Saldırı Tespit Sisteminin Bulut Bilişimde Kullanımı ve Etkileri

Fatma Didem ÖĞRETMEN, Muhammed Ali AYDIN, Ahmet SERTBAŞ

***Özet*—Teknolojinin hızla gelişmesiyle, bu hıza ayak uydurabilmek için tamamen internet tabanlı olarak geliştirilen bulut bilişimin dünyada olduğu gibi ülkemizde de kullanımı giderek yaygınlaşmaktadır. Bulut bilişimin kullanıcılarına sunduğu başta ekonomik fayda olmak üzere birçok faydanın yanı sıra, bu yeni modelle yeni güvenlik sorunları da ortaya çıkmıştır. Bulut bilişimin maruz kaldığı güvenlik tehditlerine, açıklıklarına, yetersizliklerine karşı çeşitli güvenlik mekanizmaları geliştirilmesi üzerine çalışmalar yapılmaktadır. Bu çalışmada, 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.**

***Anahtar Kelimeler*—Bulut bilişim, Güvenlik, Saldırı tespit sistemi, Sanallaştırma.**

***Abstract*— 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 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. There are many studies on improving a variety of security mechanisms for security threats, vulnerabilities and insufficiencies in cloud computing. This study is aims to use host-based intrusion detection system (HIDS) for enabling security of virtualization and virtual machine in cloud computing and thus providing secure cloud computing.**

***Index Terms*—Cloud computing, Security, Intrusion detection system (IDS), Virtualization.**

I. GİRİŞ

BİLİŞİM teknolojileri, bilgi çağı olarak nitelendirilen modern çağın getirdiği yenilikler ve kullanıcıların hızla değişen ihtiyaçları nedeniyle sürekli bir değişim ve gelişim göstermektedir. Kullanıcıların ofis bağımlılığı olmadan çalışabilme olanaklarının artması, dinamik yapıdaki ofislerin yaygınlaşması, daha az kaynak ile daha fazla hizmet sunma gerekliliğinin ortaya çıkması ve ekonomik nedenlerden dolayı kurumlarda kısıtlı miktardaki kaynakların daha etkili kullanımı önem kazanmış ve tüm iş gruplarının kendi içlerinde yeniden yapılanması gerekliliği ortaya çıkmıştır. Bilişim sektöründe ortaya çıkan bu ihtiyaçlara yönelik olarak fiziksel sistemlerinin sanal ortamlara taşınması ve bir fiziksel sistemin üzerinde birçok sanal sistem kullandırılması çözümleri geliştirilmiştir. Zamanla kullanıcıların uygulamalarını mekân, 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, yeni bir bilişim teknolojisi olarak sanallaştırma alt yapısının üzerine yapılandırılmıştır.

Bulut bilişim, kullanıcılarına ekonomik faydası başta olmak üzere birçok fayda sağlamaktadır. Buna karşın dağıtık servise dayalı mimarisi, çoklu-kullanıcılar ve çoklu-domain altyapıları nedeniyle, tehditlere ve savunmasızlıklara karşı dayanıksız olarak görülmektedir. Bulut bilişimde güvenlik sorunu kritik bir sorundur. Bu sorun nedeniyle kullanıcılar bulutları kullanmaya tereddüt etmektedirler [1]. Güvenlik sorunları, servisleri barındıran bulut sağlayıcılarını daha fazla ilgilendirmektedir. Antivirüs yazılımları, güvenlik duvarları, bekçi sistemleri ve özellikle saldırı tespit sistemleri gibi mekanizmaları kullanılarak güvenlik sorunlarının üstesinden gelinmeye çalışılmaktadır. Güvenlik mekanizmaları ele alınırken bulut bilişimin doğası gereği kaynak kullanım verimliliği konusu göz önünde tutulmaktadır.

II. BULUT BİLİŞİM

Bulut bilişim, kullanıcılara veriye daha az maliyetle ve daha hızlı bir şekilde ulaşma imkânı 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 birçok 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) [2] tarafından yapılan tanıma göre bulut bilişim, yapılandırılabilir bilişim kaynaklarından (bilgisayar ağları, sunucular, veri depolama, uygulamalar ve servisler vb.) oluşan ortak bir havuza, uygun koşullarda ve isteğe bağlı olarak her zaman, her

Fatma Didem ÖĞRETMEN, Harran Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 63000, Şanlıurfa, Türkiye. (e-mail: fdidemogretmen@harran.edu.tr).

Muhammed Ali AYDIN, İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34320, İstanbul, Türkiye. (e-mail: aydinali@istanbul.edu.tr).

Ahmet SERTBAŞ, İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34320, İstanbul, Türkiye. (e-mail: asertbas@istanbul.edu.tr).

Bu çalışma aynı zamanda İstanbul Üniversitesi, Fen Bilimleri Enstitüsü'nde **Güvenli Bulut Bilişim İçin Saldırı Tespit Sistemi Kullanımı** adlı yüksek lisans tezinin bir parçasıdır.

yerden erişime imkân veren bir modeldir. Söz konusu kaynaklar asgari düzeyde yönetimsel çaba ve servis alıcı-servis sağlayıcı etkileşimi gerektirecek kolaylıkta tedarik edilebilmekte ve elden çıkarılabilmektedir [2].

III. SANALLAŞTIRMA

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 paylaşılması veya birleştirilmesiyle yapılmaktadır.

Sanallaştırma, 1960'lı yıllarda IBM şirketinin anaçatı (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 [3]. İlk çıktığı günden bu yana sanallaştırma kavramı önemli ölçüde olgunlaşmış ve hafıza, depolama, işlemciler, yazılım, ağ servisleri gibi bilişim teknolojilerinin tüm yönlerine uygulanmıştır.

Sanallaştırma; fiziksel sınırlamaların ortadan kalkmasının sağlanması, tek bir merkezden birden çok sunucunun yönetilebilmesi ile yönetim yükünün en aza indirgenmesi, alt yapı maliyetlerinin büyük ölçüde azaltılması, 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. Bu sayede bilişim teknolojilerinde oldukça yaygın kullanım alanına sahip bir teknolojidir.

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.

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 bağdaştırıcısı, 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 hipervizör tarafından tek bir donanım sunumu vardır.

IV. BULUT BİLİŞİMDE GÜVENLİK

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 [4]. Bununla birlikte bulut sağlayıcısı veya yöneticisinin de her zaman güvenilir olacağının garantisi yoktur.

Bulut bilişimde kritik konu olan güvenlik konusunda karşılaşılan sorunlara genel olarak iki açıdan yaklaşılmaktadır. Birincisi, bulut servis sağlayıcısının sağladığı servislerin güvenli olduğunu garanti edebilmesi ve kullanıcının kimlik yönetimini başarabilmesi; ikincisi ise, kullanıcının kullandığı servislerin yeteri kadar güvenli olduğundan emin olabilmesidir.

Bulut bilişimde karşılaşılan güvenlik riskleri veri gizliliği ve mahremiyetinin korunması, yönetim yetersizliği, yönetim arayüzündeki olası güvenlik açığı, bulut çalışanlarının kötü niyetli davranışları, kullanılabilirliğin garantilenememesi, izolasyon başarısızlığı, uyum ve yasal riskler olarak belirtilmektedir [5].

A. *Bulut Bilişimde Güvenlik Mekanizmaları*

İnternet kullanımının yaygınlaşması ile birlikte bilişim sistemlerine karşı güvenlik tehditlerinde önemli artışlar ve saldırı türlerinde 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 kullanılmaktadır. Bu güvenlik mekanizmalarının hiçbirinin tek başına kullanılması güvenlik açısından tam 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 sağlanması amacıyla çeşitli yönetim modellerine odaklanmıştır [6]. Bu modellerde çoğunlukla Saldırı Tespit Sistemi, Güvenli Bilişim, veri şifreleme gibi mekanizmalar yer almaktadır.

Saldırı Tespit Sistemi (Intrusion Detection System – IDS)

Bir kaynağın veya verinin güvenilirliğini, 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 [7]. IDS, bir tür alarm sistemi olarak düşünülebilir. Saldırıların tespit edilebilmesi tetikleme mekanizmaları ile gerçekleştirilir. Bir IDS, birkaç bileşenden oluşmaktadır [8]:

Algılayıcılar (Ajanlar): Güvenlik olaylarını oluşturur.

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

Merkezi Motor: Algılayıcılar tarafından günlüğe kaydedilen kayıtları bir veri tabanında tutar ve bir güvenlik olayı

alındığında uyarıları oluşturmak için bir kurallar sistemini kullanır.

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.

Sunucu-Tabanlı Saldırı Tespit Sistemi (Host-Based IDS - HIDS)

Sunucu-Tabanlı saldırı tespit sistemi (host-based IDS – HIDS), hedef sistemin bireysel bilgisayarların [8] olduğu, tasarlanan ilk saldırı tespit yazılım türüdür. 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 (ajanlardan) oluşmaktadır. HIDS sistem durumu izlemek için işletim sistemi denetim rotalarından ve sistem log kayıtlarından faydalanarak karar verebilmektedir. Hangi kaynakların hangi programlara eriştiğini algılayabilmektedir.

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 izlemekte ve tespit etmektedir.

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

Ağ tabanlı saldırı tespit sistemleri (Network-Based IDS – NIDS), belirli bir sunucudan ziyade ağın kendisine odaklanmaktadır. Ağ üzerinden geçen trafiği veri kaynağı olarak görüntülemektedir. NIDS, ağ segmenti veya anahtarlama cihazını dinleyerek, bu ağ segmentine bağlı birden çok hostu etkileyen ağ trafiğini izleyebilmektedir.

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 tetiklemektedir.

Güvenilir Bilişim (Trusted Computing)

Güvenilir bilişim (Trusted Computing – TC), Trusted Computing Group (TCG) tarafından geliştirilen ve desteklenen bir metodolojidir [9]. 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 [9].

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 örneklerin bütünlüğünü kontrol eder [10]. 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 [10].

V. GÜVENLİ BULUT BİLİŞİM İÇİN SALDIRI TESPİT SİSTEMİ KULLANIM ÖRNEĞİ

Bulut bilişimde sistem güvenliğini arttırmak için en popüler yöntem 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. Literatürde bulut bilişim için birçok IDS mekanizması tanımlanmıştır.

Bu çalışmada güvenli bulut bilişim için farklı IDS kullanım modellerinin oluşturulması amacıyla bulut bilişimde kullanılabilecek IDS tabanlı bir hibrid yaklaşım sunan AdjointVM [11] Koruma Modeli ile AdjointVM yaklaşımıyla sunulan güvenlik mekanizmasının zayıf yönlerinin üstesinden gelmek amacıyla birtakım iyileştirmeler ekleyerek eksikliklerinin giderilmesini hedefleyen U. Oktay ve Ark. [12]'nın öneri olarak literatüre sunduğu İyileştirilmiş AdjointVM Koruma Modeli, IDS kullanım mekanizmaları açısından ele alınarak bulut bilişimde IDS kullanımının nasıl olması gerektiğinin ortaya konulması için test ortamında gerçekleştirilmiş ve bulgular karşılaştırılmıştır.

A. *İlgili Çalışmalar*

AdjointVM Koruma Modeli

J.Kong [11], 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 VMM barındırmaktadır. Her iki saldırı türüne karşı koruma sağlayan hibrid bir mimari geliştirilmiştir.

AdjointVM Koruma Modeli'nde sanallaştırma ortamı olarak açık kaynak kodlu bir platform olan Xen Hypervisor [13] seçilmiştir.

Xen hipervizörü, sanallaştırma katmanının ince ve minimal olması düşüncesine dayanarak geliştirilmiştir. Bu tasarım fikrine dayanarak, asıl sanallaştırma işi hipervizörün bir seviye üstüne devredilmesi gerekmektedir. Bu nedenle sıradan VM'lerden daha fazla ayrıcalıklara sahip Domain 0 (Dom0)

olarak adlandırılan özel bir VM yer almaktadır. 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. Xen terminolojisinde tüm VM'lere domain denilmektedir. Yönetimden sorumlu VM Dom0 iken, diğer tüm VM'ler de DomUs olarak adlandırılmaktadır

AdjointVM modelinde her bir VM, AdjointVM denilen başka bir VM tarafından izlenmekte ve korunmaktadır. AdjointVM'in kurulması ile Xen, bir VM'nin adres alanının 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 [14] ile işletim sistemi çekirdeğini izleyen Kernel Monitor Daemon (KMD) [11] kullanılmıştır. OSSEC sunucusu AdjointVM üzerine kurulur, ajanı ise korunmuş VM üzerindedir. Saldırılara ait imzalar OSSEC sunucusu üzerinde bulunmakta ve ajanlar saldırı bilgilerini sunucu bünyesindeki saldırı imza veri tabanından almaktadır.

İyileştirilmiş AdjointVM Koruma Modeli

AdjointVM, bulut ortamı için iyi bir model olarak sunulmasına karşın bazı güvenlik sorunları öngörülmektedir. 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 [12] yaklaşımına göre yeni bir model önermişlerdir. Önerilen modelde, AdjointVM modelinde koruyan VM'nin güvenliği nasıl sağlanacak sorusuna cevap bulunmaktadır. 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 modelde 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 [15].

B. Gerçekleştirme Ortamı ve Kullanılan Bileşenler

Gerçekleştirme ortamında, referans model olarak ele alınan J. Kong'un yaptığı AdjointVM Koruma Modeli uygulamasının [11] gerçekleştirme ortamına uygunluk açısından Intel Virtualization Technology (Intel®VT) [16] 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.

C. 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 çalışmada, sistem üzerindeki istenmeyen ve zararlı aktivitelere karşı HIDS'lerin nasıl konumlandırılması ve yapılandırılması gerektiği ile ilgili iki güvenlik mekanizması mimarisi oluşturulmuştur. Bunlar;

1) AdjointVM Koruma Mekanizması,
2) İyileştirilmiş AdjointVM Koruma Mekanizması.

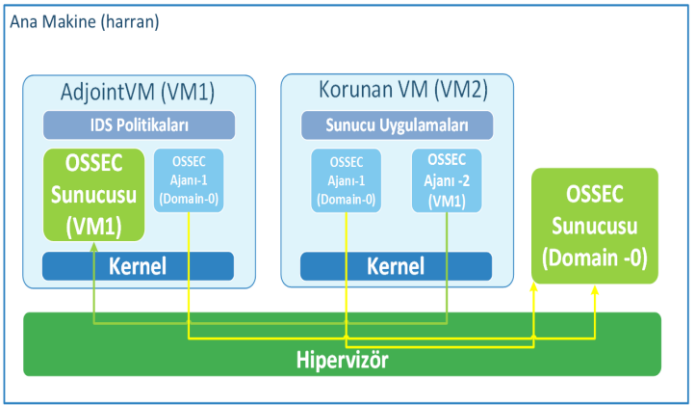
Birinci mekanizmada, sistem içerisinde yer alan VM'lerin güvenliğinin sağlanması için her bir VM'yi koruyacak AdjointVM olarak adlandırılan eşlenik bir VM oluşturulmuştur. Oluşturulan bu AdjointVM, korunan VM'nin güvenliğinden sorumludur. OSSEC'in sunucu istemci mimarisinden dolayı OSSEC sunucusu eşlenik 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.

İkinci mekanizmada ise 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.

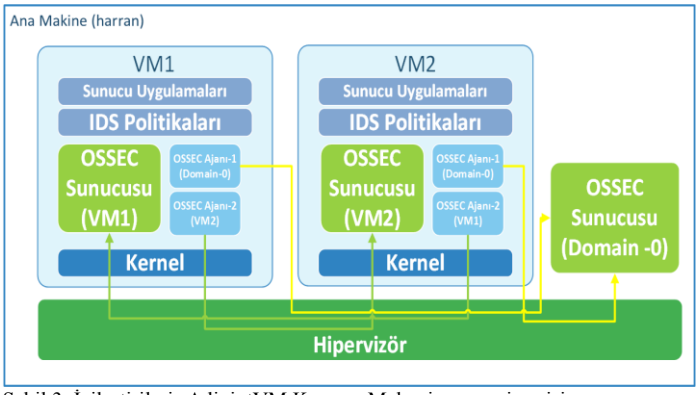
Gerçekleştirme ortamında ana makine "harran" olarak adlandırılmaktadır. AjdointVM Koruma Modelinin saldırı tespit mekanizmasının uygulanması için J. Kong [11]'un

çalışmasında belirttiği üzere sistemde en az 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.

Şekil 2'de AdjointVM Koruma Mekanizmasına ait mimari, Şekil 3'de ise İyileştirilmiş AdjointVM Koruma Mekanizmasına ait mimari gösterilmektedir.



Şekil 2. AdjointVM Koruma Mekanizması mimarisi



Şekil 3. İyileştirilmiş AdjointVM Koruma Mekanizması mimarisi

D. Bulgular

Bu çalışmada 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 durumda donanım 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.

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.

Ana makine ve VM'lerin tükettiği hafıza 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 imkân verir. 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.

xentop ve *top* yazılımında varsayılan olarak 3 saniye aralıklarla donanım kullanım miktarları gösterilmektedir. Sistemlerden elde edilen değerlerin daha güvenilir ve doğruluğunun daha yüksek olabilmesi amacıyla farklı zaman dilimlerinde elde edilen verilerin çeşitliğini arttırmak için her iki yazılımı –d opsiyonuyla çalıştırarak donanım kullanım durumu güncelleme sıklığı 1 saniye ve 10 saniye olarak değiştirilmiş olup, her bir zaman aralığı için 50 iterasyon yapılarak veriler elde edilmiştir. Bu işlem 1, 3 ve 10 saniye zaman dilimleri için birden fazla kez tekrarlanmıştır. Bu şekilde her bir durum için ayrı ayrı 1, 3 ve 10 saniye aralıklarla ölçümler yapılarak sistem üzerinden CPU ve RAM kullanım miktarları yüzde olarak alınmış; 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 1, Tablo 2, Tablo 3 ve Tablo 4'te gösterilmiştir.

TABLO 1. FARKLI FREKANSLARDA CPU KULLANIMLARINA GÖRE GÜVENLİK MEKANİZMALARININ KIYASLANMASI

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 2. FARKLI FREKANSLARDA RAM KULLANIMLARINA GÖRE GÜVENLİK MEKANİZMALARININ KIYASLANMASI

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 3. ORTALAMA CPU KULLANIMLARINA GÖRE GÜVENLİK MEKANİZMALARININ KIYASLANMASI

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. ORTALAMA RAM KULLANIMLARINA GÖRE GÜVENLİK MEKANİZMALARININ KIYASLANMASI

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**

E. Elde Edilen Bulguların Karşılaştırılması

AdjointVM Koruma Mekanizması ve İyileştirilmiş AdjointVM Koruma Mekanizmasından elde edilen bulgular göstermiştir ki genel olarak IDS kullanımı sistemdeki kaynak tüketim miktarını arttırmaktadır. İyileştirilmiş AdjointVM Koruma mekanizması, AdjointVM Koruma Mekanizmasına göre VM'lere daha fazla iş yaptırmış, bu nedenle CPU ve RAM kullanım miktarları kıyaslandığında daha fazla kaynak tüketimine neden olmuştur.

VI. SONUÇ

Bu çalışmada 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 optimum güvenlik mekanizmasının tercih edilmesi konusunda fayda sağlamak amaçlanmıştır.

Öncelikle sistemde hiçbir güvenlik mekanizması kullanılmadığı zaman donanım kullanım miktarları tespit edilmiştir. İkinci durumda güvenlik modeli olarak literatürde yer alan AdjointVM yaklaşımına ait sunucu-tabanlı saldırı tespit sistemi kullanım mimarisi yönünden model ele alınarak AdjointVM Koruma Mekanizması gerçekleştirilmiştir. Oluşturulan mekanizmanın donanım kullanım miktarları incelenmiştir.

Üçüncü durumda literatürde öneri olarak yer alan, AdjointVM IDS yaklaşımına bazı ek iyileştirmeler ekleyerek eksikliklerinin giderilmesiyle oluşturulan İyileştirilmiş AdjointVM Koruma Mekanizması sunucu-tabanlı saldırı tespit sistemi mimarisi gerçekleştirilerek donanım kullanım miktarları incelenmiştir.

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 sonucuna varılmıştır.

Gelecek çalışma olarak İyileştirilmiş AdjointVM Koruma Mekanizmasının eksik yönleri tespit edilerek bu eksikliklerin giderilmesini sağlayan, böylelikle sistemin direncinin arttırılmasını ve esnek bir güvenlik politikasının oluşturulmasını hedefleyen yeni bir güvenlik mekanizması önerilecektir.

KAYNAKLAR

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

[2] P. Mell, and T. Grance, "The NIST Definition of Cloud Computing, NIST Special Publication 800-145 (SP800-145)", National Institute of Standards and Technology, September 2011.

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

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

[5] "Security for Cloud Computing 10 Steps to Ensure Success", (2012, August), *Cloud Standards Customer Council* [Online], Erişilebilir: http://www.cloud-council.org/Security_for_Cloud_Computing-Final_080912.pdf.

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

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

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

[9] Trusted Computing Group [Online], Erişilebilir: http://www.trustedcomputinggroup.org/.

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

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

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

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

[14] OSSEC HIDS [Online], Erişilebilir: http://www.ossec.net/.

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

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