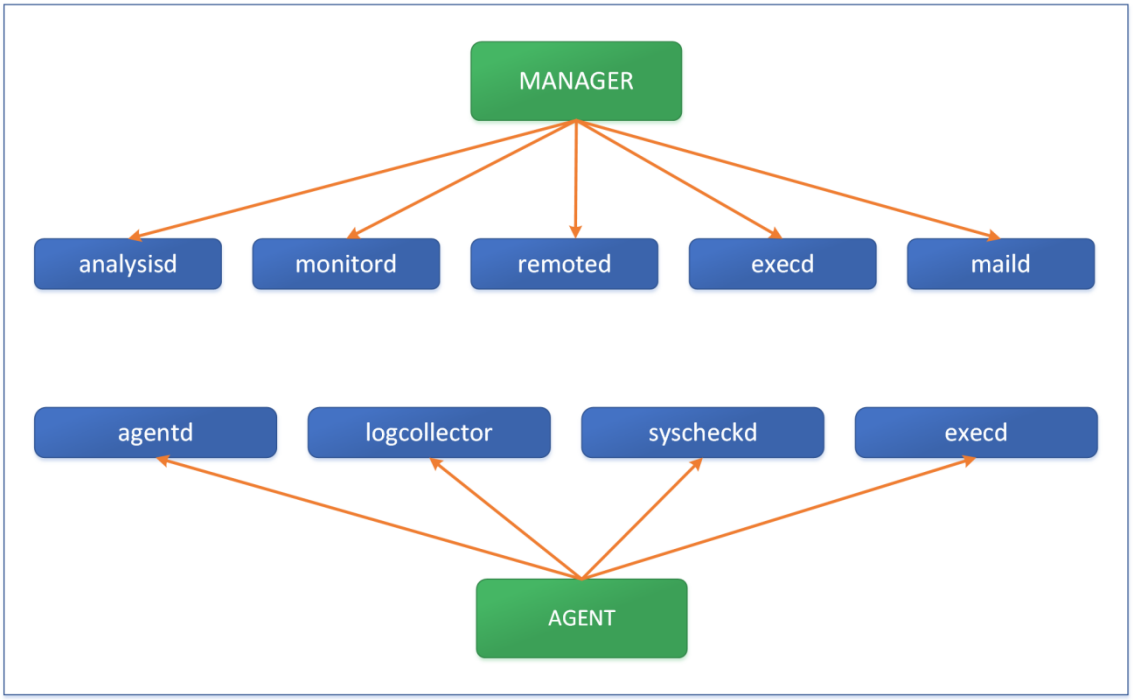
Agentless: Üzerine agent kurulamayan her türlü sistemi izlemek için kullanılan moddur. Agentless izleme, özellikle güvenlik duvarı, anahtarlama cihazı, yönlendirici gibi ağ cihazlarının ya da üzerinde agent kurma yetkisinin olmadığı bazı sistemlerin izlenmesi ve dosya bütünlük kontrollerinin yapılması için kullanılması uygundur.

OSSEC HIDS, yukarıda belirtilen görevleri yerine getirmek için daemon olarak bilinen birçok program süreci yürütmektedir. Şekil 3.7'de OSSEC HIDS çalışma durumu ve Şekil 3.8'de gösterilen temel süreçler ile ilgili açıklamalar aşağıda verilmiştir.



Şekil 3.7: OSSEC HIDS çalışma durumu.

- **Analysisd**: Sunucu taraflı çalışır; tüm analiz işlerinden sorumludur.
- **Remoted:** Agentlarla iletişim kuran süreçtir.
- **Monitord:** Agentların bağlantı durumlarını izler.
- **Maild:** Mail yoluyla alarmları göndermekten sorumlu süreçtir.
- **Execd:**Aktif cevap (active response) komutlarını çalıştırır.
- **Agentd:** Agent'da çalışır; sunucu ile iletişimden sorumludur.
- **Logcollector:**İzlenen log dosyalarını okur.
- **Syscheckd:** Dosya bütünlük kontrolünden sorumludur.

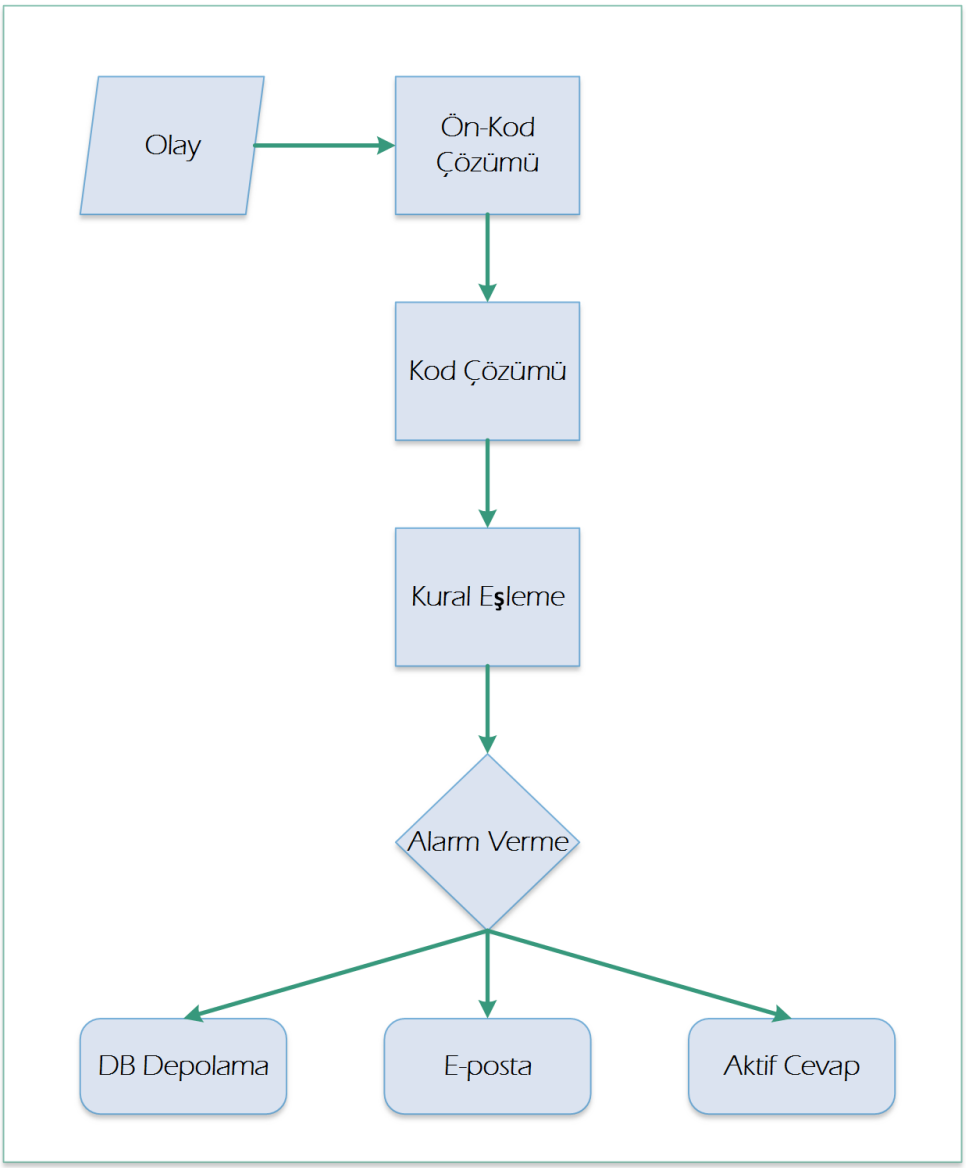


Şekil 3.8: OSSEC HIDS süreçleri.

OSSEC HIDS analiz işleminde, sistemi izlerken bir olay (event) meydana gelir gelmez en kısa sürede olayı çözmeye ve olayla ilgili değerli bir bilgi çıkarımında bulunmaya çalışır. Olayı çözme ve normalleştirme işlemi iki aşamada yapılır: ön-kod çözümü (predecoding) ve kod çözümü (decoding). Pre-decoding işleminde olaydan daha basit ve statik bilgiler çıkartılır.Bunlar tarih, saat, bilgisayar adı, log mesajı gibi bilgilerdir. Decoding işlemindeki amaç ise daha ayrıntılı, statik olmayan bilgiler çıkartarak, daha sonraki aşamada kurallarla bu bilgileri eşleştirerek alarm üretmektir.Bunlara ise IP adresi, kullanıcı adı vb. bilgiler örnek verilebilir. OSSEC HIDS'in olayı ele alışı ile ilgili akış diyagramı Şekil 3.9'da verilmiştir.

3.3.1.4. Logstash, ElasticSearch ve Kibana

OSSEC HIDS sistemi izlerken kural tabanı ve çözümleyiciler ile meydana gelen olaylar analiz edilerek üreilen alarmlar log dosyalarında kayıt altında tutulmaktadır. Büyük log dosyalarındaki veriler üzerinde kullanıcıların kolaylıkla arama, depolama, izleme ve görsel olarak zenginleştirme işlemlerini yapabilmesi için yardımcı araçlara ihtiyaç duyulmuştur. Bu nedenle log kayıtlarının yönetimi ve değerlendirilmesi için OSSEC HIDS ile beraber açık kaynak olarak sunulan Logstash, Elasticsearch ve Kibana yazılımları kullanılmıştır.



Şekil 3.9:OSSEC sunucusu iş akışı.

Logstash [41], olay ve log kayıtlarını yönetmek için kullanılan açık kaynak kodlu bir araçtır. Log kayıtlarının toplanması, ayrıştırılması (parsing) ve sonradan yapılacak arama için depolanması gibi işlemleri sağlamaktadır. Uygun yapılandırma sağlandıktan sonra log kayıtlarının toplanması istenilen sunucularüzerinde çalıştırılarak log verilerinin merkezi sunucularına aktarır ve bir web ara yüzü ile bunların görüntüler.

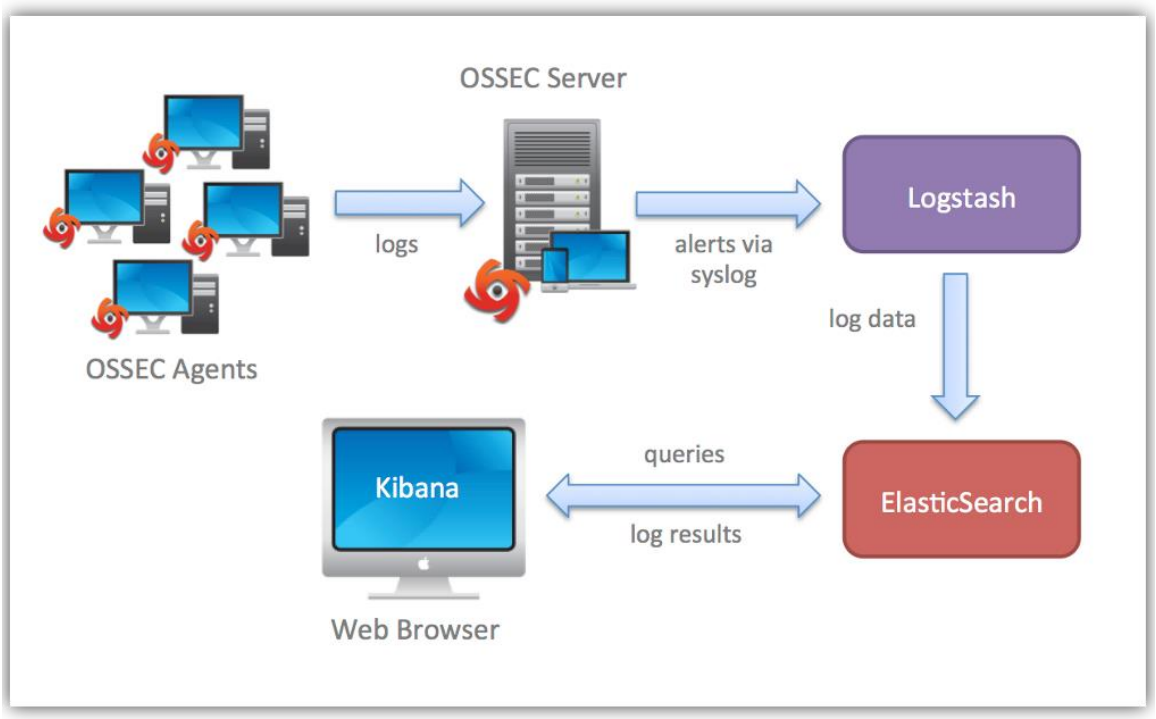
ElasticSearch [42], Apache Lucene altyapısından geliştirilmiş, hafif, kolay kurulan, açık kaynak kodlu, ücretsiz, ölçeklendirilebilen bir arama motorudur. Java tabanlı olarak geliştirilen ElasticSearch RestfulAPI üzerinden hizmet vermekte, 3.part görsel araçları ve güvenlik seçenekleri ile çok hızlı ve kullanışlıdır. RestfulAPI üzerinden

hizmet verdiği için tüm programlama dilleri ile kullanılabilmektedir. Hem Windows hem Linux işletim sistemleri ile kolayca çalışabilmektedir. İlk çıkış amacı dağıtık ve ölçeklenebilir büyük veride çalışabilen bir arama motoru oluşturmak olan ElasticSearch gerçek zamanlı olarak indeksleme yapabilmektedir.ElasticSearch belge yönelimli(document oriented) bir arama motorudur. ElasticSearch'te her kayıt, yapılandırılmış JSON belgesidir. Bir başka deyiş ile, ElasticSearch'e indekslenmesi için gönderilmiş her veri bir JSON dokümandır. Dokümanın bütün alanları varsayılan olarak indekslenir ve tek bir sorguda kullanılabilir.

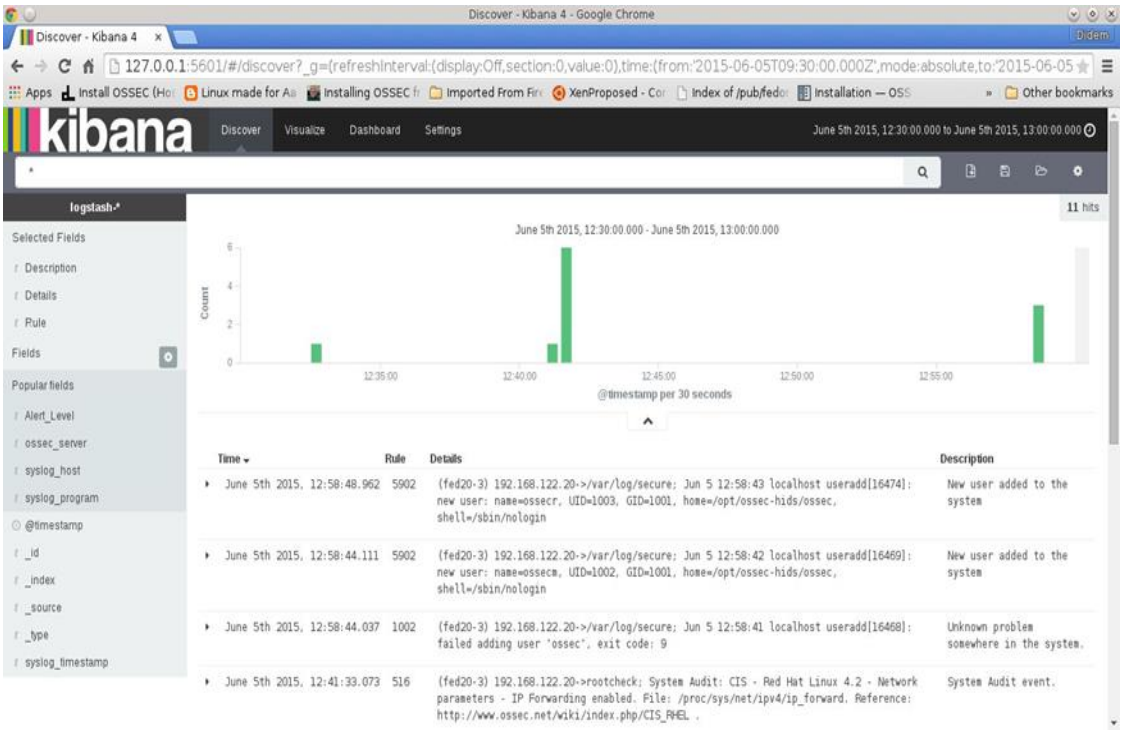
Kibana [43] ise, ElasticSearch için veri görselleştirme motorudur. ElasticSearch'daki anlamlı veriyi sorgulamak ve görselleştirmek için kullanılan bileşendir.

Elasticsearch, Logstash ve Kibana log toplama, analizi ve görselleştirme konusunda birlikte sıkça kullanılan "ELK Stack" ismi ile anılan açık kaynak kod bir çözümdür.

OSSEC sunucusundan elde edilen log kayıtlarının analizi ve görselleştirilerek değerlendirilmesi için öncelikle Logstash yazılımıOSSEC sunucusu tarafından üertilen log kayıtlarını ayrıştırabilmesi için yapılandırımıştır. Böylece Logstash, OSSEC sunucusundan alınan ham verilerin (log kayıtları ve alarmların) sorgulanarak analiz edilebilmesine olanak sağlamak amacı ile JSON formatına dönüştürmekte ve Elasticsearch'e yazılmasını sağlamaktadır. Bu şekilde Logstash üzerinden ayrıştırılarak düzenlenmiş veri analiz etmek üzere düzgün bir formata kavuşmaktadır. ElasticSearch, Logstash'ten gelen OSSEC alarm ve loglarını depolamaktadır. Kibana ise, OSSEC HIDS tarafından üretilen ve Logstash tarafından formatlanarak Elasticsearch'de depolanan verinin analiz edilerek görselleştirilmesi için kullanılmıştır.Kibana, kullanıcı sorgularının kolayca ElasticSearch'e göndermek ve sonuçları bir web ara yüzü üzerinden görsel olarak gösterge panolarıyla ekranda görüntülenmesi amacıyla tasarlanmıştır. Şekil 3.10 [44]'da OSSEC sunucusu ile ElasticSearch, Logstash ve Kibana çalışma mimarisi gösterilmektedir. Şekil 3.11'de ise bu çalışma kapsamından oluşturulan sistem üzerindeki Kibana web arayüzü yer almaktadır.



Şekil 3.10:OSSEC Sunucusu ile Logstash, ElasticSearch ve Kibana çalışma yapısı [44].



Şekil 3.11: Kibana web arayüzü.

3.3.2. Gerçekleştirme Ortamı ve Uygulama

Bulut bilişim sistemlerinde sistemin içerisinden, yani sanal altyapıda kullanılan hipervizör katmanından, bulut sistem yöneticilerinin veya operatörlerinin yetkisiz erişimlerine karşı bulut kullanıcıları tarafından alınabilecek güvenlik mekanizmalarının eksikliğin giderilebilmesi adına yapılan bu tez kapsamında sistem üzerindeki istenmeyen ve zararlı aktivitelere karşı HIDS'lerin nasıl konumlandırılması ve yapılandırılması gerektiği ile ilgili çalışmalar yapılmıştır. Bu amaç doğrultusunda önceki bölümlerde açıklanan dört ayrı koruma modeline ve tipik HIDS kullanımına görebeşgüvenlik mekanizması mimarisi oluşturulmuştur. Bunlar;

1. Sadece Ana Makinede HIDS Kullanım Mekanizması

2. AdjointVM Koruma Mekanizması,

3. İyileştirilmiş AdjointVM Koruma Mekanizması,

4. Döngüsel Zincir VM Koruma Mekanizması,

5. Mesh VM Koruma Mekanizmasıdır.

Birinci mekanizmada, tipik HIDS kullanım yapılandırmasına uygun olarak oluşturulan güvenlik mekanizmasında, sistemde yer alan sanallaştırma platformunu barındıran ana makineye OSSEC sunucusu kurulmuş, mevcut her bir VM'ye de OSSEC ajanı kurularak ana makinedeki sunucu tarafından bu VM'ler izlenmiştir. Böylelikle sistem güvenliği sadece ana makinenin sorumluluğundadır.

İkinci mekanizmada, sistem içerisinde yer alan VM'lerin güvenliğinin sağlanması için her bir VM'yi koruyacak AdjointVM olarak adlandırılan eşlenik bir VM oluşturulmuştur. Oluşturulan bu AdjointVM, korunan VM'nin güvenliğinden sorumludur. OSSEC'in sunucu istemci mimarisinden dolayı OSSEC sunucusu eşlenik olarak ortama eklenen VM üzerinde, istemci durumundaki OSSEC ajanları da asıl güvenliğinin sağlanması amaçlanan VM üzerinde teşkil edilmiştir. Saldırılara ait imzalar OSSEC sunucusu üzerinde bulunmakta ve ajanlar saldırı bilgilerini sunucu bünyesindeki saldırı imzaları veri tabanından almaktadır. Bu nedenle HIDS olarak OSSEC sunucusu AdjointVM üzerine kurulmuş, bu sunucuya ait OSSEC ajanı (agent)

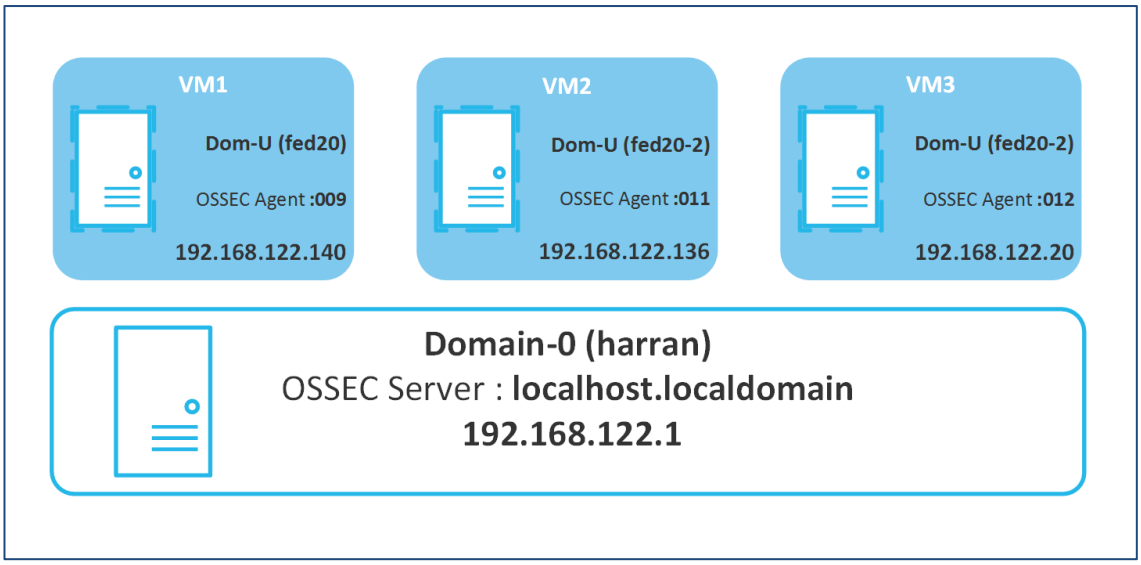
ise korunan VM üzerine kurulmuştur. Bu iki VM bir koruma çifti olarak sistemde yer almaktadır.

Üçüncü mekanizmada, AjdointVM Koruma Modelindeki OSSEC sunucularını barındıran eşlenik VM'lerin güvenliğinin de sağlanması amaçlanarak, koruma çiftlerindeki her bir VM hem OSSEC sunucusu hem de OSSEC ajanı barındırmakta, karşılıklı olarak birbirlerinin güvenliğinin sorumluluğunu almaktadırlar. AdjointVM korunan VM'yi izlerken, korunan VM üzerindeki OSSEC sunucusu da AdjointVM'yi izlemektedir. Bu şekilde sanallaştırma ortamında tüm VM'ler çiftler halinde yer almaktadır.

Dördüncü mekanizmada ise, her bir VM için güvenliğinden sorumlu ek bir VM'ye ihtiyaç duyan çift VM yapısının sisteme getirdiği ekstra VM oluşturma ve barındırma maliyetini ortadan kaldırmak amacıyla önerilen Döngüsel Zincir VM Koruma modelinin gerçekleştirilmesi amacıyla sistemdeki mevcut her bir VM, zincir halinde diğer bir VM'nin güvenliğinden sorumlu olur ve başka bir VM ile güvenli hale gelir, zincirin en son üyesi de ilk VM'in güvenliğini sağlar. Her VM, zincirin bir üyesinin güvenliğini sağlar ve diğer bir üyeyi güvenli hale getirmektedir. Bu döngüsel zincir kendi güvenliğini sağlayabilmekte ve ek VM'lere ihtiyaç duymamaktadır. Bu mekanizmanın sağlanması için her bir VM üzerine hem OSSEC sunucusu hem de OSSEC ajanı kurulmuştur.

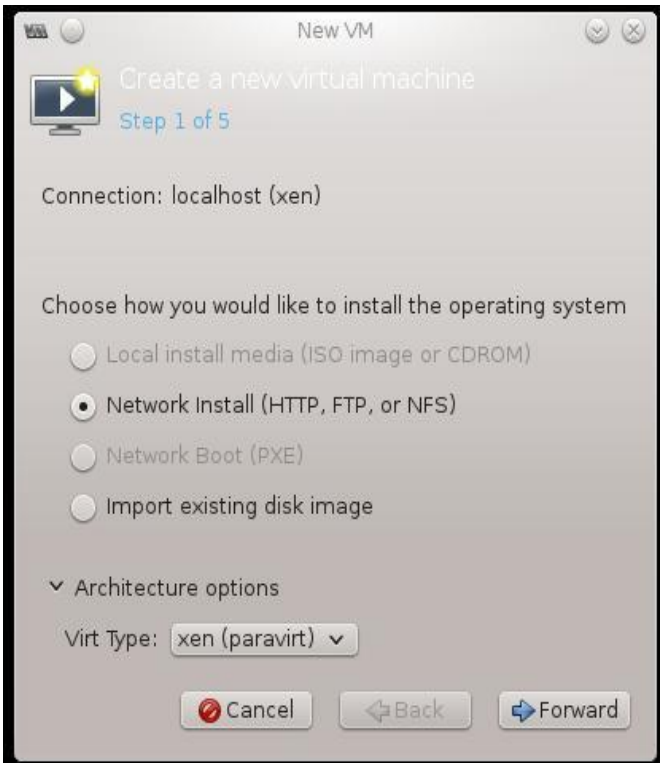
Beşinci mekanizmada ise güvenlik seviyesinin en üste çıkarılması amaçlanarak tüm VM'lerin mesh yapısına uygun olacak şekilde birbirinin güvenlik sorumluluğunu alması sağlanmaktadır. Mesh VM Koruma Mekanizması'nda her bir VM, sistemde en az iki VM tarafından korunmakta ve en az iki VM'yi korumaktadır. Her VM hem koruyan hem de korunan VM olarak davranmaktadır. Başlangıçta tüm VM'ler İyileştirilmiş AdjointVM Koruma Mekanizmasında olduğu gibi çiftler halinde birbirlerini koruyacak, ikinci bir korunan VM seçimi, VM'lerde anlık olarak elde edilen donanım kullanım miktarlarına bakılarak en az donanım harcayan VM'lerin belirlenmesiyle yapılacaktır. Sisteme yeni bir VM eklenmek istendiğinde hangi VM tarafından korunacağı yine VM'lerin donanım kullanım miktarına bakılarak verimlilik açısından en az tüketim yapan VM olarak belirlenecektir. Bu mekanizmanın oluşturulması için her bir VM üzerinde bir OSSEC sunucusu ve iki OSSEC ajanı kurulmuştur.

İlk mekanizmada belirtilen sanallaştırma ortamının üzerinde bulunduğu ana makine tüm VM'lerin güvenliğinin sağlanmasından sorumlu olduğu için genel güvenlik yapısının çalışması diğer dört mekanizmada devam ettirilmiş, bunun üzerine belirtilen yapılara ek olarak ana makineye de OSSEC sunucusu kurulmuş olup, bu sunucunun koruması altındaki ajanlar da her bir VM üzerine kurulmuştur. Gerçekleştirme ortamında ana makine "harran" olarak adlandırılmaktadır.Tüm mekanizmalardaki VM'lerin güvenliğini sağlaması için oluşturulan ana makinedeki (harran) tipiksaldırı tespit mekanizması yapısı Şekil 3.12'de verilmiştir.



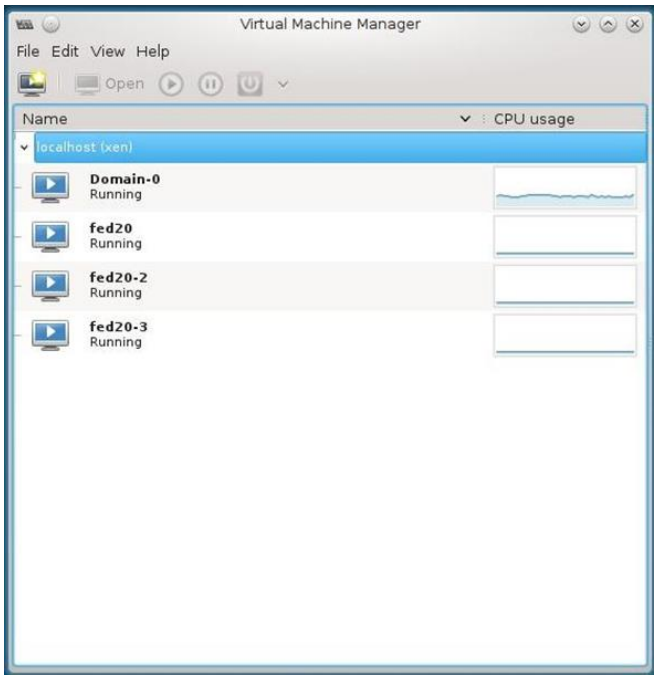
Şekil 3.12: Ana makine ve sanal makinelerin OSSEC HIDS konumlandırması.

Öncelikle ana makineye 64 bit mimariye sahip **Fedora 20** (Linux 3.19.5-100.fc20.x86_64) işletim sistemi kurulmuştur. Sanallaştırma platformu olarak Xen Hypervisor seçilmiştir. **Xen 4.3.4** versiyonu ana makineye kurularak sanallaştırma ortamı oluşturulmuştur. Ana makineye (harran – Domain-0) Xen Hypervisor ile yukarıda açıklanan tüm güvenlik mekanizmalarının modellenmesi ve test edilmesi amacıyla fed20, fed20-2 ve fed20-3 isimli üç tane VM oluşturulmuştur. Xen yarı sanallaştırma özelliği nedeniyle Şekil 3.13'te görüldüğü gibi sanallaştırma paramertesi **paravirt** seçilmiştir.Her bir VM için CPU 2 çekirdek, RAM 512 MB ve 1 GB , disk kapasitesi 8 GB yapılandırma olarak belirlenmiştir.Daha sonrasında oluşturulan her bir VM'nin Fedora 20 imajları ile konfigurasyonu yapılmıştır.



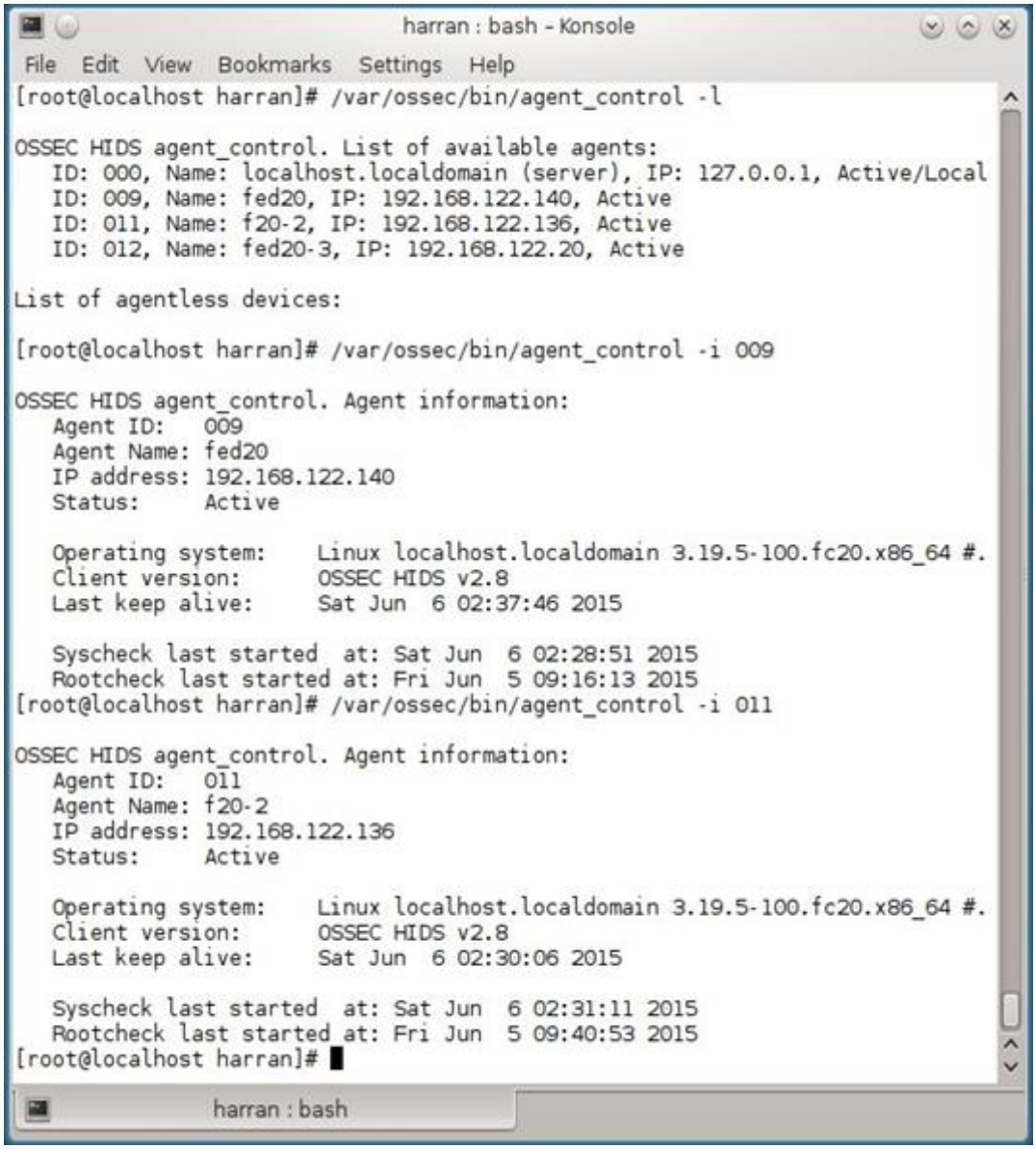
Şekil 3.13: Xen VMM ile yeni VM eklenmesi.

Şekil 3.14' te Virtual Machine Manager (VMM) arayüzü ile oluşturulan VM'ler görülmektedir.



Şekil 3.14: VMM ile VM'lerin çalışma durumu.

Genel güvenlik mekanizması için ana makineyeOSSEC sunucusu kurulmuş olup her bir VM'ye ana makinedeki OSSEC sunucusuna ait OSSEC ajanları kurulmuştur. Şekil 3.15'te ana makinedeki OSSEC sunucunun izlemesi altındaki OSSEC ajanlarının çalışma durumları verilmektedir.



Şekil 3.15: OSSEC sunucusuna ait ajanlar.

3.3.2.1. AdjointVM Koruma Mekanizması Uygulaması

AjdointVM Koruma Modelinin saldırı tespit mekanizmasının uygulanması için J. Kong [32]'un çalışmasında belirttiği üzere sistemde iki sanal makineye ihtiyaç vardır. Oluşturulan gerçekleştirme ortamında korunan VM olarak fed20-2 isimli VM, bu sanal makineyi koruyacak olan eşlenik VM yani AdjointVM ise fed20 isimli VM

yapılandırılmıştır. Birinci mekanizma olarak sistem üzerinde aktif olarak çalışan VM'lere ait yapılandırma ayarları ve VM'ler üzerinde çalışan servisler ile uygulamalar aşağıda açıklanmıştır.

Domain-0 (harran) :

- ***OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *eth0;* 192.168.2.129 (dış bacak, internet)
  virbr0; 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.1
- ***OSSEC Sunucusu:***OSSEC HIDS sunucusudur. (localhost.localdomain (192.168.122.1))
- ***Logstash***: OSSEC log verilerini ayrıştırır ve Elasticsearch üzerinde depolar
- ***Elasticsearch*:**OSSEC log verilerini indeksleme ve veri depolama sistemidir.
- ***Kibana****:* ElasticSearch ile gelen kullanıcı arayüzüdür. OSSEC alarmlarını ve log kayıtlarını görselleştirmek amacıyla kullanılmıştır.
- ***MySQL Veritabanı Sunucusu:***Hem OSSEC sunucusu log kayıtları ve alarmlarını depolaması hem de Logstash, ElasticSearch ve Kibana uygulamalarının OSSEC log kayıtlarını analiz edebilmesi ve depolaması için arka planda kullanılan veri tabanı sunucusu olarak yapılandırılmıştır.
- ***Apache Tomcat Web Sunucusu:***Web tabanlı bir uygulama olan Kibana üzerinden OSSEC alarmlarını izlemek amacıyla konfigüre edilmiştir.

fed20 (VM1):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.140
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. AdjointVM olarak diğer VM'yi izler. (localhost.localdomain (192.168.122.140))
- ***OSSEC Ajanı :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.
  Agent ID: 009
  Agent Name: fed20
  IP Address : 192.168.122.140
  OSSEC Server: 192.168.122.1

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

fed20-2 (VM2):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.136
- ***OSSEC Ajanı-1 :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 011
  Agent Name: f20-2
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.1

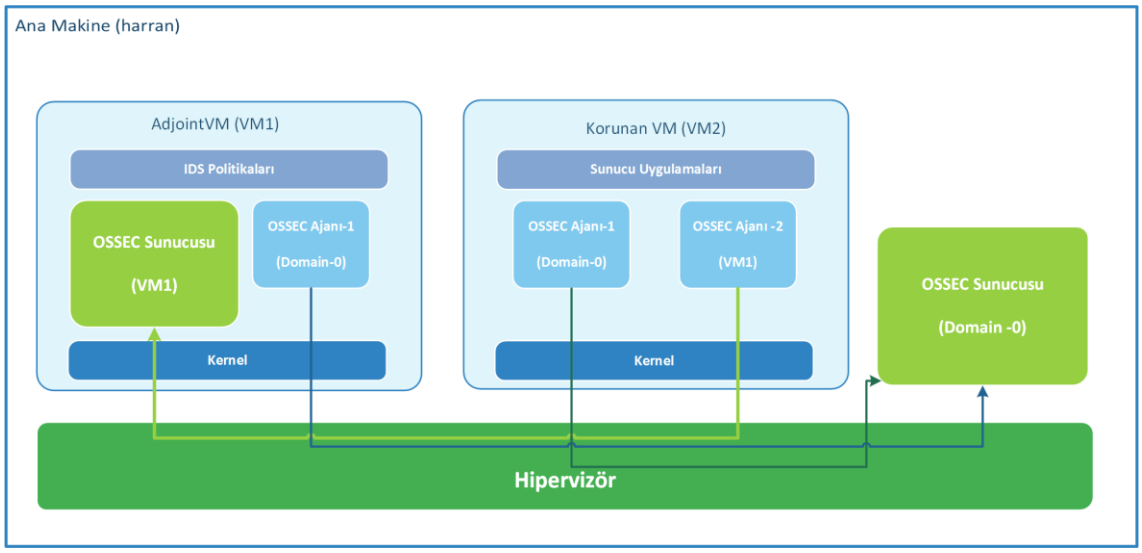
- ***OSSEC Ajanı-2 :*** fed20 (AdjointVM) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 003
  Agent Name: f20-2_agent
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.140

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

AdjointVM koruma modeline ait oluşturulan saldırı tespit sistemi güvenlik mekanizması Şekil 3.16'da gösterilmiştir.

Oluşturulan mimariye göre güvenlik mekanizmaları aktif hale getirilerek elde edilen bulgular Dördüncü Bölüm'de detaylı olarak açıklanmıştır.



Şekil 3.16: AdjointVM Koruma Mekanizması'na ait mimari.

3.3.2.2. İyileştirilmiş AdjointVM Koruma Mekanizması Uygulaması

AjdointVM Koruma Modelinin İyileştirilmesi yaklaşımına göre saldırı tespit mekanizmasının uygulanması için U. Oktay ve Ark. [35]'nın çalışmasında belirttiği üzere sistemde iki sanal makineye ihtiyaç vardır. Oluşturulan gerçekleştirme ortamında VM1 olarak fed20 isimli VM, VM2 olarak ise fed20-2 isimli VM yapılandırılmıştır. İyileştirilmiş AdjointVM koruma mekanizmasında VM çiftlerinde bir VM1, çifti olan diğer VM2'yi izlerken, VM2 de aynı şekilde VM1'i izlemektedir.İkinci mekanizma olarak sistem üzerinde aktif olarak çalışan VM'lere ait yapılandırma ayarları ve VM'ler üzerinde çalışan servisler ile uygulamalar aşağıda açıklanmıştır:

Domain-0 (harran) :

- ***OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *eth0;* 192.168.2.129 (dış bacak, internet)
  virbr0; 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.1
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. (localhost.localdomain (192.168.122.1))
- ***Logstash***: OSSEC log verilerini ayrıştırır ve Elasticsearch üzerinde depolar
- ***Elasticsearch*:**OSSEC log verilerini indeksleme ve veri depolama sistemidir.
- ***Kibana****:* ElasticSearch ile gelen kullanıcı arayüzüdür. OSSEC alarmlarını ve log kayıtlarını görselleştirmek amacıyla kullanılmıştır.

- ***MySQL Veritabanı Sunucusu:*** Hem OSSEC sunucusu log kayıtları ve alarmlarını depolaması hem de Logstash, ElasticSearch ve Kibana uygulamalarının OSSEC log kayıtlarını analiz edebilmesi ve depolaması için arka planda kullanılan veri tabanı sunucusu olarak yapılandırılmıştır.
- ***Apache Tomcat Web Sunucusu:***Web tabanlı bir uygulama olan Kibana üzerinden OSSEC alarmlarını izlemek amacıyla konfigüre edilmiştir.

fed20 (VM1):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.140
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM2'yi izler. (localhost.localdomain (192.168.122.140))
- ***OSSEC Ajanı-1:*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 009
  Agent Name: fed20
  IP Address : 192.168.122.140
  OSSEC Server IP: 192.168.122.1

- ***OSSEC Ajanı-2:*** fed20-2 (VM2) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 001
  Agent Name: fed20_agent
  IP Address : 192.168.122.140
  OSSEC Server IP: 192.168.122.136

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

fed20-2 (VM2):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64

- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.136
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM1'yi izler. (localhost.localdomain (192.168.122.136))
- ***OSSEC Ajanı-1:*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 011
  Agent Name: f20-2
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.1

- ***OSSEC Ajanı-2:*** fed20 (VM1) tarafından güvenlik altına alınması için kurulan ajan.

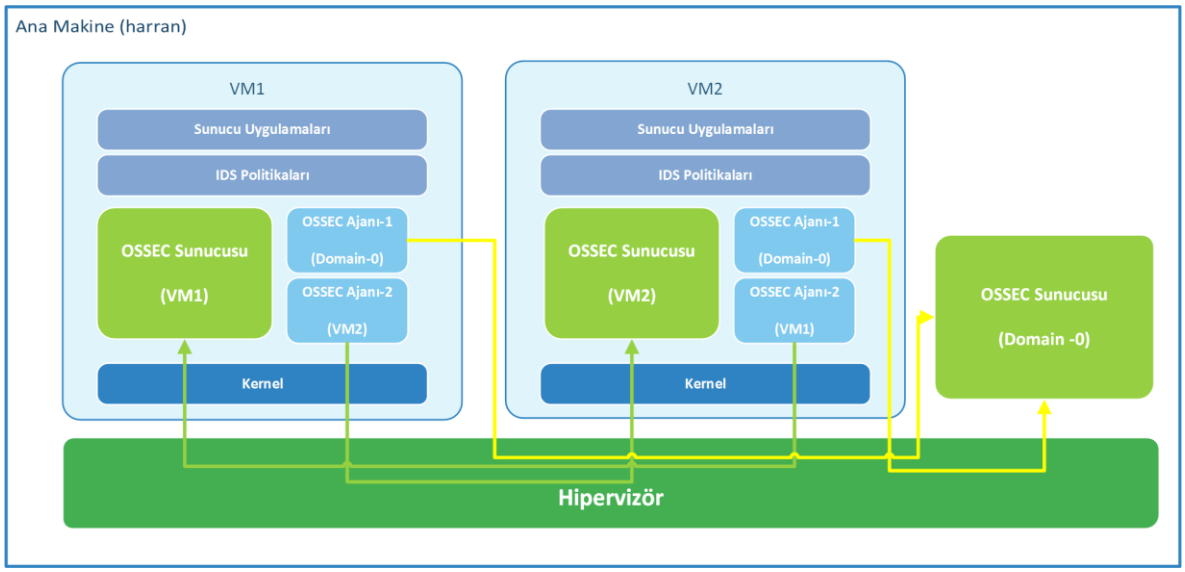
  Agent ID: 003
  Agent Name: f20-2_agent
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.140

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

İyileştirilmiş AdjointVM Koruma Modeline göre oluşturulan saldırı tespit sistemi güvenlik mekanizması Şekil 3.17'de gösterilmiştir. Oluşturulan mimariye göre güvenlik mekanizmaları aktif hale getirilmiş, elde edilen bulgular ve AdjointVM Koruma Modeli ile kıyaslanması Dördüncü Bölüm'de detaylı olarak açıklanmıştır.

3.3.2.3. Döngüsel Zincir VM Koruma Mekanizması Uygulaması

Döngüsel Zincir VM Koruma Mekanizmasının gerçekleştirilmesi için mevcut imkanlar dahilinde sistemde üç VM ile zincir yapısı modellenmiştir. Döngüsel Zincir VM Koruma Modeli'ne göre sistemdeki birinci VM yani VM1 olarak fed20 isimli VM, ikinci VM yani VM2 olarak fed20-2 isimli VM, üçüncü VM olarak yani VM3 ise fed20-3 isimli VM'dir. Bu modele göre VM1, VM2'nin güvenliğinden; VM2, VM3'ün güvenliğinden; VM3 ise VM1'in yani zincirin ilk üyesinin güvenliğinden sorumludur.



Şekil 3.17:İyileştirilmiş AdjointVM Koruma Mekanizması'na ait mimari.

Böylelikle her bir VM, zincirin bir üyesinin güvenliğini sağlar ve diğer bir üyeyi güvenli hale getirir. VM'lerin sayısının istemciler tarafından kontrol edilmektedir. Bu şekilde bir yapıyla diğer iki mekanizmaya göre sanal ortam harcaması ve VM'lerin güvenliğini sağlama çabası azaltılmıştır. Döngüsel zincir yapısı kendi güvenliğini sağlayabilir ve ek bir VM'e ihtiyaç yoktur. Bu mekanizmanın sağlanması için her bir VM üzerine hem OSSEC sunucusu hem de OSSEC ajanı kurulmuştur.Üçüncü mekanizma olarak sistem üzerinde aktif olarak çalışan VM'lere ait yapılandırma ayarları ve VM'ler üzerinde çalışan servisler ile uygulamalar aşağıda açıklanmıştır:

Domain-0 (harran) :

- ***OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *eth0;* 192.168.2.129 (dış bacak, internet)
  virbr0; 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.1
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. (localhost.localdomain (192.168.122.1))
- ***Logstash***: OSSEC log verilerini ayrıştırır ve Elasticsearch üzerinde depolar
- ***Elasticsearch*:**OSSEC log verilerini indeksleme ve veri depolama sistemidir.
- ***Kibana****:* ElasticSearch ile gelen kullanıcı arayüzüdür. OSSEC alarmlarını ve log kayıtlarını görselleştirmek amacıyla kullanılmıştır.

- ***MySQL Veritabanı Sunucusu:*** Hem OSSEC sunucusu log kayıtları ve alarmlarını depolaması hem de Logstash, ElasticSearch ve Kibana uygulamalarının OSSEC log kayıtlarını analiz edebilmesi ve depolaması için arka planda kullanılan veri tabanı sunucusu olarak yapılandırılmıştır.
- ***Apache Tomcat Web Sunucusu:***Web tabanlı bir uygulama olan Kibana üzerinden OSSEC alarmlarını izlemek amacıyla konfigüre edilmiştir.

Fed20 (VM1):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.140
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. Zincirdeki VM2'yi izler. (localhost.localdomain (192.168.122.140))
- ***OSSEC Ajanı-1:*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 009
  Agent Name: fed20
  IP Address : 192.168.122.140
  OSSEC Server IP: 192.168.122.1

- ***OSSEC Ajanı-2:*** fed20-3 (VM3) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 001
  Agent Name: fed20_agent
  IP Address : 192.168.122.140
  OSSEC Server IP: 192.168.122.20

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

Fed20-2 (VM2):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64

- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.136
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM1'yi izler. (localhost.localdomain (192.168.122.136))
- ***OSSEC Ajanı-1 :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 011
  Agent Name: f20-2
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.1

- ***OSSEC Ajanı-2 :*** fed20 (VM1) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 003
  Agent Name: f20-2_agent
  IP Address : 192.168.122.136
  OSSEC Server: 192.168.122.140

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

fed20-3 (VM3):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.20
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM1'yi izler. (localhost.localdomain (192.168.122.20))
- ***OSSEC Ajanı-1 :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 012
  Agent Name: fed20-3
  IP Address : 192.168.122.20
  OSSEC Server: 192.168.122.1

- ***OSSEC Ajanı-2 :*** fed20-2 (VM2) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 001

  Agent Name: fed20-3_agent

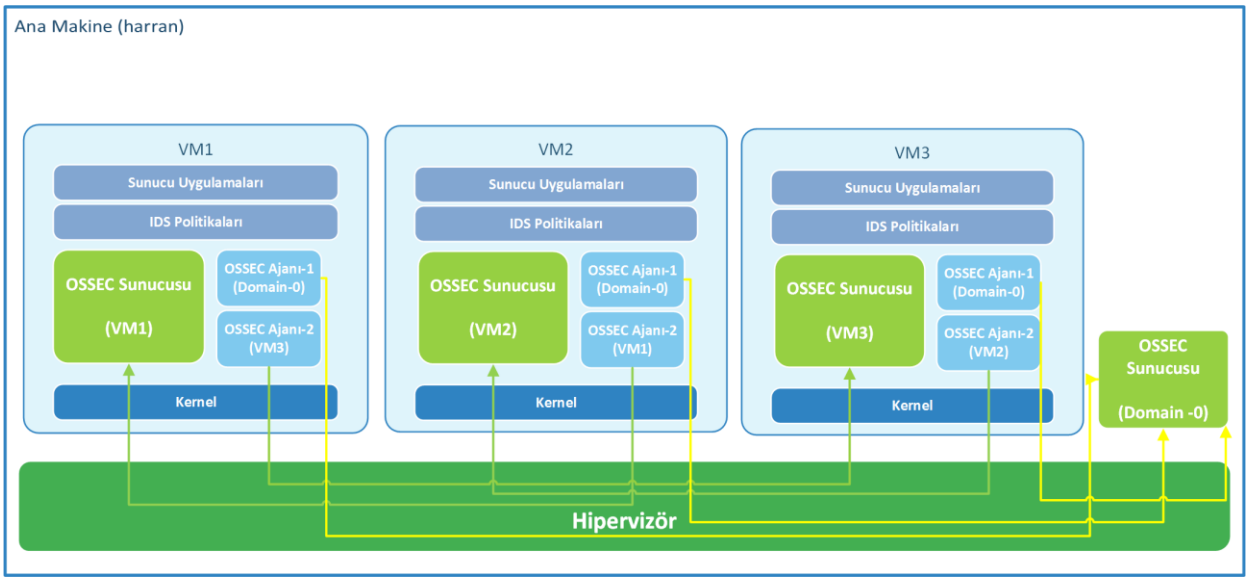
  IP Address : 192.168.122.20

  OSSEC Server: 192.168.122.136

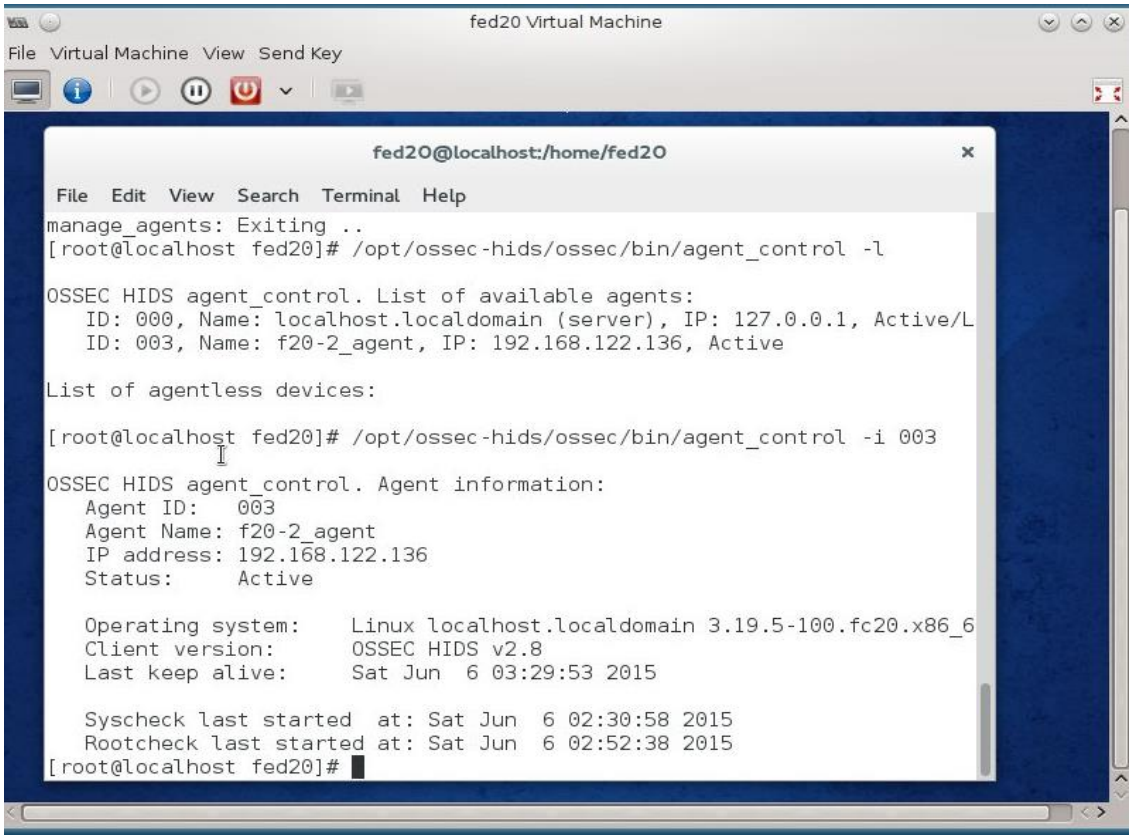
- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

Sanal makinelerin birbirini koruması için Döngüsel Zincir VM Koruma modeli, ölçeklenebilir yapısı sayesinde VM sayısının azaltılıp arttırılmasına imkan verebilmektedir. Bu sayede sistem gerçekleştirme ortamı esneklik kazanmaktadır. Döngüsel Zincir VM Koruma Modeli'ne göre gerçekleştirilen saldırı tespit mekanizmasının mimarisi Şekil 3.18'de verilmiştir.

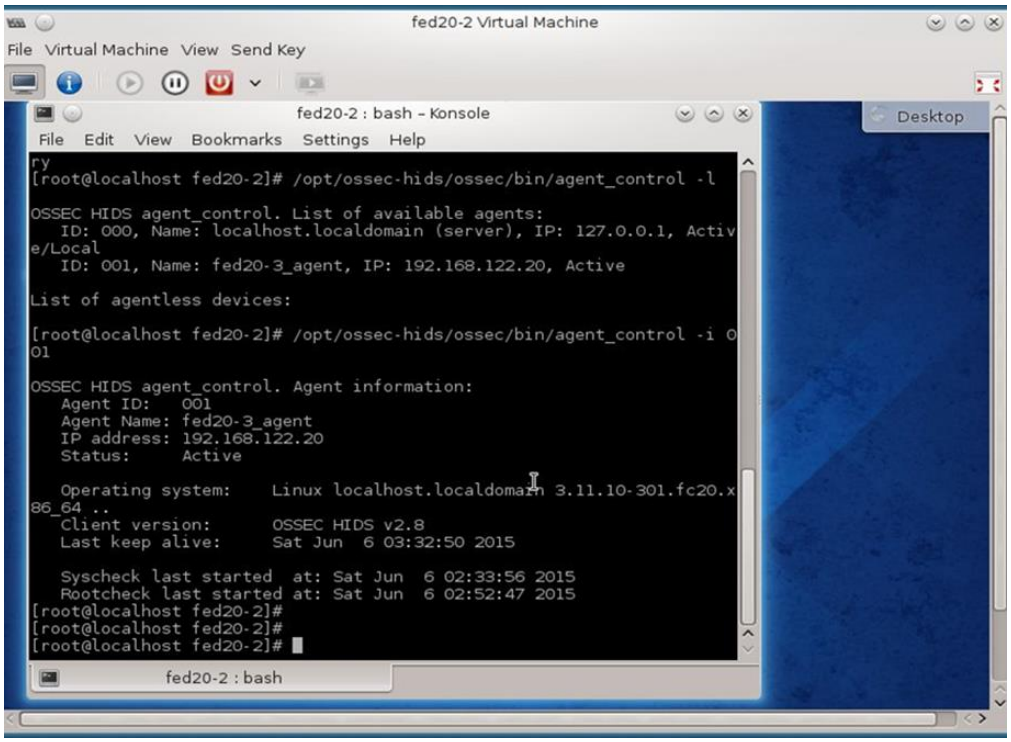
Döngüsel Zincir VM Koruma Mekanizması'ında tüm OSSEC sunucu ve ajanları aktif hale getirilmiş ve elde edilen bulgular Dördüncü Bölüm'de detaylı olarak sunulmuştur. Döngüsel Zincir VM Koruma Mekanizması'nın bileşenlerinin saldırı tespit yapılandırmalarına ve çalışma durumlarına ait ekran görüntüleri Şekil 3.19, Şekil 3.20 ve Şekil 3.21'de gösterilmiştir. Ayrıca sistemde OSSEC sunucularından gelen alarm ve log kayıtlarının analizi için kullanılan Kibana web arayüzüne ait örnek ekran alıntıları Şekil 3.22 ve Şekil 3.23'de gösterilmiştir.



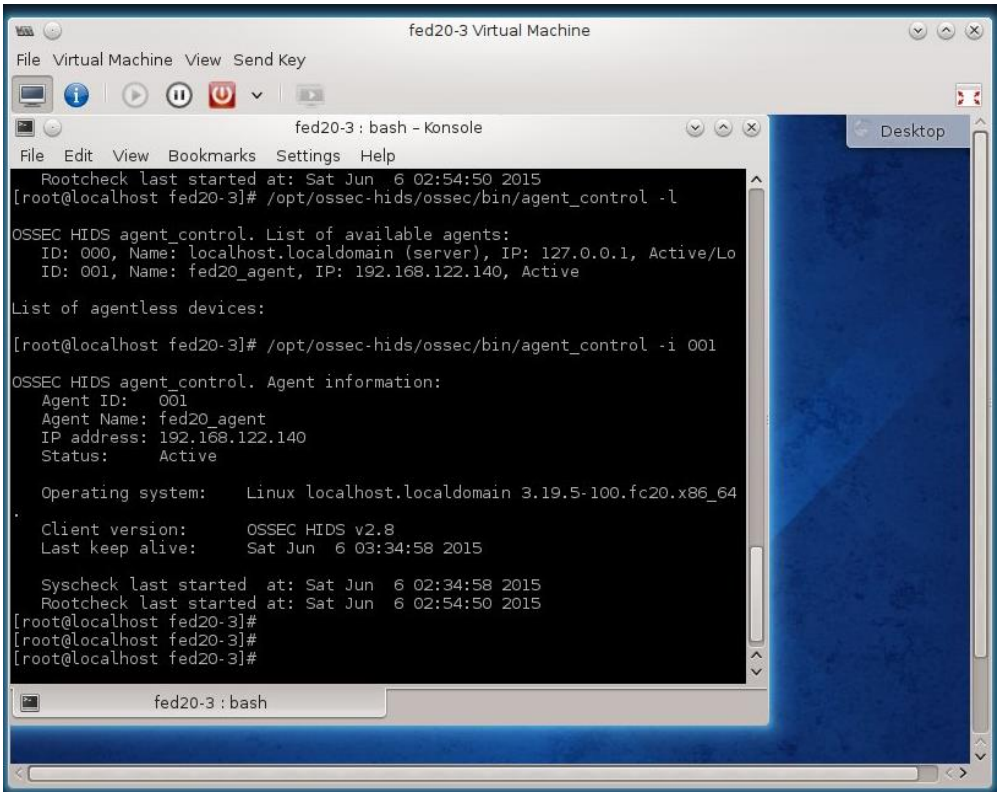
Şekil 3.18:Döngüsei Zincir VM Koruma Mekanizması mimarisi.



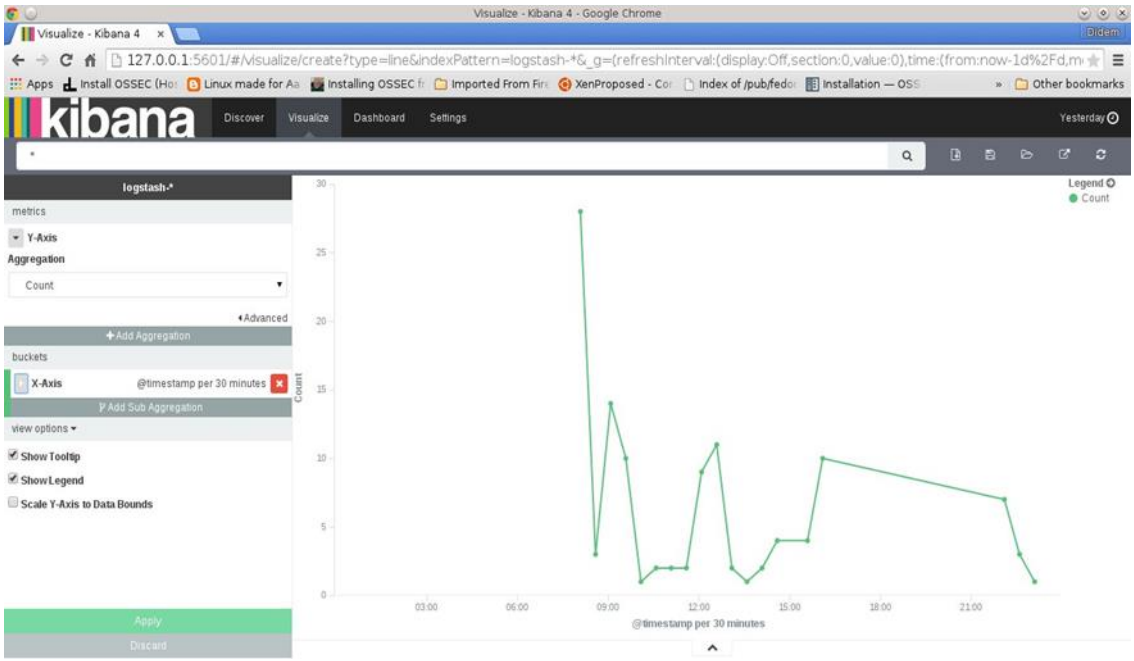
Şekil 3.19:fed20 (VM1) üzerindeki OSSEC Sunucusu çalışma durumu.



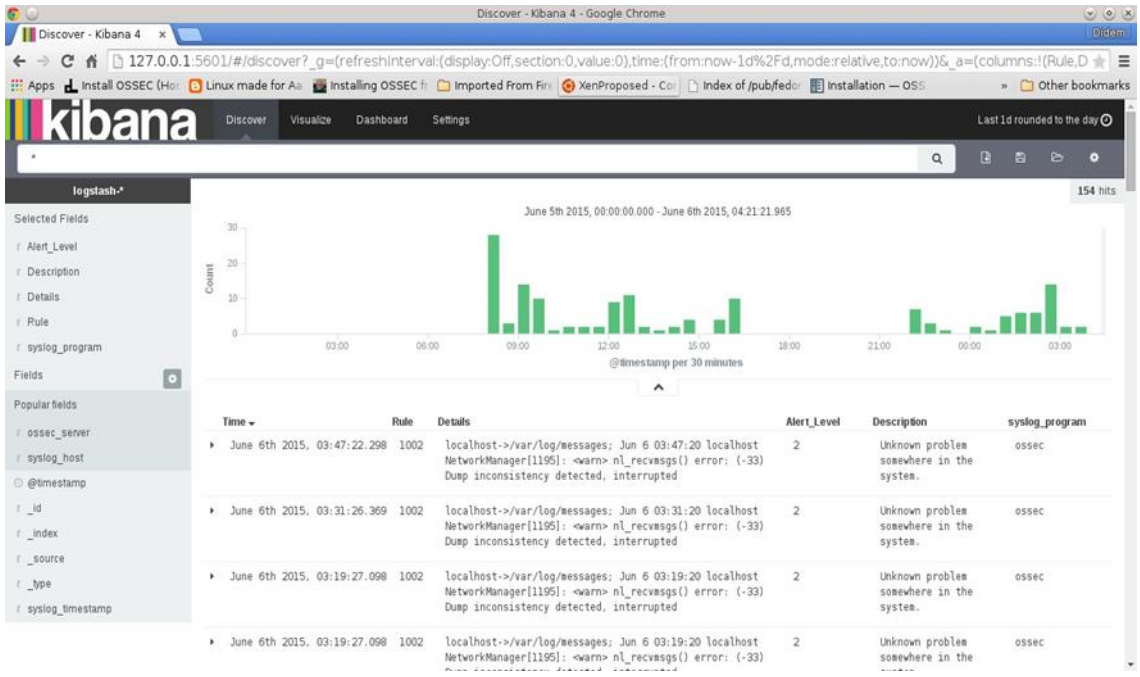
Şekil 3.20: fed20-2 (VM2) üzerindeki OSSEC Sunucusu çalışma durumu.



Şekil 3.21:fed20-3 (VM3) üzerindeki OSSEC sunucusu çalışma durumu.



Şekil 3.22: Kibana web arayüzü grafik örneği -1.



Şekil 3.23: Kibana web arayüzü grafik örneği -2.

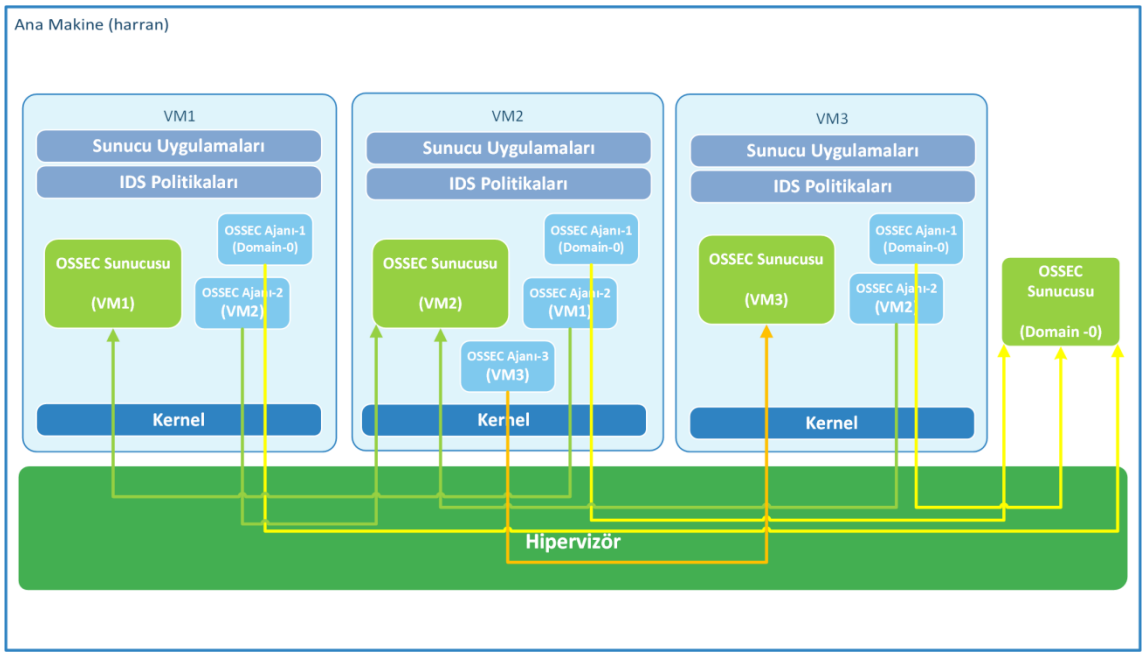
3.3.2.4. Mesh VM Koruma Mekanizması Uygulaması

Mesh VM Koruma Mekanizmasının gerçekleştirilmesinde, hem diğer mekanizmalarla test ortamında uygunluk sağlanması hem de mevcut imkanlar dahilinde sistemdeki üç VM ile model uygulanmaya çalışılmıştır.Mesh VM Koruma Modeli'ne göre sistemdeki VM1 olarak fed20 isimli VM, VM2 olarak fed20-2 isimli VM, VM3 ise fed20-3 isimli VM'dir. Bu modele göre her bir VM, en az iki VM'yi koruyacak ve kendisi de en az iki VM tarafından korunacaktır. Buna göre sistemde VM1, VM2 ve VM3'ün güvenliğinden; VM2, VM1 ve VM3'ün güvenliğinden; VM3 ise VM1 ve VM2'nin yani güvenliğinden sorumludur.

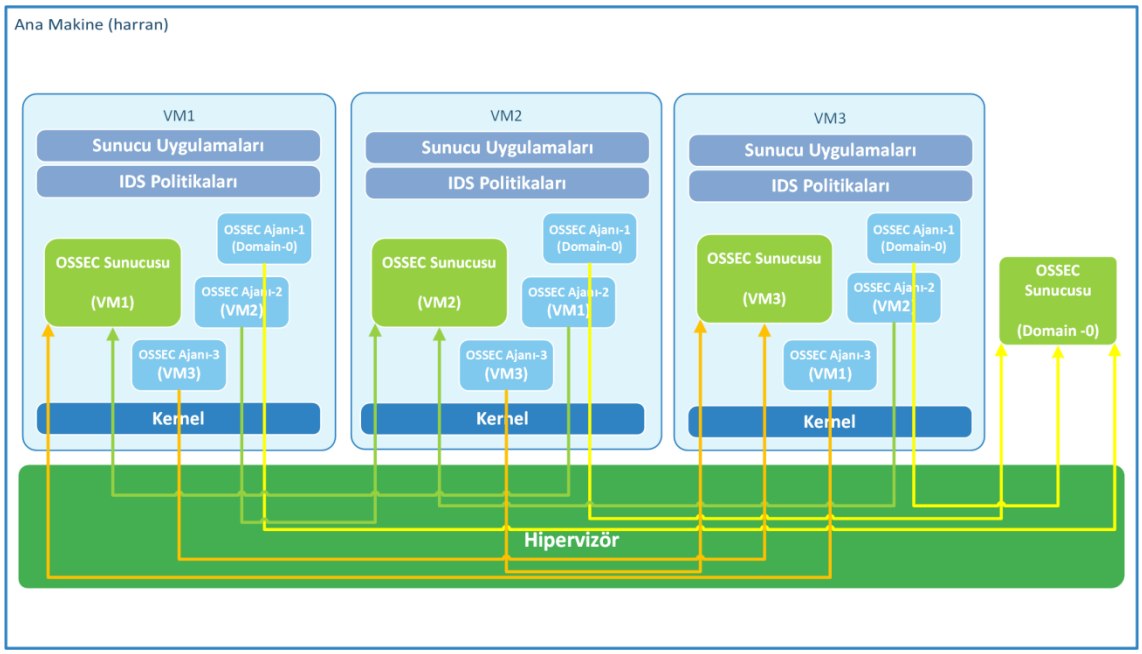
Mesh VM Koruma Mekanizması, yeni bir VM'nin sisteme eklenmesiyle güvenlik mekanizmasındaki yapılandırmanın nasıl sağlanacağını göstermek amacıyla iki aşamada gerçekleştirilmiştir. Birinci aşamada sistemde VM1 ve VM2 olmak üzere iki VM mevcut iken Mesh VM yaklaşımına göre VM'ler karşılıklı olarak birbirinin güvenlik sorumluluğunu almaktadırlar. Bu nedenle VM1 ve VM2'ye hem OSSEC sunucusu hem de OSSEC ajanı kurulmuştur. Sistemde yeni bir VM olarak VM3 oluşturulduğunda Mesh VM Koruma Mekanizmasına dahil edilebilmesi için öncelikle sistemdeki mevcut VM'lerin o andaki donanım kaynakları tüketim miktarına bakılmıştır.Hangi VM daha az meşgul ise yani daha az donanım kullanıyorsa o VM ile güvenlik ilişkisi kurulacaktır. Geliştirme ortamındaki sisteme bakıldığı zaman VM3 sorumluluğu için sistemde daha az donanım kullandığı ölçülen VM2 seçilmiş, böylelikle VM3'ü koruması için gerekli ayarlamalar yapılmıştır. Aynı şekilde VM3 de VM2'nin güvenliğini sağlaması için yapılandırılmıştır. Mesh VM Koruma Mekanizmasında birinci aşama olan yeni bir VM eklenmesi durumuna ait mimari Şekil 3.24'te gösterilmektedir.

İkinci aşamada ise Mesh VM Koruma Mekanizmasının tam olarak oluşturulması için sistemdeki tüm VM'lerin iki koruyan ve iki korunan VM ile ilişkilendirilmesi gerekmektedir. Bunun için gerçekleştirme ortamındaki VM kısıtlaması nedeniyle mevcut üç VM arasında bu güvenlik yapısı oluşturulmuştur. Öncelikle her bir VM üzerinde yeni bir OSSEC ajanı daha kurulmuştur. Daha sonra VM1 OSSEC sunucusunda, VM3'ün yeni ajanı eklenmiş, VM2 OSSEC sunucusunda VM3'ün yeni

ajanı ve VM3 OSSEC sunucusunda VM1'in yeni ajanı eklenmiştir. Oluşturulan ikinci aşama mimarisi Şekil 3.25'te verilmektedir.



Şekil 3.24: Mesh VM Koruma Mekanizması mimarisi – 1. Aşama.



Şekil 3.25: Mesh VM Koruma Mekanizması Mimarisi – 2. Aşama.

Son mekanizma olarak Mesh yapısında sistem üzerinde aktif olarak çalışan VM'lere ait yapılandırma ayarları ve VM'ler üzerinde çalışan servisler ile uygulamalar aşağıda açıklanmıştır:

Domain-0 (harran) :

- ***OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *eth0;* 192.168.2.129 (dış bacak, internet)
  virbr0; 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.1
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. (localhost.localdomain (192.168.122.1))
- ***Logstash***: OSSEC log verilerini ayrıştırır ve Elasticsearch üzerinde depolar
- ***Elasticsearch*:**OSSEC log verilerini indeksleme ve veri depolama sistemidir.
- ***Kibana****:* ElasticSearch ile gelen kullanıcı arayüzüdür. OSSEC alarmlarını ve log kayıtlarını görselleştirmek amacıyla kullanılmıştır.
- ***MySQL Veritabanı Sunucusu:*** Hem OSSEC sunucusu log kayıtları ve alarmlarını depolaması hem de Logstash, ElasticSearch ve Kibana uygulamalarının OSSEC log kayıtlarını analiz edebilmesi ve depolaması için arka planda kullanılan veri tabanı sunucusu olarak yapılandırılmıştır.
- ***Apache Tomcat Web Sunucusu:***Web tabanlı bir uygulama olan Kibana üzerinden OSSEC alarmlarını izlemek amacıyla konfigüre edilmiştir.

Fed20 (VM1):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.140
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. Zincirdeki VM2'yi izler. (localhost.localdomain (192.168.122.140))
- ***OSSEC Ajanı-1:*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 009
  Agent Name: fed20
  IP Address : 192.168.122.140
  OSSEC Server IP: 192.168.122.1

- ***OSSEC Ajanı-2:*** fed20-2 (VM2) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 001

  Agent Name: fed20_agent

  IP Address : 192.168.122.140

  OSSEC Server IP: 192.168.122.136

- ***OSSEC Ajanı-3:*** fed20-3 (VM3) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 002

  Agent Name: fed20_agent_mesh

  IP Address : 192.168.122.140

  OSSEC Server IP: 192.168.122.20

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

Fed20-2 (VM2):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.136
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM1'yi izler. (localhost.localdomain (192.168.122.136))
- ***OSSEC Ajanı-1 :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 011

  Agent Name: f20-2

  IP Address : 192.168.122.136

  OSSEC Server: 192.168.122.1

- ***OSSEC Ajanı-2 :*** fed20 (VM1) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 003

  Agent Name: f20-2_agent

IP Address : 192.168.122.136

OSSEC Server: 192.168.122.140

- ***OSSEC Ajanı-2 :*** fed20-3 (VM3) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 004

  Agent Name: f20-2_agent_mesh

  IP Address : 192.168.122.136

  OSSEC Server: 192.168.122.20

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

fed20-3 (VM3):

- ***VM üzerinde çalışan OS:*** Fedora 20 x64
- ***Ağ Konfigürasyonu:*** *virbr0;* 192.168.122.0 /24 (VLAN), *inet;* 192.168.122.20
- ***OSSEC Sunucusu:*** OSSEC HIDS sunucusudur. VM çifti olarak diğer VM1'yi izler. (localhost.localdomain (192.168.122.20))
- ***OSSEC Ajanı-1 :*** Domain-0 tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 012

  Agent Name: fed20-3

  IP Address : 192.168.122.20

  OSSEC Server: 192.168.122.1

- ***OSSEC Ajanı-2 :*** fed20-2 (VM2) tarafından güvenlik altına alınması için kurulan ajan.

  Agent ID: 001

  Agent Name: fed20-3_agent

  IP Address : 192.168.122.20

  OSSEC Server: 192.168.122.136

- ***OSSEC Ajanı-3 :*** fed20 (VM1) tarafından güvenlik altına alınması için kurulan ajan.

Agent ID: 002
Agent Name: fed20-3_agent_mesh
IP Address : 192.168.122.20
OSSEC Server: 192.168.122.140

- ***MySQL Veritabanı Sunucusu:*** OSSEC sunucusu log kayıtları ve alarmlarını depolaması için veri tabanı sunucusu olarak yapılandırılmıştır.

Mesh VM Koruma Mekanizması'ında her iki aşama için ayrı ayır tüm OSSEC sunucu ve ajanları aktif hale getirilmiş ve elde edilen bulgular Dördüncü Bölüm'de detaylı olarak sunulmuştur.

4. BULGULAR

Bu tez çalışması kapsamında oluşturulan tüm güvenlik mekanizmalarının test işlemlerinde bulut bilişimde çok önemli bir kriter olan kaynak kullanım miktarları açısından elde edilen bulguların değerlendirilmesi ve böylelikle sistemlerin verimliğin ölçülmesi hedeflenmiştir. Hem her bir güvenlik mekanizmasının çalıştırılması durumunda hem de sistemde hiçbir güvenlik mekanizmasının yer almadığı durumdadonanım kaynakları olarak sistemdeki işlemci (CPU) ve hafıza (RAM) kullanım oranları elde edilmiş ve birbiriyle kıyaslanmıştır. Böylelikle hangi güvenlik mekanizmasının tercih edileceği hakkında bulut kullanıcılarına yol göstermesi amaçlanmıştır.

4.1. CPU KULLANIM MİKTARININ ÖLÇÜLMESİ

Güvenlik mekanizmalarının denenmesi sırasında ana makine ve VM'lerin kullandığı CPU kaynaklarının kullanım miktarlarının ölçülebilmesi için Xen sanallaştırma platformunun sunduğu "***xentop***" isimli sanallaştırma ortam monitör araç yazılımı kullanılmıştır.*xentop,* bir Xen sistemi hakkında gerçek zamanlı bilgi vermektedir. Bu sayede sanallaştırma platformunda yer alan tüm çalışan VM'ler ve sistem ile ilgili anlık bilgilerin izlenebilmesini sağlamaktadır. Sistem hakkında elde edilen bilgileri yarıgrafiksel bir arayüz ile kullanıcılara sunmaktadır. Şekil 4.1'de örnek bir *xentop* ekran görüntüsü bulunmaktadır. *xentop* ekranında yer alan bilgiler ve sütun açıklamaları aşağıda verilmiştir:

- *CPU(sec)* – Saniye cinsinden Domain CPU kullanımını veriri.
- *CPU(%)* – CPU kullanımının yüzde miktarını verir.
- *VCPUS*–Sanal CPU'ların (virtual CPUs) sayısını verir.
- *NETS*–Sanal ağların sayısını verir.
- *MEM*–Geçerli hafıza miktarını verir.
- *MAXMEM(k)*– Kilobayt (KB) cinsinden en yüksek domain hafıza bilgisini verir.
- *MAXMEM(%)*–Hafıza yüzde bilgisini, geçerli domain hafızasının toplam düğüm hafızasına oranı verir.

- *NETTX*–Ağdaki toplam iletilen (tx) bayt istatistiği/1024 değerini verir.
- *NETRX* – Ağdaki toplam alınan (tx) bayt istatistiği/1024 değerini verir.
- *VBDS*–Sanal blok aygıtlarının sayısını verir.
- *VBD OO*–Toplam VBD OO istek sayısını verir. Bu sayı VBD nin kaç defa istek dışı hatasını aldığını gösterir. Bu hata meydana geldiğinde VBD için I/O istekleri gecikme yaşar.
- *VBD_RD*– VBD okuma isteği sayısını verir.
- *VBD_WR* – VBD yazma isteği sayısını verir.

Domainlerin olası durum bilgileri şöyledir:

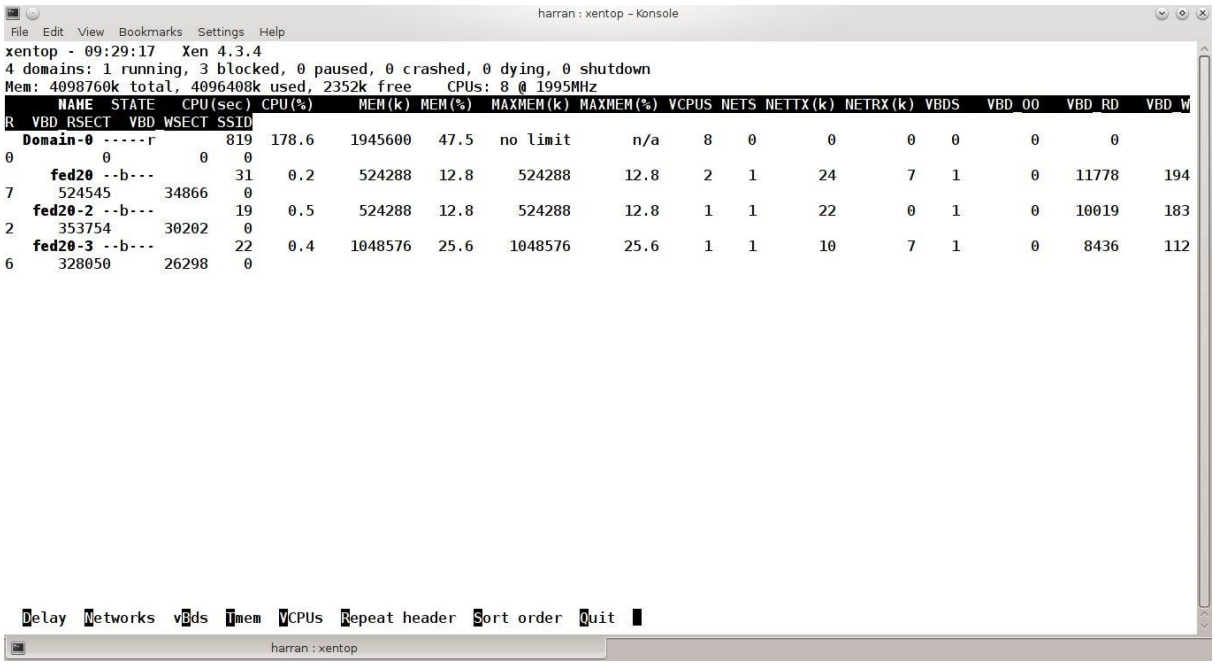
- d - domain ölüyor
- s – domain kapatılıyor
- b – engellenen domain
- c – domain çöktü
- p – domain beklemede
- r – domain aktif olarak bir CPU üzerinde çalışıyor.

xentop yazılımında varsayılan olarak 3 saniye aralıklarla donanım kullanım miktarları gösterilmektedir. Sistemden farklı zaman dilimlerinde elde edilen verilerin çeşitliğini arttırmak amacıya *xentop* yazılımını –d opsiyonuyla çalıştırarak donanım kullanım durumu güncelleme sıklığını 1 saniye ve 10 saniye olarak değiştirilmiş olup, her bir zaman aralığı için 50 iterasyon yapılarak verilerin elde edilmesi için çalıştırılmıştır. Bu işlem 1, 3 ve 10 saniye zaman dilimleri için birden fazla kez tekrarlanmıştır.Bilgilerin elde edilmesi için *xentop* yazılımı çalıştırma komutları aşağıda verilmiştir:

 # xentop –i 50 –d 1 –b

 # xentop –i 50 –d 10 –b

 # xentop –i 50 –d 3 –b



Şekil 4.1:*xentop* yazılımı ekran görüntüsü.

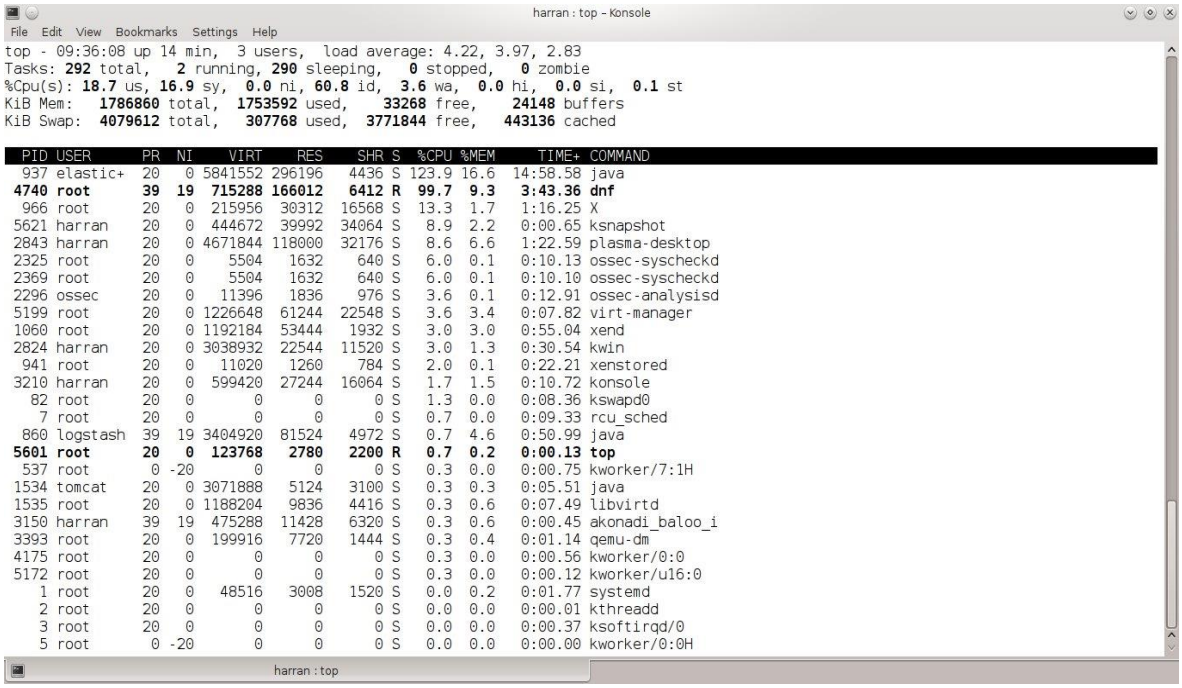
Yukarıda verilen örnek komutta *xentop* yazılımı –i parametresiyle iterasyon sayısı 50 olarak belirlenmiş, –d parametresi ile gecikme (delay) süresi 1, 3 ve 10 saniye olarak verilmiş olup –b parametresiyle arka planda (batch modda) çalıştırılmıştır.

4.2. RAM KULLANIM MİKTARININ ÖLÇÜLMESİ

Oluşturulan deneme ortamında güvenlik mekanizmalarının ayrı ayrı uygulaması gerçekleştirildiğinde ana makine ve VM'lerintükettiğihafıza yani RAM kaynaklarının kullanım miktarlarının ölçülebilmesi için Fedora OS ile varsayılan olarak sunulan "*top*" isimli sistem araç yazılımı kullanılmıştır.*top*, sistem üzerinde çalışmakta olan işlemlerin gerçek zamanlı listesini görüntüler. Aynı zamanda sistem çalışma süresi, anlık olarak mevcut CPU ve hafıza kullanımı, ya da çalışan işlemlerin toplam sayısı hakkında ek bilgileri görüntüler ve kullanıcıların işlem listesi üzerinde sıralama ya da çalışan bir işlemi öldürme gibi eylemleri gerçekleştirmesine imkan verir. Şekil 4.2'de ana makine üzerinde çalıştırılan *top* yazılımına ait örnek bir ekran görüntüsü verilmekte olup bu yazılımın çalıştırılması ile elde edilen bilgiler ve kullanılan parametreler aşağıda açıklanmaktadır:

Ekranda ilk satırda görülen "*09:36:08*", *top* komutunun çalıştırıldığı andaki zaman bilgisini, "*up 14 min*" sistemin ne kadar süredir çalışmakta olduğunu, "*3 users*" kaç kullanıcının sistem üzerinde giriş yapmış olduğunu, "*load avarage: 4.22, 3.97, 2.83*"

sistem üzerindeki 1 dakika, 5 dakika ve 15 dakikalık yük durumunu göstermektedir. Sistem üzerindeki yük durumu, belli periyottaki ortalama CPU kullanım miktarını gösteren bilgidir.



Şekil 4.2: Ana makine üzerindeki *top* ekran görüntüsü.

İkinci satırda bulunan "*tasks*" kısmı, çalışmakta olan toplam işlem sayısını ve çalışan, uyuyan, durmuş ve kullanılmayan olarak o anki işlem durumlarını gösterir.

Üçüncü satır CPU kullanım ayrıntılarını göstermektedir. "*18.7 us*" kullanıcılara ait işlem miktarı yüzdesini, "*16.9 sy*" sisteme ait işlem miktarı yüzdesini, "*60.8 id*" kullanılabilir CPU miktarı yüzdesini gösterirken, "*3.8 wa*" iseCPU'nun girdi/çıktı (intput/output – IO) için bekleme zamanını yüzde olarak göstermektedir.

Dört ve beşinci satırlar toplam kullanılabilir hafıza, boş hafıza, kullanılan hafıza, paylaşımlı hafıza ve tamponlar için kullanılan hafıza da dahil olmak üzere hafıza kullanımıyla ilgili istatistikleri göstermektedir. Dördüncü satırda fiziksel, beşinci satırda ise sanal hafıza bilgileri yer almaktadır. Hafıza bilgileri ekranına **m** interaktif komutu ile geçilebilmektedir.Hafıza bilgileri kilobayt cinsinden olup ekranda görüldüğü gibi sırasıyla "*1786868 total*" toplam hafıza miktarını, "*1753592 used*" kullanılan hafıza miktarını, "*33288 free*" hafıza üzerindeki boş miktarı, "*24148 buffers*" tampon hafıza

miktarını ve "*443136 cached*" önbellek olarak kullanılan hafıza miktarını göstermektedir.

Kullanılabilecek boşta olan RAM miktarı şu şekilde elde edilmektedir:

Boş RAM = free + (buffers + cached)

= 33288k + (24148k + 443136k)

= 500572k

Programlar tarafından kullanılan RAM miktarı ise şu şekilde hesaplanmaktadır:

Kullanılan RAM = used – (buffers + cached)

= 1753592k – (24148k + 443136k)

= 1286308k

top, CPU'yu en fazla kullanan işlemleri yukarıdan aşağı azalan sırayla göstermektedir. *top* ekranında bulunan sütun bilgileri aşağıda verilmiştir.

- *PID* – Açılımı Process Identification (ID) yani işlem numarasıdır. Her işleme ait farklı işlem numaraları bulunur.
- *USER* – İşlemi çalıştıran kullanıcıyı belirtir.
- *PR* – İşlemin önem seviyesini belirten öncelik sırasını gösterir.
- *NI* – İşlemin öncelik sırasını değiştirmek için kullanılır. (negatif bir değerişlemin öncelik sırasını arttırır, pozitif bir değer ise düşürür.)
- *VIRT* – Kullanılan sanal hafıza miktarını gösterir.
- *RES* – İşlem tarafından kullanılan yerleşik hafıza miktarını kilobayt (kb) cinsinden ifade eder.
- *SHR* – İşlem tarafından kullanılan paylaşımlı hafıza miktarını göstermektedir.
- *S* – İşlemin durumunu belirtir. ('S'= sleeping, 'D'= uninterruptible sleep, 'R'= running, 'Z'= zombies, 'T'= stopped veya traced)
- *%CPU* – İşlemin CPU'yu ne kadar kullandığını yüzde olarak belirtir.
- *%MEM* – İşlemin hafızayı ne kadar kullandığını yüzde olarak belirtir.

- *TIME+* – CPU'nun toplam kullanım zamanını gösterir.
- *COMMAND* – işlemin komutunu veya kullandığı servisi gösterir.

Bu şekilde ifade edilen bilgileri elde edebilmek amacıyla her bir güvenlik mekanizmasında hem ana makinede hem de sistemde yer alan tüm VM'lerde *top*yazılımı ayrı ayrı çalıştırılmıştır. *top* yazılımında da varsayılan olarak 3 saniye aralıklarla donanım kullanım miktarları gösterilmektedir. Daha gerçekçi veriler elde etmek amacıyla *top* yazılımının çalıştırma komutundagecikme (delay – d) parametresi kullanılmıştır.Yani sistemden farklı zaman dilimlerinde elde edilen verilerin çeşitliğini arttırmak amacıya *top* yazılımını –d opsiyonuyla çalıştırarak donanım kullanım durumu güncelleme sıklığını 1 saniye, 3 saniye ve 10 saniye olarak değiştirilmiş olup, her bir zaman aralığı için 50 iterasyon yapılarak verilerin elde edilmesi için çalıştırılmıştır. Bu işlem 1, 3 ve 10 saniye zaman dilimleri için birden fazla kez tekrarlanmıştır. Bilgilerin elde edilmesi için ana makinede kullanılan *top* yazılımı çalıştırma komutları aşağıda verilmiştir.

 # top – b –d 1 –n 50 – p 2296

 # top – b –d 1 –n 50 – p 2296

 # top – b –d 1 –n 50 – p 2296

Yukarıda verilen örnek komutta *top* yazılımı–b parametresiyle arka planda (batch modda) çalıştırılmış, –d parametresi ile gecikme (delay) süresi 1, 3 ve 10 saniye olarak verilmiş, –i parametresiyle iterasyon sayısı 50 olarak belirlenmişolup –p parametresiyle de işleme ait PID numarası verilerek hangi işlemin izleneceğini belirterek o işleme ait bilgilerin ekranda listelenmesi sağlanmıştır. Örnek komutta 2296 PID numarasına sahip *ossec-analysisd* isimli işlem izlenmektedir.

Ana makinede *top* yazılımının çalıştırılmasıyla mevcut donanıma ait hafıza kaynağının ne kadarını kullandığı ölçülürken her bir VM üzerinde *top* yazılımı ayrı ayrı çalıştırılarak VM'nin kendisine tahsis edilen hafıza bölümünün ne kadarını kullandığı tespit edilmiştir. VM için elde edilen değerler tüm sistem donanımına kıyasla elde edilen kullanım miktarı olmayıp sadece VM için tahsis edilen kaynakların kullanım miktarını göstermektedir.

4.3. ELDE EDİLEN BULGULARIN DEĞERLENDİRİLMESİ

Oluşturulan test ortamında sistemde beş ayrı güvenlik mekanizması ve hiçbir mekanizmanın kullanılmadığı başlangıç durumu olmak üzere toplamda altı ayrı durum için CPU ve RAM kullanım bilgilerielde edilmiş ve değerlendirilmiştir.

Birinci durumda, başlangıç olaraksistemde herhangi bir güvenlik mekanizması olmaksızın ana makine ve VM'lerin tükettiği CPU ve RAM kaynak miktarları ölçülmüştür. Oluşturulacak güvenlik mekanizmaları açısından sistemdeki VM sayısı değişkenlik gösterdiğinden hiçbir güvenlik mekanizması kullanılmayan durum da iki alt aşama içermektedir. İlk aşamada AdjointVM ve İyileştirilmiş AdjointVM Koruma Mekanizmalarına ait modeller ile benzerlik sağlanması amacıyla sistemde Domain-0 olarak adlandırılan ana makine ile fed20 ve fed20-2 isimli iki VM yer alırken, ikinci aşamada Döngüsel Zincir VM Koruma Mekanizması ve Mesh VM Koruma Mekanizması ile aynı ölçekteki sistem oluşturularak Domain-0 ile fed20, fed20-2 ve fed20-3 isimli üç VM yer almaktadır.

İkinci durumda, sistemde sadece ana makine üzerine OSSEC sunucusu kurulmuş olup, mevcut her bir VM üzerine kurulan OSSEC ajanları sayesinde ana makine tüm VM'leri izlemektedir. Bu şekildeki koruma mekanizmasına ait CPU ve RAM kaynak kullanımları ölçülmüştür.

Üçüncü durumda, sistem üzerinde kurulan AdjointVM Koruma Mekanizması için CPU ve RAM tüketim miktarları ölçülmüştür.

Dördüncü durumda, İyileştirilmiş AdjointVM Koruma Mekanizması için CPU ve RAM kaynak kullanım miktarları ölçülmüştür.

Beşinci durumdaDöngüsel Zincir VM Koruma Mekanizması aktifken CPU ve RAM kullanım miktarları ölçülerek bilgiler elde edilmiştir.

Altıncı durumda ise Mesh VM Koruma Mekanizmasına ait üçüncü bölümde bahsedilen iki aşama halinde oluşturulan yapıdaki CPU ve RAM kullanım miktarları ölçülerek bilgiler elde edilmiştir.

Her bir durum için ayrı ayrı donanım kaynakları olarak CPU ve RAM kullanım miktarları metin (text – .txt) formatında sistem üzerinden alındıktan sonra, donanım

kullanım miktarları aşağıda verilen tablo ve grafiklere aktarılmıştır. Sistemlerden elde edilen değerlerin daha güvenilir ve doğruluğunun daha yüksek olabilmesi için bir önceki bölümde de bahsedildiği üzere 1 saniye, 3 saniye ve 10 saniye aralıklarla ölçümler yapılmış; hem bu farklı frekans aralıklarındaki ölçüm değerlerinin ayrı ayrı ortalaması hem de tüm değerlerin genel ortalaması hesaplanmış olup ilk üç durum için elde edilen tüm sonuçlar detaylı olarak Tablo 4.1, Tablo 4.2, Tablo 4.3 ve Tablo 4.4'te gösterilmiştir. Dördüncü, beşinci ve altıncı durumlar için ise sonuçlar Tablo 4.5, Tablo 4.6, Tablo 4.7 ve Tablo 4.8'de detaylı olarakverilmiştir.

Tablo 4.1: Farklı frekanslarda CPU kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 1

Güvenlik Mekanizmaları	CPU Kullanımı (%)								
	1 sn aralık			3 sn aralık			10 sn aralık		
	Domain-0	fed20	fed20-2	Domain-0	fed20	fed20-2	Domain-0	fed20	fed20-2
OSSEC HIDS Kullanılmazken	22,816	5,330	0,346	22,218	1,956	0,336	22,418	8,714	0,372
AdjointVM Koruma Mekanizması	22,736	9,792	7,978	**22,597**	8,968	7,734	**22,922**	9,768	7,932
İyileştirilmiş AdjointVM Koruma Mekanizması	**24,213**	**10,751**	**10,669**	22,393	**11,658**	**12,574**	22,601	**12,647**	**12,381**

Tablo 4.2: Farklı frekanslarda RAM kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 1

Güvenlik Mekanizmaları	RAM Kullanımı (%)								
	1 sn aralık			3 sn aralık			10 sn aralık		
	Domain-0	fed20	fed20-2	Domain-0	fed20	fed20-2	Domain-0	fed20	fed20-2
OSSEC HIDS Kullanılmazken	54,578	59,975	39,669	54,946	57,106	39,358	54,910	57,720	39,436
AdjointVM Koruma Mekanizması	56,114	**73,125**	41,117	55,977	**73,095**	40,984	56,127	**72,696**	40,606
İyileştirilmiş AdjointVM Koruma Mekanizması	**58,996**	62,571	**44,850**	**59,234**	63,596	**44,183**	**59,924**	63,805	**44,215**

Tablo 4.3: Ortalama CPU kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 1

Güvenlik Mekanizmaları	CPU Kullanımı (%)		
	Domain-0	**fed20**	**fed20-2**
OSSEC HIDS Kullanılmazken	22,484	5,333	0,351
AdjointVM Koruma Mekanizması	22,752	9,509	7,881
İyileştirilmiş AdjointVM Koruma Mekanizması	**23,069**	**11,685**	**11,875**

Tablo 4.4: Ortalama RAM kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 1

Güvenlik Mekanizmaları	RAM Kullanımı (%)		
	Domain-0	**fed20**	**fed20-2**
OSSEC HIDS Kullanılmazken	54,811	58,267	39,487
AdjointVM Koruma Mekanizması	56,073	**72,972**	40,902
İyileştirilmiş AdjointVM Koruma Mekanizması	**59,385**	63,324	**44,416**

Tablo 4.5: Farklı frekanslarda CPU kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 2

Güvenlik Mekanizmaları	CPU Kullanımı (%)											
	1 sn aralık				3 sn aralık				10 sn aralık			
	Domain-0	fed20	fed20-2	fed20-3	Domain-0	fed20	fed20-2	fed20-3	Domain-0	fed20	fed20-2	fed20-3
OSSEC HIDS Kullanılmazken	25,953	0,388	0,410	3,846	26,769	0,300	0,429	3,565	25,376	0,318	0,614	0,626
Sadece Ana Makinede OSSEC HIDS Kullanılıyor	26,399	1,775	1,765	4,648	26,735	2,059	1,691	3,404	25,896	5,008	3,868	6,316
Döngüsel Zincir VM Koruma Mekanizması	26,932	13,456	13,953	**23,358**	26,492	10,322	13,228	**22,154**	27,806	11,797	13,433	15,738
Mesh VM Koruma Mekanizması - 1. aşama	26,407	9,080	12,640	6,854	25,647	9,994	13,508	9,708	25,569	9,948	14,484	9,882
Mesh VM Koruma Mekanizması - 2. aşama	**31,289**	**17,082**	**16,773**	18,522	**30,488**	**17,743**	**16,584**	18,022	**30,081**	**19,400**	**18,641**	**20,428**

Tablo 4.6: Farklı frekanslarda RAM kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 2

Güvenlik Mekanizmaları	RAM Kullanımı (%)											
	1 sn aralık				3 sn aralık				10 sn aralık			
	Domain-0	fed20	fed20-2	fed20-3	Domain-0	fed20	fed20-2	fed20-3	Domain-0	fed20	fed20-2	fed20-3
OSSEC HIDS Kullanılmazken	65,701	58,394	35,496	39,359	65,704	58,280	41,200	39,326	65,432	58,168	37,155	38,963
Sadece Ana Makinede OSSEC HIDS Kullanılıyor	72,638	62,481	42,637	**43,633**	72,759	62,333	44,673	**44,069**	72,909	62,193	42,775	**43,945**
Döngüsel Zincir VM Koruma Mekanizması	74,301	63,621	44,426	42,514	74,195	64,819	44,282	42,838	73,928	64,492	45,387	42,976
Mesh VM Koruma Mekanizması - 1. aşama	73,391	**65,053**	47,878	42,700	73,362	**65,276**	47,828	42,641	73,131	**65,000**	47,594	42,472
Mesh VM Koruma Mekanizması - 2. aşama	**77,924**	61,285	**49,330**	41,979	**77,951**	61,148	**49,125**	41,839	**77,424**	60,980	**48,970**	41,813

Tablo 4.7: Ortalama CPU kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 2

Güvenlik Mekanizmaları	CPU Kullanımı (%)			
	Domain-0	fed20	fed20-2	fed20-3
OSSEC HIDS Kullanılmazken	26,032	0,335	0,484	2,679
Sadece Ana Makinede OSSEC HIDS Kullanılıyor	26,343	2,947	2,441	4,789
Döngüsel Zincir VM Koruma Mekanizması	27,077	11,858	13,538	**20,417**
Mesh VM Koruma Mekanizması - 1. aşama	25,8741	9,674	13,544	8,815
Mesh VM Koruma Mekanizması - 2. aşama	**30,619**	**18,075**	**17,333**	18,991

Tablo 4.8: Ortalama RAM kullanımlarına göre güvenlik mekanizmalarının kıyaslanması - 2

Güvenlik Mekanizmaları	RAM Kullanımı (%)			
	Domain-0	**fed20**	**fed20-2**	**fed20-3**
OSSEC HIDS Kullanılmazken	65,612	58,281	37,950	39,216
Sadece Ana Makinede OSSEC HIDS Kullanılıyor	72,769	62,336	43,362	**43,882**
Döngüsel Zincir VM Koruma Mekanizması	74,141	64,310	44,698	42,776
Mesh VM Koruma Mekanizması - 1. aşama	73,295	**65,110**	47,767	42,604
Mesh VM Koruma Mekanizması - 2. aşama	**77,766**	61,138	**49,142**	41,877

4.3.1. Sistemde OSSEC HIDS Kullanılmaması Durumu (1. Durum)

Sistemde OSSEC HIDS kullanılmıyorken yani güvenlik mekanizması mevcut değilken önceki bölümde açıklandığı şekilde iki aşamada değerler elde edilmiştir. İlk aşamada sistemde iki VM varken ana makine (Domain-0) 1 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %22,82'sini, hafızanın ise %54,58'ini; 3 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %22,22'sini, hafızanın ise %54,95'ini;10 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %22,42'sini, hafızanın ise %54,91'ini kullanmıştır. Tüm bu değerlerin ortalaması alındığında sistemde hiçbir güvenlik mekanizması bulunmuyorken Domain-0 CPU'nun yaklaşık olarak %22,48'ini, hafızanın ise yaklaşık olarak %54,81'ini kullanmıştır. VM'lerin sistemde kendilerine tahsis edilen donanım kaynaklarını kullanım miktarlarına bakıldığında fed20 isimli VM 1 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %5,33'ünü, hafızanın ise %59,98'ini; 3 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %1,96'sını, hafızanın ise %57,12'sini;10 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %8,71'ini, hafızanın ise %57,72'sini kullanmıştır. Tüm bu değerlerin ortalaması alındığında sistemde hiçbir güvenlik mekanizması bulunmuyorken fed20, CPU'nun yaklaşık olarak %5,33'ünü, hafızanın ise yaklaşık olarak %58,27'sini kullanmıştır.

Aynışekilde fed20-2 isimli VM'nin 1, 3 ve 10 saniye aralıklarında ölçülen CPU ve RAM değerleri Tablo 4.1 ve Tablo 4.2'de verilmiş olup, sistemde hiçbir güvenlik mekanizması bulunmuyorken fed20-2 VM, ortalama olarak CPU'nun yaklaşık %0,351'ini, hafızanın ise yaklaşık olarak %39,49'unu kullanmıştır.

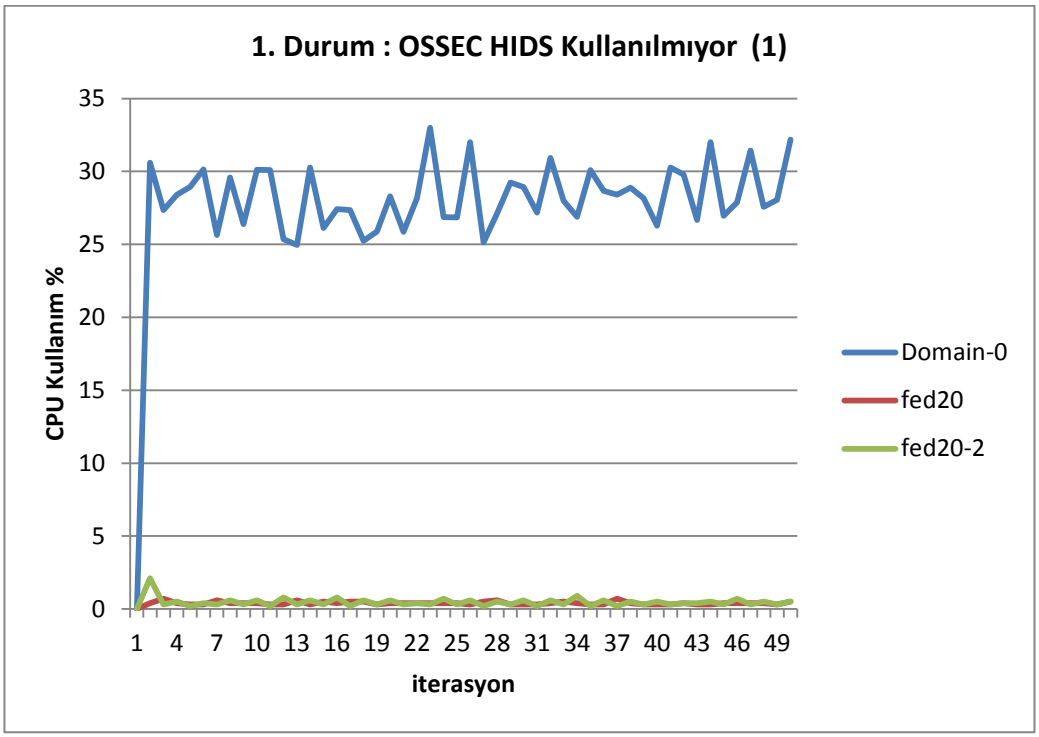
İkinci aşamada sistemde üç VM çalışıyorken Tablo 4.5 ve Tablo 4.6'da görüldüğü üzere Domain-0 1 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %25,95'ini, hafızanın ise %65,70'ini; 3 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %26,77'sini, hafızanın ise %65,70'ini;10 saniye frekans aralığında ölçüldüğünde CPU'nun yaklaşık olarak %25,38'ini, hafızanın ise %65,43'ünü kullanmıştır. Tablo 4.7 ve Tablo 4.8'de gösterilen sistemde hiçbir güvenlik mekanizması bulunmuyorken ortalama olarak Domain-0 CPU'nun yaklaşık %26,03'ünü, hafızanın ise yaklaşık olarak %65,61'ini kullanmıştır. VM'lerin sistemde kendilerine tahsis edilen donanım kaynaklarını kullanım miktarlarına bakıldığında farklı zaman aralıklarında ölçülen değerler Tablo 4.5 ve Tablo 4.6'da detaylı olarak gösterilmiş olup ortalama değerler açısından VM'ler gözlenmlendiğinde Tablo 4.7 ve Tablo 4.8'de verildiği şekilde fed20 isimli VM sistemde hiçbir güvenlik mekanizması bulunmuyorken CPU'nun yaklaşık olarak %0,34'ünü, hafızanın ise yaklaşık olarak %58,28'ini; fed20-2 isimli VM, ortalama olarak CPU'nun yaklaşık %0,48'ini, hafızanın ise yaklaşık olarak %37,95'ini; fed20-3 isimli VM ise ortalama olarak CPU'nun yaklaşık %2,68'ini, hafızanın ise yaklaşık olarak %39,22'sini kullanmıştır.

Sistemde OSSEC HIDS kullanılmıyorken Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafikler Şekil 4.3 ve Şekil 4.4'te verilmiştir.

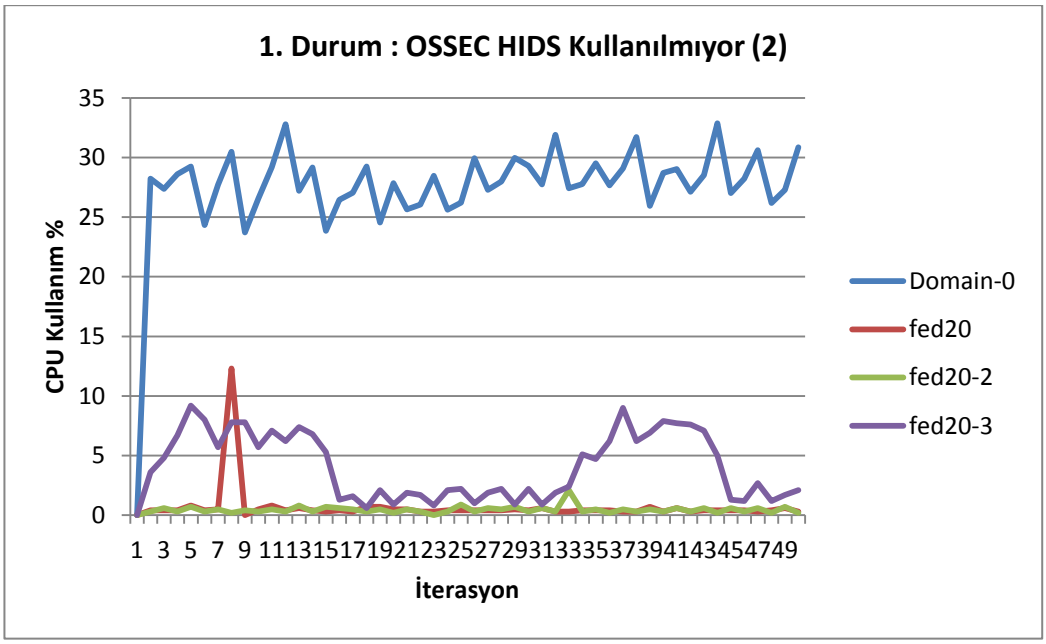
4.3.2. Sadece Ana Makinede OSSEC HIDS Kullanılması Durumu (2. Durum)

Sistemde güvenlik mekanizması olarak sadece ana makine üzerinde mevcut OSSEC sunucusu ile üç VM'nin güvenliği sağlanıyorken Tablo 4.5 ve Tablo 4.6'da farklı zaman frekanslarında elde edilen Domain-0, fed20, fed20-2 ve fed20-3 'e ait CPU ve RAM kullanımları verilmiş olup; Tablo 4.7 ve Tablo 4.8'de görüldüğü üzere ortalama olarak Domain-0 CPU'nun yaklaşık %26,34'ünü, hafızanın ise yaklaşık olarak %72,77'sini; fed20 CPU'nun yaklaşık olarak %2,95'ini, hafızanın ise yaklaşık olarak %62,34'ünü; fed20-2, ortalama olarak CPU'nun yaklaşık %2.44'ünü, hafızanın ise

yaklaşık olarak %43,36'sını; fed20-3 ise ortalama olarak CPU'nun yaklaşık %4,79'unu, hafızanın ise yaklaşık olarak %43,88'ini kullanmıştır.

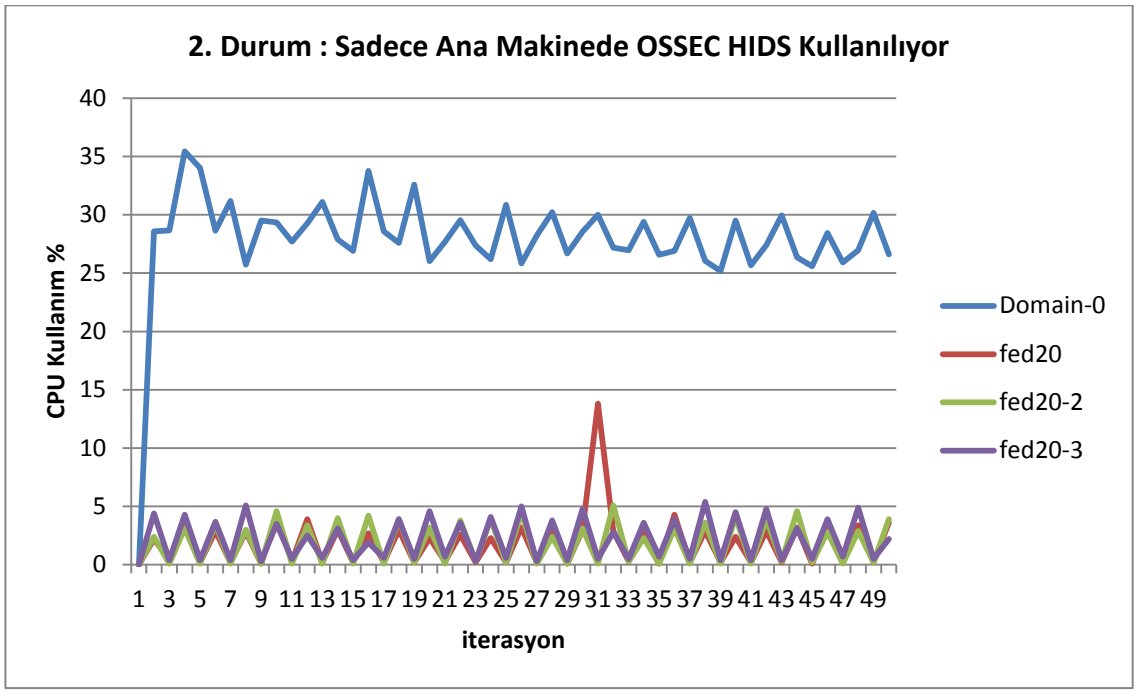


Şekil 4.3: OSSEC HIDS kullanılmazken elde edilen CPU kullanım grafiği -1.



Şekil 4.4:OSSEC HIDS kullanılmazken elde edilen CPU kullanım grafiği -2.

Sistemde sadece ana makinede OSSEC sunucusu varken Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafik Şekil 4.5'te verilmiştir.



Şekil 4.5: Sadece ana makinede OSSEC HIDS kullanılırken elde edilen CPU kullanım grafiği.

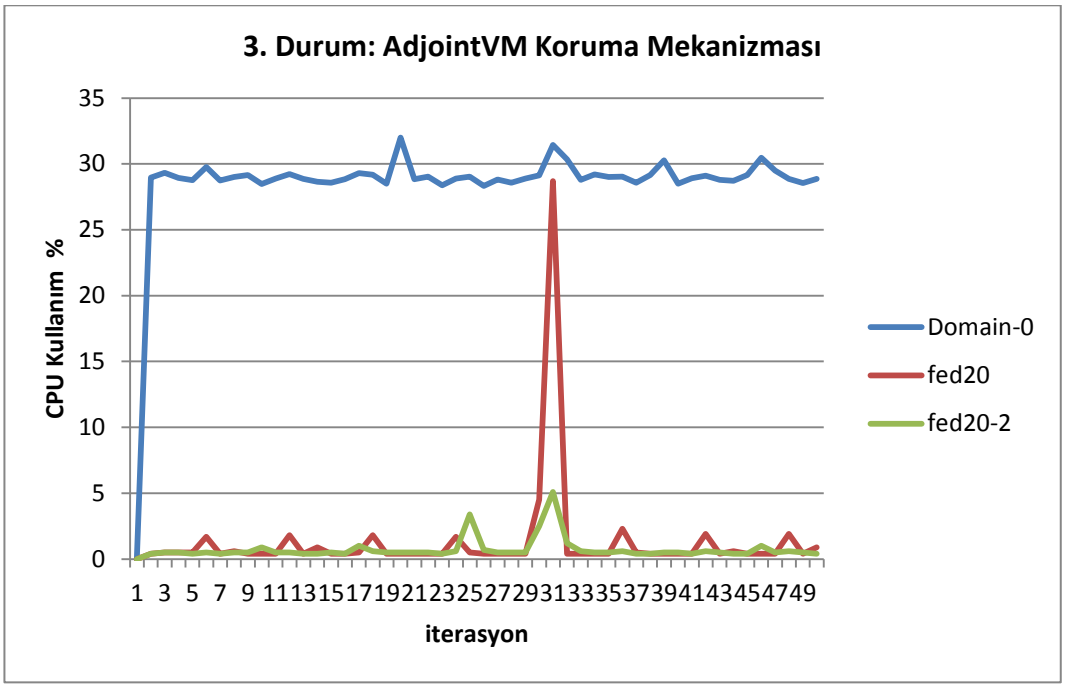
4.3.3. AdjointVM Koruma Mekanizması Kullanılması Durumu (3. Durum)

Sistemde AdjointVM Koruma Mekanizması ile güvenlik sağlanıyorkenTablo 4.1 ve Tablo 4.2'de farklı zaman frekanslarında elde edilen Domain-0, fed20 ve fed20-2 VM'lere ait CPU ve RAM kullanımları verilmiş olup; Tablo 4.3 ve Tablo 4.4'te görüldüğü üzere ortalama olarak Domain-0 CPU'nun yaklaşık %22,75'ini, hafızanın ise yaklaşık olarak %56,07'sini; fed20 CPU'nun yaklaşık olarak %9,51'ini, hafızanın ise yaklaşık olarak %72,97'sini; fed20-2, ortalama olarak CPU'nun yaklaşık %7,88'ini, hafızanın ise yaklaşık olarak %40,9'unukullanmıştır.

AdjointVM Koruma Mekanizması kurulu iken Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafik Şekil 4.6'da verilmiştir.

4.3.4. İyileştirilmiş AdjointVM Koruma Mekanizması Kullanılması Durumu (4. Durum)

İyileştirilmiş AdjointVM Koruma Mekanizması ile sistemde güvenlik sağlandığında farklı zaman frekanslarında elde edilen Domain-0, fed20 ve fed20-2 VM'lere ait CPU ve RAM kullanım miktarları Tablo 4.1 ve Tablo 4.2'de verilmiştir.



Şekil 4.6: AdjointVM Koruma Mekanizması ile elde edilen CPU kullanım grafiği.

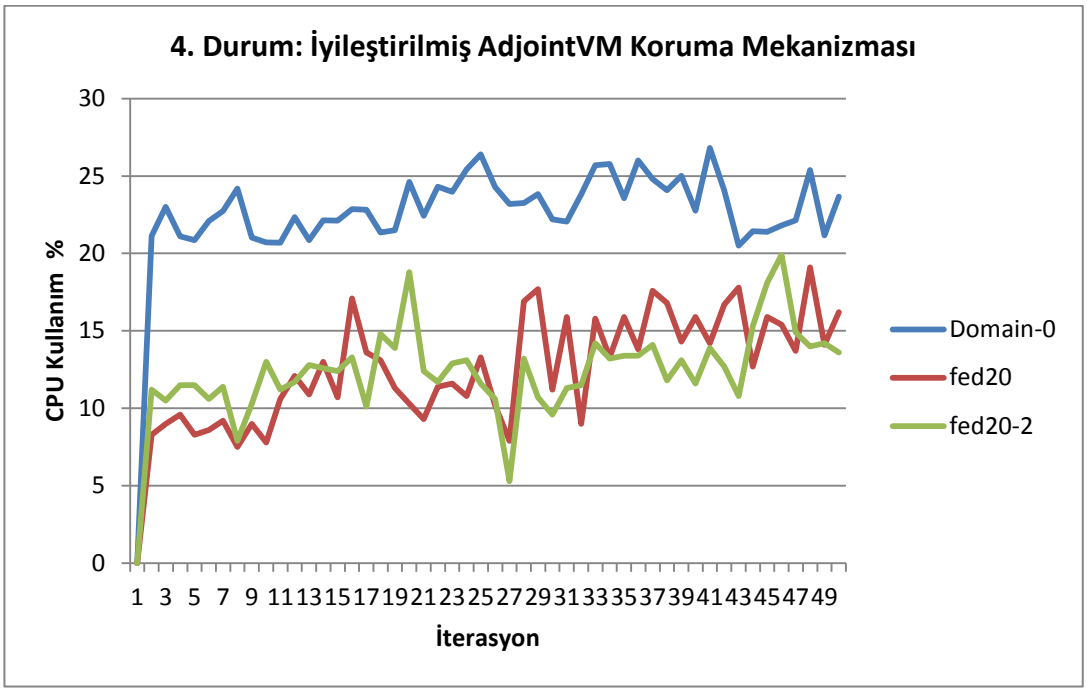
Elde edilen değerlerin ortalamasıyla oluşturulan Tablo 4.3 ve Tablo 4.4'te görüldüğü üzere Domain-0 CPU'nun yaklaşık %23,07'sini, hafızanın ise yaklaşık olarak %59,39'unu; fed20 CPU'nun yaklaşık olarak %11,69'unu, hafızanın ise yaklaşık olarak %63,32'sini; fed20-2, ortalama olarak CPU'nun yaklaşık %11,88'ini, hafızanın ise yaklaşık olarak %44,42'sini kullanmıştır.

İyileştirilmiş AdjointVM Koruma Mekanizması kullanıldığında Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafik Şekil 4.7'de verilmiştir.

4.3.5. Döngüsel Zincir VM Koruma Mekanizmasının Kullanılması Durumu (5. Durum)

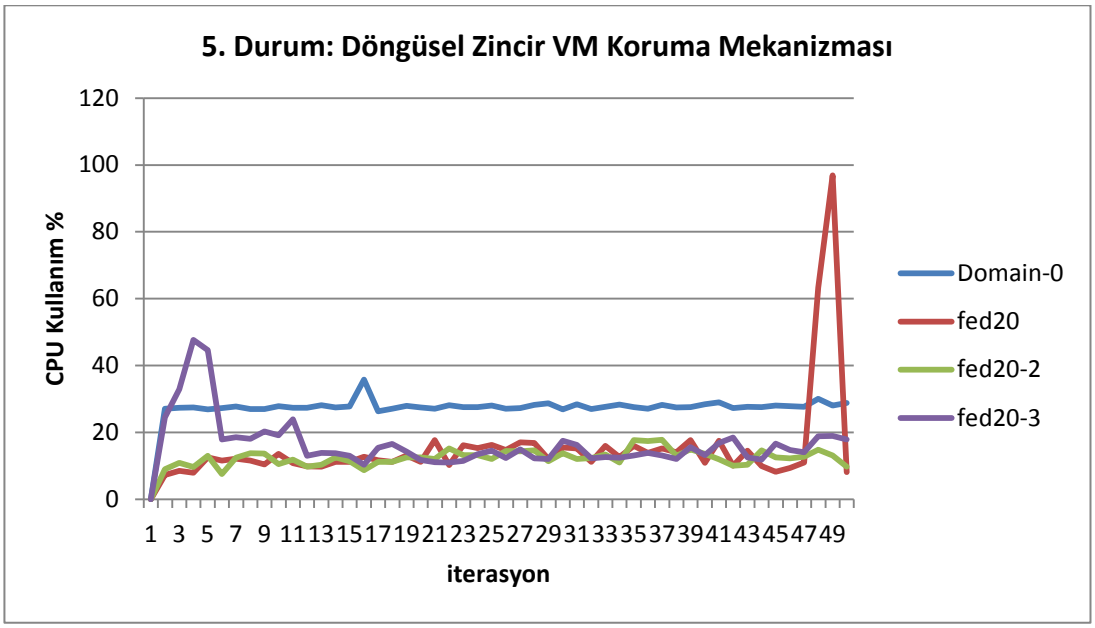
Sistemde VM'nin güvenliğinin sağlanması amacıyla Döngüsel Zincir VM Koruma Mekanizması oluşturulduğu zaman ana makine ve VM'lere ait farklı zaman frekanslarında elde edilen CPU ve RAM kullanımlarına ait bilgiler Tablo 4.5 ve Tablo 4.6'da verilmiştir. Tablo 4.7 ve Tablo 4.8'de görüldüğü üzere ortalama olarak Domain-0 CPU'nun yaklaşık %27,08'ini, hafızanın ise yaklaşık olarak %74,14'ünü; fed20 CPU'nun yaklaşık olarak %11,86'sını, hafızanın ise yaklaşık olarak %64,31'ini; fed20-2, ortalama olarak CPU'nun yaklaşık %13,54'ünü, hafızanın ise yaklaşık olarak

%44,7'sini; fed20-3 ise ortalama olarak CPU'nun yaklaşık %20,42'sini, hafızanın ise yaklaşık olarak %42,78'ini kullanmıştır.



Şekil 4.7:İyileştirilmiş AdjointVM Koruma Mekanizması ile elde edilen CPU kullanım grafiği.

Sistemde Döngüsel Zincir VM Koruma Mekanizmasi ile güvenlik sağlanıyorken Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafik Şekil 4.8'de verilmiştir.



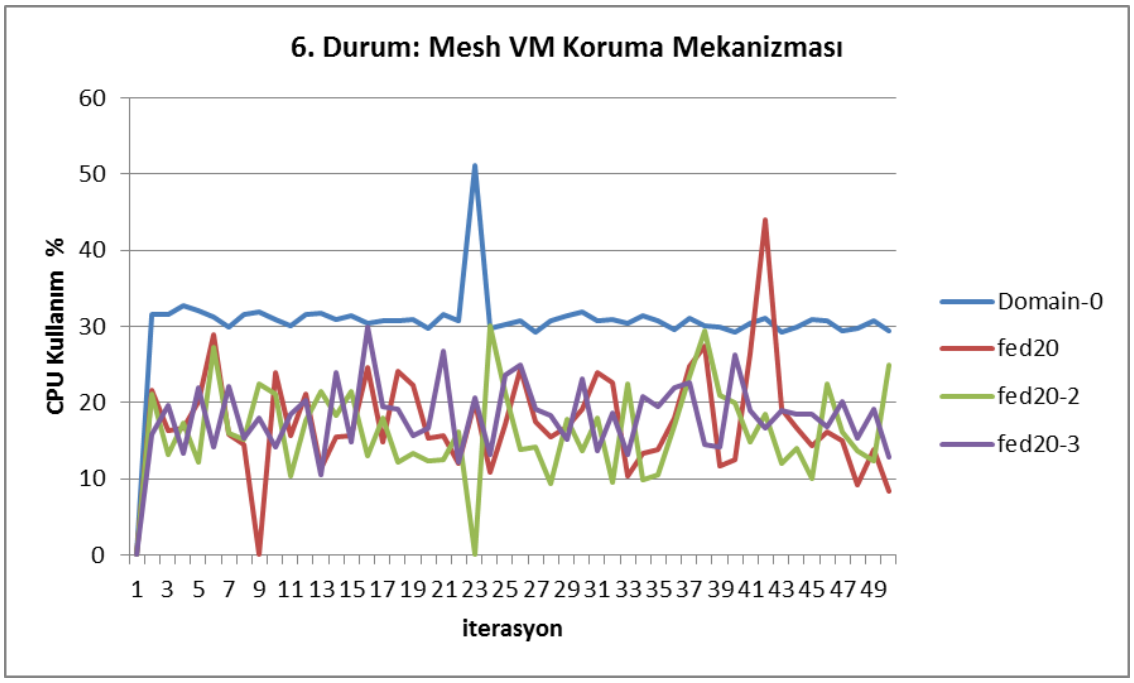
Şekil 4.8:Döngüsel Zincir VM Koruma Mekanizması ile elde edilen CPU kullanım grafiği.

4.3.6. Mesh VM Koruma Mekanizmasının Kullanılması Durumu (6. Durum)

Sistemde son güvenlik mekanizması olarak önceki bölümde açıklandığı üzere iki aşamada oluşturulan Mesh VM Koruma Mekanizması için ana makine ve VM'lere ait farklı zaman frekanslarında elde edilen CPU ve RAM kullanımlarına ait bilgiler Tablo 4.5 ve Tablo 4.6'da verilmiştir. Tablo 4.7 ve Tablo 4.8'de görüldüğü üzere bu koruma mekanizmasının ilk aşaması olan sisteme yeni bir VM eklendiği durumda ortalama olarak Domain-0 CPU'nun yaklaşık %25,87'sini, hafızanın ise yaklaşık olarak %73,3'ünü; fed20 CPU'nun yaklaşık olarak %9,67'sini, hafızanın ise yaklaşık olarak %65,11'ini; fed20-2, ortalama olarak CPU'nun yaklaşık %13,54'ünü, hafızanın ise yaklaşık olarak %47,77'sini; fed20-3 ise ortalama olarak CPU'nun yaklaşık %8,82'sini, hafızanın ise yaklaşık olarak %42,6'sını kullanmıştır.

Mesh VM Koruma Mekanizması ikinci aşamasında ise ortalama olarak Domain-0 CPU'nun yaklaşık %30,62'sini, hafızanın ise yaklaşık olarak %77,77'sini; fed20 CPU'nun yaklaşık olarak %18,08'ini, hafızanın ise yaklaşık olarak %61,14'ünü; fed20-2, ortalama olarak CPU'nun yaklaşık %17,33'ünü, hafızanın ise yaklaşık olarak %49,14'ünü; fed20-3 ise ortalama olarak CPU'nun yaklaşık %18,99'unu, hafızanın ise yaklaşık olarak %41,88'ini kullanmıştır.

Sistemde Mesh VM Koruma Mekanizmasi ile güvenlik sağlanıyorken Domain-0 ve VM'lerden elde edilen CPU kullanım yüzdelerine ait grafik Şekil 4.9'da verilmiştir.



Şekil 4.9: Mesh VM Koruma Mekanizması ile elde edilen CPU kullanım grafiği.

4.3.7. Sonuçların Karşılaştırılması

Bu çalışmada test ortamındaki mevcut sistemde hiçbir güvenlik mekanizması olmadığı durum ile beş güvenlik mekanizmasının ayrı ayrı kullanılması durumları olmak üzere toplamda altı farklı durumda sistemdeki donanım kaynaklarının kullanım miktarları gözlemlenerek bulut bilişimde çok önemli bir konu olan kaynakların verimli kullanımı açısından elde edilen bulgular karşılaştırılmıştır. Tablo 4.9'da gerçekleştirilen tüm güvenlik mekanizmalarınınkaynak tüketim miktarları bazında donanım kullanım verimliliği açısından CPU ve RAM kullanım oranları ile ilgili kıyaslama verilmiştir.

Tablo 4.9:Güvenlik mekanizmalarının donanım kullanım oranları açısından kıyaslanması.

Donanım Kullanım Oranı	Kıyaslama
CPU	1. Durum < 2. Durum < 3. Durum < 4. Durum < 5. Durum < 6. Durum
RAM	1. Durum < 2. Durum < 3. Durum < 4. Durum < 5. Durum < 6. Durum

Test ortamında donanım kullanım verimliğini ortaya koymak amacıyla CPU ve RAM kullanım oranlarının ölçülmesinde, gerçekleştirme ortamındaki kısıtlı donanım kaynağı ve mevcut ağ koşulları göz önünde bulundurulduğunda test bulgularında zaman zaman sapmalar meydana gelmiş, her bir mekanizma aktif haldeyken ana makine ve VM'lerde o anda yürütülen işlemlerin, çalıştırılan programların gerçek hayatta olduğu gibi kullanıcı tercihleri sebebiyle değişken olması da istendiğinden sistemler test edilirken CPU ve RAM kullanım oranlarında değişmeler gözlemlenmiştir. Genel durum itibarıyla ortalama sonuç değerlerine göre oluşturulan güvenlik mekanizmaların kıyaslanması yukarıdaki Tablo 4.9'da görülmektedir.

Bu bağlamda sistemde hiçbir güvenlik mekanizması kullanılmadığı durumda donanım kaynak kullanım oranının en düşük olduğu ortadadır. İkinci olarak sistemde sadece ana makine üzerinde OSSEC sunucusunun çalıştırılmasıyla sağlanan güvenlik durumu,üçüncü olarak AdjointVM Koruma Mekanizmasının kullanıldığı durum, dördüncü olarak İyileştirilmiş AdjointVM Koruma Mekanizmasının kullanıldığı durum, beşinci olarak Döngüsel Zincir VM Koruma Mekanizması durumu ve altıncı olarak Mesh VM Koruma Mekanizması durumu gelmektedir.

AdjointVM Koruma Mekanizması literatürde yer alan gerçekleştirilmiş bir modeldir. İyileştirilmiş AdjointVM Koruma Mekanizması ve Döngüsel Zincir VM Koruma Mekanizması ise literatürde önerilen modeller olup bu tez çalışması kapsamında gerçekleştirimi yapılmıştır. Mesh VM Koruma Mekanizması ise bu tez çalışması kapsamında önerilen bir modeldir. Tablo 4.9'daki kıyaslama da göstermiştir ki sistemde güvenlik mekanizmalarının gücü donanım kullanım miktarını doğrudan etkilemektedir. Güvenlik mekanizmalarındaki yapısal değişiklikler ve karmaşıklık durumu donanım kullanım verimliliğini azaltmakta, ancak sistemin saldırı tespitindeki sağlamlığı, saldırı ve tehditlere karşı dayanıklılığı ise oldukça artmaktadır. Bu nedenle sistemde hiçbir güvenlik önlemi yokken donanım kullanım verimliliğinin fazla olması güvenlik sakıncaları göz önüne alındığında sisteme olumsuz etki göstermektedir.

AdjointVM Koruma Mekanizması ve İyileştirilmiş AdjointVM Koruma Mekanizmasının bulguları göstermiştir ki sadece ana makinede OSSEC HIDS kullanımı yerine bu mekanizmaların kullanılması ana makine açısından ek yük olmadığı, sadece sanal makinelerin daha fazla iş yapmasından dolayı CPU ve RAM kullanım oranlarının

göreceli olarak arttığı, bu nedenle kıyaslama yapılırken Tablo 4.9'daki sonuç ortaya çıkmıştır. İyileştirilmiş AdjointVM Koruma Mekanizmasının AdjointVM Koruma Mekanizmasından daha fazla VM'lere iş yaptırmıştır. Döngüsel Zincir VM Koruma Mekanizmasının ise bahsi geçen bu iki mekanizmadan daha fazla güvenlik katkısı sağladığı ve saldırı tespiti açısından daha fazla sağlamlık ve dayanıklılık sunduğu düşünüldüğünde, donanım kullanımı açısından sisteme hem ana makine hem de VM'ler bazında ek yük getirmesi beklenmiş, elde edilen sunuçlar da hem CPU hem de RAM tüketimi açısından artış olduğunu göstererek bu beklentiyi doğrulayan niteliktedir. Mesh VM Koruma Mekanizması ise gerçekleştirilen diğer mekanizmalara göre karmaşıklığı daha fazla ancak güvenlik açısından daha güçlü ve dayanıklı bir yapı sunmaktadır. Bu nedenle sistemde hem ana makinede hem de mevcut VM'lerde birden fazla OSSEC ajanı aynı anda çalışmakta olmasının sisteme getirdiği ek yükler olduğu gözlemlenmiştir.

5. TARTIŞMA VE SONUÇ

Bulut bilişim, ortak kullanılan kaynaklar üzerinde her an kullanıma hazır, ihtiyaca göre ölçeklendirilebilen, kullanıcılarına kaynak ataması ve yönetimi kolaylıkla yapılabilen bilişim servislerinin bir araya geldiği yeni bir teknolojidir.Bulut platformları; donanım ve yazılım gibi mevcut bilişim kaynaklarının web servisleri üzerinden dinamik olarak paylaştırılabildiği, hemen her türlü elektronik cihazın bağlanabildiği ve ölçeklendirme kolaylığı ile yaygın hizmet sunan servis sağlayıcılardan oluşan internet ortamı olarak ifade edilmektedir.

Bulut bilişim, günümüzde geleneksel bilişim ortamlarının yerini almaya başlamış ve kullanımı oldukça yaygınlaşmaktadır. Bulut bilişim sayesinde kullanıcılar sağlanan servislere çok düşük maliyetlerle ulaşabilir duruma gelmiş ve donanım barındırma, bakım, personel, lisans gibi BT masraflarından tasarruf sağlanarak verimlilik elde edilmiştir.Bulut sağlayıcıları, bulut tüketicilerine kendi aralarında gerçekleştirdikleri anlaşmalar ile belirlenen maliyetler doğrultusunda servisler sunmaktadır. Bu şekilde *kullandığın kadar öde* modeline göre bulut tüketicilerinin servis ve altyapı kullanım miktarı ve süresine bağlı olarakücretlendirme yapılmaktadır.

Bulut bilişim teknolojilerinin sunduğu faydalar sayesinde hem bireysel hem de kurumsal kullanıcılar arasında tercih edilmesinin artmasına rağmen hala kullanıcıların bulut yaklaşımındaki en büyük endişe konusunun güvenlik olduğu görülmektedir. Bu endişe, geleneksel bilgi sistemlerinin sahip olduğu güvenlik risklerinden kaynaklanan endişelerle benzer özelliktedir.Özellikle servislerin sunulduğu altyapıda kullanılan donanım ve yazılımın, üçüncü şahısların kontrolünde bulunmasından dolayı, temel sistem gereksinimlerinde güvenlik ve gizlilik kavramları öne çıkmaktadır. Bu nedenle bilişim güvenliği konusundaki çalışmalar her geçen gün daha da artmakta, güvenlik ve gizliliğin somut bir biçimde ele alınarak kullanıcılarınbulut servislerini tercih edebilmesi için güvenlik ihtiyaçlarının karşılanmasında yeni güvenlik protokolleri ve yaklaşımlar kullanılmaktadır. Böylece kullanıcılar, bulut platformlarında yer alan veri ve uygulamalara servis kesintisi olmadan anlık olarak erişilebilmekte, verilere erişimin

güvenlik mekanizmalarıyla kuvvetlendirilmiş veri merkezleri tarafından sağlanması ile bireysel veya kurumsal kullanıcıların kendi bünyelerinde sahip oldukları güvenlik seviyesinin bile üzerinde servisleri alabilmektedir.

Geleneksel bilgi sistemlerinde olduğu gibi bulut bilişimde de güvenlik mekanizması olarak saldırı tespit sistemi kullanımı çok önem arz etmektedir. Bulut bilişim altyapısında kullanılan sanallaştırma teknolojisi sayesinde oluşturulan sanal makinelerin güvenliğinin sağlanmasında saldırı tespit sistemlerinin doğru, verimli ve etkili şekilde yapılandırılması gerekmektedir. Saldırı tespit mekanizması çalışırken, güvenliğini sağladığı bilgi sistemi varlığının yoğunluğuna bağlı olarak işlediği veri miktarından kaynaklı donanım ihtiyacı ortaya çıkmaktadır. Mekanizma doğru şekilde yapılandırılmadığı zaman fazla miktarda CPU ve bellek kullanımına neden olmaktadır.Bu nedenle, bulut bilişim bünyesinde oluşturulacak saldırı tespit sistemi mekanizması, kullanıcının kullandığın kadar öde modeliyle servis kullanım ücretlendirmesinden dolayı maliyetlerini arttırmış olacaktır.

Bu tez çalışmasında güvenli bulut bilişim için sunucu-tabanlı saldırı tespit sisteminin (HIDS) nasıl kullanılması gerektiği konusunda araştırmalar yapılmış, literatürde yer alan farklı güvenlik modellerine göre saldırı tespit mekanizmaları oluşturularak özellikle donanım kullanım maliyetleri bakımından karşılaştırılması yapılmış, böylelikle kullanıcılara en optimun güvenlik mekanizmasının tercih edilmesi konusunda fayda sağlamak amaçlanmıştır. Özellikle bulut kullanıcılarının servis aldığı sağlayıcılara karşı güven duyabilmesi bakımından bulut sağlayıcı ve yöneticilerinden gelen iç saldırılara karşı bulut güvenliğini en üst seviyeye çıkarmak için farklı güvenlik mekanizmaları oluşturulmuş ve değerlendirilmiştir.

Öncelikle sistemde hiçbir güvenlik mekanizması kullanılmadığı zaman donanım kullanım miktarları tespit edilmiştir. İkinci durumda sistem güvenliğinin ana makinenin sorumluluğuna verilmesi amacıyla sunucu-tabanlı saldırı tespit sistemi sadece ana makine üzerine konumlandırılmış ve bu şekildeki güvenlik mekanizmasının donanım kullanım miktarlarına etkisi ölçülmüştür.

Üçüncü durumda güvenlik modeli olarak literatürde yer alan güvenilir olmayan bulut sağlayıcılarına yönelik önemli bir çalışma olan AdjointVM yaklaşımına ait sunucu-

tabanlı saldırı tespit sistemi kullanım mimarisi yönünden model ele alınarak AdjointVM Koruma Mekanizması gerçekleştirilmiştir. Bu mekanizma, güvensiz sağlayıcılardan gelebilecek iç saldırılara ve gerçek veya sanal ağ üzerinden gelebilecek dış saldırılara karşı IDS tabanlı bir güvenlik mekanizması sunmaktadır. Oluşturulan mekanizmanın donanım kullanım miktarları incelenmiş, tablo ve grafiklere aktarılmıştır.

Dördüncü durumda literatürde henüz gerçekleştirilmesi yapılmamış, öneri olarak yer alan AdjointVM IDS yaklaşımına bazı ek iyileştirmeler ekleyerek eksikliklerinin giderilmesiyle oluşturulan İyileştirilmiş Adjoint VM Koruma Modeli sunucu-tabanlı saldırı tespit sistemi mimarisi gerçekleştirilerek donanım kullanım miktarları incelenmiştir.

Beşinci durumda aynı şekilde literatürde öneri model olarak sunulan, AdjointVM koruma modelinin getirdiği yeni yaklaşım sayesinde sistemin direncini arttırırken ve esnek bir güvenlik politikası sunarken yanlış-pozitif oranının azaltılması hedeflenen Döngüsel Zincir VM Koruma Modeli yaklaşımına göre sunucu-tabanlı saldırı tespit sistemi mekanizmasının gerçeklenmesi yapılmış ve bu mekanizmaya ait donanım kullanım miktarları ölçülmüş ve incelenmiştir.

Altıncı durumda ise Döngüsel Zincir VM koruma modelinin de eksikliklerine karşın ek iyileştirmeler yapılıp yapılamayacağının ortaya konması ve modelin daha da geliştirilmesinin sağlanması amacıyla bu tez çalışması kapsamında yeni bir mekanizma olarak Mesh VM Koruma Mekanizması önerilmiş ve gerçekleştirilmiştir.Mesh, merkezi olmayan ve ağ üzerindeki her düğümün en az iki düğüm ile bağlantılı olduğu bir ağ topolojisidir. Bu yapının en büyük avantajı esneklik ve merkezi olmaması yani tek bir hata noktasının bulunmamasıdır. İki düğüm arasında bağlantı kopması durumunda, düğümlerin bağlı olduğu başka düğümler olması nedeniyle ağ iletişiminde sorun oluşturmadan tolore edilebilmektedir. Mesh topolojisinden esinlenilerek önerilen Mesh VM Koruma Modeli yaklaşımı ile Döngüsel Zincir VM Modeli mesh yapısı olarak uyarlanmış ve böylelikle güvenlik mekanizması seviyesi daha yukarı seviyeye çıkarılmıştır.

Yapılan çalışmalar kapsamında oluşturulan güvenlik mekanizmalarının birinci durumdan son duruma doğru donanım kullanım miktarları karşılaştırıldığında güvenlik

seviyesinin yükseltilmesinin, daha iyi ve daha dayanıklı saldırı tespit sistemi yapılandırılmasının sistemdeki donanım kullanım miktarlarını arttırdığı gözlemlenmiştir. Bulut bilişimde kullanıcıların güvenlik endişelerini giderilmesine katkı sağlamak maksadıyla yapılan bu çalışmanın öncelikli hedefi sistemin güvenliğinin en üst düzeye çıkarılması olduğu için sisteme sunduğu güvenlik katkılarıyla donanım kullanım miktarlarındaki artışın tolore edilebileceği Döngüsel Zincir VM Koruma Mekanizması ve Mesh VM Koruma Mekanizmasının bulut bilişimde tercih edilmesi sonucuna varılmıştır. Bu güvenlik mekanizmalarının donanım kullanım verimliliği ve sisteme uygulama karmaşıklığı göz önüne alındığında ise Döngüsel Zincir VM Koruma Mekanizmasının kullanıcı beklentilerini daha iyi karşılayacağını, karmaşıklığının daha az olduğu gerekçesiyle bu tez çalışmasının sonucunda güvenli bulut bilişim için bulut sağlayıcıları tarafından bu mekanizmanın tercih edilmesi önerilmektedir.

Yapılacak donanım bazlı iyileştirmeler ve kullanılan açık kaynak kodlu sunucu-tabanlı saldırı tespit sistemi yazılımındaki modifikasyonlar sayesinde sistemdeki donanım kullanım oranlarının optimize edilmesi ile Mesh VM Koruma Mekanizmasının donanım verimliliği açısından eksikliklerinin giderileceği öngörülmektedir. Böylelikle bulut bilişim güvenliğinde etkili ve verimli bir mekanizma olarak kullanılabilecektir.

KAYNAKLAR

[1]. Mell, P., and Grance, T., September2011, The NIST Definition of Cloud Computing, *NIST Special Publication 800-145 (SP800-145)*, National Institute of Standards and Technology.

[2]. Cliff, D., June 2010, Remotely Hosted Services and "Cloud Computing", *British Educational Communications and Technology Agency*, United Kingdom.

[3]. Furht, B.,Escalante, A., 2010, *Handbook of Cloud Computing*, Springer, New York, ABD

[4]. Höfer, C.N., Karagiannis, G., 2011, Cloud Computing Services: Taxonomy and Comparison, *Journal of Internet Services and Applications*, 2, 81–94

[5]. Hogan, M., Liu, F., Sokol, A. ve Tong, J., 2011, NIST Cloud Computing Standards Roadmap, *NIST Special Publication 500-291 (SP500-291)*, National Institute of Standards and Technology, Gaithersburg.

[6]. Rimal,B.P., Jukan, A., Katsaros, D., Goeleven, Y., 2011, Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach, *Journal of Grid Computing*, 9 (1), 3–26

[7]. Varia, J., January 2010, Architecting for the Cloud: Best Practices, Amazon Web Services,http://jineshvaria.s3.amazonaws.com/public/cloudbestpractices-jvaria.pdf, Erişim Tarihi: Haziran 2012.

[8]. Kwasnıewski, T. J., Puig EJ, 2011, Cloud Computing in the Government, *The Data & Analysis Center for Software*,New York, ABD, 101.

[9]. Ergin, O., ve Ark., Nisan 2012, *Kamuda Bulut Bilişim*, 1. Çalışma Grubu, Türkiye Bilişim Derneği Kamu Bilgi İşlem Merkezleri Yöneticileri Birliği, Ankara, 36.

[10]. Newcomer, E., Lomow, G., 2005,*Understanding SOA with Web Services*, Addison Wesley. ISBN 0-321-18086-0.

[11]. Desısto, R.P., PlummerD.C., Smıth D.M.,August 2011, An Easy Way to Understand the Relationship Between Cloud Computing and SaaS, *Gartner*, Report ID: G00214956.

[12]. Vouk, M. A., June 2008. Cloud computing – Issues, Research And Implementations. *Proceedings of the ITI 30th International Conference on Information Technology Interfaces*, Cavtat, Croatia, 31–40.

[13]. Erl T., Mahmood Z., Puttini R., 2013*Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, Westford.

[14]. Bıswas, S., 2011, Cloud Computing vs Utility Computing vs Grid Computing: Sorting The Differences, Cloud Tweaks, (çevrimiçi) *http://www.cloudtweaks.com/2011/02/cloud-computing-vs-utilitycomputing-vs-grid-computing-sorting-the-differences*, son erişim: Mayıs 2012.

[15]. KORRI, T., 2009, Cloud computing: utility computing over the Internet, *Helsinki University of Technology*.

[16]. Marshall D., Beaver S. S., McCarty J. W., 2009, VMware ESX: Essentials in the Virtual Data Center, *CRC press*.

[17]. Kirch J., Eylül 2007, Virtual machine security guidelines. *The center for Internet Security*, http://www.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf.

[18]. VelteA. T., VelteT. J., ElsenpeterR., 2010, Cloud Computing: A Practical Approach, *McGraw-Hill*.

[19]. Microsoft Hyper-V'ye Genel Bakış, https://www.microsoft.com/tr-tr/server-cloud/solutions/virtualization.aspx , son erişim: Mayıs 2015.

[20].VMWare ESX and VMWare ESXi, http://www.vmware.com/files/pdf/VMware-ESX-and-VMware-ESXi-DS-EN.pdf , son erişim: Mayıs 2015.

[21]. VirtualBox, https://www.virtualbox.org, son erişim: Mayıs 2015.

[22]. Powers S., Readers' Choice Awards 2014, 2 Aralık 2014, http://www.linuxjournal.com/rc2014?page=16, son erişim: Haziran 2015.

[23]. Barham P., Dragovic B., Fraser K., Hand S., Harris T., Ho A., Neugebauer R., Pratt I., Warfield A., 2003, Xen and the art of virtualization, *ACM SIGOPS Operating Systems Review*, 37, 5, 164-177.

[24]. Bellard F., 2005, "QEMU, a fast and portable dynamic translator". *USENIX*.

[25]. Kumar, A., Kumar, V., Singh, P., & Kumar, A., 2012, A Novel approach: Security measures and Concerns of Cloud Computing. *International Journal of Computer Technology and Applications*, 3(3), 1008 -1014.

[26]. Teneyuca D.,2011, Internet cloud security: The illusion of inclusion, Information *Security Technical Report*, doi:10.1016/j.istr.2011.08.005.

[27]. Security for Cloud Computing 10 Steps to Ensure Success,Ağustos 2012, *Cloud Standards Customer Council*,http://www.cloud-council.org/Security_for_Cloud_Computing-Final_080912.pdf, son erişim: Haziran 2015.

[28]. Scarfone K., ve Mell P., 2007, Guide to Intrusion Detection and Prevention Systems (IDPS), *NIST Special Publication 800-94 (SP800-94*), National Institute of Standards and Technology, Gaithersburg.

[29]. Marinova-Boncheva V., 2007, A short survey of intrusion detection systems, *Problems of Engineering Cybernetics and Robotics*, 58, 23–30.

[30]. Trusted Computing Group, http://www.trustedcomputinggroup.org/, son erişim : Haziran 2015.

[31]. Kong J., 2010, Protecting the confidentiality of virtual machines against untrusted host, *International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, Çin, 364.

[32]. Kong J., 2011, AdjointVM: a new intrusion detection model for cloud computing, *Energy Procedia*, 13, 7902-7911.

[33]. Columbus L., 2014, Predicting The Future Of Cloud Service Providers, http://www.forbes.com/sites/louiscolumbus/2015/04/05/predicting-the-future-of-cloud-service-providers/, Forbes, son erişim: Haziran 2015.

[34]. Farcasescu M.R., 2012, Trust Model Engines in Cloud Computing, *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 465-470.

[35]. Oktay, U., Aydın, M. A., Sahingoz, O. K. Kasım 2013. A circular chain intrusion detection for cloud computing based on improved AdjointVM approach. *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium*, IEEE, 201-206.

[36]. Intel Virtualization Technology for Directed I/O Architecture Specification Rev. 1.3, 2011, *Intel Corporation.*

[37]. Oktay U., Aydın M. A., and Sahingoz O. K., 2013, Circular Chain VM Protection in AdjointVM,*International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*. 93–97.

[38]. Intel® Virtualization Technology (Intel® VT), son erişim: Haziran 2015, http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html

[39]. OSSEC HIDS, http://www.ossec.net/. son erişim: Haziran 2015.

[40]. Robertson C., Temmuz 2011, Practical OSSEC, GIAC (GCIH) Gold Certification, TheSANSInstitute.

[41]. Logstash, https://www.elastic.co/products/logstash, son erişim: Haziran 2015.

[42]. ElasticSearch, https://www.elastic.co/products/elasticsearch, son erişim: Haziran 2015.

[43]. Kibana, https://www.elastic.co/products/kibana, son erişim: Haziran 2015.

[44]. OSSEC Log Management with Elasticsearch, Kasım 2013 http://vichargrave.com/ossec-log-management-with-elasticsearch/, son erişim: Haziran 2015.

EKLER

EK 1.OSSEC HIDS (version 2.8.1) Sunucu Kurulumu.

[root@localhost harran]# *wget -U ossec* http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz

http://www.ossec.net/files/ossec-hids-2.8.tar.gz

Resolving www.ossec.net (www.ossec.net)... 150.70.191.237

Connecting to www.ossec.net (www.ossec.net)|150.70.191.237|:80... connected.

[root@localhost harran]# *tar xvfz ossec-hids-2.8.1.tar.gz*

[root@localhost harran]# *cd ossec-hids-2.8.1*

[root@localhost ossec-hids-2.8.1]# *./install.sh*

*** Para instalação em português, escolha [br].*

*** 要使用中文进行安装, 请选择 [cn].*

*** Fur eine deutsche Installation wohlen Sie [de].*

*** Για εγκατάσταση στα Ελληνικά, επιλέξτε [el].*

*** For installation in English, choose [en].*

*** Para instalar en Español , eliga [es].*

*** Pour une installation en français, choisissez [fr]*

*** A Magyar nyelvű telepítéshez válassza [hu].*

*** Per l'installazione in Italiano, scegli [it].*

*** 日本語でインストールします. 選択して下さい. [jp].*

*** Voor installatie in het Nederlands, kies [nl].*

*** Aby instalować w języku Polskim, wybierz [pl].*

*** Для инструкций по установке на русском ,введите [ru].*

*** Za instalaciju na srpskom, izaberi [sr].*

*** Türkçe kurulum için seçin [tr].*

(en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: en

OSSEC HIDS v2.8 Installation Script - http://www.ossec.net

You are about to start the installation process of the OSSEC HIDS.

You must have a C compiler pre-installed in your system.

If you have any questions or comments, please send an e-mail

to dcid@ossec.net (or daniel.cid@gmail.com).

- System: Linux localhost.localdomain 3.19.3-100.fc20.x86_64

- User: root

- Host: localhost.localdomain

-- Press ENTER to continue or Ctrl-C to abort. --

1- What kind of installation do you want (server, agent, local, hybrid or help)? ***server***

- Server installation chosen.

2- Setting up the installation environment.

- Choose where to install the OSSEC HIDS [/var/ossec]:

- Installation will be made at /var/ossec .

- The installation directory already exists. Should I delete it? (y/n) [y]: ***y***

3- Configuring the OSSEC HIDS.

3.1- Do you want e-mail notification? (y/n) [y]: ***y***

- What's your e-mail address? ***fedora.ids@gmail.com***

- We found your SMTP server as: alt3.gmail-smtp-in.l.google.com.

- Do you want to use it? (y/n) [y]: ***y***

--- Using SMTP server: alt3.gmail-smtp-in.l.google.com.

*3.2- Do you want to run the integrity check daemon? (y/n) [y]: **y***

- Running syscheck (integrity check daemon).

*3.3- Do you want to run the rootkit detection engine? (y/n) [y]: **y***

- Running rootcheck (rootkit detection).

3.4- Active response allows you to execute a specific

command based on the events received. For example,

you can block an IP address or disable access for

a specific user.

More information at:

http://www.ossec.net/en/manual.html#active-response

*- Do you want to enable active response? (y/n) [y]: **y***

- Active response enabled.

- By default, we can enable the host-deny and the

firewall-drop responses. The first one will add

a host to the /etc/hosts.deny and the second one

will block the host on iptables (if linux) or on

ipfilter (if Solaris, FreeBSD or NetBSD).

- They can be used to stop SSHD brute force scans,

portscans and some other forms of attacks. You can

also add them to block on snort events, for example.

*- Do you want to enable the firewall-drop response? (y/n) [y]: **y***

- firewall-drop enabled (local) for levels >= 6

- Default white list for the active response:

- 192.168.2.1

- Do you want to add more IPs to the white list? (y/n)? [n]: ***n***

3.5- Do you want to enable remote syslog (port 514 udp)? (y/n) [y]: ***y***

- Remote syslog enabled.

3.6- Setting the configuration to analyze the following logs:

-- /var/log/messages

-- /var/log/secure

-- /var/log/maillog

- If you want to monitor any other file, just change

the ossec.conf and add a new localfile entry.

Any questions about the configuration can be answered

by visiting us online at http://www.ossec.net .

--- Press ENTER to continue ---

5- Installing the system

- Running the Makefile

INFO: Little endian set.

**** Making zlib (by Jean-loup Gailly and Mark Adler) ****

make[1]: Entering directory `/home/harran/Downloads/ossec/ossec-hids-2.8.1/src/external'

cp -pr zlib-1.2.8/zlib.h zlib-1.2.8/zconf.h ../headers/

make[1]: Leaving directory `/home/harran/Downloads/ossec/ossec-hids-2.8.1/src/external'

…

- System is Redhat Linux.

- Init script modified to start OSSEC HIDS during boot.

- Configuration finished properly.

- To start OSSEC HIDS:

/var/ossec/bin/ossec-control start

- To stop OSSEC HIDS:

/var/ossec/bin/ossec-control stop

- The configuration can be viewed or modified at /var/ossec/etc/ossec.conf

Thanks for using the OSSEC HIDS.

If you have any question, suggestion or if you find any bug,

contact us at contact@ossec.net or using our public maillist at

ossec-list@ossec.net

(http://www.ossec.net/main/support/).

More information can be found at http://www.ossec.net

--- Press ENTER to finish (maybe more information below). ---

- In order to connect agent and server, you need to add each agent to the server.

Run the 'manage_agents' to add or remove them:

/var/ossec/bin/manage_agents

More information at:

http://www.ossec.net/en/manual.html#ma

EK 2.OSSEC-HIDS Sunucusunun Çalıştırlması.

[root@localhost ossec-hids-2.8.1]# */var/ossec/bin/ossec-control restart*

ossec-monitord not running ..

ossec-logcollector not running ..

ossec-remoted not running ..

ossec-syscheckd not running ..

ossec-analysisd not running ..

ossec-maild not running ..

ossec-execd not running ..

OSSEC HIDS v2.8 Stopped

Starting OSSEC HIDS v2.8 (by Trend Micro Inc.)...

Started ossec-maild...

Started ossec-execd...

Started ossec-analysisd...

Started ossec-logcollector...

Started ossec-remoted...

Started ossec-syscheckd...

Started ossec-monitord...

Completed.

EK 3. Monitor edilecek VM'ler üzerine OSSEC-HIDS Ajanı Kurulumu.

[root@localhost ossec-hids-2.8.1]# *./ install.sh*

1- What kind of installation do you want (server, agent, local, hybrid or help)? ***Agent***

- Agent(client) installationchosen.

2- Settinguptheinstallationenvironment.

- Choosewheretoinstallthe OSSEC HIDS [/opt/ossec-agent/ossec]:

- Installation will be made at ***/opt/ossec-agent/ossec .***

*3.1- What'sthe IP Addressorhostname of the OSSEC HIDS server?:****192.168.122.1***

*- Adding Server IP "****192.168.122.1****"*

3.2- Do youwanttoruntheintegritycheckdaemon? (y/n) [y]: ***y***

- Runningsyscheck (integritycheckdaemon).

3.3- Do youwanttoruntherootkitdetection engine? (y/n) [y]: ***y***

- Runningrootcheck (rootkitdetection).

3.4 - Do youwanttoenableactiveresponse? (y/n) [y]: ***n***

- Active responsedisabled.

3.5- Settingtheconfigurationtoanalyzethefollowinglogs:
-- /var/log/messages
-- /var/log/secure
-- /var/log/maillog
- Ifyouwanttomonitoranyother file, justchange
theossec.confandadd a newlocalfileentry.
Anyquestionsabouttheconfiguration can be answered
byvisiting us online at http://www.ossec.net .
--- Press ENTER tocontinue ---
- System is Redhat Linux.
- Initscriptmodifiedtostart OSSEC HIDS duringboot.
- Configurationfinishedproperly.
- Tostart OSSEC HIDS:
/opt/ossec-agent/ossec/bin/ossec-controlstart
- To stop OSSEC HIDS:
/opt/ossec-agent/ossec/bin/ossec-control stop
- Theconfiguration can be viewedormodified at /var/ossec/etc/ossec.conf

Thanksforusingthe OSSEC HIDS.
Ifyouhaveanyquestion, suggestionorifyoufindanybug,
contact us at contact@ossec.net orusingourpublicmaillist at
ossec-list@ossec.net
(http://www.ossec.net/main/support/).

Moreinformation can be found at http://www.ossec.net

--- Press ENTER tofinish (maybemoreinformationbelow). ---
- Youfirstneedtoaddthisagenttotheserversothey
cancommunicatewitheachother. Whenyouhavedoneso,
you can runthe 'manage_agents' tooltoimportthe
authenticationkeyfromtheserver.
/opt/ossec-agent/ossec/bin/manage_agents

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı	Fatma Didem ÖĞRETMEN
Uyruğu	T.C.
Doğum tarihi, Yeri	16.01.1986, Şanlıurfa
Telefon	0414 318 3000 - 1380
E-mail	fdidemogretmen@gmail.com

Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Yüksek Lisans	İ.Ü. Fen Bilimleri Enstitüsü/ Bilgisayar Mühendisliği Anabilim Dalı/ Bilgisayar Mühendisliği Programı	2015
Lisans	Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü	2009
Lise	Açıköğretim Lisesi	2004