



**HAVA HARP OKULU
HAVACILIK VE UZAY TEKNOLOJİLERİ
ENSTİTÜSÜ**



**KISITLI KAYNAK KULLANILARAK DÜŞÜK MALİYETLİ
DAĞITIK SİBER TUZAK SİSTEMİ
KURULMASI ve YÖNETİLMESİ**

YÜKSEK LİSANS TEZİ

Burak ÇAKMAK

Bilgisayar Mühendisliği Ana Bilim Dalı Başkanlığı

Siber Güvenlik Programı

Tez Danışmanı: Doç.Dr. Güray YILMAZ

AĞUSTOS 2016

HAVA HARP OKULU
HAVACILIK VE UZAY TEKNOLOJİLERİ ENSTİTÜSÜ

**KISITLI KAYNAK KULLANILARAK DÜŞÜK MALİYETLİ
DAĞITIK SİBER TUZAK SİSTEMİ
KURULMASI ve YÖNETİLMESİ**

YÜKSEK LİSANS TEZİ

Burak ÇAKMAK
(113109)

Bilgisayar Mühendisliği Ana Bilim Dalı Başkanlığı

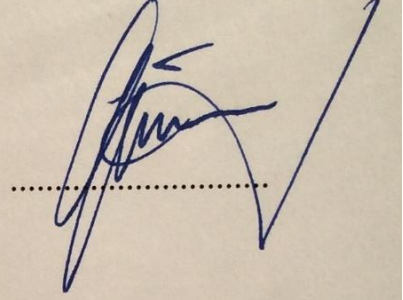
Siber Güvenlik Programı

Tez Danışmanı: Doç.Dr. Güray YILMAZ

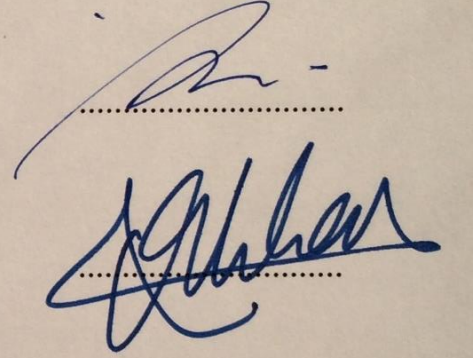
AĞUSTOS 2016

Hava Harp Okulu Havacılık ve Uzay Teknolojileri Enstitüsünün 113109 numaralı Yüksek Lisans Öğrencisi **Burak ÇAKMAK**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra **Güray YILMAZ** danışmanlığında hazırladığı “**KISITLI KAYNAK KULLANILARAK DAĞITIK SİBER TUZAK SİSTEMİ KURULMASI ve YÖNETİLMESİ**” başlıklı tezini aşağıda imzaları olan iüri önünde basarı ile sunmustur.

Tez Danışmanı : **Doç. Dr. Güray YILMAZ**
Hava Harp Okulu



Jüri Üyeleri : **Doç. Dr. İlker BEKMEZCİ**
Hava Harp Okulu



Yrd. Doç. Dr. Akhan AKBULUT
İstanbul Kültür Üniversitesi

Teslim Tarihi : **5 Ağustos 2016**
Savunma Tarihi : **19 Ağustos 2016**



Bu tez çalışmasında belirtilen görüş ve yorumlar yazara aittir. Türk Silahlı Kuvvetleri'nin ya da diğer kamu kuruluşlarının görüşlerini yansıtmaz. Ayrıca bu tez çalışması bilimsel ahlak ve etik değerlere uygun olarak yazılmış olup, yararlanılan tüm eserler kaynaklarda gösterilmiştir.

Ağustos 2016

Burak ÇAKMAK





Aileme,





ÖNSÖZ

Tez çalışmamı hazırladığım zorlu sürecin her aşamasında bana güvenen, yardımcı ve destek olan, tez çalışmasının son halini almasında, eksik yönlerinin ortaya çıkarılması ve hataların giderilmesi kapsamında gayretlerini eksik etmeyen ve değerli zamanını harcayan tez danışmanım Doç.Dr.Hv.Müh.Alb. Güray YILMAZ'a, Bilgisayar Mühendisliği Ana Bilim Dalı Başkanlığı öğretim üyelerine, çalışmam süresince büyük ilgi ve fedakarlıklarıyla her zaman yanımda olan aileme, hata düzeltmeleri için ablam Ayşenur YALINKILIÇ'a, maddi manevi desteğini esirgemeyerek motivasyonumu sağlayan arkadaşım İlyas Gökhan ŞAHİN'e, tez süreci boyunca anlayışı ve desteği ile yanımda olan müdürüm Bahtiyar TAN'a, teşekkürlerimi sunmayı bir borç bilir, sonsuz minnettarlığımı sunarım.

Ağustos 2016

Burak ÇAKMAK



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET.....	xix
1. GİRİŞ	1
1.1 Saldırıların Etkisi	5
1.2 Tezin Amacı ve Kapsamı	6
2. LİTERATÜR ARAŞTIRMASI.....	9
2.1 Araştırma Konuları.....	9
2.2 Yapılan Çalışmalar	10
3. KULLANILAN TEKNOLOJİLER.....	25
3.1 Honeypot Tuzak Sistemi	25
3.1.1 Honeypotların çalışma yapısı	25
3.1.2 Honeypot yapılandırma Çeşitleri	25
3.1.3 Honeypotların kullanım avantajları ve dezavantajları	26
3.1.4 Kippo honeypot	26
3.2 Hadoop Anaçatı Sistemi.....	27
3.2.1 Büyük veri kavramı ve meydana getiren unsurlar	27
3.2.1.1 Mobil cihazların gelişimi ve etkisi	27
3.2.1.2 Bulut bilişimin etkileri	28
3.2.1.3 Sosyal medyanın etkisi.....	28
3.2.1.4 Nesnelerin interneti kavramı	28
3.2.2 Google dosya sistemi	28
3.2.3 Hadoop dosya sistemi	29
3.2.4 Map/Reduce	29
3.2.4.1 Name node	33
3.2.4.2 Data node	33
3.2.4.3 Job tracker	33
3.2.4.4 Task tracker	33
3.2.5 Apache spark	33
3.3 Raspberry Pi	35
3.3.1 Raspberry pi çeşitleri ve özellikleri.....	36
3.3.2 Raspberry pi performansı ve karşılaştırılması.....	37
4. KISITLI KAYNAK KULLANILARAK DÜŞÜK MALİYETLİ DAĞITIK SİBER TUZAK SİSTEMİ MİMARİSİ.....	39
4.1 Donanım Altyapısının Hazırlanması.....	40
4.1.1 Malzeme seçimi ve özellikleri	42
4.1.2 Maliyet analizi.....	43
4.2 Yazılım Altyapısının Hazırlanması.....	44

4.2.1 Kippo honeypot tuzak sisteminin kurulumu ve yapılandırılması	44
4.2.2 Büyük veri mimarisinin uygulanması	50
4.2.2.1 Hadoop kurulumu ve yapılandırılması	51
4.3 Deney Sonuçlarının Analizi	58
4.3.1 Kippo honeypot ile elde edilen bulgular	58
4.3.2 Toplanan honeypot verisinin temizlenmesi	67
4.3.3 Örnek saldırı senaryosunun değerlendirilmesi	68
4.4 Deneyi Önceki Çalışmalardan Ayıran Yönleri.....	70
5. SONUÇ VE DEĞERLENDİRME	73
5.1 Karşılaşılan Problemler ve Kısıtlar	73
5.2 Çalışmanın Kullanım Alanları.....	74
5.3 Gelecek Araştırmaları ve Öneriler.....	74
KAYNAKLAR.....	76
EKLER.....	82
ÖZGEÇMİŞ.....	89



KISALTMALAR

API	: Application Programming Interface
APT	: Advanced Persistent Threat
DMZ	: Demilitarized Zone
EDI	: Electronic Data Interchange
ETL	: Extration Transformation Load
GKT	: Gelişmiş Kalıcı Tehdit
GFS	: Google File System
GUI	: Graphical User Interface
HDFS	: Hadoop Distributed File System
HP	: Honey Pot
HTML	: Hyper Text Markup Language
HTTP	: Hyper Text Transfer Protocol
IDS	: Intrusion Detection System
IP	: Internet Protocol
IPS	: Intrusion Prevention System
JSON	: JavaScript Object Notation
KİT	: Kasıtsız İçeriden oluşan Tehdit
LAN	: Local Area Network
OS	: Operating System
RDBMS	: Relational Database Management Systems
RFID	: Radio Frequency Identification
REGEX	: Regular Expression
REPL	: Read-Eval-Print-Loop
RPi	: Raspberry Pi
SÖS	: Saldırı Önleme Sistemleri
SQL	: Structured Query Language
SSH	: Secure Shell
STS	: Saldırı Tespit Sistemleri
UIT	: Unintentional Insider Threat
VPN	: Virtual Private Network
WAN	: Wide Area Network
WINE	: Worldwide Intelligence Network Environment
XML	: eXstensible Markup Language



ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Raspberry Pi Kıyaslama Tablosu	36
Çizelge 4.1 : Malzeme Seçimi ve Özellikleri	43
Çizelge 4.2 : Malzeme Maliyet Analizi	44
Çizelge 4.3 : Başarılı Giriş Sayısı ve Oranları	62
Çizelge 4.4 : En Sık Saldırı Yapan 20 IP Adresi	62
Çizelge 4.5 : En Çok Denenen 20 Şifre	64
Çizelge 4.6 : En Çok Denenen 10 Komut	65
Çizelge 4.7 : Honeypot Üzerine İndirilen Zararlı Dosyalar ve Sayıları	66



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Sıfır Gün Saldırısı Oluşumu	11
Şekil 2.2 : Büyük Veri Sistemi	13
Şekil 2.3 : Büyük Verinin İşlenmesi	18
Şekil 2.4 : İş Zekası ve Analitiği	20
Şekil 3.1 : HDFS Dosya Sistemi	31
Şekil 3.2 : Raspberry Pi	35
Şekil 3.3 : Raspberry Pi 2 ile Raspberry Pi 3 Arasındaki Görsel Farklar	36
Şekil 3.4 : Raspberry Pi Ağ Hız Testi	37
Şekil 3.5 : Raspberry Pi CPU Performansı Kıyaslama Değerleri	37
Şekil 3.6 : Raspberry Pi Hafıza Performansı Kıyaslama Değerleri	38
Şekil 3.7 : Raspberry Pi Veri Transfer ve Dosya İndirme Hızı Peformansı	38
Şekil 4.1 : Mantıksal Tasarım	41
Şekil 4.2 : Siber Olaylara Müdahale Aracı (SOMA v3.01)	42
Şekil 4.3 : Deney Ortamının Kurulması	52
Şekil 4.4 : DataNode Kontrol	56
Şekil 4.5 : Hadoop Web Arayüzü	56
Şekil 4.6 : Hadoop Web Arayüzü Çalışan Görevler	57
Şekil 4.7 : Haftalık Siber Saldırı Girişimi Dağılımı	60
Şekil 4.8 : Haftalık Ayrı IP Adreslerinden Gelen Saldırı Girişimi Dağılımı	61
Şekil 4.9 : Honeybot Üzerinde Başarılı Giriş Sayısı	61
Şekil 4.10 : En Sık Saldırı Yapan 20 Lokasyon	63
Şekil 4.12 : En Çok Denenen 20 Kullanıcı Adı	64
Şekil 4.13 : Örnek Saldırı Senaryosu	70



KISITLI KAYNAK KULLANILARAK DÜŞÜK MALİYETLİ SİBER TUZAK SİSTEMİ KURULMASI ve YÖNETİLMESİ

ÖZET

Sistem güvenlik uzmanları, her gün artarak gerçekleşen saldırılara karşı dijital varlıklarını korumak için bitmeyen bir savaş veriyorlar. Devlet ve özel kurumların bu saldırılara karşı önlem alabilmeleri için sistemin maruz kaldığı saldırılar ve saldırganların davranışları hakkında bilgileri ele geçirmeye ihtiyaçları vardır. Ancak bu değerli bilgiler sayesinde çok daha iyi tasarlanmış ve uygulanabilirliği yüksek siber güvenlik önlemleri almaları mümkün olacaktır.

Her komutan askerlerine kendilerini düşmana karşı koruyabilmeleri için öncelikle düşmanlarının kim olduğunu iyi öğrenmeleri gerektiğini emreder. Bu askeri doktrin bilgisayar ağ güvenliği sistemlerine uygulanmak için adeta hazır bir kılavuz niteliğindedir. Tıpkı askeri birlikler gibi kurumların da korumak ile mükellef oldukları kaynakları mevcuttur. Bu kaynakları korumak için sisteminize yönelik tehditleri bilmek gerekmektedir.

Ağ güvenliği, özel şirketler ve devlet kurumları için giderek artan bir tehdit haline gelmektedir. Siber saldırılar her geçen gün ciddi oranlarda artmakta bununla birlikte saldırıların karmaşıklığı da gelişmektedir. Devlet kurumları ve özel sektörde faaliyet gösteren şirketler bu saldırılara maruz kalmakta ve bu saldırılar sonucunda özel bilgiler açığa çıkmaktadır. Bu nedenle tüm kurumların bilgi kaynaklarının korunması ve savunulması için önlemler alması kaçınılmaz olarak zorunlu hale gelmiştir.

Geleneksel ağ savunma sistemi yaklaşımları bilinmeyen saldırı tehditlerini tespit etmekte oldukça yetersizdir. Bu tez kapsamında ortaya konan fikir sistem yöneticilerinin bu problemi aşmaları için bir çözüm ortaya koymaya çalışmaktadır. Bu tez, gerçek zamanlı analizin yapılabilceği, düşük maliyetli, büyük veri analitiği altyapısı sunan yazılımlar aracılığı ile kısıtlı kaynağa sahip bilgisayarlar üzerinde çalışacak bir çözüm sunmaktadır.

Bu araştırma sırasında odaklanılan ana konu saldırgan davranışının analiz edilerek ağ güvenliğinin sağlanması için bilgiler elde edilmesidir. Honeypotlar, saldırganların

zararlı davranışları hakkında bilgi toplamaya imkan sağlayan veri kaynakları olarak kullanılmaktadır. Toplanan bu veriler gözlemlenip, analiz edilerek yeni saldırı türleri ve popüler istismar eğilimlerine karşı erken uyarı sistemleri geliştirmemize yardımcı olmakta kullanılabilirler. Sistem yöneticileri bu veriler ışığında saldırı tespit sistemleri için daha verimli korunma kuralları geliştirebilir.

Sistem güvenlik uzmanlarının karşısına çıkan bir diğer zorluk ise toplanan büyük miktardaki bu verinin gerçek zamanlı olarak işlenmesidir. Tam bu noktada hadoop bu problemin aşılması için yardımımıza koşmaktadır. Hadoop, büyük ölçekli verileri işlemeye imkan veren esnek mimarisi ve aynı zamanda fazla kaynağa ihtiyaç duymaması sayesinde Raspberry Pi gibi kısıtlı kaynağa sahip donanımlar üzerinde çalışarak bu konuda sunduğumuz çözümün önemli bir parçasını oluşturmaktadır. Veri analitiği sistemleri, büyük miktardaki verileri ayrıştırarak yönetilebilir küçük veri setleri haline getirmektedir. Bu sayede geliştirilecek uygulamalar ile yığın üzerinde işlem yapmaya gerek kalmadan gerçek zamanlı tahminler ve yaklaşımlar geliştirmemize yardımcı olur.

Raspberry Pi, kredi kartı büyüklüğünde düşük maliyetli bir bilgisayardır. Önerilen sistem mimarisi çok sayıda Raspberry Pi bilgisayar üzerinde dağıtık olarak çalıştırılacaktır. Tezin içerisinde tasarım ve geliştirilme safhaları açıklanacak, kurulum ve yapılandırma aşamaları ile ilgili bilgiler verilecektir. Saldırgan yaklaşımları tanımlamak için SSH honeypot üzerinde tespit edilen zararlı davranışları analiz edilecek ve veriye dayalı mimari üzerinde analitiği yapılarak bilgi güvenliğine katkı sunacak sonuçların çıkartılması sağlanacaktır.

Honeypot üzerinde yalnızca saldırgan kullanıcıların girişimlerini gözlemleyeceğiz çünkü bu sisteme normal kullanıcıların erişim yetkisi bulunmamaktadır. Bu sayede toplanan bilgi normal kullanıcılar ile karşılaştırılacak ve saldırgan davranış sergileyen kurum içi kullanıcılar tespit edilebilecektir.

Apache Spark ve Apache Hadoop teknolojilerinin kullanımı ile veriye dayalı anlık analiz yapabilecek bir altyapı elde edilmesine imkan sağlayacaktır. Honeypot üzerinden toplanan saldırgan verisi ile kötü kullanıcı davranışı modellenenebilir ve bu sayede saldırgan profilinde tespit edilecek sapmalar ile de tanımlanmamış yeni saldırı türlerini ortaya çıkartıp incelemek mümkün hale gelecektir.

DEVELOPING DISTRIBUTED COST EFFECTIVE CYBER TRAP SYSTEM USING LIMITED RESOURCES

SUMMARY

System security professionals fight a seemingly unending battle to secure their digital assets against an everincreasing onslaught of attacks. Government and private companies need security resource whose value lies in being probed, attacked, or compromised, and provide a valuable tool to collect information about the behaviors of attackers in order to design and implement better defenses.

Any commander orders his soldiers to secure themselves against the enemy, after they get the information about who their enemy is. This military doctrine readily applies to the world of network security. Just like the military, you have resources that you are trying to protect. To help protect these resources, you need to know who is your threat and how they are going to attack.

Network security has become a growing threat to most public and private institutions. Cyber attacks are getting serious mount of rise and complexity is constantly evolving. Government departments and organizations are being breached everyday and then data is compromised by these attacks. Therefore it has become mandatory to take actions for protecting information and ensuring that the security of the information system will be inevitable.

Detection of unknown attack vectors is difficult for the traditional border systems. This thesis is based on an idea to help system administrator to overcome that problem using real time, cost effective data analytics system infrastructure based on big data analytics software and embedded cheap hardware technologies.

During the research on that network security challenge, main focus was on the analysis behaviours of attackers. Honeypots are known as security resources which trap malicious activities. It is a valuable tool to collect information about the behaviours of attackers. Collected data can be analyzed and monitored so that it will provide early warning about new attack and exploitation trends. System administrators can use that captured information to implement better Intrusion detection system rules.

Another challenge for security professional is that large amount of collected data waiting for processing real time. At this point, Hadoop is come to aid this problem for us. It is flexible architecture for large scale data processing and runs on inexpensive lower capacity hardware like Raspberry Pi. Data analytics frameworks decompose problems with a large volume of data into smaller manageable data sets to build applications that are more than just inference and can help to make predictions as well as prescriptions to problems in real time instead of batch processes.

The Raspberry Pi is a low cost, credit-card sized computer. Recommended architecture is going to run on multiple distributed Raspberry Pi computer nodes. The goal of this thesis represents the design stages and installation of the test infrastructure about described solution and proposes a data driven information security architecture that relies on both behavioral analysis of SSH profiles and bad actor data collected from an SSH honeypot to identify bad actor attack vectors.

Only bad actors are going to attempt to use the honeypot system. This systems should not be used by normal users inside an organization's environment. This thesis will collect bad actor behavior from a honeypot ensemble with normal user behavior to increase the true positive rate of bad actor detection.

Using Apache Spark and Apache Hadoop, we can create a real time data driven architecture that can collect and analyze new bad actor behaviors from honeypot data and monitor legitimate user accounts to create predictive and prescriptive models. Previously unidentified attack vectors can be cataloged for review.

1. GİRİŞ

Yeni ve henüz keşfedilmemiş saldırıların tespiti, geleneksel saldırı tespit sistemleri ile oldukça zordur. Bu problemin çözümü için sık kullanılan iki adet yaklaşım bulunmaktadır. Bunlardan birincisi bilgi tabanlı saldırı tespiti yaklaşımı, ikincisi ise davranış tabanlı saldırı tespiti yaklaşımıdır. Bilgi tabanlı yaklaşım, önceden tespit edilmiş, bilinen saldırı verilerini kullanarak önlem alınmasını sağlamaktadır [5]. Davranış tabanlı tespit yaklaşımı ise temelde olağan dışı kullanıcı davranışlarını gözlemleyerek bilinen kullanıcı işlevlerinin dışına çıkan aktiviteler için kullanılır [53]. Kullanılan verinin artmasına ve yeni saldırı türleri keşfedilmeye devam edilmesine rağmen iki yaklaşımın da kısıtları bulunmakta ve yüksek oranda false-positive değerleri üretmektedirler [34].

Siber saldırıların sayısı her geçen gün artmakla birlikte karmaşıklıkları da gelişmektedir. Her gün haber bültenlerinde siber saldırılar ile ilgili içeriklere rastlamak mümkün hale gelmiştir. Gerçekleşen saldırıların sonucu yüksek miktarda kullanıcı bilgisi saldırganların eline geçmekte yada açığa çıkmaktadır. 2014 yılında Amerika Birleşik Devletleri'nde gerçekleşen JP Morgan Chase Bank [50] ve "Home Depot" [49] olarak geçen iki büyük saldırı meydana gelmiştir. Her iki saldırının da ortak özelliği "Advanced Persistent Threat" (APT) , bilgi toplama amacı ile yapılan, hedef odaklı saldırılar olmasıdır. Bunların dışında gerçekleşen diğer dikkat çeken büyük saldırılar ise "Stuxnet" ve "Mini Duke" olarak nitelendirilebilir [62]. Bu saldırıların her biri ustaca hazırlanmıştır ve bir çok teknolojiyi kapsayan bir birikim arka planında yatmaktadır.

Yapılan araştırmalar ağ sızıntılarının aylarca tespit edilemeden sistemde varlığını devam ettirdikleri gerçeğini gözler önüne sermiştir [8]. Bazı organizasyonlar saldırıların %70'ini hiç bir zaman tespit edemezken, %30'luk bölümü ise ancak 10 ay sonra tespit edilebilmektedir [8]. Güncel rakamlar bu oranlardan çok daha yüksek olabilir çünkü birçok kurum biriken sistem kayıtlarını ancak günler sonra

incelemektedir. Saldırı tespit sistemi önceden tanımlanmış saldırıları uyarmak için yapılandırılır. Karmaşık ve ustaca hazırlanmış yeni bir APT saldırısı tamamen tespit edilmeden sisteme sızabilir. Çünkü bu saldırının IDS tarafından tespit edilebilmesi için saldırı imzasının önceden tanımlanmış olması gerekir.

Gerçek zamanlı veri üzerine inşa edilecek Apache Hadoop gibi sistemler bizlere ağ trafiği üzerinde neler olduğu ile ilgili analizler yapabilir ve değerli veriler üretebilir. Bu tez kapsamında önerdiğimiz sistem, kayıt edilen ağ trafik kayıtlarını gerçek zamanlı, maliyeti düşük ve kısıtlı kaynaklara sahip bilgisayarlar ile analiz edebilen bir altyapının hem yazılım hem donanım altyapısının kurulmasıdır. Burada üzerinde duracağımız saldırılar ise APT ve Zero-Day saldırılarıdır. Bu kapsamda incelenecek konular ise saldırgan verisinin nasıl toplanacağı, toplanacak veri üzerinde büyük veri analitiğinin nasıl yapılacağı ve bu sistemin üzerinde çalışacağı düşük maliyetli donanım altyapısının nasıl kurulacağıdır.

İşletmeler eskiden yalnızca yönetici paydaşlarına ve devlet kurumlarına sorumluluk oluşturacak bilgi ve belgeleri oluşturup, saklanırdı. Şimdi ise üretim aşamaları, müşterileri, ürünleri ve rakipleri ile ilgili tüm bilgileri bir araya getirerek bir bilgi havuzu oluşturmaya çalışmaktadırlar. Buna imkân veren ise elbette ki bilgisayar sistemlerinde ortaya çıkan gelişmelerdir. İşlemci hızlarının artması bunun yanında maliyetlerinin düşmesi işlenecek veri hacminde önemli artış yaşanmasına öncülük etmiştir. İnternet sayesinde bulut mimarisi içerisinde verinin artık her yerden kolayca erişilebilir hale gelmesi, tüm cihazların birbirleri ile iletişim halinde olması, üretilen verinin tipini ve hacmini arttıran önemli etmenlerdendir. Yeni geliştirilen analiz yaklaşımları ile verinin çıkartılması (Extraction), dönüştürülmesi (Transformation) ve yüklenmesi (Load) – (ETL) işlemleri de orta katmanda bir çok sistem arasında paylaşılmaya ve analiz edilmeye imkân sağlamaktadır. Bulut bilişim ve sanallaştırma teknolojileri ile tüm işletmeler büyük esneklik kazanmış ve daha çok veriyi toplar ve bir çok noktada barındırır hale gelmiştir.

Tüm bu teknolojiler dünyada üretilen veri miktarını da önemli ölçüde arttırmış ve büyük veri kavramının hayatımıza girmesine sebep olmuştur. Açık kaynak analiz çözümleri ve analitik sistemleri bize tüm bu verilerden anlamlı bilgiler oluşturmamız için imkan verir. Önde gelen büyük şirketler bu veri işleme yöntemlerinde yaptıkları yenilikler ile sektörde önemli rekabet avantajı yakalamışlardır. Bu şirketler

geliştirdikleri bu sistemlerin kaynak kodlarını, bunlar ile ilgili yayınladıkları makale ve çalışmaları tüm dünyaya açarak herkesin bu alana oldukça kolay bir şekilde girmelerinin de önünü açmıştır.

Saldırı Tespit Sistemleri (STS) ve Saldırı Önleme Sistemleri (SÖS) ile bilgi varlıklarının güvenliğinin sağlanması tam anlamı ile mümkün olmamaktadır. 2011 – 2014 yılları arasında gerçekleşen ve medyada öne çıkan saldırılar Gelişmiş Kalıcı Tehdit (GKT) olarak sınıflandırılmıştır. Bu saldırılar STS sistemlerini atlatarak ağa sızmayı başarmışlardır [49,50].

Kasıtsız içeriden oluşan tehditler (KİT) ve Zero-Day saldırıları da kurumların bilgi güvenliğini tehdit eden öne çıkan saldırılar olarak belirlenmiştir [51]. Burada KİT olarak bahsedilen kavram, kurum içerisinden farkında olmadan kurumu tehlikeye atacak davranış ve eylemlere sebep olan tehditlerdir. KİT'ler kurumlar için büyük endişe kaynağıdır çünkü bu tehdidi yaratan birim halihazırda ağa dahil olduğu için STS'nin bu kalıcı tehditleri engellemesi mümkün değildir [45].

Sıfır gün saldırıları, zafiyetin tespit edildiği günden bu açığı kapatacak yamanın çıkıp uygulanması aşamasına kadar gerçekleştirilebilir. Saldırganlar, internet üzerinde sürekli olarak bu tip zafiyetleri barındıran yazılım veya sistemleri araştırırlar. Ardından bu açıklara sahip olan, istismar edebilecekleri kurumları tespit ederek hedeflerine ulaşmaya çalışırlar [5]. Burada göz önünde bulundurulması gereken bir diğer tehlike de kurumların üçüncü taraflardan aldığı/verdiği servislerin de bu tip açıkları içerebileceğidir. Birçok kurum kendilerine ait ağı bu servislere güvenerek istismar edilmeye açık hale getirmektedir. Gerçekleşen saldırıların birçoğunun üçüncü taraflardan alınan servisler üzerinden olduğu belirlenmektedir.

Kurumları hedef alan ağ güvenlik ihlalleri sayısı her geçen gün artıyor. 2013 yılında SANS tarafından yapılan bir ankette katılımcıların %65'i APT saldırılarına önlem olarak yalnızca STS ile değil daha fazla işlem kayıt tutacak şekilde sistemlerini güncellediklerini belirtmişler. 2014 yılında gerçekleşen bir çok GKT saldırısı içeriden oluşan zaaflar ile meydana gelmiş olduğu tespit edilmiştir. Bu saldırıların tespit edilememesindeki temel sebepler ise kullanıcı farkındalığının düşük olması, normal kullanıcı davranışı analizlerinin yapılmaması, veri depolama ve analiz maliyetlerinin yüksek olması olarak tespit etmişler [32].

Makine öğrenmesi ve istatistiksel analiz yöntemleri bu saldırıların tanımlanması için kullanılabilir. Son 20 yıldır veri analitiği teknikleri bilgi güvenliği alanında yoğun bir şekilde kullanılmaktadır. Ancak bu yöntemler öncelikli olarak sadece STS'ler üzerinde uygulanmaktadır. STS ve SÖS'ler kuraldışı davranış tespiti için Markov Zincirleri ve davranış analiz tekniklerinden yardım alır [53].

Bu tekniklerin çoğu ağ trafiğini analiz etmek için kullanılmaktadır. Geçmiş yıllara bakıldığında, algılayıcı sistemler (STS, SÖS ve Honeypot) ile toplanan veriler ağ hızına bağlı olarak işlenmesi ve uyarı oluşturulması oldukça zor olduğu görülmektedir [2]. Ayrıca ağ trafiğindeki yoğunluklar tespit edilse bile saldırganlar saldırı hızlarını yavaşlatarak duruma ayak uydurup amaçlarına ulaşmakta sıkıntı çekmezler [8].

İşlemci gücü yüksek, depolama birimleri hesaplı küçük bilgisayarların piyasaya çıkması ve bu kısıtlı kapasiteli donanımlar üzerinde çalışacak Hadoop gibi veri analitiği yazılımlarının açık kaynak bir şekilde üretilmesi ile bilgi güvenliği açısından yepyeni bir yaklaşımın kapılarını bizlere açmış bulunmaktadır [9]. Bu çözümler ile zamanında yapılması çok zor gözü ile bakılan birçok zorluğun üstesinden gelmek mümkün olmaktadır [1]. Fakat elbette bu yeni çözümler yanlarında başka en uygun şekle sokma ve yapılandırma problemlerini de beraberlerinde getirmişler [42].

2015 yılı içerisinde yayınlanan bir çok çalışmanın odak noktası bu bilgi güvenliği problemlerinin üstesinden gelecek yöntemleri ele almaktadırlar. Bu çalışma başlıklarından bir kaç: Veriye dayanan bilgi güvenliği, veri analizi ve görselleştirme yaklaşımları, NetFlow ve IPFIX ile ağ güvenliği, bilgi güvenliği analitiği, kullanıcı davranışı ve kuraldışı aktivite tespiti gibi konularda birçok yayın oluşturulmuştur [32]. Burada bahsedilen yayınlar da veri işlerken ortaya çıkan sorunları ortadan kaldıracak yanıtlar verememektedir. Daha çok son teknolojileri, bilgi güvenliği uzmanlarını veri analitik tekniklerinde hızlandırmak için kullandırmayı amaçlamaktadırlar. Tüm bunlar göz önünde bulundurulduğunda, büyük veri analitiği ile bilgi güvenliğinin sağlanması alanı birçok yeni tekniğin denenmesi ve yeni çözümler önerilmesi için uygun bir çalışma sahası olarak nitelendirilebilir [44].

1.1 Saldırıların Etkisi

Son yıllarda bilgi güvenliği ihlalleri için yapılan haberlerin sayısı oldukça artmıştır. Büyük firmalar, kurumlar sayısız saldırının hedefi olmuşlardır ve bunlar ile ilgili bir çok yayın, haber hazırlanmıştır. Etkisi en büyük olaylara, eski Amerikan istihbarat çalışanı Edward J. Snowden'ın ortaya çıkardığı, Amerikan Ulusal İstihbarat Ajansının vatandaşını izlemek için kullandığı altyapının gözler önüne serilmesi örnek olarak verilebilir. Bu olay bilgileri sızan ve sızdıranlar tarafından dikkatle takip edilmiştir. Fakat bilgi güvenliğini bizim açımızdan daha ilginç hale getiren kısmı, yapılan sızma saldırılarının büyük bölümünün farkedilmemesi, farkedilseler bile üzerlerinden uzun bir zaman geçtikten sonra tespit edilmeleridir [8].

Büyük şirketler saldırıları önlemek için küçük şirketlere oranla çok daha büyük bütçeler ile daha fazla çalışma yaparlar. 2014 yılında yapılan bir araştırmanın sonuçlarına göre, 100 milyon dolardan düşük gelirli şirketler yaklaşık 1,091 adet saldırıyı tespit ederken, 100 milyon – 1 milyar dolar arasınd geliri olan orta büyüklükteki şirketler ortalama 4,227 adet saldırıyı tespit etmişlerdir. 1 milyar dolardan daha büyük gelirli büyük şirketler ise yaklaşık 13,138 adet saldırıyı tespit etmişlerdir [8]. Burada üzerinde durulması gereken konu küçük şirketlere yapılan saldırıların büyük şirketleri de doğrudan etkilediğidir. Çünkü diğer küçük firmalar ile iş yapan büyük firmalar da işletme altyapılarını bir noktada birbirleri ile bağlamaktadırlar. Küçük ve orta büyüklükteki firmalarda ortaya çıkacak bir sızma faaliyetinin büyük firmaları da etkileme olasılığı oldukça yüksektir.

Artan saldırılar ile işletmelerin bunlara ayırdığı bütçeler de doğru orantılı olarak artmaktadır. Ortaya çıkan her saldırı haberi ile kurumlar bilgi güvenliği yönetimi, saldırı önleme mekanizmaları ve kayıpların önüne geçmek adına bu konuya daha fazla ağırlık vermeye başladıkları görülmektedir. 2014 yılında gerçekleşmiş siber saldırıların oluşturduğu finansal kayıp yaklaşık olarak 2.7 milyar dolardır [8].

Siber saldırılar yalnızca dışardaki hackerlar tarafından gerçekleştirilmez. Kurum altyapılarını hedef alan saldırgan profilleri yabancı devletler, rekabet halinde olunan diğer firmalar, bilgi tüccarları, organize suç örgütleri olabileceği gibi ticari ortaklar, müşteriler, kurum içerisinden çalışanlar, eski çalışanlar tarafından da bilgi varlıkları hedef alınabildiği tespit edilmiştir [8]. Snowden tarafından sızdırılan

belgeler sayesinde içeriden yapılan saldırıların nasıl kontrol altına alınacağı ile ilgili önemli bilgiler elde edilmiştir. Kurumların genel yaklaşımı daha çok dışarıdan gelecek saldırılar üzerinde yoğunlaştığı için bu iç tehdidin de yadsınamaz boyutlarda olduğunu gözler önüne sermiştir.

Dışarıdan gerçekleşen saldırılar içeriden de yardım aldığı durumlarda bilgi güvenliği uzmanlarının bilgi güvenliğini sağlamaları için uçtan uca bir analiz mekanizması kurmaları gerektiği açıkça ortadadır [34]. Yapılan bir araştırma organizasyonel endişeler yaratan durumları da teknik durumlar kadar incelenmesi gerekliliğini ortaya koymaktadır. Yapılması gereken davranışsal analizin de dıştan gelecek tehditler kadar içten gelecek tehlikelere karşı da korunması bir başka önem verilmesi gereken alandır. Bu iç alana bilgisayar donanımı, ağ donanım altyapısı ve organizasyonel bilgilerin de denetlenmesi gerekliliğini ortaya çıkartmaktadır [13].

Davranışsal bilgi güvenliği yaklaşımı, normal kullanıcı hareketleri ile anormal kullanıcı hareketlerini gözlemleyerek dışarıdan yapılacak saldırılar ile birlikte incelenmesi anlamına gelir. İçeriden oluşacak saldırgan davranışlar iki grupta incelenmektedir. Bunlardan birincisi normalden kullanımdan sapan, sabotaj ihtimali, bilgi çalma ve endüstriyel casusluk olarak nitelendirilen davranıştır. İkinci grup ise hata ve bilgisizlik ile gerçekleşen, zayıf şifre seçimi, iş ile ilgili olmayan web sayfalarını ziyaret etme, gizli bilgilerin yanlışlık ile sosyal ortamlarda paylaşılması ve dikkatsizlik ile tıklanan yemleme maillerini açma gibi davranışları gerçekleştiren kullanıcılardan oluşur [13]. Bu iki farklı davranışın tespit edilmesi kurumun iç kaynaklarının ne amaçla kullanıldığının gözlemlenmesi ve farkındalık için önem arz etmektedir.

1.2 Tezin Amacı ve Kapsamı

Tez çalışmamız, Kippo Honeypot üzerinden toplanan verinin, Hadoop Spark ile işlenerek olası sistem içerisinden oluşacak tehditleri algılaması sağlanacaktır. Bu kapsamda yapılacak deney, Kippo Honeypot üzerinden toplanacak veri ile Linux Shell üzerinde toplanacak veriler bir araya getirilerek saldırgan tahmini yaptırılacaktır. Honeypot kullanmanın avantajı burada elde edilen verinin yeni ve bilinmeyen bir saldırı türüne ait olma ihtimalinin bulunmasıdır. Tüm bu sistemin çalışacağı donanım altyapısı olarak Raspberry Pi üzerinde gerçekleştirilecektir.

Bileşenlerin seçimleri ile ilgili karşılaştırma ve veriler tez içerisinde ayrıntıları ile anlatılacaktır. Tezin hedefi ise akıllı bir algılama ve analiz ortamının hayata geçirilerek tüm kurum altyapılarına bütünleştirecek bir güvenlik aygıtının oluşturulmasıdır.

Bölüm 1, Giriş ve bilgi güvenliği altyapısının dünyadaki genel durumunun incelenmesi, saldırı tipleri ve etkileri değerlendirilecektir.

Bölüm 2, Literatür araştırması, tez kapsamında ortaya konan başlıklar ile ilgili yapılmış diğer çalışmaların araştırılması ve değerlendirilmesi. Büyük veri analitiği ve gelişmiş kalıcı tehditler üzerine yapılan çalışmaların incelenmesi.

Bölüm 3, Kullanılacak teknolojilerin açıklanması ve çalışma mekanizmalarının incelenmesi.

Bölüm 4, Mimarinin tanımlanması ve kullanım senaryolarının belirlenmesi. Deney ortamının kurulması ve sonuçlarının incelenmesi konuları ele alınacaktır.

Bölüm 5, Tezin ürettiği çıktıların değerlendirilmesi, gelecek çalışmalar için tavsiyelerin paylaşılması ve kullanım alanları ile ilgili değerlendirmeler yer almaktadır.



2. LİTERATÜR ARAŞTIRMASI

Bu tez içerisinde sunulan, bilgi güvenliği probleminin çözümünde büyük veri sisteminin kullanılmasıdır. Bu sebeple literatür analizi, büyük veri analitiği ve gelişmiş kalıcı tehdit üzerine yapılmış çalışmaları birlikte ele alan çalışmaları inceleyecektir. Bu iki konuyu da ayrı ayrı ele alan bir çok çalışma bulunmasına karşı bu iki konuyu birlikte inceleyen çalışma sayısı çok azdır. Her iki alanda da aktif olarak bir çok çalışmalar yürütülmekte ve sürekli gelişmektedir. Öncelikli olarak bilgi altyapısına sızılmasının etkileri tartışılacak ve bunun sonuçlarının ne kadar ciddi bir problem yarattığı ortaya konmaya çalışılacaktır. Devamında gelişmiş kalıcı tehdit unsurları irdelenecek ve sektörde üstesinden gelmek için geliştirilen yöntemler ele alınacaktır. Sonra ağ güvenliğinin neden bir büyük veri problemi olarak bakıldığı incelenecektir. Büyük verinin tanımlaması yapılacak, halihazırda kullanılan büyük veri analitiği araçlarından bahsedilecektir. Son olarak da güvenlik araçlarının, büyük veri ortamı ile nasıl ortak kullanılabileceği ile ilgili yaklaşımlar ortaya konulacaktır.

2.1 Araştırma Konuları

Yapılan çalışmalar aşağıda bulunan,

- Gelişmiş kalıcı tehditler,
- Honeypotlar,
- Büyük veri problemi olarak bilgi güvenliği,
- Büyük veri bileşenleri,
- Veri madenciliği ve algoritmalar,
- Güvenlik verisi analizi için kullanılabilecek algoritmalar,
- Büyük veri güvenliği ve analitiği

başlıkları altında incelenmiştir.

2.2 Yapılan Çalışmalar

Gelişmiş kalıcı tehditler (GKT), fiziksel yapıları hedef alan , finansal olarak değerli altyapıları hedef alan saldırılar olarak tanımlanır. Bu saldırıların çalışma şekli genellikle yavaş ve arkaplanda gerçekleşmesi şeklindedir. Burada arkaplandan kasıt, saldırının normal ağ trafiğinden gizlenmesi ve normal trafik içerisinde kaybolmasını sağlaması anlamına gelir. Sistemi ele geçirmek için yavaş fakat uzun süreli saldırılardır. Bu özelliği sebebi ile anlık yıkıcı etki yaratan virüs, trojan ve solucanlardan farklıdır. Bu özellikleri yüzünden çoğu zaman kurumlar bu saldırılardan uzun zaman haberdar bile olmazlar. Genellikle çalınan kullanıcı yetkileri ve sıfır gün açıkları kullanılarak STS tarafından tespit edilemeden sistem içerisinde gizlenirler [1].

GKT'ler karmaşıklığı en yüksek ve ciddi tehditler olarak sınıflandırılmaktadır. Bu saldırıyı gerçekleştirenler genellikle sosyal mühendislik tekniklerini de kullanarak güvenlik duvarını aşarak sistemi istismar ederler. Burada bu saldırıları tespit etmenin önündeki en büyük engel birçok kaynaktan üretilen büyük miktardaki ağ verisinin bulunmasıdır. Bu veriler güvenlik duvarı kayıtları, STS kayıtları, uygulama kayıtları ve VPN kayıtlarıdır. Bu veriler ayrı ayrı birimler tarafından incelendikleri için aralarındaki bağıntıları tespit etmek oldukça güçtür.

Sıfır gün saldırıları bir diğer önemli tehdittir (Şekil 2.1). Bunun sebebi ise STS'nin saldırıyı tespit edebilmeleri için varolan imzalar ile güncellenmeleri gerekir. Halihazırda bilinmeyen saldırı imzaları STS'ler içermedikleri için bu saldırılar karşısında çaresizdirler. Saldırı grupları bu açıkları barındıran yazılımları kullanan kurumları tespit ettiklerinde yama çıkana kadar sistemi kolayca istismar edebilirler. Yakın zamanda ortaya çıkan Heartbleed ve Shellshock gibi açıklar bir çok kurumu etkilemiştir. Heartbleed, Linux kullanan sistemlerde OpenSSL üzerinde oluşan bir hatadan meydana gelmektedir. Çoğu açık kaynaklı yazılım OpenSSL kullandığı için büyük kayıplar oluşmuştur. Shellshock ise Bash, linux tarafından yaklaşık 20 yıldır popüler olarak kullanılan kabukta bulunan bir zafiyettir.

Symantec isimli şirkette çalışan Bilge ve Dumitras, “Zero-Day Attacks in the Real World” isimli çalışmalarında geliştirdikleri teknik ile “Worldwide Intelligence

Network Environment” kullanarak sıfır gün saldırılarını tespit yarayan bir çalışma gerçekleştirmişlerdir [5].

Sıfır gün saldırısının yaşam döngüsünün ilk aşaması yazılımcının farkında olmadan zafiyet içeren programı oluşturup dağıtmasıdır. Eğer işletme dışarısından birisi bu zafiyeti farkedirse istismar etmeye başlar. Eğer yazılımcı veya kurum bu hatayı farkedirse hemen bir yama yayınlamak için çalışmalara başlar. Yama yayınlandıktan sonra uzmanlar sistemi güncelleyerek sorunu halletmiş olurlar. Ardından STS sistemlerinde bu zafiyeti içeren imza kayıt edilerek istismar edecek saldırganlar engellenmiş olur [5].



Şekil 2.1: Sıfır Gün Saldırısı Oluşumu

Honeypotlar güvenlik duvarları ile iç ve dış ağ arasındaki zararlı aktiviteleri tespit etmekte oldukça kullanışlıdır. Bilgi güvenliği uzmanlarının dış kaynaklardan gelecek zararlı aktiviteler hakkında bilgi toplamalarına imkan sağlamaktadır. Akademide bilgi güvenliği araştırmalarında yoğun olarak kullanılmaktadırlar. Honeypotlar tek başlarına bilgi güvenliğini sağlamakta yeterli değildirler ama STS ve SÖS ile uyumlu bir şekilde çalışabilirler [54].

Honeypotların başlıca iki amaç için kullanılırlar. Bunlardan birincisi araştırma, ikincisi ise üretim. Araştırma amacı ile kullanılanlar özellikle akademide saldırı tehditlerini anlamak için kullanılırlar. Üretim honeypotları ise kurumlar tarafından riskleri azaltmak için tercih edilmektedirler [26]. Bu sistemler bir çok şekilde uygulanabilir. Bir web sayfasını modelleyebilecekleri gibi bir linux shell olarak da hizmet edebilirler. “Honeypot Project” isimli uluslararası gönüllülerden oluşan bir

organizasyon, yeni araç ve teknikleri paylaşarak bilgi güvenliğine katkı yapmak için sürekli çalışmaktadırlar [56]. Bir çok çeşitte araç bulunmaktadır. Bunlardan bir tanesi “tcpdump” analizini yapmaya imkan verir. Neredeyse tamamının kullanımı basittir ve belirli bir alan üzerinde veri toplamaya imkan sağlarlar. Bunun sonucunda da yeni taktikler ve araçlar geliştirmeye yardımcı olurlar. Yalnızca honeypot üzerinden toplanan veri ile yeni taktikler geliştirmek kolay değildir.

Honeypotlar, topladıkları ağ verileri ile saldırganların eğilimlerini anlamamıza yardımcı olurlar. Honeypot verileri organizasyon verileri ile bir araya getirilebilir. Diğer organizasyonlar üzerinden elde edilen veriler ile karşılaştırılıp içten gelecek tehditlere karşı bir uyarı sistemi kurmamıza imkan verirler [54]. Örneğin, honeypot üzerinde tespit edilen saldırgan bir IP, diğer ağlar ile paylaşılarak farkındalık yaratılabilir. Araştırma honeypotların büyük çoğunluğu dışarıdan gelecek saldırıları tespit etmek için kullanılmaktadırlar.

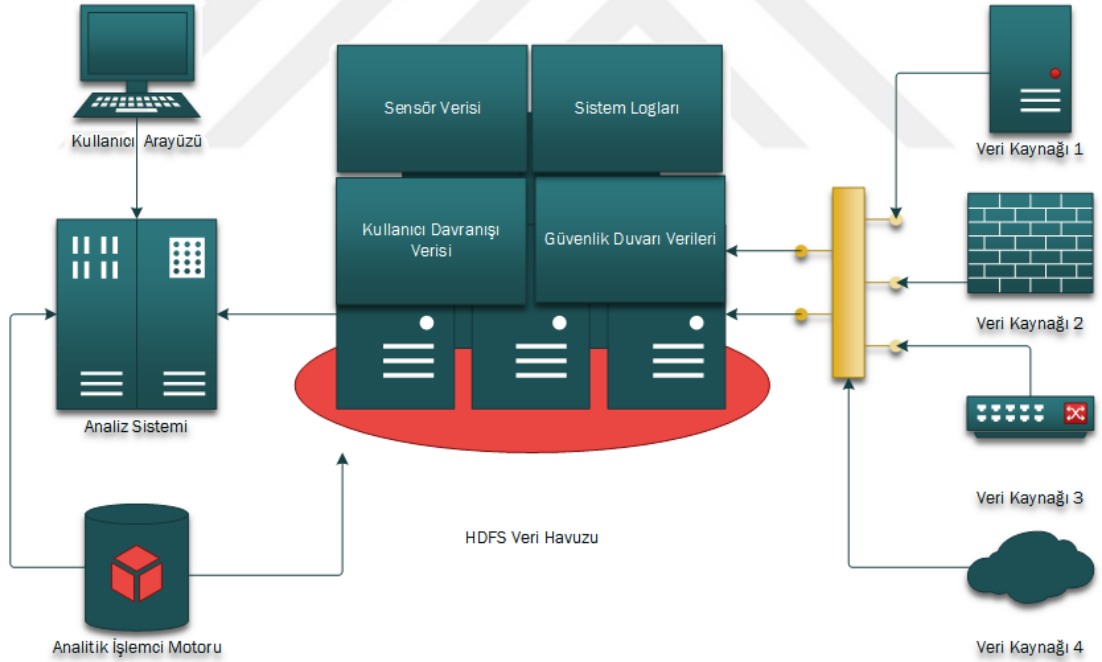
Honeypot uygularken bir çok konu ve üstesinden gelinecek zorluk bulunmaktadır. Özellikle ağ üzerinde ölçeklendirme bunlardan en önemlisidir. Honeypotların dezavantajları, yönetimi ve kullanımı sırasında toplanan verinin kaydedileceği yerin seçimi, verilerin kullanılabilir formatta kaydedilmesi, ağ üzerinde dağılımı, konumlandırılması ve elde edilen verilerin görselleştirilmesi konuları belirli bir bilgi birikimi ile gerçekleştirilmesi gereken işlemlerdir [26]. Honeypotların avantajları, az kaynak tüketimi, belirli bir probleme yönelik ayarlanabilmesi ve yeni çıkan saldırıları tespit etmeye imkan vermesi gösterilebilir.

İçten gelebilecek saldırıların tespit etmek için bilgi güvenliği politikası ve güvenlik teknolojilerinin bir arada kullanılması gerekmektedir. İyi tanımlanmış bir kullanıcı yetki kontrolü ne yazık ki saldırganı caydırmak için yeterli değildir. Bu iç tehdit probleminin çözümü birden fazla tekniğin bir arada kullanılması gerekir. Honeypotu iç ve dış tehditler için kullanmak teknik olarak bu tehdidi azaltmakta başarılı olduğu görülmüştür [55]. Veri analitiği ile birlikteliği sağlandığı takdirde honeypot kullanıcı davranışı verisi yardımı ile saldırı tespit başarısı iyileştirilebilir [13].

Büyük veri tanımı bir çok veri tipini kapsayacak şekilde yapılsa da bizim için ağ trafik bilgisi de bir büyük veri problemi olarak sınıflandırılabilir. Bir çok büyük yazılım üreticisi sundukları çözümler ile büyük veri platformlarını oluşturmaya

çalışmaktadırlar. Hadoop gibi platformlar, yazılım şirketlerinin veri yönetim sistemlerini kökten değiştirmelerine ve depolama yaklaşımlarını yeniden gözden geçirmeye yöneltmiştir [25]. Hadoop, hacmi, artış hızı ve çeşitliliği ne olursa olsun eldeki veriyi anlık olarak işlemeye ve analiz etmeye olanak sağlar [27].

HDFS, Hadoop için geliştirilen dosya yönetim sistemine denir. HDFS ile bulut bilişim depolama çözümlerinin kullanımı ile saklanan büyük miktarda verinin analiz edilmesi için uygun ortam oluşturulabilir [20]. Şekil üzerinde tavsiye edilen veri depolama ve analizi topolojisi görülmektedir. Burada üç büyük bileşen vardır: veri havuzu, kullanıcı arayüzü ve analitik işlem motoru. Kurum içerisindeki bir çok farklı kaynaktan, bir çok tipte veri ile veri havuzu beslenir. Veri, tipleri analiz edilip ayrılarak HDFS içerisinde saklanır. Kullanıcı arayüzü özel bir amaç için yapılacak analitik incelemesine ve veri kaynaklarının gözlemlenmesine imkan verecek şekilde sistemde yer alır. Analiz sistemi gerçek zamanlı veya toplu işlem teknikleri birlikte kullanılarak çeşitli veri kaynakları üzerinde işlem yapmamızı sağlar [35].



Şekil 2.2: Büyük Veri Sistemi

Şekil 2.2’de gösterilen büyük veri sistemi üzerinde veri havuzu tek bir parça olarak ifade edilmişse de aslında birden fazla bileşenden oluşmaktadır. Bunlar, uygulama logları, işletim sistemi logları, güvenlik duvarı logları ve sızma tespit loglarını içerir. Bu veri çeşitliliği, sunulan sistemi klasik veri analiz sistemlerinden ayıran en önemli

unsurdur. Klasik veri analizi sistemleri olay tabanlı çalışarak yalnızca belirli kriterler ile tanımlanmış değerler üzerinden işlem yapar. Tez kapsamında sunulan sistem ise bir çok kaynaktan elde edilen verilerin Hadoop ekosisteminde işlenip, büyük veri mantığı içerisinde değerlendirilerek belirli uyarılar oluşturması amaçlanmaktadır.

Kurumlar giderek daha fazla bilgi güvenliği verisi depolamaya başlamak durumunda kalmışlardır. Güvenlik kayıtlarının ve olay verilerinin toplanmasının ardındaki en büyük sebepler, şüpheli davranışları tespit ve takip etmek, adli analiz yapılarını desteklemek, olayları önlemek ve yasal gereksinimlere uyum sağlamaktır [33]. Geleneksel güvenlik çözümü yaklaşımları büyük veri altyapısına entegre olmaya çalışsalar da henüz bu sistemler tek başlarına bunu yapmaya yeterli değildirler. Bu birimlerin çoğu keşif amacı ile analiz yapımında yer almaya çalışırlar. Bu sistemleri kuracak kişiler bilgi güvenliği ve veri analizi konularında tecrübeye sahip olmak durumundadırlar. Ancak bu özellikler ile sistemi yapılandırmaya yetecek yeterliliği sağlarlar. Tüm bu çabalara rağmen geleneksel sistemler yapıları gereği bir çok güvenlik olayını farketme ihtimalleri oldukça düşüktür.

Artan işlemci gücü sayesinde kurumlar daha fazla veri toplamaya başlamışlardır. Bunun sonucu olarak da sunucular üzerinde yüksek miktarda veri birikimi meydana gelmiştir. Eskiden arşivlenme ihtiyacı duyulmayan veriler dahil saklanmaya başlanmıştır. Bunların dışında yeni alanların bu verinin artmasında önemli rol oynamaktadır. Örnek olarak, sosyal medya verileri, gömülü sistemler üzerinden gelen veriler ve akıllı telefonlar bu veri kaynaklarını beslemeye başlamışlardır. Tüm bunların sonucu olarak kaçınılmaz bir şekilde yeni bir yaklaşıma ihtiyaç duyulmasına neden olmuştur. Büyük veri çağı böylelikle hayat bulmuştur [30].

Büyük veri bir çok insanın dilinden düşmeyen moda bir sözcük haline gelmesine rağmen çoğu insan tam olarak anlamını kavrayabilmiş değildir. Bu yeni kavram, geleneksel ilişkisel veritabanı yönetim sistemi (RDBMS) yaklaşımı ile yönetilemeyecek kadar dev veri yığınlarını ifade eder. Elbette ki büyük veriyi yalnızca verinin hacmi ile değerlendirmek yanlış olur. Veri tipleri ve çeşitliliğin fazla olması da bir diğer önemli büyük veri özelliğidir. Her ne kadar teknoloji gelişip işlemci güçleri artsa da bu boyuttaki bir veriyi analiz etmek için tek bir bilgisayarın yetersiz kalacağı aşikardır. Bu sebeple bir kaç bilgisayara dağıtılacak, analiz edilip toplanacak yeni bir yaklaşıma ihtiyaç duyulmuştur.

2010 yılında McKinsey bir rapor hazırlamış ve bir çok teknolojinin ve akımın büyük veri ile olan ilişkisini incelemiştir. Bu akımlardan bazıları, nesnelerin interneti kavramı ve bulut bilişimin hayatımıza soktuğu servis olarak hizmetlerin sunulması oldukça etkili olmaktadır [7]. İşletmeler bu dış kaynaklar ile etkileşimlerini arttırdıkları sürece büyük veri altyapısını beslemeye devam edecektir.

Büyük veri bu tez içerisinde işlenirken sözü geçen diğer tüm teknolojiler de bu kavramı besleyen önemli unsurlar olarak üzerlerinde durulacaktır. Google ve Yahoo gibi internet şirketleri büyük veri yaklaşımının arkasındaki güçler olarak önemli rol oynamaktadırlar. Yalnızca internet tabanlı şirketler değil aynı zamanda fiziksel ürün üreten kurumlar da büyük veri teknolojisinden fayda sağlamaktadırlar [7]. İşletmeler sosyal medya, kendi iç ağları, nesnelerin interneti gibi veri kaynakları üzerinde yapacakları çalışmalar ile bir çok sektörde rekabet avantajı yakalayabilirler.

Tüm bu değerlendirmeler ve çalışmalar yapılmasına rağmen büyük veri kavramı daha başlangıç safhasındadır. Dijital veri üretiminin 2009 ve 2020 yılları arasında 44 kat artarak her yıl 35 zetabyte veri üretildiği araştırmalar sonucunda ortaya konmuştur [37]. Büyük verinin işlenmesi bugün tek bir teknoloji, platform, mimari ve kullanım etrafında düşünülemez. Google, Facebook ve Twitter gibi şirketler büyük veri kavramına öncülük edip kullansalar da bunların dışında kalan örneğin sağlık ve üretim sektörü gibi, rekabet avantajı için bu teknolojileri yapılarında aktif bir şekilde kullanmaktadırlar [39].

Bulut bilişim çözümü sunan şirketler de büyük veri avantajını kullanarak müşterilerine özel ortamlar yaratarak bu teknolojilerden yararlanmalarına olanak sağlarlar. Örneğin VMWare, ürünlerini büyük veri platformları için uygun hale getirmek için bir çok çalışma yapmaktadır. Performans testleri yaparak müşterilerin kendi donanım altyapıları yerine bulut tabanlı, büyük veri desteği sağlayan altyapılarını kullanımlarına sunmaya çalışmaktadır [6].

Büyük verinin üç adet niteliği bulunmaktadır. Genellikle 3V olarak bilinen bu kavramlar: Hacim (Volume), Çeşitlilik (Variety) ve Hız (Velocity) olarak karşımıza çıkar [16]. Gartner bu üç kavramın büyük veri analizinde önemini raporunda bize sunmaktadır. Buna rağmen 3V örneği büyük veriyi sınıflandırmak için tek kıstas değildir. Veri uzmanlarının farklı nitelik tanımlamaları da mevcuttur. Bu tez

kapsamında ele alınan konu itibari ile bizim vardığımız kanı, bilgi güvenliği verisinin kesinlikle bu üç özelliği de taşıdığı yönündedir [1]. Büyük veriyi anlamaya çalışırken 3V kavramı tüm analiz aşamalarında karşımıza çıkacaktır.

Hacim kavramı, bu üç kavram arasında anlaşılması en kolay olandır. Geleneksel veri setleri megabyteler ve gigabyteler ile ifade edilir. Bu miktar ilişkisel veritabanı yönetim sistemleri ile saklanıp analiz edilebilir. Bu hacmi arttıran verinin kaynağı ise internet teknolojileridir. Bunun yanında kurumlar artık sistemde üretilen logları bir süre sonra silmek yerine kalıcı olarak saklamaktadırlar. Veri analitiği platformlarından önce analiz birimlerinin bu verileri işlemeleri ciddi bir mesele idi. Booz & Company şirketinin yapmış olduğu araştırma çalışmasına göre, düzenli veri miktarı her geçen gün artmaktadır. Bunun yanında düzensiz veri miktarı ise dünyadaki üretilen veri miktarının %80'ini oluşturmaktadır [28].

Hız kavramı, veri üretim hızının ne kadar olduğunu bununla birlikte talepleri karşılamak için yapılan analiz hızının ne kadar olması gerektiği ile ilgilidir. Veri üretim hızı arttığında, bu veriyi analiz edip işleyecek hızın da yeni teknoloji ve algoritmalar ile artırılması ihtiyacını beraberinde getirir. Günümüzdeki sistemler veriyi yığın olarak işlemektedir. İnternet teknolojilerini kullanan son akımlar üzerinden üretilen verinin, üretilme aşamasında işlenerek gruplandırılmasına gerek duymaktadır.

Son ve en karışık nitelik olan “Çeşitlilik” kavramı ise kaydedilen verinin türleri ve grupları olarak açıklanabilir. Veri çeşitleri üç sınıf altında irdelenebilir. Bunlardan birincisi, düzenli veridir. Bu sınıftaki veriler beklenen formattadır ve ilişkisel veri tabanı yönetim sistemleri üzerinde normalleştirme teknikleri ile gerekli sayıya indirgenebilir. Uzmanlar bu veri üzerinde işlem yaparken standart sorgulama dili, SQL ile istedikleri işlemi gerçekleştirebilirler [16].

Çeşitlilik kavramı içerisindeki ikinci sınıf ise yarı düzenli verilerdir. Bu veriler XML, JSON ve EDI gibi dosyalardır. Yarı düzenli verilerden bilgiler kolaylıkla çıkartılabilir fakat çoğu durumda bir ön işleme tabi tutulmaları gerekmektedir. Bu işlemler ticari ve açık kaynaklı ayrıştırıcılar (Parser) kullanılarak kolayca düzenli veri haline getirilebilirler.

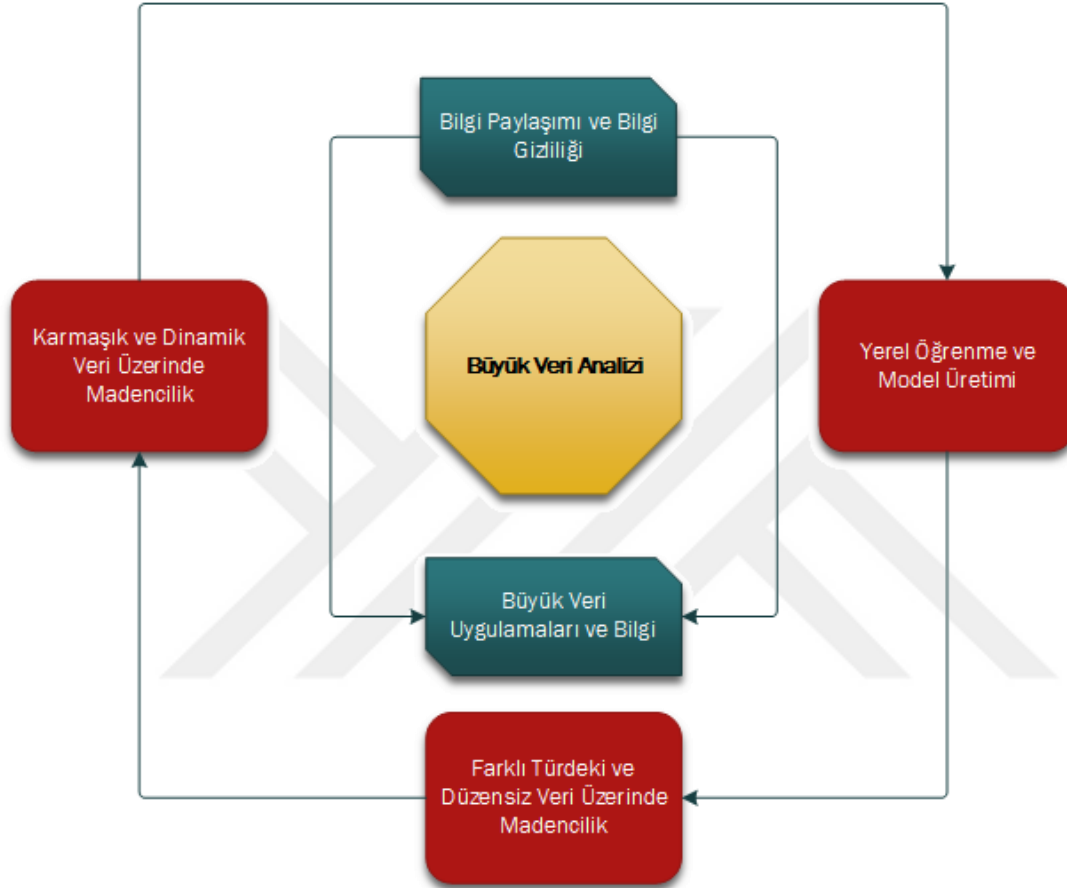
Son çeşitlilik sınıfı ise analizi en zor olan düzensiz verilerdir. Şirketler büyük veri analitiği platformlarının tüm gücünü kullanarak bu verileri derlemeye çalışırlar. Düzensiz verilere örnek vermek gerekirse bunlar: metin dosyaları, loglar, sosyal medya paylaşımları, tablo dosyaları, dökümanlar ve fotoğraflar olarak tanımlanabilir. Teknolojideki son gelişmeler yaşanana kadar bu veriler çoğunlukla gözardı edilirdiler. Kurumlar rekabet avantajı sağlamak adına bu düzensiz verileri de kazanarak sisteme dahil etmenin yollarını geliştirmeye çalışmaktadırlar [28]. Bu düzensiz verileri düzenli ve yarı düzenli veriler ile ilişkilendirerek aralarındaki bağıntılardan yararlanmaya çalışırlar. Bu sayede bu verileri de veri ambarlarına kazandırarak bilgi hazinelerine değer katmayı öngörmektedirler [11].

Gartner raporundaki “3V” yaklaşımı büyük veriyi tanımlamadaki tek yaklaşım değildir. Bir diğer model ise Wu’nun ortaya çıkarttığı “HACE” teorisidir. Burada farklı yapıda (Heterogen), otonom (Autonomous), karmaşık (Complex) ve gelişen (Evolving) terimleri ile açıklanmaya çalışmaktadır [43]. Wu’ya göre, bu nitelikler büyük veriden kullanılabilir veri elde etme sırasında nelerin seçileceğinin kararlaştırılmasına yardımcı olur. Ele alınan her iki modelde de birbirleri ile kesişen noktalar bulunmaktadır. İlerleyen bölümde bu kesişim noktaları ayrıntılı olarak açıklanacaktır. Şekil 2.3 üzerinde çalışma süreçleri belirtilmiştir.

Farklı yapıdaki veriler, 3V yaklaşımındaki çeşitlilik ile aynı anlama gelir. Bir çok tipte veri bir araya gelir çünkü her bilgi sisteminin kendi şeması ve protokolü bulunmaktadır. Bu çeşitlilik, verilerden mantıklı bir sonuç çıkarılması önündeki en büyük zorluktur. Bir çok değişik boyuttan elde edilen veriden anlamlı bir veri çıktısı üretmek oldukça karmaşık bir iştir.

Modern bilgi sistemlerinde üretilen veri dağıtılmış bir çok ortamda üretilir. İnternet kurumların yalnızca kendi altyapılarından bilgi üretilemeyeceğini kanıtlamaktadır. Her kaynak ürettiği ve toplayacağı bilgiye bir yönetim merkezi olmaksızın karar verebilir yapıda olmalıdır. Bu otonom kaynakların dağıtılmış yapı içerisinde üstesinden gelmesi gereken bir çok zorluk bulunmaktadır. Ölçeklenebilirlik ve yönetilebilirlik vasıflarını kaybetmeden bu üretim ve toplama işini halledecek yapıda olmaları gerekmektedir. Veriler arasında bağıntı veya bir ilişki bulmaya çalışmak veri analitiği sistemlerindeki en büyük zorluklardan birisidir.

HACE teorisi ile 3V yaklaşımının örtüştüğü noktalar vardır. Bunlar büyük verinin niteliği analiz edilirken oldukça faydalıdır. Ortak üç adet nokta üzerinden çalışma yapılacak olursa bunlar: veri erişimi ve işlenmesi, veri gizliliği ve etki alanı bilgisi ve büyük veri algoritmalarıdır [43]. Bu başlıklar veri madenciliği ve algoritmalar bölümünde ayrıntıları ile anlatılacaktır.



Şekil 2.3: Büyük Verinin İşlenmesi

İş zekası ve Analitiği ilk kullanımından bu yana üç aşama kaydetmiş ve gelişimini halihazırda sürdürmektedir. 2012 Aralık ayında, Chen, Chiang and Storey yayınladıkları “Business Intelligence and Analytics: From Big Data to Big Impact” isimli makaleleri ile iş zekasındaki evrimi ve kurumsal analitiği büyük veri kapsamında değerlendirmişlerdir [11]. İş zekası ve veri analitiği 1950 yılından bu yana üzerlerine çalışılan konulardır ancak iş zekası kavramı, 90’lı yılların sonuna doğru popülerleşmiştir. Büyük veri analitiği son yıllarda iş zekası alanında, kurumlar için önemli kararların alınmasına altyapı oluşturacak bilgileri sağlamak için kullanılmaktadır.

İş zekası ve analitiği 1.0 versiyonu, en az veritabanı üzerinde raporlama konusu kadar eskidir. Bu versiyondaki verilerin neredeyse tamamı düzenli verilerden oluşmaktadır. Veri genellikle dosya sistemi üzerinde veya ilişkisel veritabanları içerisinde bulunmaktadır. Bu versiyon içerisinde kullanılan istatistiksel analiz teknikleri 1970'li yılların başlarında kullanılmaya başlanmış olup, 1990'lı yıllarda yoğun bir şekilde kullanılmaktaydı. Veri ambarları, tüm veriyi bir arada tutup analiz etmek için bu versiyon içerisinde kullanılmıştır. ETL ile verilerin belirli formatlara dönüşümü ve entegrasyonu sağlanmıştır.

İş zekası ve analitiği 2.0 versiyonu ile, 1998 ve 2003 yılları arasında web teknolojilerindeki gelişmenin etkisi ile kurumların büyük miktarda verinin üretmesini sağlamıştır. İnternet teknolojileri verinin yaratılması, saklanması ve analiz edilmesi için eşsiz bir yol sunmuştur. Amazon ve eBay gibi şirketler bu yeni yöntemleri sayesinde veriyi geleneksel yolların dışında saklamaya ve analiz ederek sektörde avantaj elde etmişlerdir. Web üzerinde kullanıcıların ürettiği düzensiz veri şeklindeki içerikler veri analitiği yapanlar için hem bir zorluk aynı zamanda da bir fırsattır. İş zekası ve analitiği 1.0 ve 2.0 versiyonları arasındaki en büyük fark, düzensiz veri miktarının ilişkisel veritabanları üzerinde tutulamayacak kadar büyük miktarda olmasıdır. Web 2.0 uygulamalarında üretilen veriler, sosyal medya verileri, forumlarda oluşturulan içerikler, web bloglarında üretilen içerikler, görselleri videolar ve ses dosyaları olarak karşımıza çıkmaktadır. Bu verilerin işlenebilmesi sonucunda arama ve istenilen sorguların yapılmasına imkan sağlanmış olur.

İş zekası ve analitiği 3.0'ın kullanıldığı sırada sektörde ciddi değişimler meydana gelmiştir. 2011 yılı sonrası akıllı telefon ve tablet satışlarının, dizüstü ve masaüstü cihazları geçmesi ile mobil dünya kaçınılmaz bir şekilde yepyeni bir yola girmiş oldu. Sürekli internete bağlı olan insanların ürettiği verilerin boyutu çok daha yüksek boyutlara yükselmiştir. Bu noktada her yerde yazılımın hakimiyetin arttığı görülmektedir. Bir çok alanda uçtan uca paket yazılımların etkisi, mobil oyunlardan kurumsal uygulamalara kadar her yerde hissedilmiştir. Radyo frekansı üzerinden tanımlama (RFID) teknolojisi ile nesnelerin interneti kavramı hayatımıza kazandırılmıştır. Giyilebilir teknolojiler ve bu nesnelerden toplanan veriler ile kişilerin yaptıkları spor bilgilerine kadar ulaşmak mümkün hale gelmiştir. Konum bazlı veri toplanması ve sensörler sayesinde mobil analitik veri miktarları oldukça

artmıştır. Bir çok kaynaktan akan bu verileri bir araya getirmek ve görselleştirmek analiz sonucu mümkün olmaktadır [11].

Chen'e göre, öne çıkan araştırma alanları, büyük veri analitiği, metin analitiği, web analitiği, ağ analitiği ve mobil analitiği alanlarıdır. Yazılım tabanlı ağ ve en son ağ teknolojileri bir bütün olarak büyük veri analizini kapsayacak şekilde yapılmalıdır [32]. Günümüzde yapılan en son araştırma alanlarının büyük bir bölümü internet teknolojileri tabanlıdır. Bu alanlar yalnızca üniversitelerde deneysel amaçlı araştırma için değil aynı zamanda kuruluşlar ve devlet tarafından da ticarileştirilmiştir [12].



Şekil 2.4: İş Zekası ve Analitiği

Şekil 2.4 temel teknolojiler ve araştırma alanları ile ilgili bilgileri göstermektedir. Chen, veri analitiği ile ilgili güncel çalışma alanlarına ve fırsatlara dikkat çekmiştir. Araştırma alanlarını beş ayrı önemli teknik alanı işaret etmiştir.

Büyük veri analitiği için MapReduce işlemlerini uygulamak için öncesinde bir takım istatistiksel işlemler yapılması gerekebilir. Veri setleri üzerinde yapılacak işlemler için kullanılabilecek yaklaşımlar ve algoritmalar sınıflandırma, kümeleme, regresyon ve birliktelik analizi gibi teknikler kullanılabilir [23]. Metin analitiği, istatistiksel doğal dil işleme tabanlı yaklaşım ile kullanıcılar tarafından üretilen verilerde kapsam dışı verileri önemli ölçüde azaltır.

Web analitiği, HTTP üzerinde, HTML tabanlı veriler, arama logları, web servisleri ve API'lar üzerinde gerçekleştirilmektedir. Ağ analistleri çeşitli sosyal ağ yaklaşımları ve topolojilerini betimlemeye çalışmaktadırlar [36]. Son olarak mobil analitiği, nesnelerin interneti üzerinden oluşturulan verilerin analizi üzerine

yoğunlaşır. Her bir alan iş zekası ile sınırlı kalmayıp bir çok sistem üzerine veri üretmek için uygulanabilir.

Bilgi güvenliğini sağlamaya çalışan uzmanların siber tehditlere karşı mücadele ederken kullanabilecekleri bir çok analiz tekniği bulunmaktadır. STS'ler üzerinde en çok tercih edilen yöntemlerden bir tanesi makine öğrenmesi yaklaşımıdır. Bilgi güvenliği topluluğu üyeleri, düzenli olarak yeni makine öğrenmesi yaklaşımları geliştirmeye ve bunu diğerleri ile paylaşmaya çalışmaktadırlar.

STS'ler çeşitli eğitimci ve eğitimci olmayan öğrenme tekniklerini kullanabilirler. İmza tabanlı tespit yaklaşımı ise genelde en çok kullanılan yöntemlerden bir tanesidir. Olağandışı davranışları inceleyen STS'ler bunları engellemek için makine öğrenmesi ile bir karar verme mekanizması sağlayabilir [53].

Bir çok makine öğrenmesi tekniği bulunmaktadır. Bunlar: yapay sinir ağları, karar ağaçları, genetik algoritması, bulanık mantık ve bayes ağları olarak verilebilir [24]. Karar ağaçları, önceden tanımlı imzalar üzerinden sınıflandırma yaparak çalışır fakat yeni saldırı türlerini tespit etmekte oldukça etkisizdir. Genetik algoritması, STS üzerinde önceden tanımlanan imzalar üzerinden yaklaşık bir tespit bildirimi çıkarmak için kullanılır [53]. Bulanık mantık, önceden tanımlanmış kuralları baz alarak tespit ettiği tehditi sınıflandırmaya çalışır fakat büyük veri setleri üzerinde sorunlar yaşayabilmektedir [1]. Bayes ağları modeli, önceki bilgileri kullanarak olası tehdit durumlarını önceden haber vermeye çalışır. Buna rağmen yeni saldırılar karşısında zayıf yönleri bulunmaktadır. Bayes en iyi olmamakla birlikte diğer analitik öngörü araçları ile uyumlu bir şekilde çalışmaktadır [44].

Makine öğrenmesi tekniği önceki veriler üzerinden yaptığı yüksek başarılı çıkarımlar ile saldırı tespiti için dikkat çeken bir yöntemdir [10]. Endüstri içerisinde, siber güvenlik üzerine odaklanan topluluklar yeni makine öğrenmesi yaklaşımları ile yeni saldırıların tespiti için önemli gelişmeler kaydetmiştir. Bu teknikler için en büyük problem öğrenme tabanlı bir yaklaşım içerisinde olmasıdır. Yukarıda bahsedilen algoritmaların tümü bilinen saldırı vektörleri üzerinden işlem yaparlar. Saldırganlar bunun farkında olup saldırı yöntemlerini karartarak bunu atlatmayı başarabilirler.

En çok bilinen makine öğrenmesi tekniği davranışsal analizdir. Temel çözüm yaklaşımları, eğitici öğrenme ve sınıflandırma algoritmasıdır. İki temel yöntem, Naive Bayes ve Gizli Markov Modelidir [57]. UNIX kabuk komutları içerisinde maskelenen saldırıları tespit eden çok bilinen bir veri seti vardır. Matthias Schonlau tarafından oluşturulmuş, 50'den fazla kullanıcıyı içeren maskelenmiş saldırıları tespit etmekte kullanacağımız bir veri seti bulunmaktadır [58]. Bu veri seti, bilgi güvenliği topluluğu tarafından geniş bir kullanıma sahiptir. Tek eksik yanı komutların parametrelerini barındırmamasıdır [59]. Maxion'a göre, saldırganlar tarafından olağandışı bayrakların kullanımı tespit edilme olasılıklarını %18'den, %70 seviyesine çıkartmaktadır.

Halihazırda kullanılan araçlar ihlali önlemek üzerine odaklanmışlardır. Pratikte bu güzel bir sonuçtur fakat bu davetsiz misafirleri tamamı ile durdurmaz. Piyasadaki araçlar genellikle güvenlik duvarı, STS ve gatewaylerin güvenliği üzerine odaklanmaktadır. Sıfır gün saldırılarını içeren zararlılar için bir çözüm sunan uygulama neredeyse yoktur. Bu şuan bulunan uygulamaların ağı korumadıkları anlamına gelmez. Sadece bilinen ve yaygın saldırılara karşı koruma sağlamaktadır. Bununla birlikte bu araçlar yüksek oranda yanlış alarmı sebep olmaktadır. Bu hataları en aza indirmek için vir çok işlemin el ile yapılması gerekmektedir.

Büyük veri güvenliği analitiği yöntemi, güvenlik olaylarına bakış açımızı değiştirmemizi sağlayacaktır. Büyük veri analitiği ile bir çok farklı kaynaktan toplanacak veri sayesinde otomatikleştirilmiş bir savunma düşüncesi hedeflenmektedir. Bu yöntem bilgi güvenliğinin izlenmesi ve kayıt altına alınması görevini tek bir merkezde toplar. Kurumlar bu yöntem ile dinamik ve kendi kendine öğrenen, özel saldırıları tespit edebilme gücüne sahip, halihazırda kullanılan sistemlerden daha hızlı aksiyon alan bir altyapıya sahip olabilirler [38]. Önerilen sistemin hedefi, gelişmiş kalıcı saldırılar için geçmişte gerçekleştirilen bilgileri kullanarak saldırının amacını anlamaya çalışarak kaynağını tespit etmektir.

Büyük veri analitiğinin avantajı, kaydedilen tüm sunucu loglarını, ağ üzerindeki trafiği, IP adreslerini, zamanlarını ve oturum sürelerini kullanmasıdır. Alan kontrolü sağlayan birimler, belirli kullanıcılara, belirli IP adresleri üzerinden, belirli zaman ve sürelerde işlem yapmalarını sağlar. İçerik sunucuları da kullanıcıların aktivitelerinin detaylarını ve giriş yetkisi bulunan birimleri sistemimize sağlayabilir.

Kurumlar biriktirdikleri kullanıcı kayıtlarını belirledikleri süre sonunda silerler. Bu veriler silinmeden tek bir depo üzerinde saklanması sağlanabilir. İleri veri analitiği bir çok güvenlik alanında uygulanabilir. Bunlardan bir kaçı: Sistemin saldırı altında olup olmadığı, saldırının amacının ne olduğu ve saldırının nereden geldiği gibi konular tespit edilebilir. Tüm bunları tespit edecek tek bir algoritma ne yazık ki bulunmamaktadır. Gelişmiş kalıcı tehditlere yol açan en büyük sorunlardan bir tanesi, güvenlik analistinin kendisini bir çok sağlayıcıdan kullanılan yazılımlar sebebi ile savunmasız hissetmesidir [39]. Ağ ve uygulama verilerinin tümünün analitik için kullanılması ile güvenlik analistleri, tehditlere karşı daha hazır, önceden haber alabileceği ve potansiyel tehditlere karşı hızla aksiyon alabileceği bir yapıya kavuşmasıdır.

Kurumun büyük veri analitiğini kullanarak geliştireceği bir çok güvenlik yaklaşımı bulunmaktadır. Bu kullanım senaryolarına örnek vermek gerekirse: dışarıdan gelecek kötü amaçlı yazılım tabanlı tehditlerin bertaraf edilmesi, gelişmiş kalıcı tehditlerin önüne geçilmesi, kasıtsız içeriden oluşacak tehditlerin önlenmesi, risk yönetimi, veri üzerinde gerçek zamanlı raporlama, geçmişte yaşanan saldırılara ait raporlama, korelasyon kuralları/modelleri, dolandırıcılık tespitlerinin yapılması [34].

Büyük veri güvenliğinin uygulanması yeni bir yaklaşımdır ve bu alanların her biri üzerinde bir çok çalışma gerçekleştirilebilir. Organizasyonun ihtiyaçlarına göre bu alanlardan belirli bir tanesi üzerine yoğunlaşıp çalışmalar bu kapsamda gerçekleştirilebilir.



3. KULLANILAN TEKNOLOJİLER

3.1 Honeypot Tuzak Sistemi

Kaynaklarımız iri, şişman ve hareket edemeyen birer hedef gibidirler. Saldırganlar bu sistemlerimize istedikleri zaman ve istedikleri şekilde saldırabiliyorlar. Üstelik saldırırganlar girişimci ve her geçen gün yeni teknikler bularak saldırı yöntemlerini geliştirmektedirler. Honeypotlar, bilgi sistemlerinin kullanımında yetkisiz veya kötü amaca sahip erişimleri tespit etmek, zararlarından kaçınmak ya da önlemek amacıyla kurulan bir çeşit tuzaktır. Genelde, bir ağın parçasıymış gibi görünen bilgisayar veya veri barındıran herhangi bir sunucu makine olabilir. Aslında saldırırganlara göre, saldırırmak için sebep olabilecek bilgi veya değer taşıyan bir hedef gibi duran, izole edilmiş ve hareketleri özellikle izlenen bir kaynaktır. Bilgi güvenliği yönetimi için ilk adımı sizin atmanızı sağlayarak saldırırgan için işleri tersine çevirmenize olanak sağlar. Güvenlik duvarları ve Saldırı Tespit Sistemleri (STS) hali hazırda kullanılan en yaygın koruma ve savunma yöntemleridir. Honeypotları bu iki sisteme alternatif olarak düşünmek yerine, bilgi güvenliği altyapısını destekleyecek ek bir katman olarak değerlendirmek yerinde olacaktır. 1990 yılından bu yana kullanılmaktadır.

3.1.1 Honeypotların çalışma yapısı

Ağ taraması ve benzeri yöntemler ile ağa sızmaya çalışan saldırırganlar, dikkatlerini çekecek kadar bilgi barındıran bu sistemlere, bilerek bırakılan açıklar sayesinde erişim sağlarlar. Honeypot üzerinde yaptıkları işlemler ve etkileşimler kayıt altına alınarak saldırırganın davranışı hakkında bilgi edinmemize yardımcı olurlar. Burada dikkat edilmesi gereken nokta saldırırganın erişmeye çalıştığı sistemin honeypot olduğunu anlayacak kadar kolay kullanıcı adı ve şifrelerin belirlenmemesi ve sahte bilgisayar olduğu anlaşılamayacak kadar veri ve servisleri barındırmasıdır.

3.1.2 Honeypot yapılandırma çeşitleri

Honeypotlar iki grup altında toplamak mümkündür. Birincisi etkileşim seviyelerine göre, ikinci olarak da uygulama türlerine gruplandırılabilirler.

Etkileşim seviyelerine göre: Yüksek etkileşim ve Düşük etkileşim olmak üzere ikiye ayırmak mümkündür. Düşük etkileşimli honeypotlar, sistemin bazı özellikleri ve servislerinin taklit edilmesi ile oluşturulur. Hazırlanmaları oldukça kolaydır. Tespit edilme riskleri azdır. Saldırgan hakkında edinilecek bilgi sınırlıdır. Yüksek etkileşimli honeypotlar, sistemin tüm özelliklerini taklit ederek gerçek bir sistem gibi davranırlar. Saldırgan tarafından tespit edilme olasılıkları yüksektir bu yüzden risk fazladır. Saldırgan hakkında elde edilecek bilgi daha kapsamlı olacaktır.

Honeypotları, uygulama türlerine göre Fiziksel ve Sanal olmak üzere ikiye ayırmak mümkündür. Fiziksel honeypotlar, gerçek sunucular üzerinde yapılandırılırlar. Gerçek IP ve MAC adresleri kullanırlar. Gerçek birer bilgisayar oldukları için etkileşimleri yüksektir. Bu sebeple de bakım ve kullanım maliyetleri fazladır. Sanal honeypotlar, gerçek bilgisayarlar üzerinde birden fazla sanal makineler olarak hizmet vermektedirler. Gerçek adreslere sahip olmadıklarından tespit edilmeleri kolaydır. Bu sebeple etkileşimleri düşüktür. Bakımları kolaydır ve birden fazla kolaylıkla oluşturulabilirler. Maliyetleri de fiziksel yapılandırılanlara göre düşüktür.

3.1.3 Honeypotların kullanım avantajları ve dezavantajları

Honeypotların avantajları, bilgi güvenliği konularında araştırma yapmak için güzel bir başlangıç noktasıdır. Kurması, ayarlanması, uygulanması ve bakımı oldukça kolaydır. Maliyetleri düşüktür. Esnek bir yapıdadır ve ağ üzerinde istenen noktaya konumlandırılabilir. Riski çok azdır. Honeypotun saldırı tarafından ele geçirilmesi üzerinde gerçek veriler barındırmadığı için sorun teşkil etmemektedir.

Dezavantajları ise saldırı ile etkileşimin zor olmasıdır. Burada bahsedilen zorluk saldırının ağ üzerinde tarama yaparken honeypotu bulması ve onunla etkileşime girmesi ihtimalidir. Elde edilen logların analizlerinin iyi yapılması gerekir. İleri düzeydeki saldırılar tarafından tespit edilme ihtimalleri yüksektir bu sebeple yapılandırmalarının bu durumlar göz önünde bulundurularak yapılması gerekmektedir.

3.1.4 Kippo honeypot

Kippo SSH honeypot, python dili ile geliştirilmiş açık kaynaklı bir honeypottur. Kaba kuvvet saldırılarını ve kabuk etkileşimlerini kayıt altına alır. Debian tabanlı

linux dosya sistemini başarılı bir şekilde taklit eder. Oturum kayıtları UML ile uyumlu formatta kaydedildiği için yönetimi kolaydır. Saldırgan tarafından indirilen dosyaları saklayarak incelememize imkân sağlar. Raspbian işletim sistemi ile tam uyumludur. Tez kapsamında Raspberry Pi üzerinde çalıştırılacağı için bizim için tercih edilmesinde büyük etkene sahiptir. Dokümantasyonu ve yardım alma konusunda açık kaynak yazılım topluluğu tarafından iyi bir şekilde desteklenmektedir.

3.2 Hadoop Anaçatısı

Apache Hadoop, Doug Cutting tarafından oluşturulmuştur. Aslında açık kaynak kodlu bir web gezgini olan Apache Nutch denilen bir proje üzerine çalışmaktaydı. Zorluğu büyük miktarda veriyi depolayabilmesi ve bunları işleme kapasitesine sahip olmasını mümkün kılmaktı. 2003 - Google File System and 2004 – MapReduce raporlarını okuduktan sonra, Cutting bir veri işleme ve depolama çerçevesi olarak ele alınabilecek kombine sistem içine iki sistem birleştirmek için harekete geçti. Hadoop Common olarak bilinen Hadoop'un çekirdeği HDFS, Hadoop Distributed File System, ve MapReduce çerçevesine sahiptir [41].

3.2.1 Büyük veri kavramı ve meydana getiren unsurlar

Apache Hadoop dört ana mekanizma üzerinde kurgulanmıştır. Bunlar ölçeklenebilir olması, hesaplı bir çözüm, esneklik ve hatadan kurtarma altyapısının bulunmasıdır. Ölçeklenebilirlik, ihtiyaç duyulduğunda, verinin kendisini, formatını, yerini değiştirmeden yeni düğüm noktası eklenebilir olmasıdır. Hesaplı çözüm, Yüksek hacimli verinin, fazla CPU gücü ile işlenmesini gerektiren paralel çözüm ihtiyaçlarını, daha ucuz veya hesaplı bilgisayar altyapısı ile gerçekleştirilmesini sağlamasıdır. Hadoop'un şemasız, yapısal olan veya olmayan bütün veri tiplerini, her kaynaktan işleyebilen bir mimarisi vardır. Esneklik, farklı kaynaklardan gelen, farklı veri tipleri birbirleriyle birleştirmesi, özetlemesi ve işleyebilmesidir. Hatadan kurtarma yaklaşımı ise düğüm noktalarından biri ulaşılamaz olduğunda, sistem, gelen yükü diğer düğüm noktalarına paylaştırarak kesintisiz hizmete devam etmesidir.

3.2.1.1 Mobil cihazların gelişimi ve etkisi

Mobil cihazların insanların hayatında daha fazla yer alması ile uygulamalar üzerinden üretilen düzensiz ve çeşitli veriler (Örneğin: Lokasyon, video, fotoğraf, döküman vb.) büyük veri kavramını meydana getiren unsurların en başında gelmektedir.

3.2.1.2 Bulut bilişimin etkileri

Bulut bilişim teknolojileri sayesinde bir çok kişi ve kurum farklı noktalardan kullandıkları servisler ile hayatlarına devam etmektedirler. Bu durum, oluşturulan verinin kaynaklarının sayısının artmasına ve verinin uçtan uca kullanımı için bir çok noktada bulunmasına sebep olmaktadır. Veri erişimindeki bu artış hiç kuşkusuz büyük veri kavramının hacminin artışında etkisi oldukça büyüktür.

3.2.1.3 Sosyal medyanın etkisi

Sosyal medya kavramının hayatımıza girmesi ile insanlar gündelik hayatlarında yaşadıkları ve etkileşimde bulundukları tüm gelişmeleri anlık olarak paylaşmaya başlamışlardır. Paylaşılan bu bilgilerin yoğunluğu ve çeşidi oldukça fazladır. Aynı zamanda üretilen bu verilerin çoğu düzensiz haldedir ve işlenmeye muhtaçtır.

3.2.1.4 Nesnelerin interneti kavramı

Akıllı cihaz sayısındaki artış kaçınılmaz olarak hayatımıza çevremizdeki tüm araçlar ile iletişimde bulunma fırsatını sunmaktadır. Evimizdeki buzdolabı ve ısıtma sistemlerinden, garajımızda bulunan aracımıza kadar bir çok aygıt ile internet altyapısı veya doğrudan iletişim kurmak mümkün hale gelmiştir. Tüm bu aygıtların internet ortamına bağlanması ile nesneler akıllanmakta ve veri üreten birimler olarak son dönemde karşımıza çıkmaktadır.

3.2.2 Google dosya sistemi

Google'ın araştırma ve geliştirme topluluğu sık sık çeşitli alanlarda yeni araştırma belgeleri yayınlıyor. Geniş bir bilimsel topluluk ile işbirliği kurmayı ve çeşitli alanlarda katkıda bulunmayı amaçlamaktadırlar. Onların Ar&Ge ekibi, Algoritma'dan Dağıtılmış Bilgi İşlem'e değişen Bilgisayar biliminde çeşitli doktora

geçmişine sahip bir çok araştırmacıdan oluşmaktadır. Bu ekipler Google Altyapısı uygulanması üzerine araştırmadan uygulamaya kadar yeni ürünler geliştirmek için kullanılır. 2003, 2004 ve 2007'te Sanjay Ghemawat, diğerleri ve Jeffry Dean bilgisayar mimarlarının tasarladığı büyük veri uygulamaları şekli değiştiren, ki aşağıda incelenecek, tezler yayınladı.

"Google Dosya sistemi" [18] başlıklı tez, Google Ar-Ge birimi tarafından 2003 yılında yayınlandı. GFS veya Google GFS, dosya sistemi, çok sayıda yararlı donanım kullanarak verilere güvenilir ve verimli bir erişim sağlamaya tasarlanmış dağıtılmış bir dosya sistemidir. Bu dosya sistemi Google'da özel bir şekilde uygulanmaktadır, ancak tez yayımlandığından beri belirtilen niteliklere dayalı benzer bir sistem inşa etmek için çeşitli girişimleri olmuştur.

İşleme ihtiyaçları zamanında hızla arttığından Google'ın Arama etkinliğini ve güvenilirliğini artırmak için Google GFS hayata geçirildi. Dağıtılmış dosya sistemlerinin yararlı donanımı üzerinde çalışan Ölçeklenebilirlik, Güvenilirlik ve Kullanılabilirlik kendi tasarımlarının temel ilkelerindendir. Tasarım birçok ucuz yararlı donanım bileşeninden oluştuğunu, çok sayıda büyük dosyayı, kısa yazıları ve ileri okumaları sakladığını ve API kullanıcılarını kullanıma kolaylığını öngörür [18]. Onların tezinden büyük veri analitiği çerçevelerine iyi dönüştürülmüş anahtar fikirler vardır.

3.2.3 Hadoop dosya sistemi (HDFS)

HDFS, sıradan sunucuların diskleri bir araya gelerek büyük, tek bir sanal disk oluşturmasıdır. Bu sayede çok büyük boyutta bir çok dosya bu dosya sisteminde saklanabilir. Bu dosyalar bloklar halinde (varsayılan 64MB) birden fazla ve farklı sunucu üzerine (varsayılan 3 kopya) dağıtılarak yedeklenir. Bu sayede veri kaybı önlenmiş olur.

3.2.4 Map/Reduce

Google 2004'ün sonunda kamuya başka bir araştırma makalesi yayınladı. Bu sefer büyük bir küme yararlı donanım üzerinde büyük veri setlerini işlemek için kullanılan bir programlama modeline dayandırıldı. MapReduce özünde, veri işlemenin durum değişkenleriyle değil de bu süreçte kullanılan işlevler ile ilişkili giriş ve çıkışlar

üzerine uğraşan bir yer olan fonksiyonel programlama tarzı paradigmasına dayanmaktadır. Bu şekilde yazılmış programlar otomatik olarak benzerlikleri bulunur, dağıtılır ve büyük bir küme üzerinde yürütülür. Tez MapReduce'in sadece teorik programlama paradigması amacıyla olmadığını, "her gün Google üzerinden binlerce MapReduce işinin sunulduğunu belirtti" [14,15].

Bu sistem depolanan veri miktarındaki artan karmaşıklık sebebiyle dizayn edilmiştir ve işleme her görev için çok özel sistemleri ile yapılmıştır. MapReduce amacı ortaya ortak süreçler çıkarmak, paralelleştirme, hata toleransı, dağıtım ve çoklu sistemler arasında yürütme ve kod derleme gibi birçok karmaşık programlama ayrıntısını gizlemektir. Sistemin temeli Lisp programlama dilinde kod doğasının the `_map_` and `_reduce_` parçalarına dayanır.

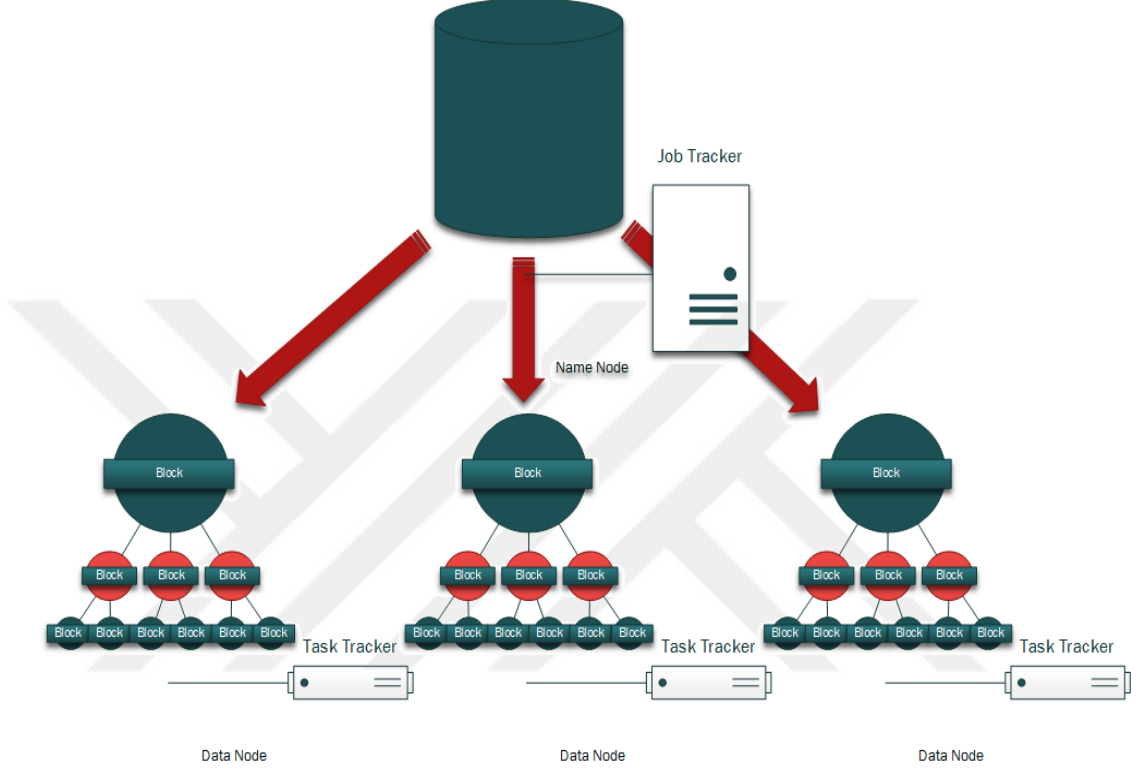
Google'ın hem GFS hem MapReduce çerçeveleri uygulaması C / C++ ile yazılmış. Hadoop Java ile yazılmıştır ve Tüm önemli bileşenlere kendi Java API üzerinden bir API sağlar. Makine koduna derlenmiş C++ ile karşılaştırıldığında Hadoop'un JIT dilinde yazılması nedeniyle doğal bazı performans tartışmaları vardır.

HDF'ler öncelikle büyük miktarda yararlı donanımı çalıştırmak için tasarlanmıştır; ancak Hadoop'u kullanıcıların kullandıkları CPU ve bellek miktarına göre işlem için ödeme yapabildikleri bir çevrimiçi servis olarak sunan birden fazla bulut örgütü var. Dosya sistemi mevcut dağıtılmış dosya sistemlerine çok benzerdir; ama bu daha fazla hataya toleranslıdır. Ayrıca da POSIX dosya sistemleri kullanan ve birkaç arayüz değiştiren dosya sistemi sözdizimini kullanan benzerlikler ile tasarlanmıştır.

Dosya Sistemi, Google Dosya Sistem tezi sunduğu için aynı Master / Slave mimarisi üzerine inşa edilmiştir. Küme müşteriler tarafından dosya sistemini ve dosyalara erişimi yöneten tek bir ana sunucu, NameNode'dan oluşur. Birincil olanın başarısız olması durumunda, bir yedek NameNode

oluşturmak mümkündür. Çoklu DataNodes karşısında dosyalar bloklar halinde saklanır. Veriler fazlalılık oluşturan ve hata toleransı geliştiren çoklu DataNodes'a çoğaltılır, varsayılan çoğaltma değeri üç ayrı veri düğümüdür. Her dosya için meta verileri NameNode'da bellekte depolanır, ve hatta sistemin yeniden başlatılması için NameNode için bir günlük dosyası oluşturulur, belleğe geri meta veri yükleyebilir.

MapReduce Hadoop müşteregin ikinci çekirdek bileşenidir. Hem HDF'ler hem de MapReduce işleme diskten belleğe doğrudan donanım üzerine yüklenecek şekilde aynı sistemler üzerinde çalışır, ki bu veri işleme performansını artırır. MapReduce çerçevesine Master/Slave mimarisi vardır. Bir temel kaynak yöneticisi vardır ve her düğüm kendi düğüm yöneticisi olacaktır.



Şekil 3.1: HDFS Dosya Sistemi

Müşteri, iş sunacaktır, kaynak yöneticisi işlenecek verilerinin yerini almak için NameNode'a konuşacaktır, sonra işi HDFS'lerden veri çekmek için düğüm yöneticisine sunar. Sonra düğüm yöneticisi MapReduce görevlerini başlatacak olan Java programını başlatacak. Şekil 3.1 üzerinde HDFS'in bileşenleri görülmektedir.

MapReduce ilk aşaması eşleştirilmesidir. Veri ve ardından bu veri elementi çıktısı üzerinde bazı dönüşümü veya hesaplama gerçekleştirecek olan girdinin her satırı tek tek Mapper adında bir fonksiyona işlenecektir. Harita faz bir giriş çifti alır ve bir dizi anahtar/değer çiftleri üretir. Her çıktıya bir Reducer gönderilecektir. Harita, müşteriye anlamlı gelen kayıtlardan birşeyler alır, karıştırır ve sonucu sıralar, sonra onu redüktöre verir.

Azaltılmış fonksiyon anahtar/değer çiftini alır ve anahtarla değerleri birleştirir ve bunları birbirine birleştirir. Azaltılan fazdan çıkan, HDF'ler üzerine bir dosyaya gönderilebilir veya veriyi daha fazla analiz etmek için MapReduce aşamasının başka bir yinelenmesiyle gönderilebilir. İstenen çıktıyı üretmeden ister bazı büyük hesaplama yapmak için isterse de verileri dönüştürmek için MapReduce aracılığıyla birkaç kez veri göndermek mümkündür. Azaltma verileri birleştirir, özetler, filtreler ve dönüştürür. Veriler daha sonra başka bir MapReduce iş, bir dosya veya çıkış ekranına gönderilebilir.

Harita ve Azaltma hem paralel olarak yapılır hem de Master düğüm tarafından kontrol edilir. Girdinin her satırı yapılandırmasında varsayılan bir boyutta parçaların içinde X haritası görevleri ayrılmıştır. Azaltma fazı Y bölmeleri görevleri azaltır. Görevler dinamik olarak işçilere atanır. Her işçi Master'a geri kalp atışı işi gönderir, ve eğer herhangi bir nedenden bir işçi çevrimdışı kaldıysa ve ana düğüm işçiden geri kalp atışı sinyali almazsa, sonra usta önceki görev için yeni bir işçiyi tayin edecektir. Ana düğüm işçi verinin bulunduğu aynı düğümün üzerinde diskten okuyabilsin diye verilerin konumunu göz önünde bulundurabilir. Usta Kaynak yöneticisi işçilerin görevlerini izlediğinden işçi düğümlerinden birinin herhangi bir nedenle başarısız olması durumunda Hadoop'un MapReduce kısmı güvenilirlik oluşturur.

MapReduce işleri güçlüdür ve Hadoop'un kalbidir; ama onları yazması çok zordur. Hadoop'tan eklenti olarak geliştirilmiş dış ürün sayısı sebebiyle bir ekosistem olarak bahsedilir. Yahoo, Facebook, Twitter ve diğer erken uyarlayıcılar bir şekilde Hadoop ekosistemine katkıda bulunmuştur. Pig, Mahout, R Connectors, Hive, ve Oozie Hadoop ekosisteminin farklı bileşenlerin içinden belki de en popülerleridir. Pig ve Hive programcı olmayanlara dost bir ortam sağlayan MapReduce'un tepesine kurulan kısaltmaları düşünür [17]. Storm gibi Twitter'ın gerçek zamanlı analitik motoru olan bir çok daha küçük veya özel olarak yapılmış ürün vardır. Hadoop hala büyüme aşamasında geçiyor ve büyük olasılıkla yakın gelecekte yeni bir dizi ürüne sahip olacaktır.

3.2.4.1 Name node

Ana (Master) süreç olarak blokların sunucular üzerindeki, dağılımından, yaratılmasından, silinmesinden, bir blokta sorun meydana geldiğinde yeniden oluşturulmasından, her türlü dosya erişiminden sorumludur.

Kısacası HDFS üzerindeki tüm dosyalar hakkındaki bilgiler (metadata) NameNode tarafından saklanır ve yönetilir. Her kümede yalnızca bir adet NameNode olabilir.

3.2.4.2 Data node

İşlevi blokları saklamak olan işçi (Slave) süreçtir. Her DataNode kendi yerel diskindeki veriden sorumludur. Ayrıca diğer DataNode'lardaki verilerin yedeklerini de barındırır. DataNode'lar küme içerisinde birden fazla olabilir.

3.2.4.3 Job tracker

Yazılan MapReduce programının küme üzerinde dağıtılarak çalıştırılmasından sorumludur. Dağıtılan iş parçacıklarının çalışması sırasında oluşabilecek herhangi bir problemde o iş parçacığının sonlandırılması ya da yeniden başlatılması da JobTracker'ın sorumluluğundadır.

3.2.4.4 Task tracker

DataNode'ların bulunduğu sunucularda çalışır ve JobTracker'dan tamamlanmak üzere iş parçacığı talep eder. JobTracker, NameNode'un yardımıyla DataNode'un lokal diskindeki veriye göre en uygun Map işini TaskTracker'a verir.

Bu şekilde verilen iş parçacıkları tamamlanır ve sonuç çıktısı yine HDFS üzerinde bir dosya olarak yazılarak program sonlanır.

3.2.5 Apache spark

Apache Spark Hadoop Framework'un gücünü kullanan üst düzey bir projedir. MapReduce doğasında Spark'ın üstesinden gelmek için oluşturulduğu bazı sınırlamalar vardır [3]. Spark her MapReduce işi arasındaki bazı performans boşlukları sebebiyle MapReduce paradigmasını güçlendirmez. Bu Hadoop dosya sistemi API'yı kalıcı depolama ve geri alma için kullanır; fakat Spark akılda

soyutlama ile dizayn edilmiş ve çerçevenin Hadoop ile sıkıca birleştirilmeyeceğini ve gelecek dağıtık dosya sistemlerine potansiyelde bağlanabileceğinden emin olmuştur. Bu arada Spark topluluğunda yer alan mevcut projeler: Spark, Spark SQL, Spark Streaming, MLib (Machine Learning), ve GraphX (Graph based system). Spark, Bilgi Güvenliği verilerine Makine öğrenme teknikleri uygularken belirgin bazı konuları çözmeye yardımcı olabilir [42].

Spark'ın iç parçaları Berkeley'in ortaya koyduğu Resilient Distributed Datasets (RDD) olarak bilinen dağıtık veri soyutlamaya dayanır. RDDler In-Memory Küme Bilgi İşlem için hataya dayanıklı soyutlamalardır. Onlar depolama veya bir şekilde değişime uğramış başka RDD üzerinden veriler arasından yaratılmış, değiştirilemez. Bir RDD başka RDD'den oluşturulduğu zaman hafıza-içi süreçler sayesinde, MapReduce gibi diskler üzerine depolanmaz. Bu Hadoop doğasında olan MapReduce işleme üzerinde bir avantaja sahiptir. Dağıtık veri kümesinin değişmez doğası gereği, bunların üzerinde yapılan dönüşüm ve işlemler veri kümesindeki her eleman için uygulanması gerekir. Bu soyutlamanın dezavantajı değişken durum gerektiren veri kümeleri üzerinde çalışma bu tür sistem içinde tamamıyla farkedilmeyecek olmasıdır [46].

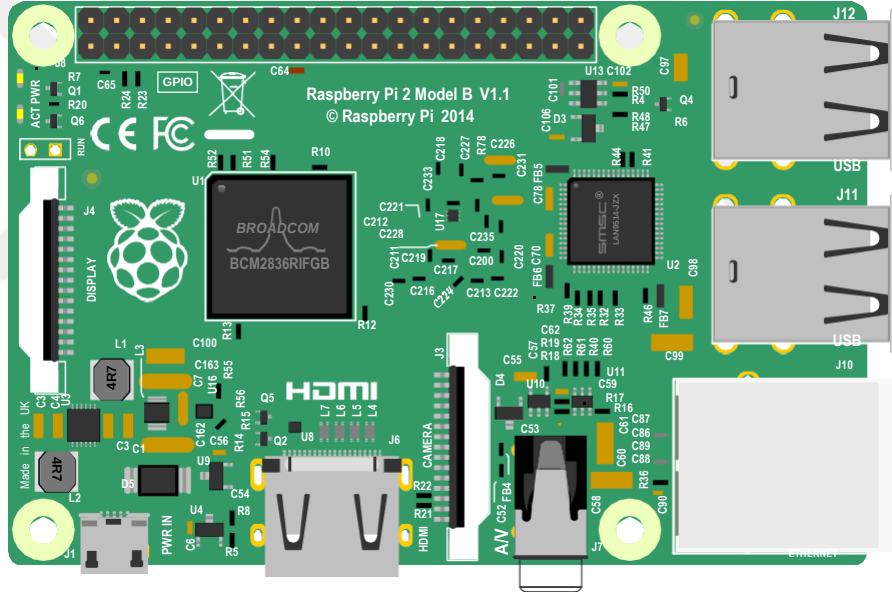
Spark'ın avantajı verilerin işlenmesindeki hızıdır. Algoritmanın doğası sabit ve kullanışlıysa eğer hızların 20-40x daha fazla çıkabileceğini gösteren Spark'ı Hadoop'un MapReduce'u ile karşılaştırmak için uygulanan çok sayıda kriter olmuştur. Spark'ın hesaplama modeli daha hızlı gerçek zamanlı veri analizi için izin verir [40]. Spark ile uygulanmış kullanım durumları: Conviva Inc. geniş In-Memory Analizini Hadoop'un 20 saatine karşılık 30 dakikada yapabildi; Mobile Millennium GPS sensörleri ile trafik modelleyebildi; ve Twitter'da bağlantıları spam olarak tespit edebilen Spam Sınıflandırma algoritması yapmayı başardı.

Discretized Streams (D-Streams) Berkeley'den çıkan başka bir modeldir; büyük veri uygulamalarını yakın gerçek zamanlı gelen veri işlemesine imkan sağlamıştır ki, bu sayede veri analizi toplumunda büyük verilere gerçek zamanlı analiz için önemli bir sorunun üstesinden gelinmiştir [4]. D-Akımlar gerçek zamanlı veriler analiz edildiğinde zorlukların çoğunun üstesinden gelmeye yardım eden hataya dayanıklı ve tutarlı akış programlama modeli için izin verir. DStreams bir dizi deterministik parti hesaplamalarını küçük zaman çıktıları üzerine parçalara ayıran RDD'lerin üzerine

kurulur [47]. Spark ölçeklenebilir gerçek zamanlı analitik işleme sistemlerine için izin vermek hem D-Streams hem de RDD'ler uygular. Şu anda, D-Akımlar zaman aralıklarını yarım saniyeden daha fazla işlemesi gerekir. Eğer gereksinim verileri alt saniyelik aralıklarla işlemek ise, o zaman başka teknik uygulanmalıdır.

3.3 Raspberry Pi

Raspberry Pi Birleşik Krallık'ta Raspberry Pi Vakfı tarafından okullarda bilgisayar bilimini öğretmek amacıyla geliştirilmiş kredi kartı büyüklüğünde bir bilgisayardır. Şekil 3.2 üzerinde bileşenleri görülmektedir. Fiyatı modeline göre değişiklik gösterir ve 5\$ ile 35\$ arasındadır. Raspberry Pi 2 modeli Şubat 2015'te çıkmıştır. Sonraki model Raspberry Pi Zero, Kasım 2015'te çıkmıştır. Son model olan Raspberry Pi 3 ise Şubat 2016'da çıkmıştır.



Şekil 3.2: Raspberry Pi

Bu proje 2006 yılında Cambridge Üniversitesinde başlatılmıştır. Temeldeki düşünce, düşük maliyet ile herkesin ulaşabileceği bir bilgisayar üretilmesidir. Raspberry Pi'nin popülerliği arkasındaki topluluğun bu bilgisayar üzerinde geliştirip paylaştığı boyundan büyük işler ile dikkat çekmektedir. Eğitim amacı ile çıkılan bu yolda yapılan bu işler, Raspberry Pi'yı sadece küçük bir bilgisayar olmaktan çıkarmıştır. Üzerinde yapılan bir çok çeşitte proje olması ile birlikte bilgi güvenliği alanında da bir çok uygulama alanı bulabileceği fikri bizleri bu bilgisayarı kullanmaya sevk etmiştir.

3.3.1 Raspberry pi çeşitleri ve özellikleri

Tez kapsamında kullanılacak Raspberry Pi 2 ve Raspberry Pi 3'e ait özelliklerin kıyaslama tablosu Çizelge 3.1 üzerinde görülmektedir. Kullanılacak kartlar arasındaki görsel farklılıklar Şekil 3.3 üzerinde işaretlenmiştir.

Çizelge 3.1: Raspberry Pi Kıyaslama Tablosu

Birim	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B
İşlemci Modeli	Broadcom BCM2837 64Bit Quad Core ARM Cortex A53	Broadcom BCM2836 32Bit Quad Core ARMv7
İşlemci Hızı	1.2 GHz	900 MHz
İşlemci Çekirdek Sayısı	4 Çekirdek	4 Çekirdek
Görüntü İşlemci	Videocore IV 400MHz	Videocore IV 250Mhz
RAM	1 GB SDRAM 400 MHz	1 GB SDRAM 400 MHz
Depolama Birimi	Mikro SD Kart	Mikro SD Kart
USB	4 Adet USB 2.0	4 Adet USB 2.0
Enerji Tüketimi	2.5A 5V	1.8A 5V
GPIO	40 pin	40 pin
Ethernet	10/100 Mbps	10/100 Mbps
Wifi	Var	Yok
Bluetooth	Var	Yok
Görüntü Çıkış Arabirimi	HDMI	HDMI
Ses Çıkış Arabirimi	3.5mm	3.5mm

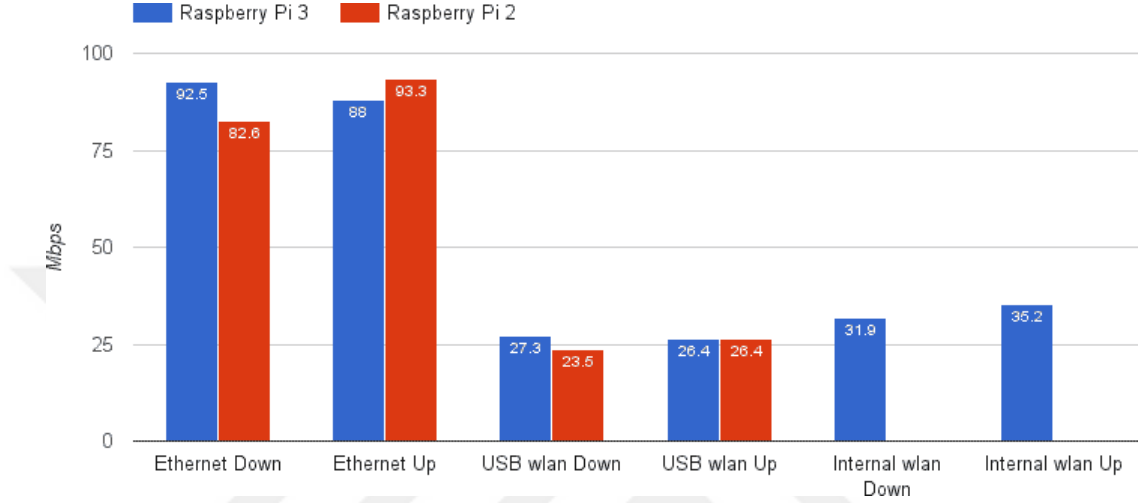
Her ne kadar Raspberry Pi 2 ve Raspberry Pi 3 aynı gibi gözükseler de aralarında büyük farklar bulunmaktadır. RPi 3, 64Bit mimarisi, 1.2 GHz CPU hızı, 400MHz GPU hızı ve dahili Wifi/Bluetooth modülleri ile tercih edilirken ön plana çıkmaktadır.



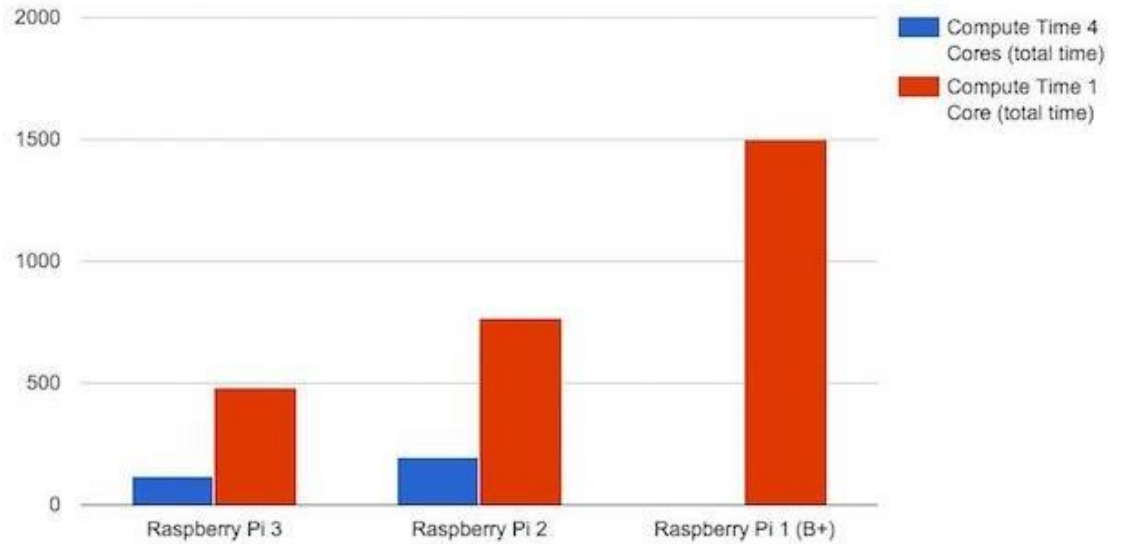
Şekil 3.3: Raspberry Pi 2 ile Raspberry Pi 3 arasındaki Görsel Farklar

3.3.2 Raspberry pi performansı ve karşılaştırılması

Raspberry Pi Ağ Hızı test değerleri Şekil 3.4 üzerindeki grafikte, CPU hız değerleri Şekil 3.5’de bulunan grafikte, Hafıza performansı değerleri Şekil 3.6 üzerindeki grafikte ve Dosya Transfer hızı karşılaştırma değerleri Şekil 3.7 üzerindeki grafikte gösterilmektedir.

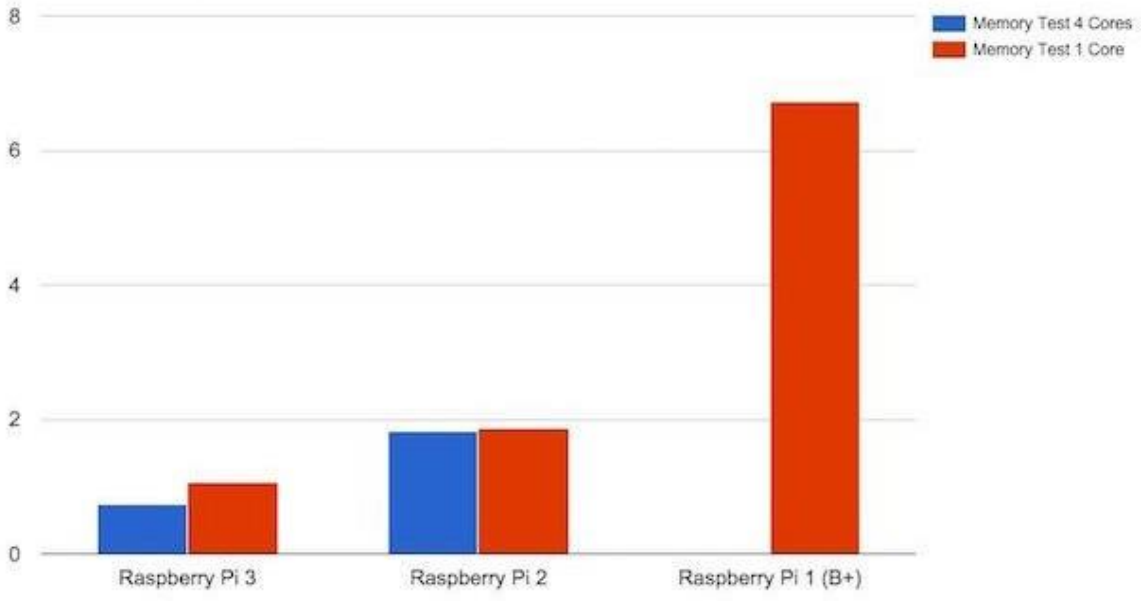


Şekil 3.4: Raspberry Pi Ağ Hız Testi

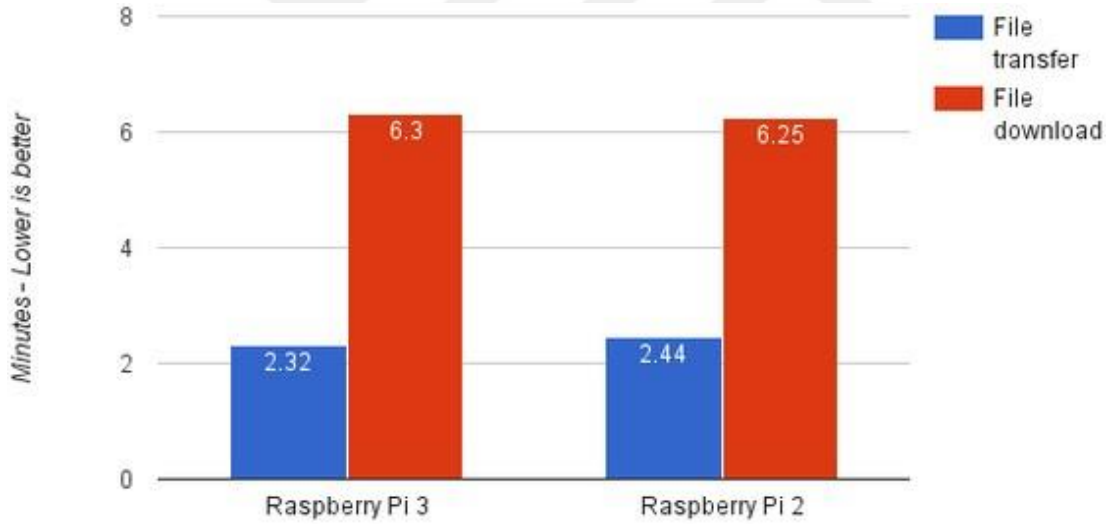


Şekil 3.5: Raspberry Pi CPU Performans Kıyaslama Değerleri

Şekil 3.5 üzerinde bulunan değerler milisaniye cinsinden olup işlem tamamlama süresi az olan Raspberry Pi’ın daha hızlı olduğu sonucu değerlendirilmektedir.



Şekil 3.6: Raspberry Pi Hafıza Performans Kıyaslama Değerleri



Şekil 3.7: Raspberry Pi Veri Transfer ve Dosya İndirme Hızı Performansı

Aynı ortam ve değişkenler altında yapılan testler bariz bir şekilde Raspbbery Pi 3'ün Raspberry Pi 2'den CPU, Memory ve dosya transfer işlemlerinde daha iyi sonuçlar ürettiği ve daha başarılı olduğu sonucuna ulaşılmıştır. Raspberry Pi 3 sonradan piyasaya sürülmesine rağmen fiyatının Raspberry Pi 2 ile aynı olması, gelecek çalışmaları için Raspberry Pi 3'ün kullanılmasının daha mantıklı olacağı gerçektir.

4. KISITLI KAYNAK KULLANILARAK DÜŞÜK MALİYETLİ DAĞITIK SİBER TUZAK SİSTEMİ MİMARİSİ

Şuan kullanımda olan saldırı tespit metodları günümüzdeki saldırı tehditlerinin tespit etmekte yeterli değildir [1,10,62]. Klasik saldırı tespit yaklaşımlarımızın saldırganları caydırdığı günler geride kalmıştır. Saldırganlar sosyal mühendislik teknikleri ile birlikte teknik yeteneklerini bir araya getirip kurumların sistemlerine sızmayı başarmaktadırlar. Gerçekleşen saldırıların çoğu artık belirli bir hedef yazılıma veya donanıma yönelik yapılmaktadır. Güvenlik tehditlerini analiz etmek için bir çok yaklaşım bulunmaktadır. Bu teknikler, büyük verinin getirdiği çoklu kaynaktan güç alma imkanından yoksundurlar.

Burada tavsiye edilen mimari, büyük veri analitiği çatısı altında gerçekleştirilecektir. Büyük veri üzerine yapılan çalışmalar sonrasında ortaya çıkan teknikler sayesinde, daha önce elimizde olmayan bir çok imkana şuan sahibiz. Burada tavsiye edilen altyapı varolan güvenlik araçlarına bir alternatif olarak değil, ek bir tespit katmanı olarak tasarlanmıştır.

Davranışsal analiz ve honeypotlar çeşitli saldırıları tespit etmek için kullanılabilir. Örneğin: web sayfası aktiviteleri, dolandırıcılık tespiti, Linux SSH oturumları analizi gibi. Bu deney ortamı SSH honeypotu kullanarak saldırgan davranışı hakkında veri toplamayı hedeflemektedir. Kullanıcı verisi “pacct” ve “bash_history” kullanılarak yapılacaktır.

Web üzerinde kısıtlı miktarda herkese açık verisetleri bulunmaktadır. Fakat SSH honeypot verisi bulmak oldukça zordur. Bu zorluk sebebi ile deney ortamı belirli bir süre boyunca kullanıcılardan bu veriyi toplamak üzere ayarlanacaktır. Bash profil verisi bulmak ise oldukça kolaydır. Popüler veri seti maskelenmiş saldırıların davranışlarını analiz etmek için kullanılacaktır [58]. Bu verinin en büyük dezavantajı, komutlarda parametrelerin olmamasıdır. Komutların argümanları ile şüpheli aktiviteleri tespit etme olasılığı daha yüksektir.

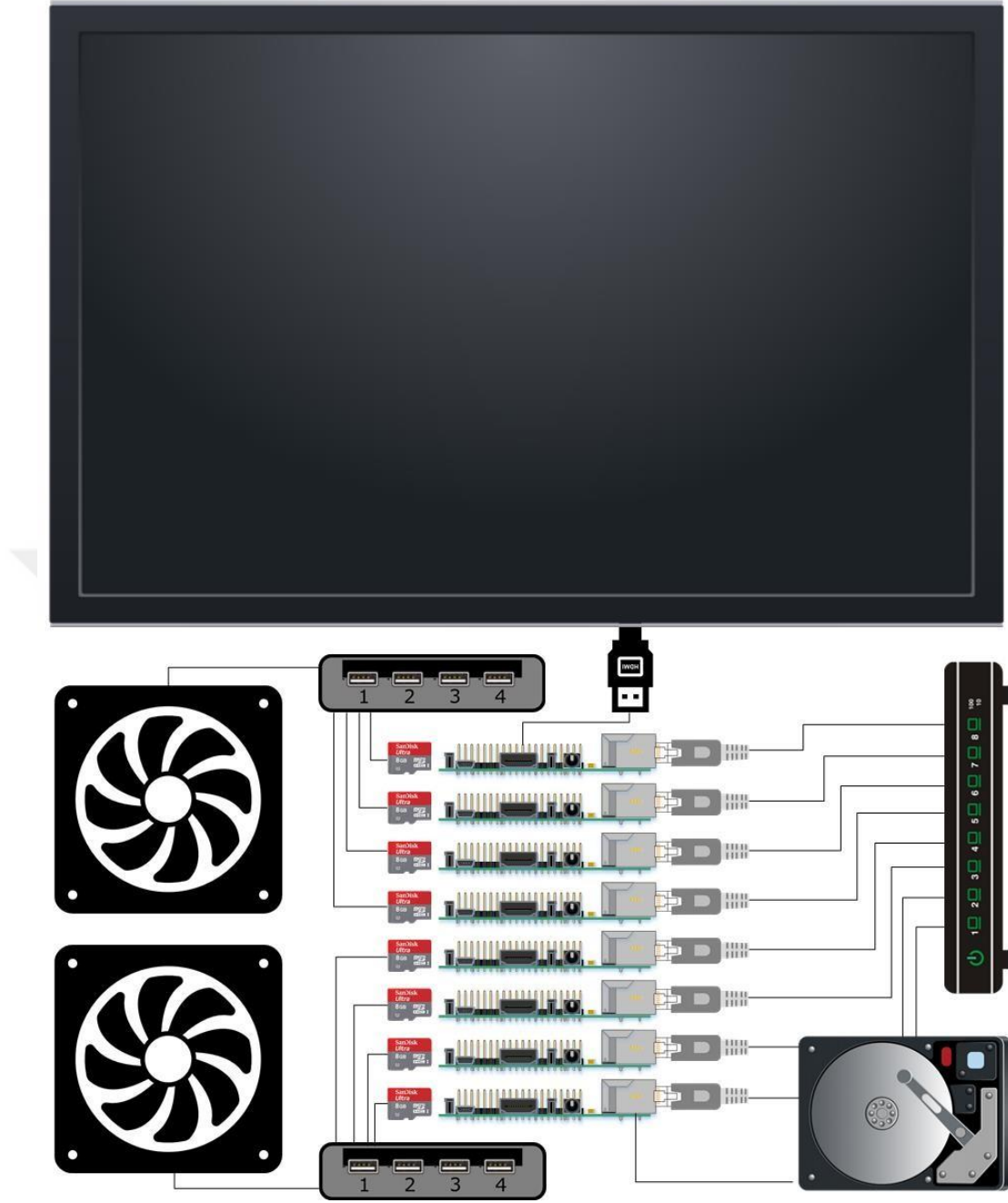
Veri seti toplandıđında HDFS dosya sistemi ierisinde saklanacaktır. Veri temizleme ve istenen formata dnüştürme işlemleri düzenli ifadeler (Regular Expression) ile yapıldıktan sonra tüm analiz tamamlanacaktır.

Naïve Bayes yöntemi kullanılırken, veri seti “iyi huylu” ve “kötü huylu” olmak üzere iki gruba ayrılacaktır. Her kullanıcının ilk 5000 komutu iyi huylu davranışı temsil edecektir. Honeypot üzerinde toplanan komutların ise tamamı kötü huylu olarak işaretlenecektir. Bu analizin kısıtlarından bir tanesi bayrak ve argüman bilgilerinin eksik olmasıdır [59].

4.1 Donanım Altyapısının Hazırlanması

Tezin testlerini ve başarısını ölçmek için kullanacağımız donanım altyapısını hazırlarken, düşük maliyet ile en yüksek performansı alabileceğimiz parçalar kullanılmaya dikkat edilmiştir. Bu kapsamda seçilen parçalar sayıları ve özellikleri ile birlikte ayrıntıları ile açıklanacaktır.

Temin edilen parçaların bağlantıları yapıldıktan sonra elde edilmesi planlanan donanım Şekil 4.1 üzerinde görüldüğü gibi olacaktır. Burada eğer aynı sistemi kurgulamak istenecek olur ise malzemelerdeki farklılık aynı özelliklerde olduğu sürece herhangi bir sorun teşkil etmeyecektir. Biz tez kapsamında elimizde olan ve maliyet konusunda tezin hedefi doğrultusunda çalışmayı destekleyecek parçaların seçilmesine özen gösterdik.



Şekil 4.1: Mantıksal Tasarım

Tez çalışmamız için tüm bileşenlerin bir araya getirilmesi, gerekli bağlantıların yapılması ve düzenlenmesi sonrasında son görünümü ise Şekil 4.2’de görüldüğü gibidir.



Şekil 4.2: Siber Olaylara Müdahale Aracı (SOMA v3.01)

4.1.1 Malzeme seçimi ve özellikleri

Tez çalışmasına başladığımızda Raspberry Pi'nin son sürümü Model B 2 idi. Fakat çalışmanın sürdüğü sırada yeni Raspberry Pi 3 Model B tanıtılmıştır ve aynı fiyat ile piyasaya sürülmüştür. Bu sebep ile sistemde her iki tür bilgisayar da aynı anda kullanılacaktır. Aynı zamanda bu durum performanslarını da karşılaştırma fırsatı yakalamamıza sebep olmuştur. Çizelge 4.1 üzerinde seçilen malzemeler listelenmiştir.

Çizelge 4.1: Malzeme Seçimi ve Özellikleri

Adet	Malzeme	Özellikleri
4	Raspberry Pi 2 Model B	-
4	Raspberry Pi 3 Model B	-
1	TP-Link Switch	10/100 Mbit, 8 Port
8	Ethernet Kablosu	30 cm, Kısa, CAT6
2	USB Güç Kaynağı	6 Port, 5.4 Amper
8	Mikro USB Kablo	20 cm, Kısa
8	Mikro SD Hafıza Kartı	8GB, Class 10
1	Sabit Disk	500GB, SATA2
4	Soğutucu Fan	8 cm, 0,18 Amper
1	AOC Monitör	18.5 inch, 1366x768, VGA
1	VGA-HDMI Dönüştürücü	OEM
1	Microsoft All-in-one Klavye	Dokunmatik Mouse + Klavye
28	Stand Off Vida	19,05 mm
1	Taşıma Çantası	19 inch

Burada güç kaynağı ve hafıza kartı seçimi üzerinde özellikle durulması gereken bir kaç nokta bulunmaktadır. Hafıza kartları işlem yoğunluğu olacağı için sınıfının en yüksek veri transfer değerine sahip olan “Class 10” tercih edilmiştir. Dikkat edilmesi gereken diğer nokta ise güç kaynaklarının akım değerleridir. Raspberry Pi, her ne kadar bekleme durumunda düşük güç tüketiyor olsa da, yoğun işlem altında RPi başına 2 ampere kadar yükselmektedir. Bu tercihin düşük yapılması durumunda RPi kendini resetleyerek yeniden başlamakta ve yapılandırma sırasında sıkıntı yaşanmasına sebep olmaktadır.

4.1.2 Maliyet analizi

Çizelge 4.2 üzerinde kullanılan malzemeler için ortalama fiyatları ve kullanılan malzeme adeti ile gösterilmektedir. Burada verilen rakamlar 2016/Haziran ayı içerisinde yapılan alımlar için gösterilmektedir. Her ürün için değişkenlik gösterebilir. Çizelge 4.2’den de görüleceği üzere kuracağımız sistemde maliyeti arttıran en büyük kalem Raspberry Pi’lardır. Kurulması planlanan sistemdeki Raspberry Pi sayısı artırılıp azaltılarak kurumun ihtiyaçlarına göre şekillendirme yapmak mümkündür. Tezimizde 8 adet RPi kullanılıyor olması sadece deney amaçlıdır. Az sayıda Raspberry Pi ile de uygun iyileştirmeler yaparak yüksek performans elde edilmesi mümkündür. Daha fazla sayıda Raspberry Pi kullanımı ile de çok daha büyük miktarlardaki veri kümeleri çok hızlı bir şekilde analiz edilebilir.

Çizelge 4.2: Malzeme Maliyet Analizi

Adet	Malzeme	Maliyet
4	Raspberry Pi 2 Model B	4 x 150 = 600 TL
4	Raspberry Pi 3 Model B	4 x 150 = 600 TL
1	TP-Link Switch	60 TL
8	Ethernet Kablosu	8 x 5 = 40 TL
2	USB Güç Kaynağı	2 x 75 = 150 TL
8	Mikro USB Kablo	8 x 10 = 80 TL
8	Mikro SD Hafıza Kartı	8 x 20 = 160 TL
1	Sabit Disk	150 TL
4	Soğutucu Fan	4 x 10 = 40 TL
1	AOC Monitör	300 TL
1	VGA-HDMI Dönüştürücü	40 TL
1	Microsoft All-in-one Klavye	130 TL
28	Stand Off Vida	30 TL
1	Taşıma Çantası	200 TL
		TOPLAM : 2580 TL

4.2 Yazılım Altyapısının Hazırlanması

Öncelikle yazılım altyapısının kurulması için RPİ'lerin üzerine, RPİ için tasarlanan Debian tabanlı “Raspbian” işletim sisteminin kurulumu gerçekleştirilmelidir. Bu kurulum tez kapsamı dışında olduğu için anlatılmamıştır. İlgili kurulum dökümanları ve işletim sistemi imajına ürüne ait internet sayfası üzerinden erişebilirsiniz. Biz kurulumumuzu Raspbian Jessie versionu ile gerçekleştirdik.

Toplanan saldırgan davranışı verisi üzerinde işlem yapacağımız bilgileri elde edeceğimiz, ayrıştırıp belirli dönüşümleri gerçekleştirecek kaynak kodlar **EK** içerisinde paylaşılmıştır. Biz geliştirme işlemlerini Scala dilini kullanarak gerçekleştirdik.

4.2.1 Kippo honeypot tuzak sisteminin kurulumu ve yapılandırılması

Burada honeypot için Raspberry Pi 3'ü kullanacağız. Üzerinde çalışacak işletim sistemi linux tabanlı “Raspbian” olacaktır. RP3'ün işlemci Broadcom BCM2837 olarak tanıtıldı. Üzerinde 4 adet Cortex-A53 çekirdeği barındıran 64-Bit işlemci 1.2GHz saat hızında çalışıyor. 1GB RAM ile gelen RP3, WiFi N ve Bluetooth 4.1 kablosuz iletişim yongaları da dahili olarak yer almaktadır. Dahili depolama olarak ise Class 10, 8 GB'lık MicroSD kart kullanılacaktır. RP'ı honeypot olarak kullanmak oldukça avantajlıdır. Maliyetinin düşük olması, düşük enerji tüketimi ve Linux

tabanlı bir işletim sistemi ile çalışıyor olması en büyük avantajlarıdır. Tüm bu özellikler dikkate alındığında, oldukça ucuz olan bu çözüm ile bir çok noktadan veri toplayacak honeypotlar yaratmak bizim için mümkün olacaktır.

Halihazırda bir çok açık kaynaklı honeypot dağıtımı bulunmaktadır. Bu proje için Kippo SSH seçilmiştir. Kippo basit bir Python betiği ile kabuğu taklit ederek gerçek bir linux bilgisayar gibi davranmaktadır. Kippo dökümantasyonunda dinlenecek portun 2222 olmasını tavsiye eder fakat bu çok yaygın olduğu dolayısı ile bilindiği için biz 4576 numaralı portu kullanacağız. Kippo genelde kaba kuvvet saldırılarını tespit etmek için kullanılmaktadır. Burada bunun seçilmesindeki bir diğer önemli sebep ise tıpkı bir UNIX dosya sistemi gibi davranmasıdır (/etc/passwd ve /etc/proc gibi birimleri mevcuttur). Kippo'nun POSIX dosya sistemi yeterince iyidir. Saldırganları bu sistem üzerinde daha fazla tutabilmek için bir kaç ayarlama yapmamız gerekmektedir.

Raspberry Pi 3 üzerine Kippo honeypotun kurulumu ve yapılandırılması aşağıda adım adım açıklanacaktır.

Öncelikle apt deposunu güncelleyeceğiz.

```
apt-get update
```

Varsayılan olarak Kippo port 2222 üzerinde çalışır. Port 22 üzerinde çalıştırmak için standart SSH kurulumunu harekete geçirmemiz gerekmektedir. Port 2222'nin üzerinde bırakmak size tarayıcılardan daha az saldırı getirecek, çünkü varsayılan SSH tarayıcıları 22 numaralı olan varsayılan SSH portunu tararlar. Bu nedenle hala makineyi uzaktan yönetebilmeniz için varsayılan SSH bağlantı noktanızı başka bir port ile değiştirmeniz gerekecektir (Örneğin 4567).

Bunun için aşağıdaki adımları gerçekleştirmek gerekmektedir.

Bağlantı noktasını değiştirmek için SSH sshd_config dosyası yapılandırılmalıdır.

```
vi /etc/ssh/sshd_config
```

“Port 22”yi, “Port 4567” veya seçtiğiniz bağlantı numarası ile değiştirilmelidir. SSH sunucusu yeniden başlatılmalıdır.

```
/etc/init.d/ssh restart
```

Bu noktada eğer SSH kullanarak ana bilgisayara bağlıysanız, SSH oturumunuzu kesmeniz gerekmektedir. Belirttiğiniz yeni bağlantı noktasına bağlanmalıdır.

Kippo'nun çalışması için gerekli bir önkoşulları yükleyin.

```
apt-get install python-dev openssl python-openssl  
python-pyasn1 python-twisted
```

Önkoşullar yüklü ve hazırsa, Kippo örneğimizi indirecek alt versiyonları yüklememiz gerekmektedir.

```
apt-get install subversion
```

Kullanmak istediğimiz port ile ilgili diğer bir konu ise Linux içinde sadece tüm yetkili kullanıcı port 1024 altındaki portların kullanılmasına izin veriyor olması ve güvenlik nedeniyle Kippo'yu tüm yetkili olarak kullanma gerekliliği problemidir. Kippo web sitesiport 22 üzerinde sanal sunucunun nasıl çalıştırılacağına dair çeşitli çözümler sunmaktadır, ancak en basiti authbind'in uygulamasının kullanmaktır:

```
apt-get install authbind
```

Kippo'yu çalıştırmak için yeni bir tüm yetkili olmayan kullanıcı oluşturulmalıdır:

```
adduser kippo
```

Yeni oluşturduğunuz kullanıcıyı sudo komutunu kullanabilen kullanıcılar listesine eklenmelidir:

```
Visudo
```

Satırı "tüm yetkili" kullanıcı altına eklenmelidir.

```
kippo ALL=(ALL:ALL) ALL
```

Port 22'yi kullanmak için gerekli adımları tamamlamak için:

```
touch /etc/authbind/byport/22
```

```
chown kippo:kippo /etc/authbind/byport/22
```

```
chmod 777 /etc/authbind/byport/22
```

Bu noktada sisteme 'Kippo' kullanıcı olarak girer ve /home dizinine gitmek gerekir. SSH sanal sunucu olarak kullanılacak Kippo'nun en son sürümünü indirmemiz gerekmektedir.

Kippo kullanıcısı olarak giriş yaptıktan sonra /home/kippo dizini içinden kippo'nun alt versiyonlarını indirilmelidir:

```
svn checkout http://kippo.googlecode.com/svn/trunk/  
./kippo
```

Kippo yapılandırma dosyasındaki bağlantı noktası 4567'den 22'ye değiştirilmelidir:

```
mv kippo.cfg.dist kippo.cfg
```

```
vi kippo.cfg
```

Son olarak kippo başlangıç komut dosyası düzenlenmelidir:

```
twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid
```

```
authbind --deep twistd -y kippo.tac -l log/kippo.log --  
pidfile kippo.pid
```

Sanal sunucu başlatılmalıdır ve çalışıp çalışmayacağını test edilmelidir.

```
./start.sh
```

Çalışıyorsa görmek için, bağlantı noktası açıksa ve twistd süreci üzerinde dinliyorsa görmek için netstat komutu kullanılacaktır.

```
sudo netstat -antp
```

Çıktı bunun gibi görünmelidir:

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 22627/python
```

Sanal sunucunuzun çalıştığından emin olunuz ve tüm verilerini bir MySQL veritabanına giriş yapmak için yapılandırmayı başlatamazsınız.

Burada veritabanı ve web sunucusu için bazı ekstra ön koşullara ihtiyaç duyacağız.

```
apt-get install python-mysqldb mysql-server
```

MySQL veritabanında oturum açılmalı, veritabanı ve tablolar oluşturulmalıdır:

```
mysql -u root -p
```

```
shell> CREATE DATABASE kippo;
```

```
shell> GRANT ALL ON kippo.* TO 'kippo'@'localhost'  
IDENTIFIED BY 'choose-your-password';
```

```
shell> exit
```

Şimdi 'kippo' kullanıcıyı kullanarak belirlediğiniz şifreyle henüz erişim sağladığınız MySQL veritabanına yeniden giriş yapmalısınız ve oluşturduğunuz aşağıdaki komutlarla veritabanını doldurmaya başlayabilirsiniz:

```
mysql -u kippo -p
```

```
shell> USE kippo;
```

```
shell> source /home/kippo/kippo/doc/sql/mysql.sql;
```

```
shell> exit
```

Şimdi veritabanınız kippo sonuçları ile doldurulmaya hazırdır.

Eğer kippo çalışıyorsa, lütfen ortadan kaldırın. Çalışıp çalışmadığını kontrol etmek için aşağıdaki komutlar ile yapılacaktır:

```
ps -e | grep twistd
```

Şimdi ortadan kaldırın:

```
killall twistd
```

Şimdi kippo dizinindeki kippo.cfg dosyayı düzenleme veritabanına sonuçlarını giriş yaptırmasını söylemek için kippo yapılandırma dosyasını düzenlemeniz gerekmektedir ve aşağıdaki gibi değiştirilmelidir:

```
vi kippo.cfg

[database_mysql]

host = localhost

database = kippo

username = kippo

password = Kippo-DB-pass
```

Kippo.cfg dosyasını kaydettikten sonra, hala herşeyin çalışıp çalışmadığını görmek için kippo sunucusunu başlatabilirsiniz. Başlangıçta kippo içindeki veritabanı modülünü aradığı görülecektir:

```
./start.sh
```

Görsel web arayüzü için “kippo-graph” kullanıyor olacağız. Kippo-Graph’ın düzgün çalışabilmesi için yüklememiz gereken bazı önkoşullar bulunmaktadır:

```
apt-get install -y libapache2-mod-php5 php5-gd php5-
mysql

/etc/init.d/apache2 restart
```

Şimdi kippo-graph’ın en son sürümünü indirin ve /var/www dizininizin içine çıkartın.

```
wget http://bruteforce.gr/wp-content/uploads/kippo-
graph-0.7.6.tar
```

Paketi çıkartmak için “tar” komutu kullanılmalıdır. ‘kippo-graph’ adında dosya oluşturulacaktır.

```
tar xvf kippo-graph-0.7.6.tar
```

Kippo-graph dosyası içinde, küresel okuma/yazma ayrıcalıklarına sahip olmak için oluşturulan-grafikler dosyası ayarlayın. Aksi takdirde boş grafikler elde edilecektir.

```
chmod 777 generated-graphs
```

Şimdi kippo-graph dosyası içinde config.php dosyası bulunmaktadır, o dosyayı düzenlemeli ve önceden ayarladığınız veritabanı bilgisini sağlamalısınız:

```
vi config.php
```

Kippo örneğinizi ./start.sh ile başlatarak ve ana bilgisayar üzerindeki web sunucusuna göz atabilirsiniz.

```
http://<ip_address>/kippo_graph
```

Grafiklerinin içine veritabanı bilgilerini doldurmak için `GENERATE_THE_KIPPO_GRAPHS()` üzerine tıklanmalıdır.

Şimdi SSH'den port 22'ye "user" ve "123456" şifresiyle giriş yapın.

Daha fazla kullanıcı adı ve şifresi `kippo/data/userdb.txt` dosyası içinde ayarlanabilir.

4.2.2 Büyük veri mimarisinin uygulanması

Bu deney ortamı için 7 adet Raspberry Pi kullanılacaktır. 1 tanesi NameNode/Master olarak, 6 tanesi de DataNode/Slave olarak ayarlanmıştır. Burada bellek ile ilgili biraz problem yaşanması olasıdır çünkü Master Node olarak seçilen RP, HDFS tanımlanmış tüm dosyaların izlerini indexleyeceği için yapılacak iş biraz yavaş çalışması muhtemeldir. Her node 8 GB kapasiteli Class 10 depolama birimine sahiptir. Bu bize toplamda 56 GB'lık bir depolama alanı sağlamaktadır ve bu test ortamımız için yeterlidir. Ayrıca slave node'lardan bir tanesi 500 GB'lık bir diske sahiptir. Böylece toplamda yaklaşık 555 GB'lık bir depolama alanına ulaşılmış oluruz. Büyük veri kapsamında düşündüğümüzde bu alanlar elbette ki küçüktür. Fakat buradaki amacımız kavramın ispatı olduğu için daha sonra büyük boyutlara ölçeklenebilmesi mümkündür. Tüm bunlar göz önünde bulundurulduğunda MapReduce ve Spark için yeterli bir test ortamı sağlanmış olur [48].

Tüm bileşenler Raspbian işletim sistemini kullanmaktadır ve bir kaç bağımlılıkları bulunmaktadır. “/etc/host” dosyası düzenlenerek cluster oluşturulacaktır. Bu ayarlamaların kolaylığı sayesinde daha sonradan yeni node’lar eklenmesi oldukça kolaydır ve farklı sayıda nodelar kullanarak performans analizi yapmamızı da kolaylaştırır. Tüm nodeların depolama alanları dağıtılmış olsa da tek bir dosya sistemi olarak davranacaktır. Master üzerinde yapılacak değişiklikler cluster içerisindeki tüm slave nodelar üzerine gerekli güncellemeleri yapacaktır.

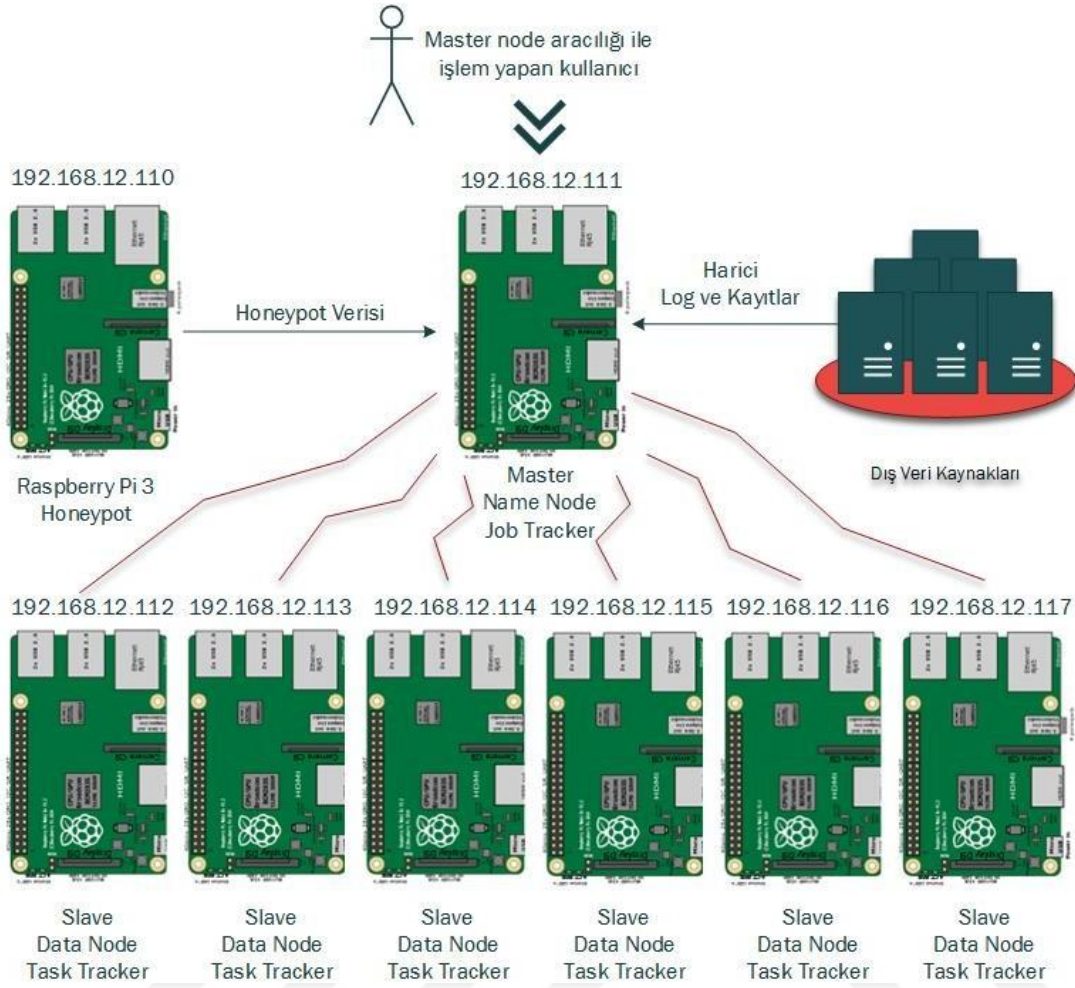
Apache Spark, Scala 2.10 ve üzeri, Python 2.6 ve üzeri ve Java 6 ve üzeri versiyonları tarafından geliştirme desteğine sahiptir. Bu üç dilden herhangi bir tanesi ile geliştirmelerimizi yapabiliriz. Python bu sistem için popüler bir seçenektir çünkü bir çok ileri seviye hazır kütüphane bulunmaktadır. Biz burada öncelikli olarak Scala ile geliştirme yapacağız.

4.2.2.1 Hadoop kurulumu ve yapılandırılması

Halihazırda iki ayrı Hadoop versiyonu bulunmaktadır. Hadoop 1, işlem modeli MapReduce ve batch odaklı işler arasındadır. Hadoop’un bu iki versiyonu arasındaki mimari neredeyse aynıdır. Tek fark Hadoop 1, HDFS dosya sistemini kullanırken, Hadoop 2 içerisinde YARN dosya sistemini kullanılmaktadır.

Hadoop 1 içerisinde, metadataları saklayan yalnızca 1 adet NameNode kullanımına izin vermektedir. Hadoop 2 ise birden çok NameNode kullanımına izin vererek çoklu alanlar üzerinde ölçeklendirme yapılmasına izin vermektedir.

Raspberry Pi kümenize Hadoop’u yapılandırmak için yapılması gerekenler adım adım anlatılmaktadır. (Şekil 4.2)



Şekil 4.3: Deney Ortamının Kurulması

NameNode'unuzu ayarlamak için (192.168.0.111 node1) planladığınız / sonraki düğümlerinizi eklemek için aşağıdaki dosyaları ayarlayın. Sisteme ait dosyaları düzenlemek için kullanıcıyı ya da 'sudo'yu değiştirmeniz gerekir. Linux'un bu çeşidi üzerinde kullanmak için en basit düzenleyici 'nano'dur:

```
$ sudo nano /etc/hosts
```

Kümenizde olmasını planladığınız tüm düğümleri ekleyin:

192.168.0.111 node1 Name Node – Job Tracker

192.168.0.112 node2 Data Node – Task Tracker

192.168.0.113 node3 Data Node – Task Tracker

192.168.0.114 node4 Data Node – Task Tracker

192.168.0.115 node5 Data Node – Task Tracker

192.168.0.116 node6 Data Node – Task Tracker

192.168.0.117 node7 Data Node – Task Tracker

Hadoop özel ayarlar dosyaları, 2.x versiyonları için aşağıdaki dizininde bulunmaktadır.

/opt/hadoop/etc/hadoop

Master ve slave olarak iki çeşit dosya bulunmaktadır. Node1'in ana dosyanızda listelenen tek satır ve tüm düğümlerin (eğer Node1'i DataNode olması için ayarladıysanız) köle dosyası içinde olduğundan emin olun.

NameNode, MicroSD kartını kopyalayın ve yeni bir MicroSD kart üzerine imaj dosyasını çoğaltın. Linux üzerinde bunu yapmak için komutlar aşağıdaki gibidir. MicroSD kartınızı takarken cihazınızın adının değişebileceğine dikkat ediniz:

/dev/sd*

```
$ sudo dd if=/dev/sdb of=node1.img $ sudo dd  
if=node1.img of=/dev/sdb
```

USB depolama aygıtını 'hdfs' etiketi vererek formatlayın. Yeniden cihazı formatlayıp EXT4 dosya sistemini kullanacağız.

Kopyaladığınız MicroSD kartınızı ve USB depolama belleğinizi yeni Raspberry Pi'nize takmalısınız.

Ethernet ağ kablosu & güç kablolarını bağlayın.

the hduser/password credentials kullanılarak kopyalanan Pi'nin IP adresini kullanarak SSH'ı RPi'nin içine atamalısınız:

```
$ ssh hduser@192.168.0.11
```

Raspi-config yardımcı programı kullanarak Raspberry Pi'yi ayarlamanız gerekmektedir. Bu değişiklikleri yapmayı tamamladığınızda yeniden başlatmanız gerekmektedir:

```
$ sudo raspi-config
```

Düğüm ismini değiştirin (Örneğin Node1'den Node2'ye)

SSH sunucusunu etkinleştirildi mi kontrol etmelisiniz.

Bölünmüş bellek 16Mb olmalıdır, buna dikkat edilmelidir.

Çıkan `raspi-config` yeniden başlattıktan sonra, bir kaç saniye bekleyin ve sonra yeniden geriye SSH'a dönün. `/etc/network/interfaces` dosyasını düzenleyin ve IP adresini sonraki uygun değer ile değiştirmelisiniz:

```
$ sudo nano /etc/network/interfaces
```

`/etc/hosts` dosyasının tüm IP adreslerini ve tüm Pi'lerinizin düğüm isimlerini içerip içermediğini kontrol edin:

```
$ cat /etc/hosts
```

RPi'yi yeniden başlatmalısınız:

```
$ sudo reboot
```

Yeni IP adresini kullanarak SSH'ı Pi'ye geri dönün:

```
$ ssh hduser@192.168.0.111
```

Yeni Pi düğüm SSH'ından NameNode'a – soruya 'evet' yanıtını vermelisiniz:

```
$ ssh node1
```

NameNode SSH üzerinde yeni düğüm ismine – soruya 'evet' yanıtını vermelisiniz – ve sonra node2'ye geri dönmelisiniz:

```
$ ssh node2 $ exit
```

/hdfs veri paylaşımı oluşturma. Dizini HDFS bölümü içine değiştirin. /tmp dosyası oluşturunuz ve sahipliğini değiştiriniz:

```
$ sudo mkdir -p /hdfs/tmp
```

```
$ sudo chown -R hduser:hadoop /hdfs $ sudo chmod 750 /hdfs/tmp
```

/etc/fstab dosyasına bir giriş ekleyin ki böylece /hdfs paylaşımı açılışta monte edilmelidir.

```
$ sudo nano /etc/fstab
```

Bununla eşlemesini sağlayın ama /dev/sd* farklılık gösterebilir, dikkat etmelisiniz.

Gerekli /opt/hadoop/etc/hadoop dosya değişimlerini yapın - NameNode + DataNode'tan sadece DataNode'a bu makinenin amacını değiştirmelisiniz.

NameNode (node1) üzerinde DFS ve YARN'ı başlatın:

```
/opt/hadoop/sbin/start-dfs.sh
```

```
/opt/hadoop/sbin/start-yarn.sh
```

DataNode üzerinde, node2, /hdfs/tmp dizinine bir göz atın ve NameNode'un içerisinde bir sürü alt dizinler oluşturulacaktır.

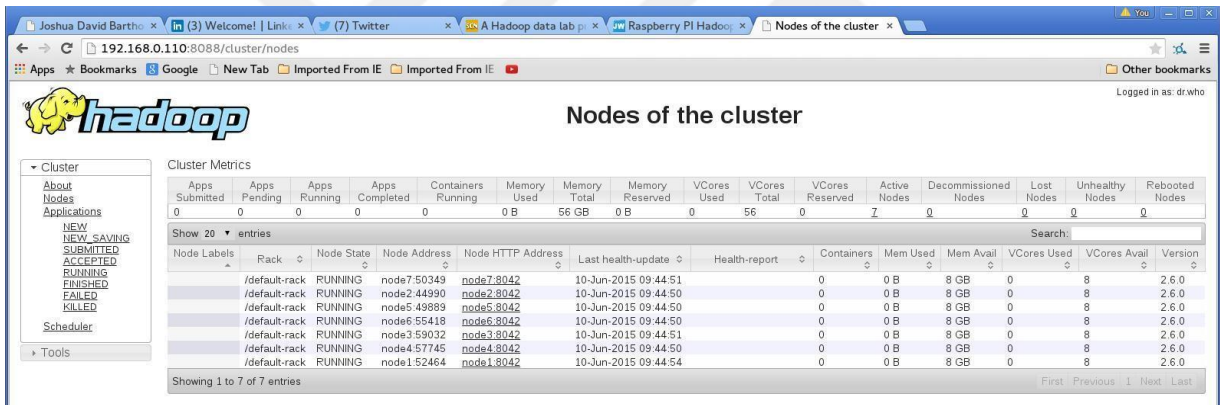
```
$ ls -lrt /hdfs/tmp
```

DataNode görevlerinin 'jps' komutuyla çalıştığını görmek için kontrol edin. (Şekil 4.4)

```
hduser@node7: ~  
File Edit View Search Terminal Help  
  
hduser@node7 ~ $ jps  
2240 DataNode  
2481 Jps  
2338 NodeManager  
hduser@node7 ~ $
```

Şekil 4.4: DataNode Kontrol

NameNode web arayüzüne göz atın: <http://192.168.0.111:8088/> ve yeni düğümü görebildiğinizden emin olun. İşte içinde tüm 7 düğümü gösteren ekran görüntüsü Şekil 4.5 üzerinde görüldüğü gibi olacaktır.

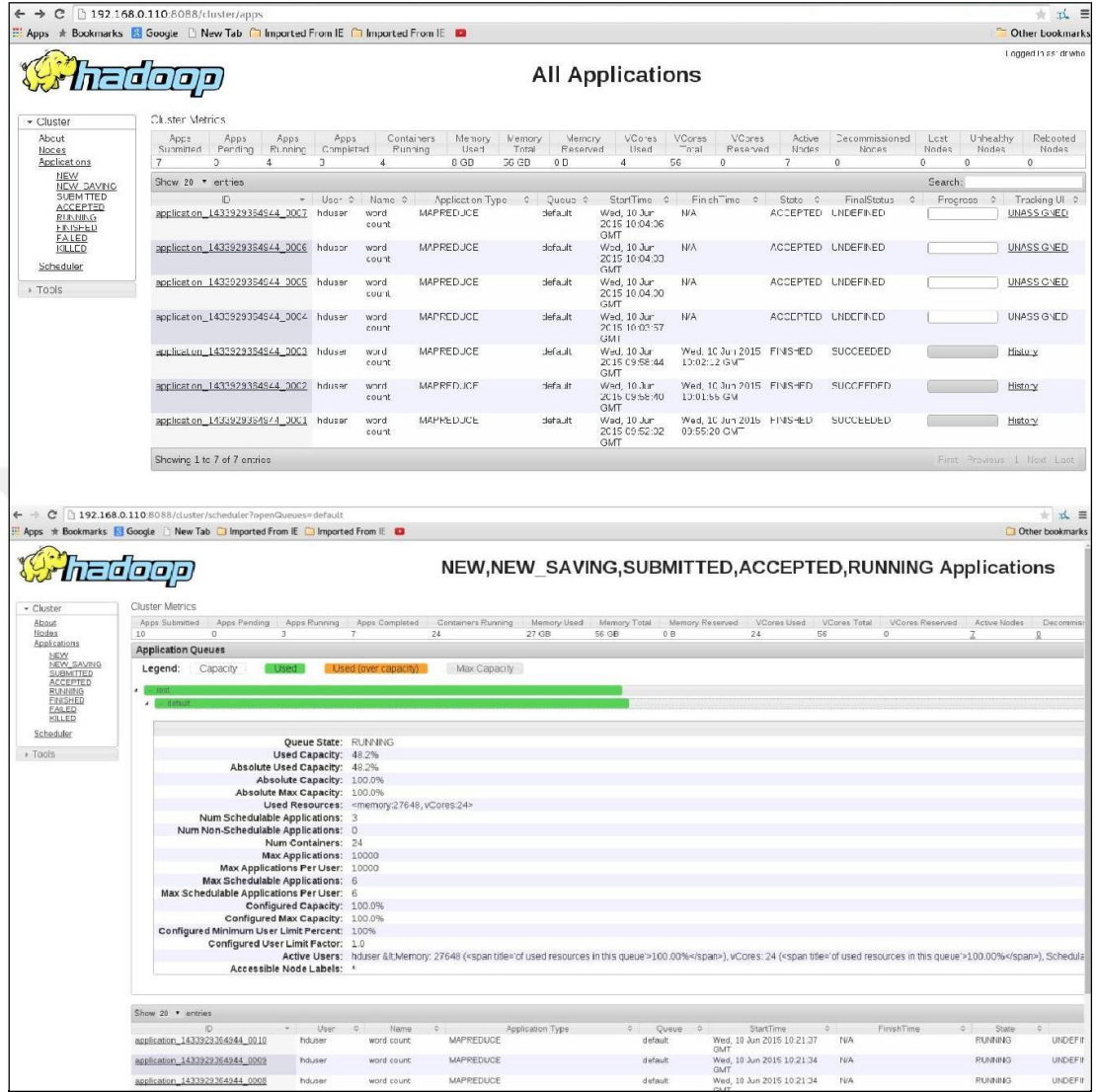


Şekil 4.5: Hadoop Web Arayüzü

Çalışan MapReduce görevleri – talimatları Hadoop 2.x versiyonu için `hdfs` komutlarına bazı ince güncellemeler aynıdır. Kullandığım adımlar şunlardır:

```
$    hadoop    fs    -copyFromLocal    mediumfile.txt  
/mediumfile.txt  
  
$    yarn    jar    /opt/hadoop/share/hadoop/mapreduce/hadoop-  
mapreduce-examples-2.6.0.jar    wordcount    /mediumfile.txt  
/mediumfile-out
```

NameNode web arayüzünden çalışan görevleri izleyebilirsiniz. Test için bazı ekran görüntüleri Şekil 4.6'daki görünmektedir.



Şekil 4.6: Hadoop Web Arayüzü Çalışan Görevler

Birinci DataNode'unuzla mutlu olduğunuzda, ve başka bir tane eklemeye hazırsınızdır, her bir alt düğüm oluşturmak için ilgili olan adımları takip etmeniz gerekmektedir.

Herhangi bir test uygulaması, veri yüklemiş iseniz, tüm düğümlerin sync içinde olduğundan emin olmak için /hdfs/tmp dizinini temizlemek zorundasınız. NameNode (Node1)'tan DFS başladığınız zaman her bir düğüm üzerinde gerekli olan dosya yapısını yeniden oluşturacak.

4.3 Deney Sonuçlarının Analizi

Kippo Honeypot, öncelikle saldırgan olarak görülen davranışları tespit etmek için kullanılmaktadır. Burada honeypot ile yapılmak istenen öncelikli amaç, gerçek kullanıcı verisini toplamak. Sahte kullanıcı verisi üretmek değildir. Bu kapsamda yapılan deney sonunda ilgi çekici sonuçlara ulaşılmıştır. Detayları ayrıntılı olarak incelenecektir.

HDFS dosya sistemini kullanırken aşılması güç bir kaç zorluk ile karşılaşılmasına sebep olmuştur. Bunlardan bir tanesi, sistemin dağıtılmış modda çalışabilir şekilde ayarlanması olmuştur. Aşılması en güç problem ise NameNode'un kaybedilmesi veya çalışmayı durdurması durumunda içerisinde bulunan ikili formattaki bilginin de kayıp olmasıdır. Bu veriyi ancak kayıp NameNode'un kendisi yorumlayıp kullanılabilir hale getirebilir.

Yapılacak deneyde Apache Spark, HDFS dosya sistemini gerçekleştirdiğimiz Raspberry Pi cluster üzerinde MapReduce işlemini gerçekleştirmek diğer bir ifade ile analiz işlemini yapmak üzere kullanılacaktır. Burada kullanılacak PySpark bir Spark türevidir olup python destekli ve java ile zorlu gerçekleşen işlemleri kolaylaştırmak üzere seçilmiştir. Spark kullanmanın bir diğer kazanımı ise küçük veri setleri üzerinde yapılan analizlerin büyük verisetleri ile yapılanlar kadar başarılı sonuçlar vermesidir. Birden fazla kaynaktan sağlanan verinin işlenmesi sırasında yapılacak ölçeklendirmelere uyumu da seçiminde etken sebeplerden bir tanesidir.

Deneyde kullanılan veri HDFS dosya sistemi üzerinde ham formatta saklanacaktır. Geleneksel veritabanlarında ise bu işlemi yapmak için veriyi temizlemeli ve belirli bir formata dönüştürdükten sonra dosya sisteminde saklamamız gerekmektedir. HDFS ve Spark/MapReduce sayesinde veri ham halde saklamak ve formatlama işlemini çalışma zamanında uygulamak oldukça kolaydır. Bu çalışma zamanında formatlama işlevi büyük veri mimarisi ile geleneksel ilişkisel veritabanı mimarisi arasındaki en büyük fark olarak nitelendirilebilir.

4.3.1 Kippo honeypot ile elde edilen bulgular

Kippo Honeypot ile SSH üzerinde çalışarak çok sayıda ilginç bulgular elde edilmiştir. Bunlardan bazıları bu tezin inceleme kapsamı dışındadır. Herhangi bir

iyileştirme yapmadan, varsayılan ayarlar ile toplanan bu veriler üzerinde biraz madencilik yapıldığında saldırı eğilimlerini ve tiplerini anlamakta oldukça faydalı bilgiler elde edileceği değerlendirilmektedir.

Basit Google aramaları ile bu saldırıların amaçlarının ne olduğu ile ilgili bilgilere ulaşmak oldukça kolaydır. Özellikle bazı hazır saldırı araçlarının geride bıraktıkları izler ile yapılacak küçük bir araştırma sizlere saldırının tipi hakkında bilgiyi kolayca ulaşmanızı sağlayacaktır.

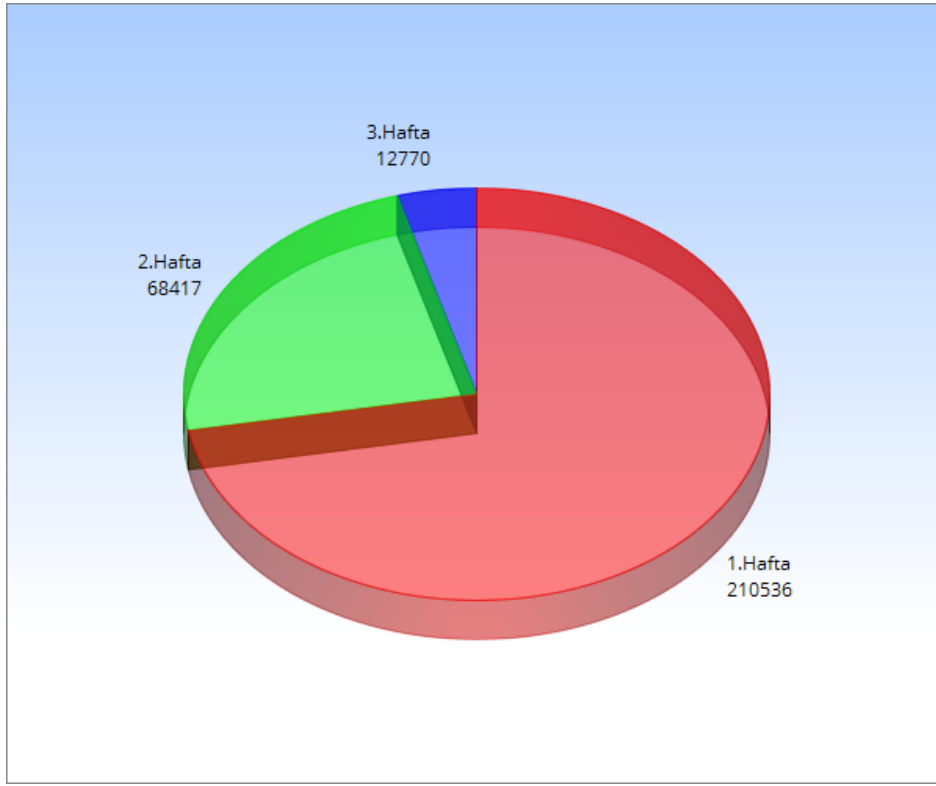
Bir çok saldırı sisteminize IRC Bot kurmaya çalışır. Bunlardan en yaygın olarak kullanılanı is psyBNC programıdır. Saldırganların bu tip programlar kullanmaya çalışmalarının amacı ise IP adreslerini gizlemeye çalışmalarıdır. Aynı zamanda zararlı saldırı girişimlerine ev sahipliği yapan bilgisayarı da perdelemek için kullanılmaktadır. Bu yaygın araçların kurulmaya çalışılması için her saldırı türü için çeşitli farklı yöntemler mevcuttur.

Saldırıların bir çoğu Perl dili ile üretilen zararlı kodlar ile yapılmaktadır. Bu gerçekleşen saldırıların çoğu da Perl'in yapısından kaynaklanan bölünme hataları sebebi ile başarısız sonuçlanmaktadır. Kippo Honeypot, yapabildiğinin en iyisi şekilde bu sahte paketlerin kurulumunu barındırır. Saldırganlar honeypot üzerinde bu şekilde bir bölünme hatası ile karşılaştıklarında genellikle pes ederler. Çünkü adımları yinelemeleri durumunda tespit edilme ihtimallerinin oldukça yüksek olduğunu bilmektedirler. Yapılan yaygın saldırı girişimlerinden bir tanesi de "Bitcoin" madenciliği yapmak için işlemci gücünün kullanılmaya çalışılmasıdır. Saldırgan bu girişimlerini ise yaygın olarak kullanılan httpd ve MySQL çağrıları ile maskeleye çalışmaktadırlar. Bu ve benzeri saldırı girişimlerini örtbas etmek ve tespit edilmesini engellemek için mevcut daha bir çok akıllı yöntem bulunmaktadır.

Kippo honeypot kullanımının bize getirdiği kısıtlardan bir tanesi ise saldırgan ile etkileşimin düşük olmasıdır. Kippo honeypot, linux kabuğunu taklit etmek için yapılandırılmış olsa da düşük bağlantı süresi sebebi ile saldırganların sistemden derhal ayrıldıkları gözlemlenmiştir. Deney sona erdiğinde toplanan kayıtlar değerlendirildiğinde, varsayılan kullanıcı adı ve şifre atanması, tecrübeli saldırganların sistemi terketmesine sebep olduğu değerlendirilmiştir. İlerleyen zamanlarda yapılacak deneylerde daha güçlü kullanıcı adı ve şifrelerinin seçiminin

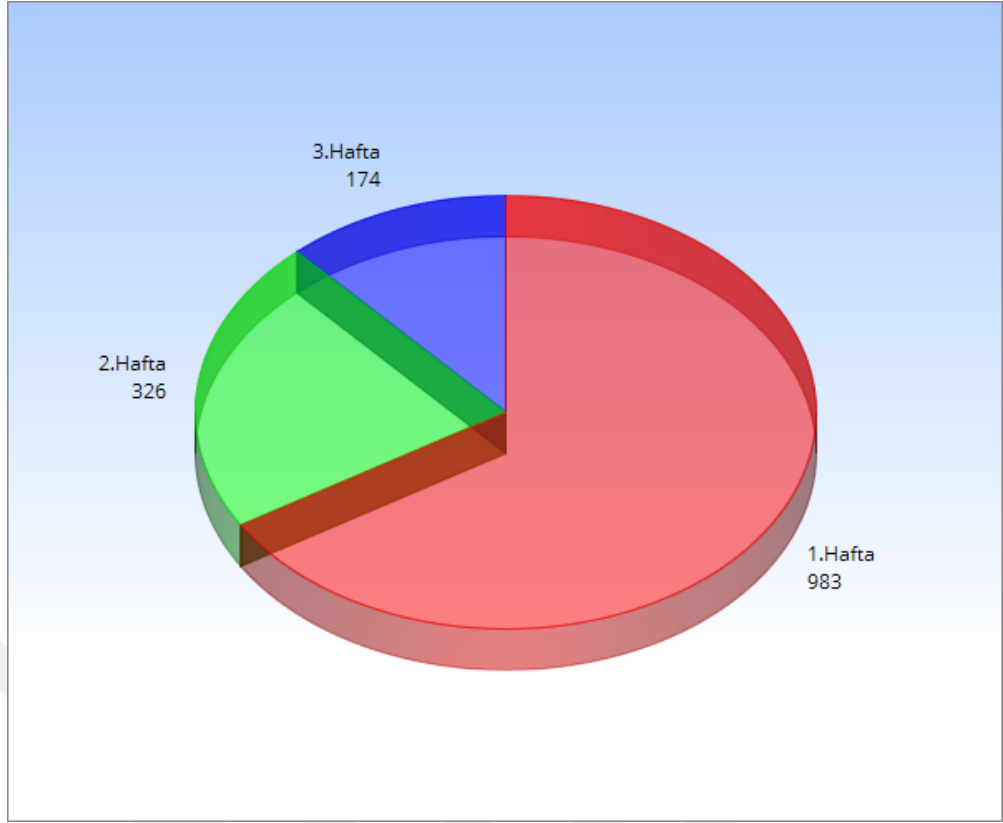
saldırganların honeypot endişelerinin önüne geçeceği ve gerçek sistemin daha iyi taklit edileceği belirlenmiştir. İleride yapılacak çalışmalar için bir diğer tavsiye ise daha sistemi daha iyi modelleyen honeypotların seçimi olabilir. Her geçen gün honeypotların kabiliyetleri artmaktadır. Bu artış, saldırganın ürkerek sistemden ayrılmasının önüne geçecek ve sistem ile daha fazla etkileşimde bulunup, komut çalıştırarak toplayacağımız veri miktarını arttıracak böylece toplanmış veriden çıkartılacak sonuçlar çok daha isabetli olmasına yardımcı olacaktır.

Kippo SSH Honeypot toplamda üç hafta faaliyet göstermiştir. Elde edilen bulgulara ait grafikler aşağıda gösterilmektedir.

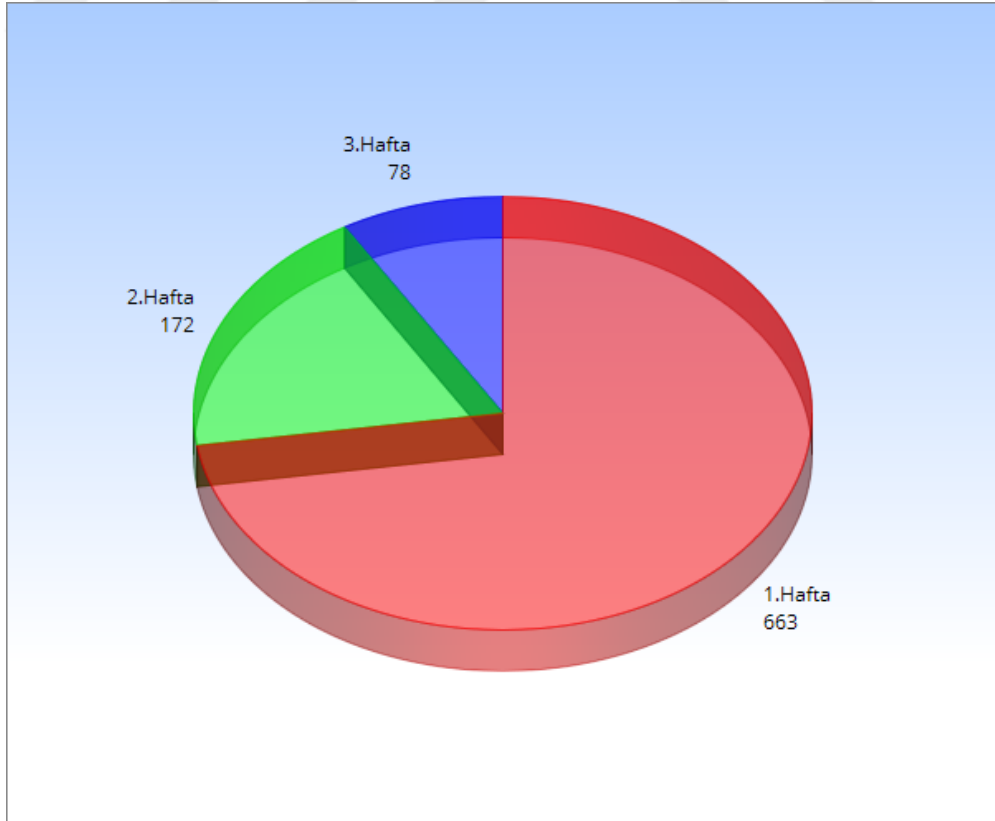


Şekil 4.7: Haftalık Saldırı Girişimi Dağılımı

Şekil 4.7 üç hafta boyunca yapılan testler sonucu sisteme yapılan giriş denemelerinin sayısını göstermektedir. Şekil 4.8 ise haftalık eşsiz IP adresleri üzerinden sistemimize gelen saldırganların dağılımları görülmektedir. Şekil 4.9’da ise sisteme yapılan saldırıların haftalık başarılı giriş sayıları verilmiştir.



Şekil 4.8: Haftalık Ayrı IP adreslerinden Gelen Saldırı Girişimi Dağılımı



Şekil 4.9: Honeypot Üzerine Başarılı Giriş Sayısı

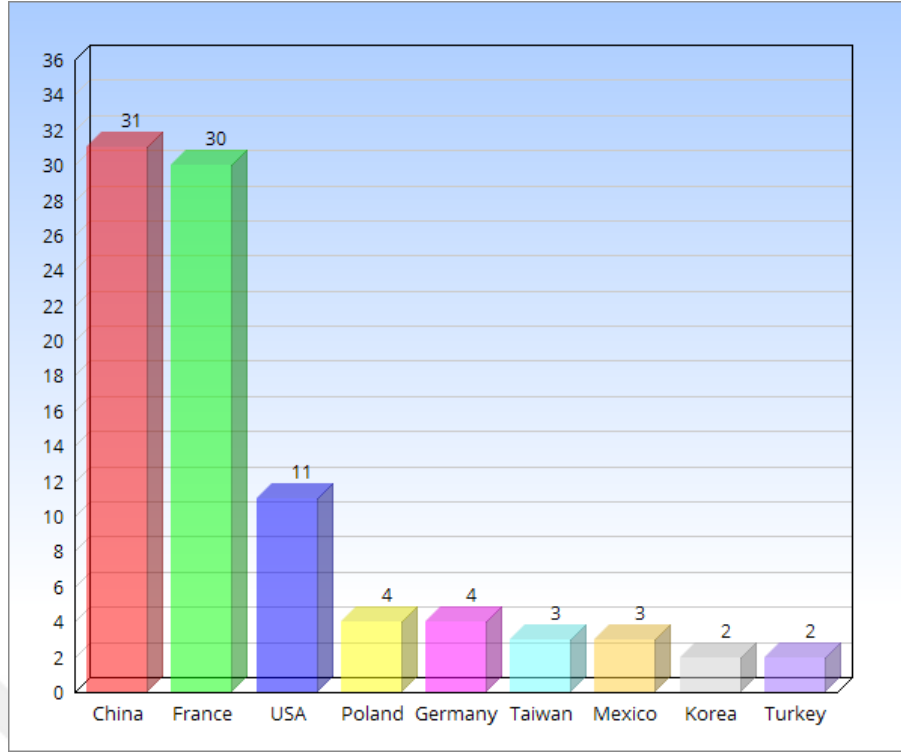
Çizelge 4.3: Başarılı giriş sayısı ve oranları

Hafta	Giriş Sayısı	Oran
1	663	Tüm giriş sayısına oranı: %0.314
2	172	Tüm giriş sayısına oranı: %0.251
3	78	Tüm giriş sayısına oranı: %0.610

Çizelge 4.3'te görüleceği üzere sisteme yapılan saldırıların başarı oranları oldukça düşüktür. Burada honeypot kullanıcı adı ve şifresinin daha kolay ve basit seçilmesi bu başarı değerlerini arttıracaktır. Çizelge 4.4 üzerinde en çok saldırı yapan 20 IP adresi ve bu IP adreslerinden gelen toplam saldırı sayısı görülmektedir. Çizelge 4.4'te dikkatimizi çeken bir durum söz konusudur. 176.31.85.59 IP adresli saldırgan ilk iki hafta boyunca sisteme yoğun bir şekilde giriş yapmayı denemiştir. Fakat 3.haftada sistem ile herhangi bir etkileşimde bulunmamıştır. Başarılı giriş yapamadığı da elimizdeki bilgilerde mevcut olduğuna göre bu adresten muhtemelen dağıtık saldırıya (DDOS) maruz kalmış olduğumuz sonucu çıkartılmıştır. Bu kadar çok girişim denemesi ancak sistemi yavaşlatma veya erişimini durdurma hedefi ile gerçekleştirilmektedir.

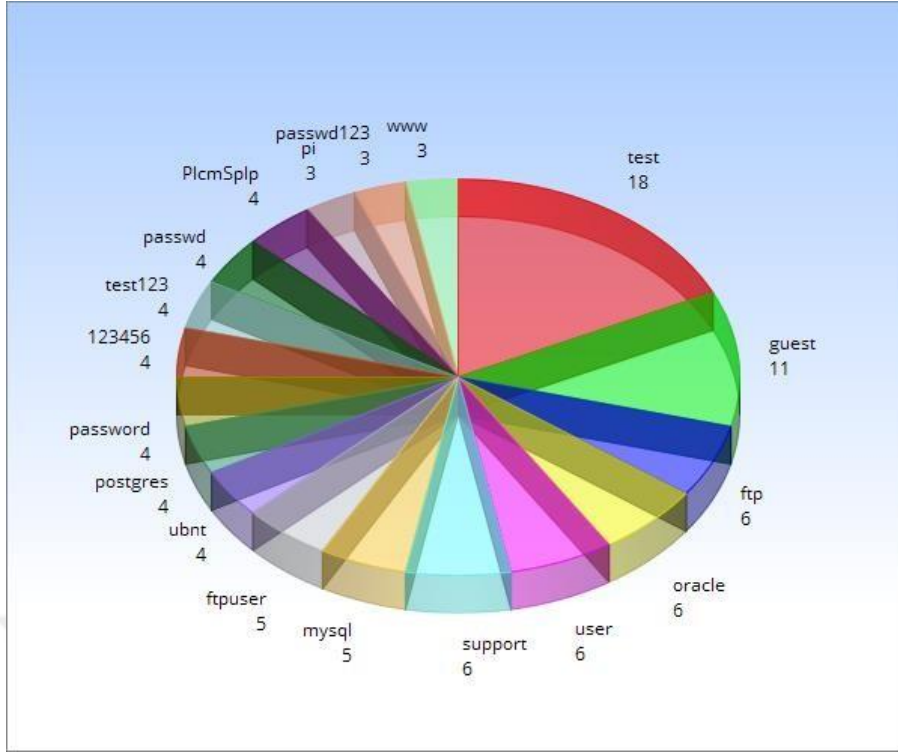
Çizelge 4.4: En Sık Saldırı Yapan 20 IP Adresi

1.Hafta		2.Hafta		3.Hafta	
IP Adresi	Saldırı Sayısı	IP Adresi	Saldırı Sayısı	IP Adresi	Saldırı Sayısı
176.31.85.59	25927	176.31.85.59	28565	94.127.2.85	1605
223.4.94.69	13322	173.224.221.197	5310	211.191.168.180	1604
223.4.241.4	10771	46.105.189.201	4703	188.132.206.233	974
74.63.238.68	9507	5.9.200.90	3282	121.9.221.102	974
223.4.171.195	8434	203.93.215.101	2238	211.110.44.113	771
223.4.175.77	6616	94.127.2.85	1605	101.44.1.135	725
59.125.208.244	6097	188.132.206.233	1461	174.140.167.238	603
220.172.107.211	6048	121.9.221.102	974	218.206.117.57	439
201.116.36.180	5777	58.240.17.250	961	74.84.89.106	286
122.158.235.33	4848	114.112.21.15	953	114.80.202.30	275
116.229.239.242	4772	218.206.117.57	854	84.22.32.2	249
223.4.182.71	3535	91.102.16.156	798	31.40.76.183	222
200.214.143.4	3520	166.78.27.149	725	220.161.148.178	216
174.140.167.238	3442	208.115.207.140	627	112.231.23.68	201
188.125.103.177	3211	128.140.1.19	597	124.160.194.27	174
180.168.83.54	3094	220.161.148.178	593	192.157.220.84	169
5.9.200.90	3075	120.101.5.209	546	62.193.192.167	163
223.4.145.38	2888	218.77.178.3	545	63.137.151.184	163
203.69.139.179	2664	59.151.5.236	511	173.208.218.70	161
218.213.234.232	2519	211.191.168.180	449	125.39.8.142	141



Şekil 4.10: En Sık Saldırı Yapan 20 Lokasyon

Şekil 4.12 bulunan grafik Kippo Honeypot üzerindeki SSH sistemine yapılan yetkisiz girişlerde en çok kullanılan 20 kullanıcı adını göstermektedir. Toplam 291.723 adet saldırı girişimi tespit edilmiştir. Şekil 4.10 üzerinde en sık saldırı yapılan 20 lokasyon gösterilmektedir. Çizelge 4.5 üzerinde en çok denenen 20 şifre ve deneme sayılarını göstermektedir.



Şekil 4.12: En Çok Denenen 20 Kullanıcı Adı

Tabloda belirtilenler dışında giriş denemelerinin yarısında “admin” ve “root” kullanıcı adı ile deneme yapılmıştır. “pi” kullanıcı adı ise bu girişimlerden yalnızca %3’ünde kullanılmıştır.

Çizelge 4.5 : En Çok Denenen 20 Şifre

1.Hafta		2.Hafta		3.Hafta	
Denenen Şifre	Deneme Sayısı	Denenen Şifre	Deneme Sayısı	Denenen Şifre	Deneme Sayısı
123456	7350	123456	1632	123456	597
password	6204	password	803	password	157
1234	3481	changeme	539	1234	125
changeme	2757	test	421	12345	101
abc123	2592	1234	369	1	90
12345	2414	123	361	123	89
test	2300	12345	311	test	84
123	1904	qwerty	291	root	80
qwerty	1342	admin	279	abc123	70
P@sswOrd	1102	test123	276	qwerty	68

Çizelge 4.6 üzerinde sisteme başarılı bir şekilde giriş yapan saldırganların kullandığı 10 komut gösterilmekte ve kullanım sayılarını vermektedir. Çizelge 4.7 ise honeypot

üzerine indirilen dosyaların listesini göstermekte ve kaç adet indirildikleri ile ilgili bilgileri göstermektedir.

Çizelge 4.6: En Çok Denenen 10 Komut

1.Hafta		2.Hafta		3.Hafta	
Denenen Komut	Deneme Sayısı	Denenen Komut	Deneme Sayısı	Denenen Komut	Deneme Sayısı
ls	110	exit	3	ls	31
w	70	ls al	2	cd ..	14
exit	54	ls	1	exit	11
ls a	43	cat /proc/cpuinfo	1	ls al	6
cd ..	25	cat /etc/issue	1	cd /	4
uname a	24			ls l	4
rm rf .bash_history	24			ls lah	4
echo "WinSCP: this is end of file:0"	23			clear	3
ls al	21			ls all	3
chmod +x *	21			ps	3

Çizelge 4.7: Honeypot Üzerine İndirilen Zararlı Dosyalar ve Sayıları

İndirilen Zararlı Dosya	İndirme Sayısı
http://www.uni.w.com/flood.tar	5
http://download.microsoft.com/download/win2000platform/SP3/NT5/EN_US/W2Ksp3.exe	2
http://iceagewar.hi2.ro/psyBETA.tgz	2
mariusica.altervista.org/lib.tt	2
www.xdutzx.altervista.org/udp.pl	2
dinte.webng.com/unic.tgz	1
doiaru.clan.su/clean.tgz	1
dracdeinger.altervista.org/mm.tgz	1
dracdeinger.altervista.org/rk.tgz	1
ftp://mestahost:mestahost2013@188.120.237.29/Smith.tgz	1
http://183.60.202.196:521/hello.exe	1
http://198.2.192.204:22/disknyp	1
http://31.170.164.152/abc/fast.jpg	1
http://abc.hol.es/abc/fast.pdf	1
http://cachefly.cachefly.net/100mb.test	1
http://d4ng3r.byethost6.com/stuff/FDSK.zip	1
http://geocities.ws/map/perl.pdf	1
http://hacker.linux.mi.clan.su/gosh.tgz	1
http://pinky.clan.su/sniff/rk.jpg	1
http://psybnc.at/download/beta/psyBNC_2.3.2_7.tar.gz	1
http://rapk1d.webs.com/100mb.tgz	1
http://rapk1d.webs.com/key.pl	1
http://root.arhive.clan.su/scanner/debian.jpg	1
http://sacele.ucoz.ro/Mech/Mech.tgz	1
http://system.comule.com/bnc/psybnc.linux.ro.tgz	1
http://tehgame.altervista.org/rk2012.tgz	1
http://wOrmer.altervista.org/bot.txt	1
http://wmbro.webs.com/x.tgz	1
linuxtrade.us/	1
LinuxTrade.Us/Arhive/Others/pico.tgz	1
linuxtrade.us/authorized.keys	1
linuxtrade.us/mech.tgz	1
linuxtrade.us/mh.tgz	1
linuxtrade.us/x.tgz	1
marius.altervista.org/scanner/HaitaTeam.jpg	1
smithboy.webs.com/emech/psybnc.gz	1
wOrmer.altervista.org/arhiva.tgz	1
wbo.clan.su/chx/ssh.gz	1
www.parazit.eu/p/psy.tgz	1
www.unix.com/flood.tar	1
xx.ucoz.com/xx.pdf	1
zuby.go.ro/boti.tgz	1

"wget" ve "ftp" ile indirilen dosya adresleri, indirme sayısı ve inen dosyalar ŞEKİL üzerinde görülmektedir. İndirilen dosyalar analiz edildiğinde "rk.tgz" dosyası rootkit,

“flood.tar” dosyası DDOS, “psyBNC.tgz” dosyası ise IRC aracı olduğu tespit edilmiştir.

Saldırı girişimlerinde kullanıcı adından daha önemli olan sisteme gönderilen komutlardır. Aşağıdaki grafik ise sisteme yapılan girişimlerde kullanılan komutların dağılımlarını göstermektedir. Grafikten de görüleceği üzere, en çok kullanılan iki adet komut “ls” ve “wget” komutlarıdır. Burada karşımıza çıkan davranış, saldırganın sistemden bir web sunucusuna bağlanmaya çalışması ve ardından zararlı bir yazılımı indirmeye çalışmasıdır. Bağlanmaya çalışılan sunucular genellikle bir alan adı üzerinden değil doğrudan bir IP adresi üzerinden gerçekleştirilmeye çalışılmaktadır. “wget” komutu ile bağlanmaya çalışılan bu web sunucuları üzerinden dosya indirme işlemini gerçekleştirmek için kullanılmaktadır. Geleneksel saldırı maskeleyme tekniklerini kullanan saldırganlar komutları argümansız olarak kullanmaktadırlar [58]. Çünkü bu argümanların kullanılması halinde sistem indirilen yazılım hakkında daha fazla bilgiye ulaşmamıza yardımcı olur. Bu durumdan uzak durmak isteyen saldırganlar komutları yalın ve amaca yönelik olarak kullanmaktadırlar.

4.3.2 Toplanan honeypot verisinin temizlenmesi

Toplanan bilgiler ışığında bir analiz yapacak olursak, kolay kullanıcı adı ve şifre seçimi saldırganı tedirgin ederek sistemden ayrılmasına sebep olduğu tespit edilmiştir. Bazı saldırganlar sisteme sızmalarına rağmen yalnızca bir yada iki adet komut çalıştırarak sistemi terk etmişlerdir. Ancak saldırganların çoğu sistem ile çok fazla etkileşimde bulunmadan ayrılmalarına rağmen elimizde analiz etmeye yetecek kadar kayıt bırakmışlardır.

Kippo Honeypot verisi yarı düzenli bir yapıdadır. Bu toplanan verileri MySQL veritabanında saklayacak bir çok proje geliştirilmiştir fakat honeypot ağır bir saldırı altında iken bu işlemi gerçekleştirmesi açısından bir çok performans problemi ortaya çıkmaktadır. Burada Spark tercihimizi neden yaptığımız ile ilgili de bir ipucu elde etmiş olacaksınız. Spark, düzensiz ve yarı düzensiz veriler üzerinde oldukça verimli çalışmaktadırlar. Spark-shell ve PySpark interaktif bir şekilde konsol üzerinden dağıtık veri setleri üzerinde hızlı bir şekilde sorgular yapmaya imkan verirler. Düzenli ifadeler (Regex) toplanan veriler üzerinden kullanılabilir formatta bilgiler

elde etmeye imkan verirler. “wget” ve “curl” honeypot üzerinde saldırganlar tarafından kullanılan yaygın komutlardır. Bu komutlar IP adresini kayıt etmeden yalnızca zararlı yazılımın adı verilerek indirilmesini sağlarlar.

Toplanacak honeypot verisinin önündeki tek kısıtlama kullanılacak honeypotun tipine bağlıdır. Kippo’nun tercih edilmesindeki en büyük sebep hakkında ayrıntılı belgelendirmesinin bulunması, kullanımı ile ilgili bir çok örneğin bulunması ve en önemlisi üzerinde çalıştıracağımız donanım olan Raspberry Pi ile uyumlu ve sorunsuz çalışıyor olmasıdır. Honeypotların kurumlar tarafından bilgi güvenliğinde kullanılmasının zamanla artmaktadır. Burada üzerinde durulması gereken en önemli konu bu kullanım sırasında honeypotun kimliğinin iyi bir şekilde gizlenmesi ve sistemin parçası gerçek bir bilgisayar gibi davranmasının sağlanmasıdır.

Sistemden veri toplanırken dikkat edilmesi gereken, saldırgan tarafından sisteme verilen komutların detayları ile (argümanlar, set ettikleri bayraklar vb.) saklanmaya çalışılmasıdır.

Scala dili ile komutların standart formata nasıl sokulacağını açıklamaktadır (EK-A).

Veri temizliği sırasında, toplanan verinin Naive Bayes formülünde kullanacağımız belirli bir formata dönüştürülmesi önemlidir (EK-C). Naive Bayes kuralının uygulanması oldukça dar bir kapsamda yapılmıştır çünkü burada elimizde olan veri seti oldukça küçüktür (EK-B).

4.3.3 Örnek saldırı senaryosunun değerlendirilmesi

Toplanan verinin %80’i sistemi eğitmek için kullanılacaktır. Kalan %20’lik dilimdeki veri ise test amaçlı kullanılacaktır. Komutlar parametreleri ile kullanıldığında doğruluk oranı %80, parametresiz kullanıldığında ise doğruluk oranı %30’lara düşmektedir. Komutların parametreleri, saldırganların sistemi ne şekilde kullandıkları ile ilgili değerli bilgiler vermektedirler.

7 Node’tan oluşan Apache Spark kümesinin 500 Megabaytlık bir veri setini analizi 1 saniyeden az sürmektedir. Spark kümesindeki düğüm sayısını ölçeklendirmek mümkündür. Fakat bu nedenle saldırgan verisini işlerken kabul edilebilir miktarda bir yanıt gecikmesi yaşanabilir.

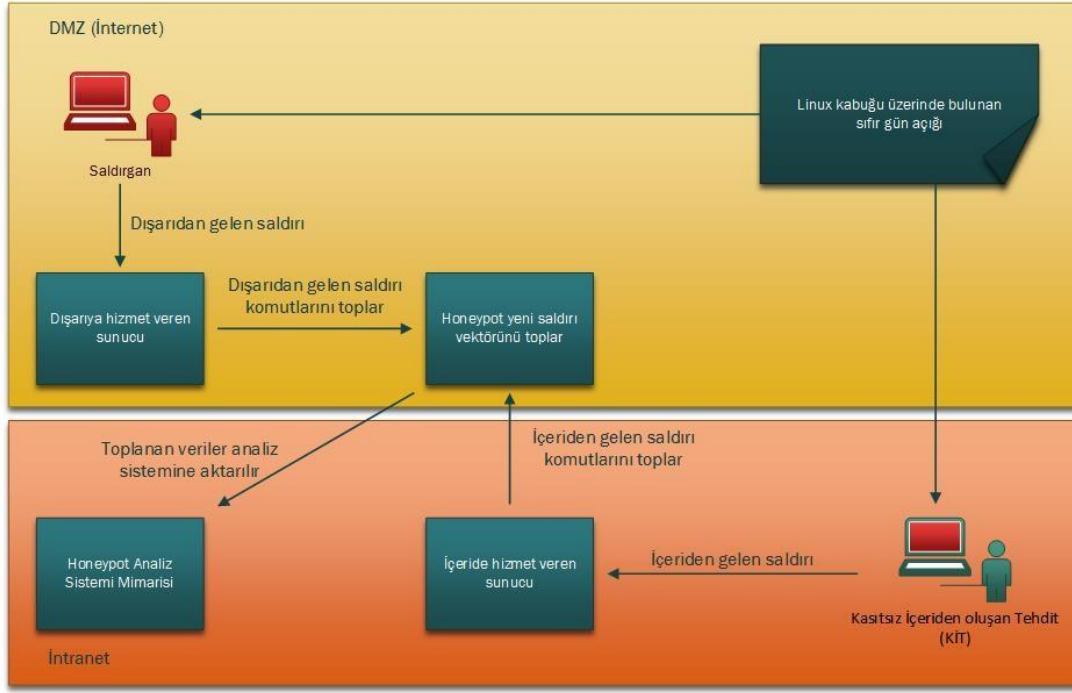
Sistem normal kullanıcının verdiği komutların geçmişini kayıt altına alır. Kullanıcı kendi kullanıcı geçmişinde kullandığının dışında bir komut veya parametre kullanırsa zararlı olarak değerlendirilecektir.

Topluluk davranış analizi için yalnızca tek bir yöntem yoktur. Tüm ağ ve uygulamalardan elde edilen bilgiler kullanılarak uygun bir analiz ortaya çıkartılabilir. Bu verilerin de dahil edilmesi ile doğruluk değeri daha yüksek bir sonuç elde edilmesine yardımcı olur. Örneğin sıradışı artan bir ağ trafiği mevcut ise bu durumu da dikkate alarak yalnızca honeypot üzerindeki etkileşim bilgi değil, davranış bilgisi de değerlendirilerek çok daha yüksek oranda zararlı kullanıcı davranışı olduğu tespit edilebilir.

Gerçek zamanlı analiz sunulan çözümün yalnızca bir bölümünü oluşturur. Analiz geçmiş kullanıcı verisi üzerinde çalışılırken, yeni saldırı vektörleri de sürekli güncellenmektedir. HDFS üzerinde ham olarak saklanan veri değiştirilmemiştir. Gerekli olan veri program ile uygun formata dönüştürülerek analiz için kullanıma hazır hale getirilir. Yeni bir saldırı vektörü tanımlandığında, bu saldırı türünün geçmiş davranışlar üzerinde araştırması yapılır. Bu sayede yeni tespit edilen saldırıya daha önceden maruz kalınıp kalınmadığı bilgisine ulaşılabilir.

Bu örnek senaryoda kabuk sistemi üzerinde yeni bir zafiyet tespit edildiğini varsayar. Saldırganlar bu zafiyeti istismar etmek için IP adresleri üzerinden büyük miktarda bir trafik yaratırlar [54]. Honeypot normal kullanıcılar tarafından kullanılmayan bir sistemdir. Bu nedenle bu sistem üzerindeki herhangi bir aktivite saldırganlar tarafından gerçekleştirildiği anlamına gelir [26]. Honeypot üzerinden toplanan komut imza verileri analiz edilmek üzere analiz altyapısına gönderilir.

Bu sırada içeriden KİT olarak tanımlanan kullanıcı, bilerek bırakılan zafiyeti istismar etmek için honeypot üzerindeki benzer komutlar kullanacaktır. Bu işlemler sonucunda KİT kullanıcısı sisteme giriş yetkisine sahip olacaktır. KİT kullanıcısının burada gerçekleştirdiği aktiviteler kayıt altına alınarak analiz edilmek üzere analiz sistemine aktarılacaktır. EK G'de açıklanan komut ayrıştırıcı sayesinde KİT kullanıcısının komutları ayrıştırılarak analiz edilecektir. Bu kullanıcı tarafından daha önceden kullandığı komutlar da zararlı olduğu tespit edilmemiş olsalar da inceleme altına alınır.



Şekil 4.13: Örnek Saldırı Senaryosu

Örnek saldırı senaryosu Şekil 4.13 üzerinde görülmektedir. Bu senaryodan bir çok sonuç çıkartılabilir. Bunlardan birincisi, elde edilen saldırı verilerinden belirlenen seviye ve ağırlıklara göre içerideki kullanıcı hesaplarını denetlemeleri ve alarmlar oluşturmaları sağlanabilir. İkincisi, belirlenen süre içerisinde gerçekleştirilen komut sayısından belirli hesaplar gözlem altına alınabilir. Üçüncüsü, saldırgan tutuma benzer tutum sergileyen kullanıcının şifresi sıfırlanarak destek ile iletişime geçmesi sağlanabilir. Bu durum eğer kullanıcı farkında olmadan bir saldırıya zemin hazırlıyor ise kayıplar engellenmiş olur. Tüm bu sonuçların herbiri birer yol ayrımıdır fakat tüm sonuçlar değerlendirilmelidir. Bu tez kapsamında uyarlanacak Naive Bayes yöntemi diğer algoritmalar kadar zengin içerikli değildir. Örneğin, destek vektör makineleri verilebilir [11]. Dışarıdan yapılan saldırılarda doğrudan kullanıcı hesaplarını kilitleyecek kadar güçlü değildir.

4.4 Deneyi önceki çalışmalardan ayıran yönleri

Bu deneyin oluşturulması 2013 yılında yayınlanan “Future directions for behavioral information security research” isimli yayından esinlenerek gerçekleştirilmiştir. Bu yazıda da ele alındığı gibi, davranış analizi için ileride kullanılacak üç adet temel prensip belirlenmiştir. İçeriden oluşan saldırgan davranış ile kullanıcı hatasından

kaynaklanan yanlış davranışların birbirinden ayırt edilmesi, bilgi güvenliği ile ilgili verilerin toplanması ve ölçüm yöntemlerinin geliştirilmesi ve normal kullanıcı davranışının sürekli gözlenmesidir [13].

“Honeypots: Catching the Insider Threat”, isimli çalışmada ise honeypot ile saldırgan verisinin toplanması üzerine yazılmıştır. Bu çalışma, içeriden gelecek saldırılara odaklanmak yerine güvenlik duvarının önüne konumlandırılmış bir honeypot ile analiz yaparak saldırganları uzak tutmak üzerine kurgulanmıştır [54]. Bu iki makaledeki çalışmaları bir araya getirerek ulaştığımız sonuç: saldırgan davranışını normal kullanıcı davranışından ayırt etmek için yalnızca saldırgan verisinin incelenmesi ile ulaşılmaya çalışıldığı olmuştur.

Hadoop ve makine öğrenmesi teknikleri önceki çalışmalarda da otomatik saldırı tespit sistemi tasarımlarında kullanılmıştır. Bu çalışmalardan bir tanesi honeypot üzerinde otomatik analiz gerçekleştirerek ağa sızan botları yakalamak üzerine yapılan çalışmadır [62]. Bir diğer çalışma ise makine öğrenmesi tekniği ile zararlı yazılımları tespit etmeyi hedeflemiştir [61]. Her iki sistem de Hadoop teknolojisinin davranışsal analiz için kullanılması için araştırmacıları teşvik etmiştir.

Davranışsal analiz, bilgi güvenliği konusunda üzerinde sık çalışılan konulardan bir tanesidir. Saldırı tespit sistemleri ağa giriş modellerinden yararlanarak kuraldışı davranışları tespit etmeye çalışmaktadır [53]. Bu yaklaşımın önündeki en büyük sakınca STS’lerin yalnızca uygulama ve sistemi kullanan kullanıcıların ağ trafiğindeki kullanıcı modelini değerlendirerek işlem yapmasıdır. Uygulama kullanım profilleri daha güçlü veri setleri elde etmeye imkan sağlar fakat kullanıcı etkileşimi ile ilgili veri toplamada herhangi bir standart bulunmamaktadır.

Linux sistemlerde kabuk üzerinde gerçekleştirilen işlemlerin kayıtları belirli bir standartta saklanmaktadır. Davranış analizi hakkında yapılan sayısız çalışmada linux kabuğundaki kullanıcı profilleri sıradışı davranışları tespit etmek için kullanılmaktadır. Maskelenmiş saldırıları tespit etmek için bir çok çeşit algoritma bulunmaktadır (Örneğin Naive Bayes ve Hidden Markov Models) [59]. Bu çalışmaların bir kaç sakıncaları vardır. Kullanıcı verileri normal davranışın modellenmesi amacıyla öncelikle güvenilir bir profil oluşturmak için toplanır. Bu güvenilir veri elde edilirken kullanıcının gerçekten normal bir davranışta olduğu garanti altına

alınmalıdır. Bunun için kilit kullanıcı profilleri değerlendirilebilir. Bu davranış baz alınarak diğer davranışlardaki sapmalarda bir problem olup olmadığı denetlenebilir [60].

Tez kapsamında gerçekleştirdiğimiz deney daha önceki çalışmalardan farklıdır. Çünkü burada saldırgan olarak nitelenen davranış verisi saldırgan olduğundan emin olduğumuz honeypot verisi üzerinden sağlanmaktadır [54,56]. Burada elde edilen saldırgan kullanıcı verisi bilinen saldırı yöntemleri ile karşılaştırılıp sınıflandırmaya imkan vermektedir. Eğer kullanıcı bir sıra eylemi normal davranışından dışında gerçekleştirmeye çalışır ise ve bu sıradışılık daha önceden belirlenen saldırgan eylemin tutumu ile özdeşleşiyorsa, bu kullanıcının KİT kullanıcısı olduğunun tespitinin yapılmasını sağlamış olur.

5. SONUÇ VE DEĞERLENDİRME

Uzmanlar, bilgi güvenliği üzerine yapılacak çalışmalarda potansiyel güvenlik ihlallerini tespit etmek için heterojen veri setleri üzerinde büyük veri analitiği mimarisinin getirdiği faydalardan yararlanabilirler. Saldırıları her geçen gün daha karmaşık ve çok yönlü hale gelmektedir. Bu nedenle tespit ve analiz sistemlerinin bu saldırıları tam olarak tespit etmeleri için kurumun ağında gerçekleşen tüm aktiviteleri yakından takip etmeleri gerekmektedir.

Bu tez kapsamında gerçekleştirilen deney ile büyük veri mimarisi ve ağ üzerindeki davranışlar analiz edilerek potansiyel saldırgan kullanıcıların tespit edilmelerine olanak sağlayacak bir çözüm geliştirilmeye çalışılmıştır. Honeypot üzerinden toplanan veri ile gerçek zamanlı davranışsal analiz yapılabileceği kanıtlanmıştır. Honeypot üzerinden ve gerçek zamanlı sistem üzerinden davranış analizinin birlikte yapılmasının iki büyük getirisi bulunmaktadır. Birincisi güvenlik uzmanlarının normal kullanıcının zararlı eylemlere başladığında bunu tespit edip kıyaslayacak veriye sahip olmasıdır. İkincisi, saldırgan eylemlerinin analizi ile sıradışı yeni bir yöntemin kullanılmaya başlanması durumunda yeni tip saldırıları tespit etmelerinin kolaylaşacak olmasıdır.

5.1 Karşılaşılan Problemler ve Kısıtlar

Bu tezin yazımında izlediğimiz yol itibari ile bir çok kısıtı barındırmaktadır. Bunlardan bir tanesi yalnızca SSH kabuğu üzerinden gerçekleştirilecek saldırılara odaklanmasıdır. Daha fazla kaynak üzerinden toplanılacak bilgilerin bir araya getirilip analiz edilmesi ile saldırgan tespitindeki doğruluk ve başarının artacağı değerlendirilmektedir. Test ortamında honeypot üzerinde 2 milyona yakın olay yakalanmasına rağmen yalnızca komutlar ve yapıları incelenmiştir. Honeypot üzerinde yakalanan komutlar bize saldırgan kullanıcı profili ile ilgili detaylı bilgilere ulaşmamızı sağlamıştır. Burada gözlenen en önemli nokta ise saldırgan profili ile normal kullanıcı profillerinin birbirlerinden tamamen farklı olduklarıdır. Bu ayrım her ne kadar iki kullanıcıyı ayırmada kolaylık sağlayacakmış gibi görünse de normal

kullanımın dışına çıkan kullanıcı ile bu saldırgan verisini özdeşleştirmek oldukça fazla işlem yapılması gereken bir süreçtir.

Tez kapsamında kullanılan Kippo honeypot açık kaynaklı bir yazılımdır. Bilgi güvenliği kullanıcı topluluğu tarafından geliştiriliyor olması bir çok noktada güçlü yeteneklere ve çözümlere sahip olmasını beraberinde getirmektedir. Geniş dökümantasyonunun bulunması, kullanımı ve yapılandırılması konusunda sorun yaşanmamasına yardımcı olmuştur. Ayarların gerçekleştirilmesi aşamasında linux kabuğunu tamamen taklit etmesine yetecek yetenekler eklenmesine imkan sağlamaktadır. Gelecekteki bilgi güvenliği araştırma mimarisinin, dinamik olarak maskelenmiş saldırıları gerçek sunuculardan, honeypot sunuculara aktaracak kadar yetenekli olacakları öngörülmektedir. Böylece potansiyel saldırıların otomatik olarak doğru veya yanlış tespit edildiğine karar verilmesinde etkili olacaktır.

5.2 Çalışmanın Kullanım Alanları

Bilgi güvenliği çalışmalarının yüksek maliyet ortaya çıkartacağını düşünen kurumlar tez kapsamında ortaya konan sistem ile ağ ve bilgi kaynaklarını saldırganlardan koruyabilecek altyapıyı kolayca bilgi sistemlerinde kullanmaya başlayabilirler. Böylece gerçekleştirilecek olası saldırı ve istismar durumlarından kaçınmaları mümkün olacaktır.

Siber saldırıların denetimi ve tanımlanması için herhangi bir çalışma yapmamış veya yetersiz olduğunu düşünen tüm özel ve devlet kurumları, düşük bütçe ile dağıtık siber tuzak sistemini altyapılarına kurup veri analizine başlayabilecekleri değerlendirilmektedir.

Özellikle üniversiteler, bilgi güvenliği araştırma konularında çalışma yapan kişi ve kurumlar tasarladığımız sistemi bilgisayar altyapılarına dahil ederek saldırganların davranışları üzerinde gözlem yapabilirler. Tespit edilecek anormal davranışlardan çıkarttıkları sonuçları açık kaynak olarak araştırmacılar ile paylaşabilir ve bu konuda önlem almaya çalışan kişi ve kurumlara fayda sağlayabilirler.

5.3 Gelecek Araştırmaları ve Öneriler

Hem kurumlar hem de üniversiteler için honeypot üzerinde toplanan veriden yapılacak davranışsal analiz ile veri güdümlü güvenlik sistemlerinin kurulması, bilgi

güvenliği konusunda yapılacak gelecekteki araştırmalar için gelişime açık bir çalışma alanı olduğu değerlendirilmektedir. SSH honeypotlar dışında HTTP honeypotlar ile kurulacak bir araştırma ortamı ile bilinmeyen saldırı türlerinin keşfedilmesi için faydalı bir çalışma alanı olacaktır. Bu çalışmalar ile “Heartbleed” gibi saldırıların tanımlanmasında oldukça büyük fayda sağlayacağı düşünülmektedir.

Veri güdümlü bilgi güvenliğinin Hadoop ve Spark gibi anaçatı sistemleri sayesinde popülerliğinin zamanla artacağı açıkça görülmektedir. Veri bilimindeki yükseliş ile makine öğrenmesi kullanımına odaklanılarak geliştirilen ürünler, önleyici ve tanımlama yetenekleri çok daha başarılı araçların kullanıma girmesine yardımcı olacaktır. Bu tez kapsamında yapılan araştırma ile veri analitiği anaçatısı ve honeypot verisi kullanılarak bilinmeyen saldırıların tespitinde öngörü sağlayabilecek bir yaklaşım ortaya çıkartılmıştır.

ODROID-C2 veya Orange Pi gibi alternatif donanımların da özellikleri araştırılıp uygun çeşitli farklı yapılandırmalar yapılabilir. Ayrıca tasarımdaki düğüm sayısından az ya da çok düğüm ile sistem ölçeklendirilerek kurum için en uygun çözüme ulaşılabileceği değerlendirilmektedir. Bu şekilde maliyet üzerinde de ölçeklendirme yapılması sağlanabilir.

KAYNAKLAR

- [1] **Anonymous**, Analytics, Big Data. "Big Data Analytics for Security." (2013).
- [2] **Andersen, David G., and Nick Feamster**. "Challenges and opportunities in Internet data mining." *Parallel Data Laboratory, Carnegie Mellon University, Araştırma Raporu CMU-PDL-06-102* (2006).
- [3] **Bansal, G., Gupta, A., Pyne, U., Singhal, M., & Banerjee, S.** A Framework for Performance Analysis and Tuning in Hadoop Based Clusters. In *Smarter Planet and Big Data Analytics Workshop (SPBDA 2014), International Conference on Distributed Computing and Networking, Coimbatore, INDIA*. (2014)
- [4] **Bifet, Albert**. Mining Big Data in Real Time. *Yahoo Research*. Çevirmici: http://www.informatica.si/PDF/37-1/03_Bifet%20-%20Mining%20Big%20Data%20in%20Real%20Time.pdf (2012)
- [5] **Bilge, et al.** "Before we knew it: an empirical study of zero-day attacks in the real world." *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, (2012)
- [6] **Buell, Jeff**. "A Benchmarking Case Study of Virtualized Hadoop Performance on VMware vSphere 5." *Teknik makale. VMware* (2011).
- [7] **Bughin, J., Chui, M., & Manyika, J.**. Clouds, big data, and smart assets: Tentech-enabled business trends to watch. *McKinsey Quarterly*, 56(1), 75-86. (2010)
- [8] **Burg, David, et al.** Managing Cyber Risks in an Interconnected World. Key Findings from the Global State of Information Security® Survey 2015. Çevirmici: <http://www.pwc.com/gsis2015> (2015)
- [9] **Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., & Gruber, R. E.**, Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2), 4. (2008)
- [10] **Chaudhuri, S., Dayal, U., & Narasayya, V.**, An overview of business intelligence technology. *Communications of the ACM*, 54(8), 88-98. (2011)
- [11] **Chen, H., Chiang, R. H., & Storey, V. C.**, Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly*, 36(4). (2012)

- [12] Courtney, M., "Data on Demand". *Engineering & Technology*, 9(1), 64-67. (2014).
- [13] Crossler, Robert E., et al. "Future directions for behavioral information security research." (2013)
- [14] Dean, J., & Ghemawat, S., "MapReduce: simplified data processing on large clusters", *Communications of the ACM*, 51(1), 107-113. (2004)
- [15] Dean, J., & Ghemawat, S., "MapReduce: a flexible data processing tool", *Communications of the ACM*, 53(1), 72-77. (2010)
- [16] Gartner Group., "Pattern-Based Strategy: Getting Value from Big Data" Çevrimiçi: <http://www.gartner.com/it/page.jsp?id=1731916> (2011)
- [17] Gates, Alan F., et al., "Building a high-level dataflow system on top of MapReduce: the Pig experience." *Proceedings of the VLDB Endowment* 2.2 (2009)
- [18] Ghemawat, S., Gobioff, H., & Leung, S. T., "The Google file system" In *ACM SIGOPS Operating Systems Review* (Vol. 37, No. 5, pp. 29-43). ACM. (2003)
- [19] Greitzer, Frank L., et al., "Analysis of Unintentional Insider Threats Deriving from Social Engineering Exploits." *Security and Privacy Workshops (SPW)*, IEEE, (2014)
- [20] Hashem, Ibrahim Abaker Targio, et al., "The rise of "big data" on cloud computing: Review and open research issues." *Information Systems* 47, 98-115. (2015)
- [21] Jagadesh, H.V., et al., "Big Data and Its Technical Challenges", *Communications of the ACM* 57.7 86-94. *Computers & Applied Sciences Complete*. (2014)
- [22] LaValle, Steve, et al., "Big data, analytics and the path from insights to value." *MIT Sloan Management Review* 21 (2013)
- [23] Lee, Y., & Lee, Y. "Toward scalable internet traffic measurement and analysis with Hadoop". *ACM SIGCOMM Computer Communication Review*, 43(1), 5-13.(2013)
- [24] Louridas, P., & Ebert, C., "Embedded Analytics and Statistics for Big Data" *IEEE Software*, 30(6), 33-39. (2013)
- [25] Manyika, James, et al., "Big data: The next frontier for innovation, competition, and productivity", (2011)

- [26] **Mokube, Iyatiti, and Michele Adams.** "Honeypots: concepts, approaches, and challenges." *Proceedings of the 45th annual southeast regional conference*. ACM, (2007)
- [27] **Mone, G.** "Beyond Hadoop" *Communications of the ACM*, 56(1), 22-24. (2013)
- [28] **Nair, Ramesh, & Narayanan, Andy.** "Strategy and Benefitting from Big Data: Leveraging Unstructured Data Capabilities for Competitive Advantage", Çevrimiçi:
http://www.strategyand.pwc.com/media/file/Strategyand_Benefitting-from-Big-Data.pdf (2012)
- [29] **Qin, H. F., & Li, Z. H.** "Research on the Method of Big Data Analysis", *Information Technology Journal*, 12(10). (2013)
- [30] **Sakr, S., et al.** "The Family of MapReduce and Large-Scale Data Processing Systems", *ACM Computing Surveys*, 46(1), 11-11:44. (2013)
- [31] **Sallam, Rita L., et al.,** "Magic quadrant for business intelligence platforms." *Gartner Group, Stamford, CT* (2011)
- [32] **Simoncelli, D., Dusi, M., Gringoli, F., & Niccolini, S.** "Stream-monitoring with BlockMon: convergence of network measurements and data analytics platforms", *ACM SIGCOMM Computer Communication Review*, 43(1), 29-36. (2013)
- [33] **Shackelford, Dave.** "Security Intelligence in Action: A review of LogRhythm's SIEM 2.0 Big Data Security Analytics Platform", Çevrimiçi:
<http://www.sans.org/readingroom/whitepapers/analyst/security-intelligence-action-review-logrhythm-039-ssiem-20-big-data-security-analytics-35155> (2012)
- [34] **Shackelford, Dave.,** "SANS Security Analytics Survey" Çevrimiçi:
<http://www.sans.org/reading-room/whitepapers/analyst/security-analytics-survey-34980> (2013)
- [35] **Suthaharan, Shan.** "Big data classification: Problems and challenges in network intrusion prediction with machine learning." *ACM SIGMETRICS Performance Evaluation Review* 41.4 70-73. (2014)
- [36] **Taherimonfared, Aryan, Tomasz Wiktor Wlodarczyk, and Chunming Rong.,** "Real-Time Handling of Network Monitoring Data Using a Data-Intensive Framework.", *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*. Vol. 1. IEEE, (2013)
- [37] **Villars, R. L., Olofson, C. W., & Eastwood, M.** "Big data: What it is and why you should care", *IDC* (2011)

- [38] **Virvilis N., Gritzalis D.**, “The Big Four - What we did wrong in Advanced Persistent Threat detection?”, 8th International Conference on Availability, Reliability and Security (ARES-2013), pp. 248-254, IEEE, (2013)
- [39] **Virvilis N., Gritzalis D.**, “Trusted Computing vs. Advanced Persistent Threats: Can a defender win this game?”, 10th IEEE International Conference on Autonomic and Trusted Computing (ATC-2013), pp. 396-403, IEEE Press, Italy, (2013)
- [40] **Wang, Chengwei, Infantdani Abel Rayan, and Karsten Schwan.** "Faster, larger, easier: reining real-time big data processing in cloud." *Proceedings of the Posters and Demo Track*. ACM, (2012)
- [41] **White, T.** “Hadoop: The Definitive Guide: The Definitive Guide.”, O'Reilly Media.(2009)
- [42] **Whitworth, Jeff, and Shan Suthaharan.** "Security problems and challenges in a machine learning-based hybrid big data processing network systems." *ACM SIGMETRICS Performance Evaluation Review* 41.4: 82-85. (2014)
- [43] **Wu, Xindong, et al.** "Data mining with big data." *Knowledge and Data Engineering, IEEE Transactions on* 26.1: 97-107. (2014)
- [44] **Yang, Zhen, et al.** "An approach to spam detection by naive Bayes ensemble based on decision induction." *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*. Vol. 2. IEEE, (2006)
- [45] **Young, Williams T., et al.** “Detecting Unknown Insider Threat Scenarios.” Security and Privacy Workshops (SPW), 2014 *IEEE*. IEEE, (2014)
- [46] **Zaharia, Matei, et al.** "Spark: cluster computing with working sets." *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. (2010)
- [47] **Zaharia, Matei, et al.** "Discretized streams: Fault-tolerant streaming computation at scale." *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. ACM, (2013)
- [48] **N. Marz and J. Warren.** “Big Data: Principles and best practices of scalable real time data systems.” Manning Publications, (2013)
- [49] **Home Depot reports findings in Payment Data Breach Investigation.**
Çevrimiçi:
<https://corporate.homedepot.com/MediaCenter/Documents/Press%20Release.pdf> (2014).

- [50] **Scharr, Jill.** "Chase Bank Security Breach May Not Be That Bad." Çevirmiş: <http://www.tomsguide.com/us/chase-bank-breach-update,news-19545.html> (2014)
- [51] **CERT Insider Threat Team.** "Unintentional insider threats: A foundational study.", Çevirmiş: <http://resources.sei.cmu.edu/library/assetview.cfm?AssetID=58744> (2013)
- [52] **Clearswift.** "The enemy within." Çevirmiş: <http://www.clearswift.com/sites/default/files/images/blog/enemy-within.pdf> (2013)
- [53] **Liao, Hung-Jen, et al.** "Intrusion detection system: A comprehensive review." *Journal of Network and Computer Applications* 36.1 (2013): 16-24. (2013)
- [54] **Spitzner, Lance.** "Honeypots: Catching the insider threat." *Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE,* (2003)
- [55] **Hunker, Jeffrey, and Christian W. Probst.,** "Insiders and insider threats—an overview of definitions and mitigation techniques." *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* 2.1: 4-27. (2011)
- [56] **Watson, David, and Jamie Riden.** "The honeynet project: Data collection tools, infrastructure, archives and analysis." *WOMBAT Workshop on Information Security Threats Data Collection and Sharing.* (2008)
- [57] **Stamp, Mark.** "A Revealing Introduction to Hidden Markov Models." *Department of Computer Science San Jose State University* (2012)
- [58] **Schonlau, Matthias, et al.** "Computer intrusion: Detecting masquerades." *Statistical science* (2001): 58-74. (2001)
- [59] **Maxion, Roy A.** "Masquerade detection using enriched command lines.", *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).* IEEE Computer Society, (2013)
- [60] **Rieck, Konrad, et al.** "Automatic analysis of malware behavior using machine learning." *Journal of Computer Security* 19.4 (2011): 639-668. (2011)
- [61] **Haltas, Fatih, et al.** "An automated bot detection system through honeypots for largescale." *Cyber Conflict (CyCon 2014), 2014 6th International Conference On.* IEEE, (2014)

- [62] **Virvilis, Nikos, Dimitris Gritzalis, and Theodoros Apostolopoulos.** "Trusted Computing vs. Advanced Persistent Threats: Can a defender win this game?." *Ubiquitous Intelligence and Computing, IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*. IEEE, (2013)



EKLER

EK A: Komut Analizörü Kaynak Kodu

EK B: Komut Ayristici Kaynak Kodu

EK C: Naive Bayes Eğitici Kaynak Kodu



/**

83

```

// Log olmayan satirlarin tumunu temizle
val non_log_ons = honeypotInput.filter(!_.contains("Login"))
    .filter(_.contains(" ")) .map(line => (line.split(" ")(0,
line.substring(line.indexOf(" "))))))

// Komut icin kullanan tum argumanlar ile birlikte komutu al
val allCommands =
honeypotInput.filter(_toUpperCase.contains("COMMAND"))
    .map(line => """Command
\[ [^\]]+\]""".r.findAllIn(line).toList) .map(line =>
line.mkString)

// Komutlari listele
val individualcommands =
allCommands.filter(!_.isEmpty).map(line => """\[ [^
]+""".r.findFirstIn(line).
toList).filter(!_isEmpty).map(_>mkString.replace("[", "").repla
ce("]", "")).collect.foreach( parseEnrichedCommands(_))
}

```

EK B

```
/**
 * Author: bcakmak
 * HUTEN 2016
 * */

package com.hho.soma301.parser

object CommandParser {
    def parseHttpRequest(input:String) : String = {
        val index = input.lastIndexOf("/")
        if (index >= 0) input.substring(index + 1) else input
    }

    //Cift tirnak icerisindeki her seyi al ve tek bir komuta cevir
    def quotesToSingleCommands(input:String) : String = {
        val regexlist = "\"\"\"[^\"]+\"\"\".r
        val otherCommands =
        regexlist.findAllIn(input).toList.map(line =>
        line.replace("\"", ""))
        var returnlist = List[String]()
        for(i<-0 to otherCommands.size-1) {
            returnlist =
returnlist:+transformCommand(otherCommands(i))
        }
        returnlist.mkstring("||")
    }

    //Cift tirnak arasindaki her seyi sil
    def removeQuotesFromCommand(input:String) : String = {
        val regexlist = new Regex("\"\"\"[^\"]+\"\"\"")
        regexlist.replaceAllIn(input, "").mkString
    }

    //Saldirgani tanımlamak için komutu argumanlari ile cevir
    def transformCommand(input:String) : String = {
        var returnValue = ""
        var firstArgExist = false
        if (!input.isEmpty) {
            val inputlist = input.split(" ")
            returnValue = inputlist(0)
            for(i <-1 to inputlist.size-1) {
                var value = inputlist(i)
            }
        }
    }
}
```

```

        if (value.size > 0 && (value(0) == '-' ||
!firstArgExist)) {
            if (value(0) != '-') {firstArgExist = true}
            if (inputlist(0).contains("wget") ||
inputlist(0).contains("curl")) {
                value = parseHttpURL(value)
            }
            returnValue = returnValue + " " + value
        }
    }
}
returnValue
}

// Argumanlari bulunan komutlari ayristir
def parseEnrichedCommands(input:String) : String = {
    var returnValue = ""
    if (!input.isEmpty) {
        val otherCommand = quotesToSingleCommands(input)
        val removedCommands = removeQuotesFromCommand(input)
        val removedCommandsTransformed =
transformCommand(removedCommands)
        if (!otherCommand.isEmpty) {
            returnValue = otherCommand + "||" +
removedCommandsTransformed
        } else {
            returnValue = removedCommandsTransformed
        }
    }
    returnValue
}
}

```

EK C

```
/**
 * Author: bcakmak
 * HUTEN 2016
 * */

package com.hho.soma301.naiveBayesTrainer

class naiveBayesTrainer {
    def main(args:Array[String]) {
        val badactorCommands =
        Source.fromFile("allcommands.txt").getLines.toList
        val goodactorCommands =
        Source.fromFile("trainingGoodActor.txt").getLines.toList

        // Kotu aktor için egit
        println("Kotu aktor egitimi: ")
        for(i <- 0 to badactorCommadns.size -1) {
            val enrichedCommands =
            parseEnrichedCommands(badactorCommands(i).split("\\|\\|"))
            for (x <- 0 to enrichedCommands.size-1) {
                val probability =
                calculateProbabilityBadActor(badactorCommands,
                goodactorCommands, enrichedCommands(x))
                println(enrichedCommands(x) + " , " + probability)
            }
        }

        // Iyi aktor için egit
        println("Iyi aktor egitimi: ")
        for (i <- 0 to goodactorCommands.size - 1) {
            val enrichedCommands =
            parseEnrichedCommands(goodactorCommands(i).split("\\|\\|"))
            for (x <- 0 to enrichedCommands.size -1 ) {
                val probability =
                calculateProbabilityBadActor(badactorCommands,
                goodactorCommands, enrichedCommands(x))
                println(enrichedCommands(x) + " , " + probability)
            }
        }
    }

    // Olasilik hesabi
```

```

def calculateProbabilityBadActor(goodactorList:List[String],
badactorList:List[String], command:String) : Double = {
    val countinBad =
goodActorList.map(parseEnrichedCommands(_)).count(_.contains(c
ommand))
    val countinGood =
badActorList.map(parseEnrichedCommands(_)).count(_.contains(co
mmand))
    calculateBayesTheorem(countinBad, countinGood)
}

def calculateBayesTheorem(first:Int, second:Int): Double = {
    (first * 1.0) / (first + second * 1.0)
}
}

```



ÖZGEÇMİŞ

Ad Soyad : Burak ÇAKMAK
Doğum Yeri ve Tarihi : Manisa / 07.01.1990
Adres : Hava Harp Okulu, Yeşilyurt/İstanbul
E-Posta : bcakmak@live.com
Lisans : Sakarya Üniversitesi Bilgisayar Mühendisliği

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- Cakmak B., Yilmaz G., *Akademik Bilişim*, Honeypot ve Hadoop Tabanlı Bilgi Güvenliği Yönetim Sistemi, 4-6 Şubat 2015, Eskişehir, Türkiye
- Cakmak B., Yilmaz G., *International Conference on Advanced Technology & Science*, Developing Test Infrastructure for Distributed Cost Effective Network Security System Using Limited Resources, 1-3 Eylül 2016, Konya, Türkiye