
GAZİ ÜNİVERSİTESİ
Gazi
1926

GELİŞMİŞ KALICI TEHDİT SALDIRILARININ AĞ AKIŞ ANALİZİYLE TESPİT EDİLMESİ

Mehmet Emin BAYRAK

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2015

Mehmet Emin BAYRAK tarafından hazırlanan "GELİŞMİŞ KALICI TEHDİT SALDIRILARININ AĞ AKIŞ ANALİZİYLE TESPİT EDİLMESİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Yrd.Doç.Dr. Mehmet DEMİRCİ

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Başkan : Prof.Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Üye : Yrd.Doç.Dr. Murat AYDOS

Bilgisayar Mühendisliği Anabilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Tez Savunma Tarihi: 06/07/2015

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……

Prof. Dr. Şeref SAĞIROĞLU

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Mehmet Emin BAYRAK

06/07/2015

GELİŞMİŞ KALICI TEHDİT SALDIRILARININ AĞ AKIŞ ANALİZİYLE TESPİT EDİLMESİ

(Yüksek Lisans Tezi)

Mehmet Emin BAYRAK

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2015

ÖZET

Son yıllarda, bilgi güvenliği manzarasında zararlı faaliyetlerin kaynağı bağlamında bireysel saldırganlardan büyük suç örgütlerine ve devletlere doğru bazı değişiklikler yaşandı. Gelişmiş kalıcı tehdit saldırıları bu trendin bir tezahürüdür. Bu saldırılar dikkatli şekilde planlanan ve uzmanlıkla gerçekleştirilen siber saldırılardır. Bu tez çalışması, gelişmiş kalıcı tehditler hakkında kapsamlı bir araştırma sunmaktadır. Gerçek hayatta gerçekleştirilen gelişmiş kalıcı tehdit saldırılarından yola çıkarak yaşam döngüleri tanımlanmış ve ortak hedefleri belirlenmiştir. Potansiyel olarak gelişmiş kalıcı tehdit saldırılarına maruz kalabilecek hedefler için literatürden araştırılan saldırının etkilerini azaltma stratejileri ve savunma metotları gözden geçirilmiştir. Bu çalışma ile gelişmiş kalıcı tehdit saldırılarını belirleyecek bütünsel bir güvenlik politikası oluşturulmuştur. Gelişmiş kalıcı tehdit saldırıları, ağ akış trafik bilgileri kullanılarak tespit edilmeye çalışılmıştır. Kullanıcı bilgisayarlarının ağ akış trafik bilgileri ile oluşturulan matris üzerinde tekil değer ayrıştırması metodu uygulanarak boyut küçültme işlemi yapılmış ve kosinüs benzerliği kullanılarak kullanıcı bilgisayar davranışları karşılaştırılmıştır. Bu çalışmanın yapılan testler neticesinde gelişmiş kalıcı tehdit saldırılarının tespitinde etkili olabileceği, gerçek bilgi sistem altyapılarında kullanılabileceği değerlendirilmektedir. Gelişmiş kalıcı tehdit saldırılarına karşı yeni bir alternatif güvenlik yaklaşımı getirmesi nedeniyle çalışmanın literatüre önemli katkılar sağlayacağı düşünülmektedir.

Bilim Kodu	:	902.1.014
Anahtar Kelimeler	:	Gelişmiş Kalıcı Tehdit, Siber Güvenlik, Siber Savaş, Tekil Değer Ayrıştırması, GKT, Ağ Akış Verisi, Kosinüs Benzerliği.
Sayfa Adedi	:	89
Danışman	:	Yrd. Doç. Dr. Mehmet DEMİRCİ

DETECTING ADVANCED PERSISTENT THREATS
WITH NETWORK FLOW ANALYSIS
(M.Sc. Thesis)

Mehmet Emin BAYRAK

GAZİ UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2015

ABSTRACT

In recent years, information security landscape has seen a shift in the source of malicious activity from individual attackers to large criminal organizations and governments. Advanced persistent threats are a manifestation of this trend. They are carefully planned and expertly executed attacks, usually employed as an important weapon in cyber warfare. This thesis provides a thorough investigation of advanced persistent threats. We describe their life cycle, classify their nature and common objectives using information from real examples. We also review mitigation strategies and defense methods, extract strategic principles from the literature for strengthening potential targets. This work constitutes a holistic security policy to determine advanced persistent threats. APT attacks will be tried to detect using network flow data. Flow data matrix is created using information of a user's computer network traffic flow. Singular value decomposition is applied on the matrix. In this way dimension reduction process is carried out. Later on user computer behavior is compared using cosine similarity. The result of the study shows that the proposed model can effectively be used to detect advanced persistent threats in real information system infrastructures. It is thought that this study provides an important contribution to the literature bringing a new alternative security approach against advanced persistent threats.

Science Code : 902.1.014
Key Words : Advanced Persistent Threats, Cyber Security, Cyber War, Singular Value Decomposition, APT, Network Flow Data, Cosine Similarity.
Page Number : 89
Supervisor : Assist. Prof. Dr. Mehmet DEMİRCİ

TEŞEKKÜR

Yardımlarını her konuda esirgemeyen sayın danışman hocam Yrd.Doç.Dr. Mehmet DEMİRCİ'ye, bu tez çalışması hakkında yorumlarını, fikirlerini ve katkılarını sunan Prof.Dr. Şeref SAĞIROĞLU ve Yrd.Doç.Dr. Murat AYDOS'a, "baba bilgisayardan oyun aç" diyerek bilgisayarı açmama sebep olan ve oyun bitiminde tezime biraz daha vakit ayırmamı sağlayan oğullarım Cenker Eren ile Celal Eray'a ve moral-motivasyon desteği sağlayan sevgili eşime çok teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
APT	Advanced Persistent Threats
GKT	Gelişmiş Kalıcı Tehdit
DoS	Denial of Service
DDoS	Distributed Denial of Service
TDA	Tekil Değer Ayrıştırması
GSM	Global System for Mobile
IDE	Integrated Development Environment
DMZ	Demilitarized Zone
RRD	Round Robin Database
UDP	User Datagram Protocol
TF	Term Frequency (Terim Frekansı)
ITF	Inverse Traffic Frequency
IDF	Inverse Document Frequency
THAF	Ters Hedef Adres Frekansı
MB	Megabyte
GB	Gigabyte
TB	Terabyte
LSI	Latent Semantic Indexing (Saklı Anlam Indeksi)
SVD	Singular Value Decomposition (Tekil Değer Ayrıştırması)
DPI	Deep Packet Inspection (Derin Paket İnceleme)
OSI	Open Systems Interconnection
PHP	Hypertext Preprocessor
SSH	Secure Shell
HTTP	Hypertext Transfer Protocol

Kısaltmalar	Açıklamalar
HTTPS	Secure Hypertext Transfer Protocol
GUI	Graphical User Interface
IOS	Internetwork Operating System
BSD	Berkeley Software Distribution
USB	Universal Serial Bus

1. GİRİŞ

Kurumlar fiziksel güvenliğin yanında bilgi sistemleri güvenliği için de önemli miktarlarda para harcamaktadırlar. Fakat günümüzde ilginç bir durum oluşmaktadır, 1990 ve 2000 yılları arasında bilgi sistemleri güvenliği üzerine ne kadar para harcanırsa kurumlar o kadar daha güvenli olmakta ve daha az risk yaşamaktaydılar. Günümüzde ise, kurumlar güvenlik bütçelerini artırmakta fakat risk yaşamaya devam etmekte ve önemli bilgilerini farklı siber saldırı aktörlerine kaptırmaktadırlar. Önemli verilerini kaybetmeseler bile saldırılar nedeniyle, verdikleri hizmetlerde aksamalar meydana gelmektedir.

Günümüzde güvenliğe ne kadar çok para harcanırsa kurum o oranda güvenlidir anlayışı doğru değildir. Yaşanan bu problem, kurumların, siber tehdide ve güvenliğe karşı yaklaşımlarının değişmemesinden kaynaklanmaktadır. Geleneksel tehditler (virüsler, trojanlar, solucanlar, zararlı yazılımlar vb.) hala endişe vericidir. Ancak kurumlar geleneksel tehditlerin yanında artık yeni bir tehditle mücadele etmek zorundadırlar, Gelişmiş Kalıcı Tehditler (GKT), İngilizce isimlendirmesi ile Advanced Persistent Threats (APT). GKT'ler iyi finanse edilen organize gruplarca oluşturulup, devlet kurumlarının ve ticari şirketlerin önemli bilgilerini ele geçirmekte veya kurumların bilgi sistemlerini sabote etmektedirler.

Günümüzdeki güvenlik manzarası

Günümüzde ne zaman bir gazete sayfası çevrilse, ne zaman televizyonda haberler izlense devamlı surette bir kurumun, bir üniversitenin, bir organizasyonun veya bir devletin siber saldırıya maruz kaldığı, bazı bilgilerinin istismara uğradığı ve önemli bilgilerinin ele geçirildiği duyulmaktadır. Büyüklüğü, yapısı, güvenliği ne olursa olsun farklı kurumların ve organizasyonların sistemlerinin ele geçirildiğine şahit olunmaktadır. Bu durum şimdilik sona erecekmiş gibi gözükmemektedir. Kurbanlar arasında çeşitli devletler, ticari ve ticari olmayan organizasyonlar, üniversiteler, ulusal ve uluslararası yapılar bulunmaktadır.

Hedefi olan ve kararlı yapıdaki bu hacker grupları, organizasyonlar üzerinde ciddi bir panik havası oluşturmaktadır. Örneğin medya haberlerine göre 2013-2014 yılları arasında

sadece Amerika Birleşik Devletlerinde toplam 130 milyon kişinin (Target, Home Depot firma müşterileri vb.) kart ve kişisel bilgileri çalındı [40].

Organizasyonların ve kurumların güvenlik korkusu ile yaşamaları doğru değildir. Kurumlar için birinci öncelik, hedef olmaktan kaçınmaya çalışmaktır. Böyle gelişmiş ve kararlı düşmana karşı mücadele etmek çok zor olacaktır. Kurumlar, ellerindeki değerli şeylerle şehrin karanlık sokaklarında göstere göstere ilerlemektense, değerli şeylerle şehrin nispeten güvenli yerlerinde gizli gizli gitmek zorundadırlar. Değerli veriye sahip olan kurum çalışanlarının internet üzerinde ve medyada ne söylediğine, sosyal paylaşım sitelerinde ne yüklediğine çok dikkat etmesi gerekmektedir. Çünkü GKT saldırılarının hedeflerinden biri de kurum çalışanlarıdır. Yetkiye sahip kurum çalışanlarının, tehlikenin her zaman farkında olması ve bu bilince sahip olmaları gerekmektedir.

Klasik güvenlik anlayışında geleneksel saldırılara karşı yamaların yüklenmesi (patching), zararlı yazılım koruması kurulumu ve güncellenmesi (anti-virus), güvenlik duvarı, saldırı tespit sistemleri vb. yeterli görülmesine rağmen bu durum GKT saldırıları için yeterli değildir. Klasik güvenlik tedbirlerini uygulayan kurumlar güvenlik için milyon dolarlar harcasalar bile değerli bilgi sızmaları devam edecektir. Organizasyonların günümüzde tehdit algılaması değişmedikçe daha büyük siber olaylar yaşanacaktır. Çünkü GKT saldırıları 90'lı yıllarda yaşanan ve günümüzde de devam eden klasik saldırılara benzememektedirler.

Çağımızda tüm kurumlar ve organizasyonlar günün birinde değerli verilerinin ele geçirileceği ve sızdırılacağı gerçeğini kabul etmek zorundadır. Önemli olan bu durumun ne kadar erken fark edildiği ve veri kaybını nasıl telafi ettiğidir. Günün birinde bilgilerin ele geçirileceğini bilmek korkutucu gözükebilir ancak gerçeği bilerek yaşamak reddederek yaşamaktan daha iyidir. Hiçbir zaman kurumun bilgilerinin çalınmayacağını iddia etmek insanın hiçbir zaman hasta olmayacağını iddia etmesi kadar gerçektir. GKT ile mücadelede önemli olan risk büyümeden önlemeye çalışmak ve sorunu mümkün olduğunca erken tespit edebilmektir.

Kurumların güvenliğe bakışı

Güvenlik bağlamında devletlerin, kurumların ve firmaların geldiği nokta bir gün mutlaka siber saldırıda bir şekilde mağlup olacaklarıdır. Bu gerçeği bilerek yola devam etmek çok önemlidir. Çok önemli veriye sahip olan ve internete bağlı bir organizasyonun şuanda bilgilerinin ele geçirildiğini farz etmesi en doğru yaklaşımdır. Çağımızda hiçbir sistemin güvenli olmadığı bir noktaya gelmiş bulunmaktayız. Şuanda bilgilerin ele geçirildiğini farz edip sistemi dikkatli bir şekilde takip etmek yapılacak en doğru hareket tarzıdır. Veri sızdırılmasını kurum ne kadar erken tespit edebiliyorsa o kadar başarılıdır. Kurumun 6 ay boyunca veri sızdığını fark edemeyip yeni fark etmesi gerçekten çok üzücü olacaktır. Siber güvenlik konusunda her zaman en kötüsü düşünülmeli, en iyi sonuç için ümit edilmelidir.

GKT saldırılarının diğer siber saldırılardan farkı

GKT saldırılarının, diğer siber saldırılardan farkını belirtmeden önce GKT saldırısının ne olmadığı konusunda bilgi vermekte fayda vardır. Öncelikle GKT saldırıları, DoS veya DDoS gibi tek bir saldırı tipi değildir. Aynı şekilde GKT, Nimda gibi tek bir zararlı yazılım da değildir. GKT saldırısı aktörleri tek bir saldırı grubundan oluşmamaktadır. Sosyal mühendisler, zararlı yazılımı yazan gruplar, finansal destek sağlayan personel, dil bilimciler gibi birçok gruptan kişinin bulunduğu organize bir ekipten oluşmaktadır. Kısaca belirtecek olursak, GKT saldırısı tek bir saldırı tipi, tek bir zararlı yazılım türü ve tek bir saldırı grubundan oluşmamaktadır, belki bunların hepsinden çok fazla sayıda unsurun bir araya gelmesiyle GKT saldırıları yapılmaktadır. GKT'nin diğer siber saldırılardan farkını özetleyecek olursak;

- Saldırının her aşaması büyük bir gizlilik içinde yapılır, diğer saldırılar gibi bilişim dünyasında unvana sahip olayım, devletlerarasında gücümü dünyaya ispatlayayım, arkadaşlarım arasında itibar kazanayım gibi kaygılarla gerçekleştirilmez.

- Geleneksel saldırılarda genellikle bilinen veya yaması mevcut olan açıklıklar kullanılır, GKT saldırılarında ise zorunlu olmamakla birlikte genellikle bilinmeyen veya sıfırıncı gün açıklıkları kullanılır.

- GKT saldırılarının her zaman net hedefleri vardır. Örneğin bir ülkenin büyük elçilikleri veya bir devletin X kurumu gibi. Geleneksel saldırılarda ise hedef, genellikle elde bulunan açıklıktan faydalanılabilecek kurumlar veya kişilerdir. Örneğin Wordpress 4.x sürümünü kullanan kurumlar, Adobe 9 yazılımını bilgisayarına yüklemiş olan kişiler gibi.

- GKT saldırı aktörleri, hedef aldığı kurumun bilgi sistemine sızana ve amacına ulaşana kadar her türlü yolu deneyecektir. Geleneksel saldırılarda ise eğer saldırganın elinde Wordpress 4.x ile ilgili bir açıklık mevcut ise ve siz Wordpress'i kullanmıyorsanız hedef değilsinizdir.

Bilişim dünyasının yeni yeni gelişmeye başladığı 1990'lı yıllarda Melissa Virus, ILOVEYOU Worm, Anna Worm gibi zararlı yazılımlar net hedefleri olmamakla birlikte genel olarak insanlara bir süre rahatsızlık verdiler. GKT saldırıları ile birlikte siber saldırıların ticari avantaj elde etmek, teknoloji kopyalamak, casusluk, sabotaj gibi çok çeşitli amaçları bulunmaktadır. Bahsettiğimiz bu amaçlara ulaşıldığı takdirde kurumlar ve şirketler için hayati derecede yıkımlara ve zararlara sebep olmaktadır. Bu yüzden GKT saldırılarının tespit edilmesi ve önlenmesi, kurumlar ve şirketler için büyük öneme sahiptir. Bu tez çalışmasında bu doğrultuda, ağ akış trafiğinin matematiksel bir yöntem ile kosinüs benzerliği kullanılarak normal kullanıcı davranışının diğer kullanıcılardan ayrılmasını saptamaya çalıştık.

Bu tez çalışması yedi bölümden oluşmaktadır. İkinci bölümde, gelişmiş kalıcı tehdit (GKT) saldırıları hakkında bilgi verilmiştir. GKT saldırılarının gerçek aktörlerinin, otomatik saldırı yazılımlarından ziyade insan olmasından dolayı var olan tehdidin öneminden bahsedilmiş ve bazı önemli gerçek GKT saldırıları açıklanmıştır. Bu GKT saldırıları, Stuxnet, Operation Aurora, DuQu, Red October, Miniduke, Regin ve Gauss'tur.

Üçüncü bölümde GKT saldırı tespitinde kullanılacak olan tekil değer ayrıştırması (TDA) metodu anlatılmış ve kullanım alanları açıklanmıştır.

Dördüncü bölümde GKT saldırı tespitine yönelik olarak yapılan bilimsel çalışmalar incelenmiş, literatürdeki GKT ile ilgili konular ele alınmıştır.

Beşinci bölümde TDA ile beraber saldırı tespitinde kullanılan ağ akış verisi anlatılmış, ağ akış verisinin nasıl toplanacağı ve NfSen yazılımı ile nasıl incelenebileceği ile ilgili örnek bir uygulama yapılmıştır.

Altıncı bölümde ağ akış verisi ile TDA kullanarak gelişmiş kalıcı tehdit saldırısının tespitinin nasıl yapılacağı ayrıntılı bir şekilde anlatılmış, çıkan sonuçlar açıklanmıştır.

Son olarak yedinci bölümde sonuç ve öneriler ile gelecekte yapılabilecek çalışmalarda sonuçları geliştirmeye katkı sağlayabilecek metotlar konusunda bilgi verilmiştir.

2. GELİŞMİŞ KALICI TEHDİT

Gelişmiş kalıcı tehdit saldırıları, bilgi sistemlerine yapılan gizli, devamlı ve uzun süreli kalıcı olacak şekilde planlanan saldırıları ihtiva etmektedir. Saldırı aktörleri organize suç örgütleri olabildiği gibi, bazı devletler de bu tür saldırılar gerçekleştirmektedirler. Yüksek derecede gizlilik bu tür saldırıların en önemli özelliğidir. Saldırı aktörlerinin teknik kabiliyetlerini başkalarına gösterme gibi bir dertleri bulunmamaktadır. Yüksek derecede gizlilik saldırının her aşamasında bulunmaktadır.

Bu bölümde GKT saldırılarının tanımı yapılmış ve GKT teriminin terminolojisi hakkında bilgi verilmiştir. Daha sonra GKT saldırısı yaşam döngüsü anlatılmış, gerçek hayatta kurumların maruz kaldığı GKT saldırı örneklerinden bazıları hakkında bilgi verilmiştir. Son olarak da GKT saldırılarının tespitine yönelik olarak literatür araştırması yapılmış ve bu alanda yapılan çalışmalardan örnekler verilmiştir.

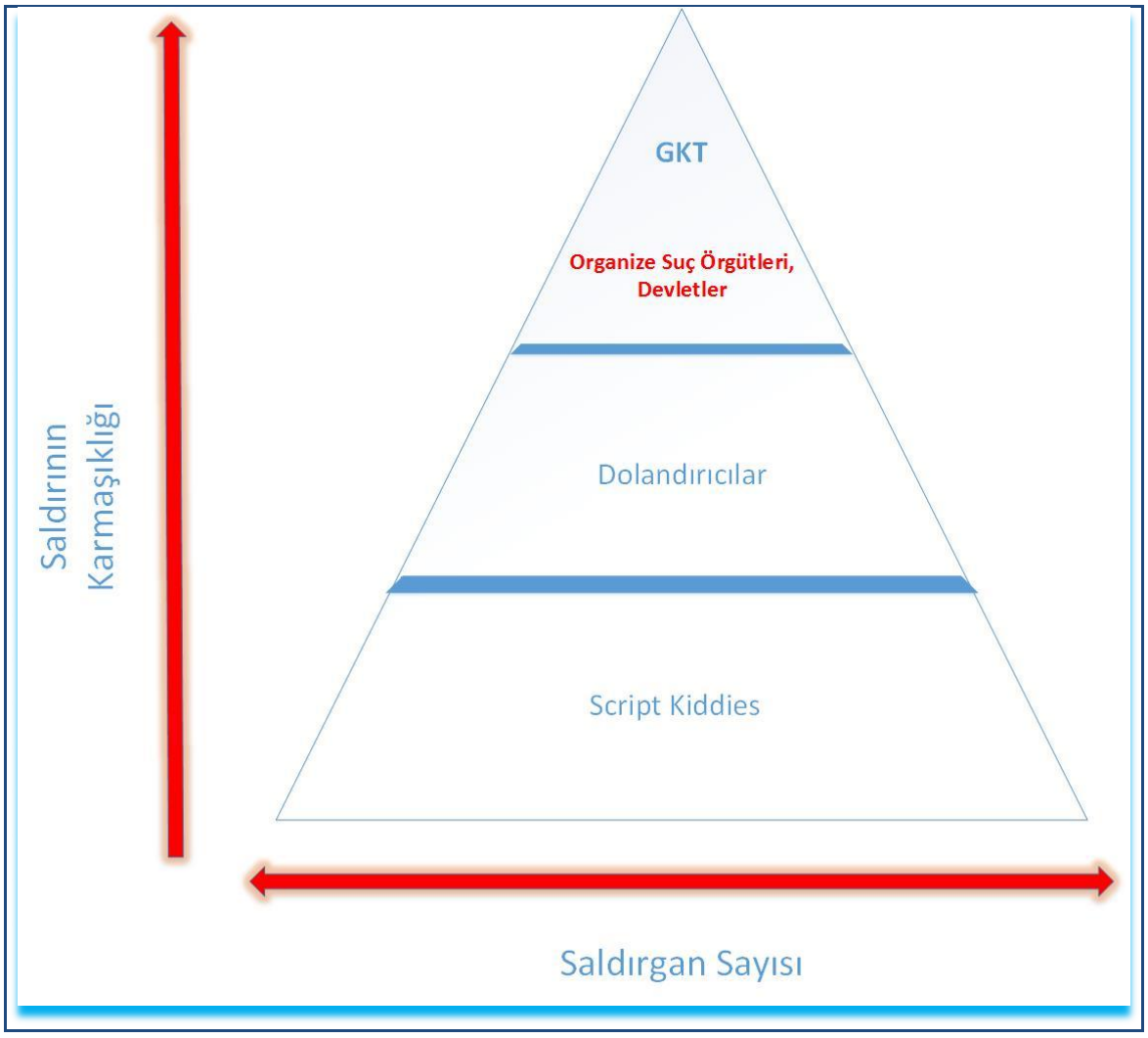
2.1. Gelişmiş kalıcı tehdit (GKT) tanımı ve amacı

Terim ilk olarak Amerikan askeri birimlerine yapılan Çin bağlantılı bilgisayar ağlarına sızma girişimlerinin kod adı olarak kullanılmaya başlanmıştır. GKT terimi daha sonra zamanla gelişerek ileri düzey düşman unsurların gizli bir tarzda bilgi sistemlerine sızarak kritik bilgileri kurumlardan çalan ve istismar eden unsurlar için kullanılmaya başlandı [36].

GKT terimini, çeşitli kesimler farklı farklı yorumlamakta olup, bazen sadece Çin'den gelen saldırılar, bazen tüm geleneksel siber saldırılar GKT olarak adlandırılmaktadır. Bu tanımlamalar yanlıştır. GKT tanımı genel olarak anlaşılan tanımdan farklı olduğu gibi buna karşı alınacak önlemler de geleneksel tehditlere karşı alınan önlemlerden farklıdır. Geleneksel tehditlere (virüsler, trojanlar, solucanlar vb.) karşı alınan tedbirler GKT saldırılarında etkisiz kalmaktadır. Ancak geleneksel tehditler hala sorun olduğu için onlara da tedbir alınmaya devam edilmesi gerekmektedir.

Şekil 2.1'de siber saldırı tehdit piramidi gösterilmektedir. Yatay eksen saldırgan sayısını, dikey eksen ise saldırının karmaşıklığını göstermektedir. Piramidin altında yer alan grup, zararlı yazılım yazmaya kabiliyeti olmayan, başkalarının yazdığı zararlı yazılımların ve

araçların teknik ayrıntısını bilmeden kullanan kişileri göstermektedir. Piramidin orta bölümünde yer alan saldırı aktörleri, başkalarının yazdığı saldırı araçlarını yeterli teknik bilgi ile kullanabilen kişilerdir. Bu kişilerden bazıları karmaşık olmasa da kendileri de saldırı araçları yazabilmekte veya var olan araçlara küçük eklemeler yapabilmektedirler. Piramidin en üst bölümünde bu tezin konusunu oluşturan GKT aktörleri yer almaktadır. GKT aktörleri devletler olabildiği gibi organize suç örgütleri de GKT aktörü olabilmektedirler. Bu aktörler saldırıyı gerçekleştirmek için her türlü teknik bilgiye araçlara sahiptirler. Ayrıca ihtiyaç olması durumunda kendi saldırı araçlarını yapabilmektedirler [12].



Şekil 2.1. Siber saldırı tehdit piramidi

GKT saldırılarının amaçlarını ticari avantaj elde etmek, teknoloji kopyalamak, casusluk ve

sabotaj gibi dört grupta toplamak mümkündür;

- *Ticari avantaj elde etmek:* Büyük teknoloji firmalarının veya devletlerin güçlü rakiplerine karşı avantaj elde etmek maksadıyla GKT saldırıları düzenleyebilmektedirler. Rekabete dayanan bir ortamda ticari avantaj elde etmek için rakiplerin değerli verilerini almak, ticari ilişkilerini öğrenmek, ihaleler vb. hususlarda rakamları öğrenmek büyük önem kazanmaktadır.

- *Teknoloji kopyalamak (transfer):* GKT saldırganının sahip olmadığı teknolojiyi, saldırdığı kurumlardan haberleri olmadan transfer etme girişimidir. Bu surette bilimsel veriler, teknolojik gelişmeler, uygulamalar, servisler ve planlar hedef kurumdan gizlice çalınacak ve teknoloji kopyalama işlemi gerçekleştirilmiş olacaktır.

- *Casusluk:* GKT saldırıları ile askeri, teknoloji, politika vb. alanlarda casusluk faaliyetleri yapılabilmektedir. Kurumların ve şirketlerin ne yaptığı, fikirleri, planları, stratejik kararları bilgi sistemleri üzerinde icra edilen casusluk faaliyetleri vasıtası ile öğrenilebilmektedir.

- *Sabotaj:* GKT saldırıları, kurumların verilerini ele geçirmek için kullanılabildiği gibi deney sonuçlarını değiştirmek, kaynak kodlara müdahale etmek, verileri bozmak gibi sabotaj faaliyetlerinde de kullanılabilmektedir. Bu suretle kurumların ilerlemeleri veya işleri sekteye uğratılabilmektedir.

Siber saldırılar günümüzde, eskiden olduğu gibi geniş çaplı hedeflere karşı değil, dar çerçevede belli hedeflere karşı yapılmaktadır. Hedef tahtasında kurban olarak yüksek profildeki teknoloji şirketleri, devlet kurumları yer almaktadır. Ayrıca bireysel saldırganlar yerini devletlere veya büyük suç örgütlerine bırakmaktadır. Basit saldırıların yerini tek kişinin yapamayacağı karmaşık saldırılar almaktadır.

2.2. GKT terminolojisi

GKT'lerin terim olarak tek tek anlamlarına bakacak olursak [36];

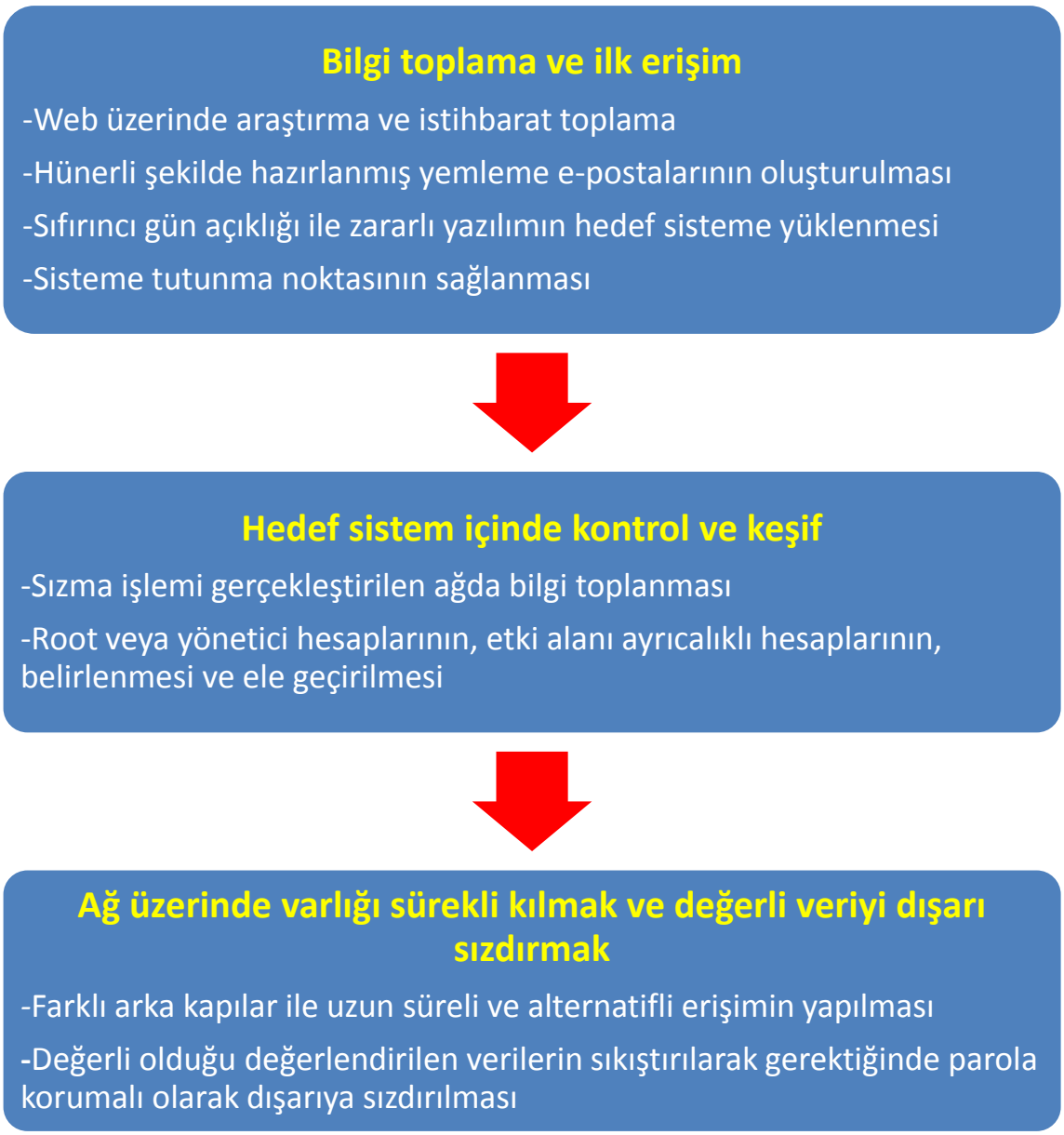
"Gelişmiş" kelimesi saldırı tekniklerinin ileri düzey olması olarak algılanmamalıdır. Kuruma veya organizasyona saldıran düşman gruplarının ileri düzeyde teknik bilgi ve beceriye sahip oldukları manasında anlaşılmalıdır. Kuruma yapılan siber saldırının veya sızmanın her zaman ileri düzey olması gerekmemektedir, çok basit bir teknikle de içeriye girilmiş olabilir. Burada önemli olan düşman unsurların gelişmiş teknikleri kullanabilecek yeteneğe sahip olmalarıdır. Basit bir teknikle de hedef kuruma sızılabiliyor ise bu basit yöntem GKT aktörleri tarafından mutlaka uygulanır.

"Kalıcı" kelimesi GKT aktörlerinin icra edecekleri görevi gerçekleştirmek için kararlı olduklarını, ihtiyaç olması durumunda tekrar sisteme ulaşmak için çeşitli arka kapılar oluşturabileceklerini belirtmektedir. Hedef kurum, devamlı surette bu aktörler tarafından izlenir. İhtiyaç olması durumunda bu arka kapılar tekrar işletilerek kuruma sızmak ve bilgi almak için kullanılır.

"Tehdit" kelimesi düşman unsurların kuruma zarar veya kayıp oluşturabilecek potansiyele ve motivasyona sahip olmaları anlamına gelmektedir. GKT saldırılarında kurumların karşı karşıya geldiği aktörler otomatik saldırı yapan bilgisayar programlarından ziyade kurumun sistemlerine sızmak için her yolu deneyen düşmandır. Yani GKT saldırısında asıl tehdit insandır, zararlı yazılımlar asıl tehdit değildir.

2.3. GKT saldırısı yaşam döngüsü

GKT saldırısının yaşam döngüsü Şekil 2.2'de de görüldüğü gibi üç ana safhaya ayrılabilir. Bu safhalar tüm GKT saldırılarında gözlemlenmekte olup, ayrıntılı olarak açıklanacaktır.



Şekil 2.2. GKT saldırısı yaşam döngüsü

2.3.1. Bilgi toplama ve ilk erişim

Web üzerinden araştırma yaparak kurum hakkında istihbarat toplama en çok kullanılan yöntemdir. İnternetten yeterli bilgi toplama işleminin yapılamadığı durumlarda fiziksel olarak temasa geçme işlemi de yapılmaktadır. Hedef kurumda işe girme, kurumun çalışanları ile internetten veya fiziksel olarak arkadaşlık kurma bilgi toplamak için kullanılan diğer yöntemlerdir [11].

Kurum hakkında bilgi toplamanın gerçek amacı hedef kuruma ilk erişimi, ilk sızmayı gerçekleştirmektir. GKT saldırılarında ilk sızma işlemi genelde iyi korunan sunucu bilgisayarlara değil, kurum çalışanların bilgisayarlarına yapılmaktadır. Sızma işlemini sağlayan unsur hünerli şekilde hazırlanmış olan ve kurum çalışanlarına gönderilen zararlı e-postalardır. Hedef kurum çalışanı zararlı içerik barındıran bu e-postayı çalıştırdığı takdirde, GKT aktörleri ilk tutunma noktasını kuruma sızmak için yakalamış olmaktadırlar. Zararlı içerik barındıran bu e-posta, kullanıcıya özel, hünerli bir şekilde hazırlanmış olmasından dolayı kullanıcı için herhangi bir şüphe oluşturmamaktadır. Hedef kuruma ilk erişim ve tutunma noktası sağlandıktan sonra diğer aşama başlamaktadır.

2.3.2. Hedef sistem içinde kontrol ve keşif

Sızma işlemi başarı ile gerçekleşen hedef kurum içinde ilk yapılacak işlem, bilgi sistem altyapısı hakkında bilgi toplama işleminin yapılmasıdır. Root veya yönetici hesaplarının, etki alanı ayrıcalıklı hesapların belirlenmesi ve ele geçirilmesi öncelikli hedeftir.

Hedef kurumun sahip olduğu önemli veriler bu aşamada belirlenir. Bu değerli verilere ulaşmak için bilgi sistem altyapısı içerisinde ilk erişim noktasından başka yatay veya dikey hareketler gerçekleştirilir. Yatay hareketler, bir kullanıcı bilgisayarından başka bir kullanıcı bilgisayarına erişmeyi veya ele geçirmeyi ifade etmektedir. Dikey hareketler ise sunucu parkının olduğu, kurumun değerli verilerinin bulunduğu bilgisayarlara erişmeyi anlatmaktadır. Ele geçirilen ayrıcalıklı hesaplar kullanılarak değerli veriye sahip olan bilgisayarlara ulaşılmaya çalışılır. Bu aşamanın asıl amacı değerli veriyi keşfetmek ve ona ulaşmaya çalışmaktır.

2.3.3. Ağ üzerinde varlığı sürekli kılmak ve değerli veriyi dışarı sızdırmak

Hedef kuruma istenildiği takdirde sürekli olarak erişebiliyor olmak GKT saldırılarının en önemli özelliklerinden biridir. Bu amaç doğrultusunda GKT aktörleri, bilgi sistemlerine sızdıkları kurumların içerisine ileride ihtiyaç olması durumunda kullanılmak üzere arka kapılar bırakmaktadırlar. Bu arka kapılar sayesinde hedef kurum fark etmediği sürece GKT aktörleri, kurumun değerli verisini belirli aralıklarla sızdırmaktadırlar.

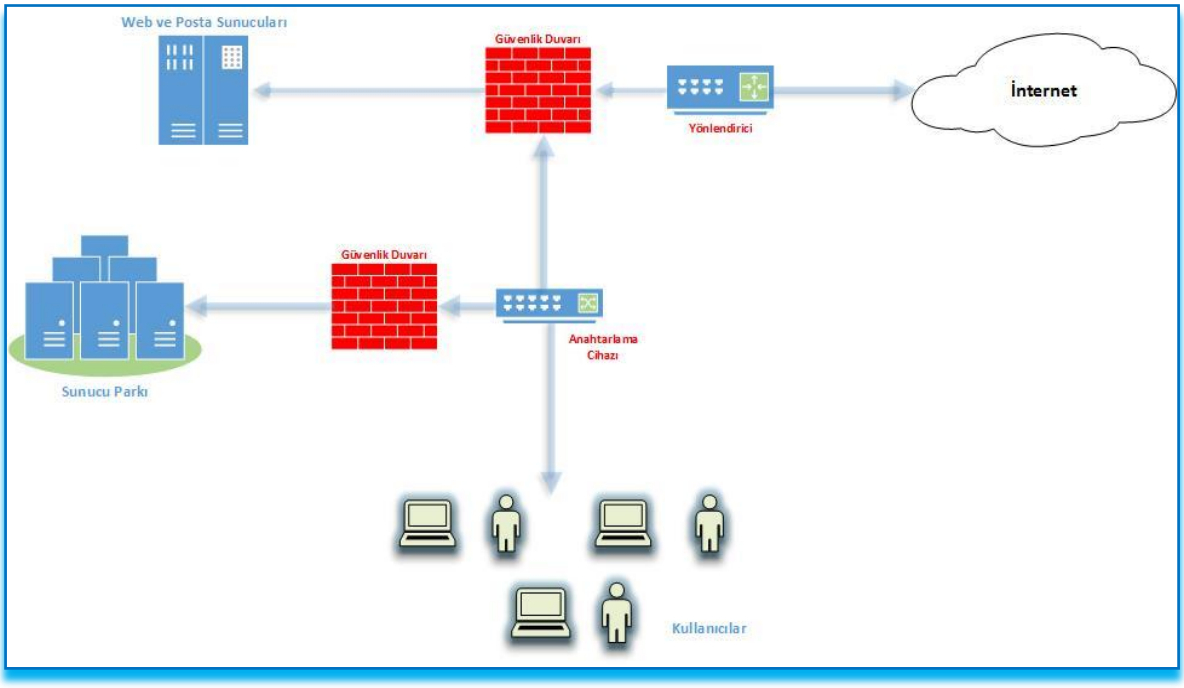
Aşağıdaki GKT saldırısı örnekleri bölümünde açıklandığı gibi bazı GKT saldırıları,

kurumlar fark etmediği sürece yıllarca devam etmektedir. Hedef kurumun ağı üzerinde varlığı sürekli kılmak ve veriyi dışarı sızdırmak GKT saldırısının son aşamasıdır ve bu aşama fark edilmediği sürece yıllarca veri sızdırma işlemi devam edecektir.

2.4. GKT saldırısı örnekleri

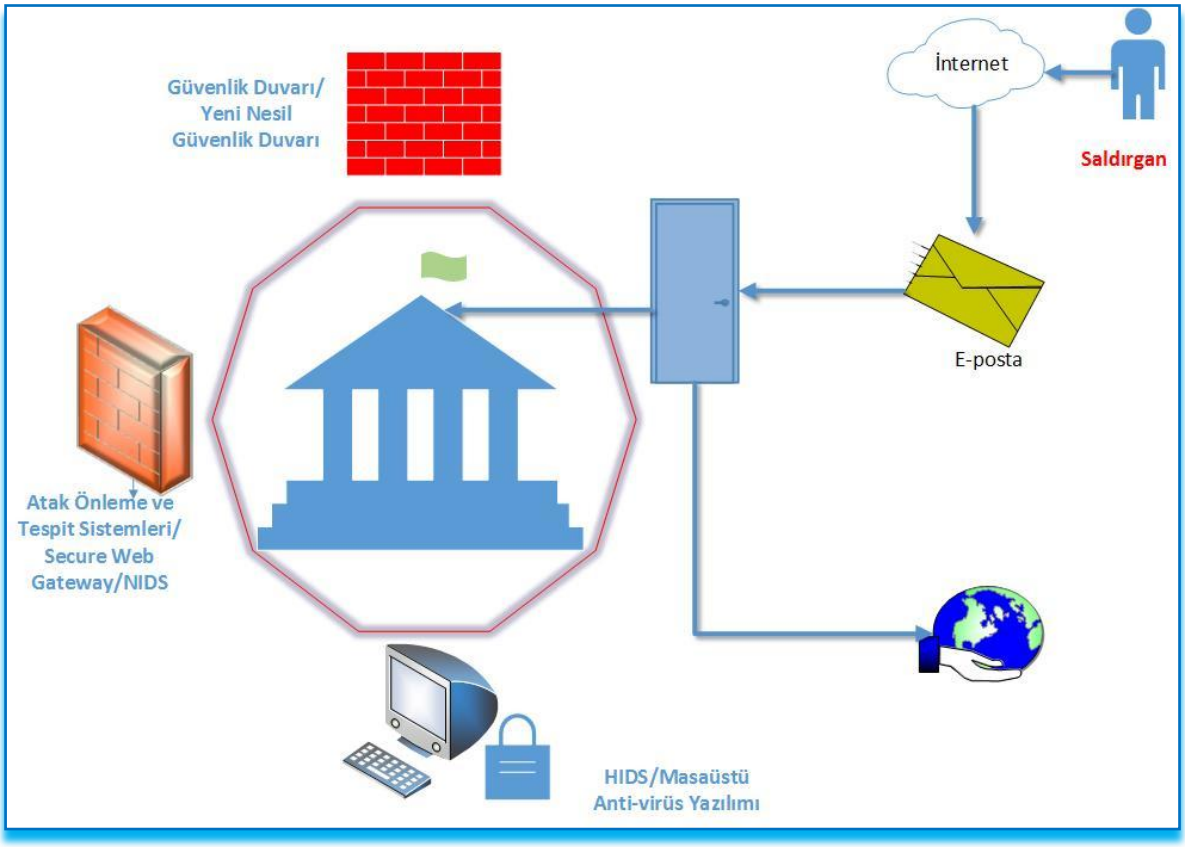
90'lı yıllarda siber güvenlik konusunda yöneticiler herhangi bir endişe taşımıyordu. Kurulan birçok sistem de güvenlik kaygıları ile tasarlanmıyordu. 2000'li yılların başlarına doğru güvenlik anlayışı yavaş yavaş yöneticilerin gündemine girmeye başladı. Geleneksel saldırılar dediğimiz zararlı yazılımlara karşı tüm kurumlar önlemlerini almaya başladı. Günümüzde ise GKT saldırılarını bilen yöneticiler tehlikenin farkındalar fakat ne yapılacağını tam olarak bilmemektedirler.

GKT'lerin kötü tarafı saldırı yaptığı organizasyonu sessizce öldürüyor olmasıdır. Organizasyon her şeyin yolunda gittiğini düşünmekte fakat bir yandan da tüm değerli verileri bir yerlere sızmaktadır. Şekil 2.3'te birçok kurumun genel ağ yapısını gösteren şablon resmedilmiştir. Kurumun internet ortamına açık olan web, posta vb. sunucuları ayrı bir bölgede, kurumun değerli verilerini barındıran sunucular ayrı bir bölgede (sunucu parkı) ve kullanıcı bilgisayarları ayrı bir bölgede bulunmaktadır. Bu bölgeler arasındaki iletişim kontrollü olarak güvenlik duvarları üzerinden geçmektedir. Güvenlik bilinci tam yerleşmemiş kurumlarda ise kimi zaman sunucu parkının önünde güvenlik duvarı bulunmamaktadır. Kurumun kullanıcı bilgisayarları, sunucu parkında bulunan sunuculara arada güvenlik duvarı olmadan doğrudan erişmektedirler.



Şekil 2.3. Kurumların genel ağ yapısı

GKT aktörleri ağda uzun süreli olarak kalabilmek için siber güvenlik tedbirlerini atlatarak çok karmaşık saldırılar gerçekleştirmektedirler. GTK saldırıları gizli, hedefi olan ve değerli veri odaklı olan saldırılar olup geleneksel saldırılardan farklıdırlar. GTK unsurları iyi organize olmuş birimlerdir. Genel olarak yabancı düşman odakları ve organize suç örgütleridir. Öncelikli amaç değerli veriyi çalmak, uzun süreli olarak ağda varlığını devam ettirmek ve gelecekte de ihtiyaç olması durumunda tekrar bilgi sızdırmak için arka kapılar bırakmaktır. GKT unsurları, hedef aldığı kurumun veya organizasyonun tüm güvenlik tedbirlerini atlatarak, genellikle kurumun kullanıcılarını giriş noktası olarak kullanarak saldırılarını gerçekleştirmektedirler. Geleneksel savunma mekanizmaları bu tür saldırılar üzerinde etkili değildirler. Şekil 2.4'te gösterildiği gibi web ve e-posta kapıları kurumlar için her zaman açık bir kapıdır. Saldırganlar genel olarak web ve e-posta yolu ile hedef kurumun kullanıcılarına ve bu sayede de kurumun bilgi sistemleri kaynaklarına ulaşabilmektedirler [16, 20, 51].



Şekil 2.4. Kurumların güvenlik altyapısı

GKT saldırılarının tespit edilmesi ve yapısının ortaya çıkarılması işlemi çok büyük uzmanlık gerektirmektedir. Kurumlara yapılan GKT saldırıları çok uzun zaman sonra fark edilmekte ve GKT saldırısı olup olmadığı ise uzman incelemeler sonunda anlaşılabilmektedir. Medya raporları, güvenlik firmalarının yayınladığı yıllık bültenler, itibar sahibi güvenlik danışmanı firmaların hazırladığı raporlar doğrultusunda GKT olarak nitelendirilen siber saldırılardan bazıları aşağıda açıklanmıştır. Teknik yönden çok karmaşık olan bu saldırıların her biri, ayrı bir kitap oluşturacak kadar ayrıntıya sahiptirler. Sadece önem arz eden özellikleri hakkında kısa bilgiler verilmiştir.

2.4.1. Stuxnet

Stuxnet, endüstriyel kontrol sistemlerinin ve dış dünyaya kapalı sistemlerin de hedef olabileceğini göstermesi açısından siber güvenlik konusunda önemli bir yere sahiptir. İlk incelemelerde virüsün standart bir zararlı yazılım olmadığı anlaşılmış ve uzayan incelemeler devam ettikçe karmaşıklığı fark edilmiştir. Haziran 2010'da varlığı açığa çıkan

virüs İran'ın Buşehr ve Natanz'daki nükleer tesislerini etkilemiştir [30, 45].

2.4.2. Operation Aurora

Saldırıyı Google tespit etmiş olup kamuoyuna 12 Ocak 2010 tarihinde açıklama yapmışlardır. Saldırı 2009 yılı Haziran aylarında başlamış, dünyaca ünlü teknoloji şirketlerini hedef almıştır. Medya raporlarına göre etkilenen kuruluşlar arasında Adobe, Symantec, Yahoo, Dow Chemical, Northrop Grumman, Google bulunmaktadır. MS IE 6 sıfır gün açıklığında faydalanılmıştır. Açıklığın etkili olabilmesi için kullanıcı müdahalesi gerekmektedir. Güvenlik şirketi McAfee'den George Kurtz'a göre komuta kontrol sunucuları Tayvan'da bulunmaktadır. Aurora'nın asıl amacının kaynak kod hırsızlığı ve değiştirilmesi olduğu düşünülmektedir [19, 37, 46].

2.4.3. DuQu

Kod temeli ve yapısı açısından Stuxnet ile büyük benzerlikler göstermektedir. DuQu, Stuxnet değildir fakat tasarım ve kodlandırma filozofisi bunu düşündürmektedir. Ancak amaçları açısından bakıldığında birbirlerine hiç benzememektedirler. Stuxnet özellikle sabotaj için kullanılmış, DuQu ise uzaktan erişim metodu ile kurumlardan bilgi toplamak ve çalmak için kullanılmıştır. DuQu modüler bir yapıya sahip olup, bilgi çalma (infostealer), tuş kaydedici, haberleşme gibi modüllerden oluşmaktadır. Saldırı adını tuş kaydedici modülünün yarattığı ~DQ dosyalarından almaktadır. Modüler yapıda yazılmış zararlı yazılımlara iyi bir örnektir.

DuQu zararlı yazılımı komuta kontrol sunucuları ile 30 gün içinde bağlantı kuramaz ise tespit edilmekten sakınmak maksadıyla kendisini otomatik olarak kaldırmaktadır. Stuxnet'te olduğu gibi geçerli bir dijital sürücü imzasına sahiptir [35, 38].

2.4.4. Red October

Uluslararası seviyede diplomatik hizmet birimleri hedef alınmıştır. Diplomatik hizmet birimleri hakkında bu zararlı yazılım vasıtası ile istihbari bilgi toplanmıştır. Zararlı yazılımın etkili olduğu bölgeler arasında Doğu Avrupa ve Merkez Asya bulunmaktadır. Saldırının 2007 ve 2013 yılları arasında çok aktif olduğu değerlendirilmektedir. Az da olsa

hala aktif olduğu makinelerin olduğu görülmektedir [34].

Resim 2.1. Red October saldırısı aktörleri tarafından gönderilen zararlı e-posta örneği

Saldırının arkasında bulunan aktörler sadece kişisel tip bilgisayarlardan veya dizüstü bilgisayarlardan değil, mobil cihazlardan da bilgi çalmıştır. Hatta taşınabilir disk ünitelerinde bulunan silinmiş dosyalardan bile kurtarma yaparak bilgi çaldığı tespit edilmiştir. Zararlı kodlar e-posta vasıtası ile Excel, Word, PDF formatında ek olarak gönderilmiştir. Ayrıca Java tarayıcı eklentisi açıklığı kullanılarak da sistemlere sızılmaya çalışılmıştır. 60 farklı etki alanında çalışan komuta kontrol sunucuları, daha sonra gönüllü olan hosting firmaları vasıtası ile etkisiz hale getirilmiştir.

2.4.5. Miniduke

Son zamanlarda ortaya çıkan GKT saldırılarından biridir. Diğer GKT saldırılarının kullandığı komuta kontrol sunucularını kullandığı gibi kurbanlara komut göndermek için Twitter gibi sosyal ağ hesaplarını kullanabilmektedir. Kurban bilgisayarlar genel olarak diplomat, bilgi sistem, enerji, askeri ve telekom şirketlerine aittir. İlk kurban bilgisayar Nisan 2012 tarihinde belirlenmiştir. Miniduke GKT saldırısı Adobe 9, 10, ve 11 yazılımlarına ait sıfırıncı gün açıklığından faydalanarak kurban sistemlerine sızmayı başarmıştır. 20 KB büyüklüğündeki backdoor yazılımı makine dilinde yazılmıştır. Kodu yazanların iyi derecede makine dili birikimine sahip olduğu düşünülmektedir [33].

2.4.6. Regin

Regin ilk olarak 2012 yılı bahar aylarında fark edilmiştir. Kamuoyu tarafından bilinmesi 2014 yılının sonlarına denk gelmektedir. Regin zararlı yazılımının başlangıcının 2003 yılına kadar gittiği düşünülmektedir. Kurban bilgisayarların finans, devlet ve araştırma enstitüleri (özellikle kripto araştırma enstitüleri), politik kuruluşlar ve haberleşme şirketleri olduğu görülmektedir. Bireysel kurbanlardan birisi de ünlü kripto araştırmacısı Jean Jacques Quisquarter'dır. Medya kuruluşlarına göre Regin saldırısının arkasında Amerikan ve İngiliz istihbarat ajanslıklarının olduğu değerlendirilmektedir. Regin saldırısı ile Avrupa Birliği ülkeleri ve şirketleri hakkında geniş çaplı bilgi toplama işlemi gerçekleştirilmiştir. Alman gazetesi Bild'e göre Regin zararlı yazılımı, Angela Merkel'in bir yardımcısının taşınabilir diskinde bulunmuştur. Regin'in en büyük özelliği 10 yıl gibi uzun bir süre fark edilmeden bilgi toplaması ve GSM ağları üzerinde istihbari faaliyetler yapabiliyor olmasıdır [32, 43, 44].

2.4.7. Gauss

Gauss, kodlama yapısı açısından Flame ve DuQu zararlı yazılımlarına benzemektedir. Ancak Gauss her iki zararlı yazılımdan da daha fazla oranda yayılmıştır. Kurban bilgisayarlar genel fiziksel olarak Orta Doğu bölgesinde yer almaktadır. C++ programlama dilinde yazılmıştır. Çoğu yazılım güvenlik şirketi Stuxnet, DuQu, Flame ve Gauss'un aynı saldırı aktörleri tarafından yazıldığını belirtmektedir. Bu dört zararlı yazılım yapı olarak birbirine çok benzemektedir. Gauss zararlı yazılımı Fransabank, Byblos Bank ve Bank of

Beirut gibi bankaların çevrimiçi servis hizmetlerini kullanan kullanıcıların, kullanıcı yetkilerini ele geçirmeye çalışmıştır [31].

Yukarıda bahsedilen GKT saldırılarının bazı önemli özellikleri Çizelge 2.1'de özet olarak gösterilmiştir. GKT saldırılarının yapılarına bakıldığında kullandıkları araçların çoğunlukla özel araçlar olduğu görülmektedir. Komuta kontrol sunucuları ile haberleşmek için güvenlik duvarlarından çoğunlukla izin verilen HTTP/S protokollerinin kullanıldığı görülmektedir. Kurumların iç ağlarında proxy sunucuları kullanıyor olması nedeniyle komuta kontrol sunucuları ile haberleşmek için proxy'i sezebilen yapıda zararlı yazılımlar geliştirilmiş olduğu gözlemlenmektedir. Sabotaj amaçlı kullanılmış olan Stuxnet dışında diğer saldırıların kurumların değerli verisini çalmak için kullanıldığı söylenebilir. İlk sızma işlemini gerçekleştirmek için GKT unsurların genelde sıfırıncı gün açıklığını kullandığı çizelgeden anlaşılmaktadır. Bunun yanında Stuxnet'te yapıldığı düşünülen, içeriden bir ajan vasıtası ile de sızma işleminin gerçekleştirildiği görülmektedir.

Çizelge 2.1 Gelişmiş kalıcı tehdit saldırılarının karşılaştırılması.

GKT Saldırıları	Hedef	Kurban Bölge	Kullanılan uzaktan erişim araçları	İlk sızma metodu	Komuta Kontrol Haberleşmesi
OPERATION AURORA (Haziran –Aralık 2009)	Google'ın yer aldığı yüksek teknoloji firmaları.	İngilizce konuşulan yüksek teknoloji ülkeleri	Özel	Internet Explorer 0-day açıklığı	HTTP ve HTTPS, proxy-aware.
STUXNET (2009-2010)	Endüstri kontrol sistemleri	İran	Özel	İçerideki bir ajan veya taşınabilir disk vasıtası ile yapıldığı tahmin edilmektedir	HTTP
DUQU (2010-2011)	Bazı şirketlerin dijital sertifikalarını çalmak (McAfee raporuna göre).	Fransa, Hollanda, Amerika, İngiltere, İran, Sudan, Vietnam ve diğer bazı ülkeler	Özel	Word dokümanı 0-day açıklığı	HTTP ve HTTPS, proxy-aware.
RED OCTOBER (2007-2013)	Diplomatik birimler	Geniş çapta birçok ülke	Genel (internette bulunabilir)	Excel, Word ve PDF dokümanlarının bilinin güvenlik açıklıkları	HTTP protokolü ile birlikte 60 farklı etki alanı
MINIDUKE (COSMICDUKE) (2012-2014)	Devlet kurumları, diplomatik, enerji, telekom şirketleri, askeri	Avusturalya, Belçika, Fransa, Almanya, Macaristan, İspanya, Ukrayna, Amerika	Genel (internette bulunabilir)	Sahte Java, Acrobat Reader Updater, Chrome programları	Sosyal ağ hesapları, Twitter.
REGIN (2003-2014)	GSM şirketleri, devlet kurumları, teknoloji şirketleri	Afganistan, Cezayir, Belçika, Brezilya, Fiji, Almanya, Hindistan, Endonezya, İran, Kiribati, Malezya, Pakistan, Rusya, Suriye	Özel	Tam olarak bilinmemekle birlikte man-in-the-middle attacks ile browser zero-day açıklığı olduğu değerlendirilmektedir	HTTP, HTTPS ve P2P protokolleri.
GAUSS (2011-2012)	Banka ve finans şirketleri	Ortadoğu ve Lübnan	Özel	USB sürücüsü .LNK açıklığı	HTTPS

2.5. GKT saldırılarına karşı alınması gereken güvenlik önlemleri

Saldıran tarafın işi her zaman savunan tarafın işinden daha kolaydır. Saldıran düşmanın sadece bir tane başarılı olacak saldırıyı biliyor olması yeterlidir ancak savunan tarafın başarılı olabilmesi için her zaman daha fazla şey biliyor olması gerekmektedir.

Saldırının erken fark edilmesi, hemen reaksiyon verilmesi, zararın hızlı bir şekilde telafi edilmeye başlanması çok büyük önem arz etmektedir. 6 ay gibi bir süre fark edilmeyen bir GKT saldırısı kurum için çok acı verici olacaktır. Siber saldırı ile kurumun tüm bilgilerinin ele geçirileceği gerçeği çoğu yönetici için kabul edilmesi zor bir durumdur.

Fonksiyonellik ve güvenlik arasında ters bir ilişki bulunmaktadır. Ne kadar çok fonksiyonel bir sistem kurulursa, o oranda da güvenlik azaltılmış olur. Bilgisayar elektrik pirizine takıldığında, bilgisayara klavye ve fare bağlandığında, veri paylaşımı için bir ağa bağlandığında, hele bu ağ internet ise, bu yapılan hareketlerin her biri ile güvenlik %100'ün altına düşmeye başlamış demektir. Hemen değil ama mutlaka zamanla bir güvenlik sorunu oluşacaktır.

GKT saldırılarını yapan aktörler mağazaya normal bir müşteriymiş gibi giren fakat hırsızlık yapan kimselere benzetilebilirler. Mağaza içerisinde bulunan tüm müşterilerin yaptığı hareketlerin takip ediliyor olması mağaza güvenliği için çok önemlidir. Mağazada %100 güvenlik için içeriye ve dışarıya kimsenin gitmesine izin vermemeniz gerekir. Bu da firmanın batması anlamına gelmektedir. Böyle bir önlem alınamayacağına göre tek yöntem hırsızlığı tespit etmeye çalışmaktır. Buradaki önemli nokta eğer hırsız mağazaya geldiğinde 5 dakika kalıyor ve hırsızlığı yapıyor ise sizin 5 dakikadan daha az zamanınız var demektir. Bu durum GKT saldırılarında da geçerlidir. Hırsızlık olduktan sonra akşam kamera kayıtlarında olayı tespit etmek çoğu zaman iş işten geçti demektir.

GKT saldırılarına karşı başarılı olmanın tek yolu önleyici tedbirlerden daha çok tespit edici sistemlerin bulunmasıdır. Çünkü GKT aktörleri sizin izin verdiğiniz yollardan normal bir kullanıcıymış gibi sisteme girecektir. Her kurumun sistem yapısını önüne koyup, önleyici sistemlere Ö, tespit edici sistemlere T dersek kaç tane Ö ve kaç tane T'ye sahip olduğuna bakması gerekmektedir. Sistemde Ö'lerin bulunması tabiki önemlidir fakat GKT

saldırılarında Ö'lerin bir önemi yoktur çünkü saldırgan izin verilen, yasal olan normal kullanıcının kullandığı yoldan sisteme giriş yapacaktır. GKT saldırısını tespit edecek olan sistemde yer alan T'lerdir. Bu T'lerin de sık sık kontrol edilmesi ve izlenmesi gerekmektedir.

Güvenliği tek bir noktada toplayıp bu nokta aşıldığı takdirde savunmasız kalınan bir güvenlik anlayışı doğru bir güvenlik anlayışı değildir. Birçok noktada güvenlik önlemi almak ve en değerli veri önündeki güvenliği artırmak doğru bir yaklaşım olacaktır. Organizasyonun güvenlik yapısı bir kale gibi sağlam olmalı ve sistem yöneticilerinin yaptıkları güvenlik yapılandırmalarını çok iyi biliyor olmaları gerekmektedir. Kaleyi örnek alacak olursak, kale yüksek duvarlardan oluşmalı, etrafı derin hendeklerle çevrili olmalıdır. Giriş dar bir yoldan sağlanmalı ve girenlerin mümkünse eğilerek giriyor olması gerekmektedir. Yukarıya giden merdivenler soldan sağa doğru kıvrılmalı, basamaklar aynı yükseklik yerine farklı yüksekliklerde olmalı ve genişlikleri her zaman değişmelidir. Merdivenlerin olduğu bölüm tam aydınlatma yerine loş olmalıdır. Merdivenlerin soldan sağa doğru yukarıya çıkıyor olmasının nedeni çoğu insanın sağ elini kullanıyor olması nedeniyle kılıcı sallarken zorluk yaşamasını sağlamaktır. Yukarıdan gelen insan ise tam tersine daha rahat edecektir. Merdiven basamaklarının ise çeşitli boylarda yapılmasının nedeni bu basamakları yeni kullanan düşman unsurların zorluk çekmelerini sağlamaktır. Bu merdivenleri her gün kullanan dost unsurlar boyutlara alışkın olduğu için zorluk yaşamayacaklardır.

Bir kale bile kimsenin belki de fark etmediği bu kadar ayrıntılı olan güvenlik önlemlerine sahipken kurumların da mutlaka güvenlik hususunda ayrıntılı şekilde yapılandırmaları olmak zorundadır. Her önlemin saldırganın işini zorlaştıracak şekilde birbirini tamamlıyor olması gerekmektedir. Değerli olan veriye gidildikçe güvenliği artırılması ve log kayıtları tutan sistemlerin bulunması saldırı tespitinin yapılması için hayati önemi vardır. GKT saldırıları önleme yapan cihazlar vasıtasıyla değil de daha çok izleyen, loglara bakan sistemler vasıtası ile tespit edilebilecektir [63, 64]. Son zamanlarda oldukça yaygın bir şekilde kullanılmaya başlanan bulut bilişim teknolojisi de güvenlik konusunda başka zorluklar ortaya çıkarmıştır. Bulut bilişimin getirdiği büyük ağ trafiği, çok büyük veri boyutu güvenliği zorlaştıran durumlardan bazılarıdır [62].

Alınan güvenlik önlemlerinin kale örneğinde olduğu gibi savunma yapanın işini kolaylaştıracak, hücum yapanın işini zorlaştıracak şekilde olmalıdır.

GKT saldırıları konusunda kurum proaktif mi, reaktif mi olacak sorusu önemlidir. Bu soruyu şuna benzetebiliriz; bina yangına karşı korumalı olacak ve önlemler alınacak mı yoksa yangın olduktan sonra binayı tekrar mı inşa edeceğiz. Yangına karşı önlem almak mutlaka yangından sonra tekrar inşa etmekten daha az maliyetli olacaktır.

Küçük yaşlardan itibaren yolda yürürken gördüğümüz yiyecekleri yememem iz, tanımadığımız insanlara güvenmememiz, onlardan yiyecek almamamız gerektiği anlatılır. Biliriz ki bu yiyecek bizim sağlığımızı bozabilir veya bu tanımadığımız kişiler bize zarar verebilir. Konu siber güvenlik olduğunda bazı konularda sağduyu kaybı yaşanabiliyor. Çoğu şirket çalışanı yolda bulduğu USB cihazını bilgisayarına içinde ne var diye bağlayabiliyor, tanımadığı insanlardan gelen e-postalara bakıp eklentileri açabiliyor. Bu sağduyu kaybını siber tehdit algılarının hayatımıza sonradan girmesine bağlamak gerekir. Çünkü küçük yaşlardan itibaren hayatımızda yeri olan ve uyarı aldığımız konular değildir. Peki hangisi daha önemli siber güvenlik mi yoksa sağlık mı? Daha açık düşünmek gerekirse yolda bulduğunuz çikolatayı yiyip bir hafta hastalanmak mı yoksa yolda bulduğunuz USB çubuğu bilgisayara taktıktan sonra bulaşan zararlı yazılım ile kredi kartı bilgilerizi çaldırıp, kart limiti olan 30 bin TL kaptırmanız mı daha kötü?

GKT saldırılarının daha çok oranda bilinmesiyle beraber kurumlar ve bireylerin siber güvenlik algısı da daha farklı bir boyut kazanacaktır. İnternet, kurumların ve bireylerin hayatını değiştirdiği gibi onları birçok yeni tehlikeler ile tanıştırdı. Bu yeni tehlikeleri fark eden ve güvenlik konusunda sağduyusunu kaybetmeyen kurum ve bireyler daha az zarar göreceklerdir.

Siber güvenlik konusunda iyi şeyler yapmak ile doğru şeyler yapmak arasında farklar vardır. Güvenlik konusunda para harcamak iyidir fakat doğru tedbirler almıyorsanız güvenlik için para harcamanız kurumu daha güvenli yapmayacaktır. Gelişmiş yeteneklere sahip olan bir düşmana karşı geleneksel güvenlik önlemleri ile (güvenlik duvarı, saldırı önleme/tespit sistemleri, anti-virüsler vb.) başarı sağlanamaz. Değerli olduğu düşünülen veriler için güvenlik tedbirleri alınmadan önce risklerin tespit edilmesi, en yüksek önceliğe sahip riskin belirlenmesi, riski azaltacak en etkin yolun seçilmesi gerekmektedir. Riskler

belirlendikten sonra sistemde var olan zayıflıkların giderilmesine öncelikle en yüksek önceliğe sahip riskten başlanmalıdır. Eğer riskin olmadığı bir zayıflığın giderilmesinde çaba harcanıyor ise iyi bir şey yapılıyor fakat doğru şey yapılmıyor demektir.

Çoğu kimse ele geçirilen sistemleri veya sızdırılan bilgileri haberlerden okuduğunda saldırıya maruz kalan firmaların bariz güvenlik hataları yaptığını düşünebilir. Ancak durum düşünüldüğü gibi değildir. Saldırı anında firmaların hemen hemen hepsinde güvenlik politikaları, güvenlik için bütçeleri, güvenlik ekibi, güvenlik duvarları, uygulama güvenlik duvarları, saldırı tespit/önleme sistemleri, anti-virüs yazılımları vb. güvenlik unsurları bulunmaktadır. Çoğu güvenlik elemanına sorsanız bunların kesinlikle bulunması gerektiğini söyleyecektir. GKT saldırılarında bu alınan güvenlik tedbirleri sadece iyi olan şeylerdir ama yeterli değildir, aynı zaman doğru şeyleri yapmak gerekmektedir. Önleyici önlemlerin yanında tespit edici sistemlerin de kullanılması ve 7/24 izlenmesi gerekmektedir. 50 tane risk oluşturmayan açıklığın kapatılması için uğraşmak yerine 1 tane yüksek risk taşıyan açıklığın giderilmesi için çaba harcanmalıdır. Kurumun taşıdığı risklerin mutlaka önceliklendirilmesi yapılmalıdır.

Devletlerin, kurumların, organizasyonların icra ettikleri faaliyetlerinden dolayı aldıkları riskler vardır. Aslında iş yapmanın taşıdığı risklerdir bunlar. Örneğin bir bankanın internet ortamında çalıştırdığı bir servis iş gereği alması gereken bir risktir. İş yaptığı sistemden kaynaklanan bir sorun nedeniyle saldırıya uğraması normaldir. Ancak banka, iş gereği kullanmadığı bir sistemden kaynaklanan sorun nedeniyle saldırıya uğruyorsa üzülmekte son derece haklıdır. Organizasyonlar saldırı alanlarını çok dar tutmalıdırlar sadece gerçek iş maksatlı servisleri açık olmalıdır. Diğer gereksiz servisler kapalı olmalıdır.

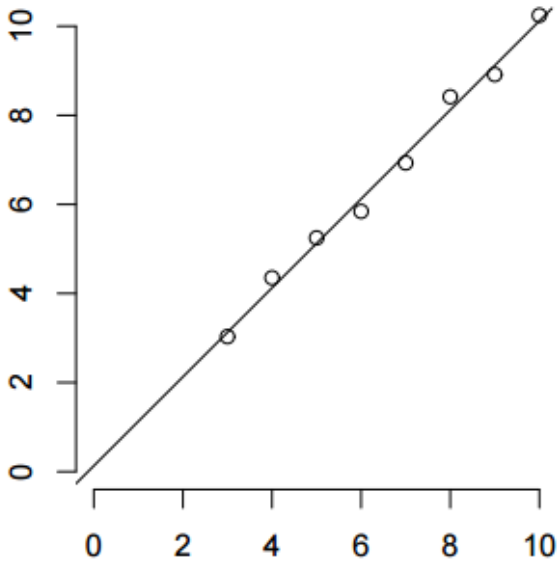
Birçok yemleme e-postası, html gömülü özellik aktif ise daha kolay kurum personelini kandırabilmektedir. Kurumun bu özelliğe ihtiyacı yok ise mutlaka kapatılmalıdır. Bu özelliğin kapatılması ile birçok yemleme e-postasının önüne geçilebilecektir. GKT aktörleri yemleme e-postalarını çok fazla kullanmaktadırlar. Saldırgan, kullanıcıyı e-posta içerisindeki bir link ile zararlı kodun olduğu veya zayıflığı tetikleyen siteye yönlendirmektedir [21-29, 39].

3. TEKİL DEĞER AYRIŞTIRMASI

Tekil değer ayrıştırması (TDA) (Singular Value Decomposition–SVD), bir matrisin çarpanlarına ayrılma türlerinden biridir. Gürültülü veriyi azaltmada ve boyut küçültmede geniş ölçüde kullanılan bir tekniktir. TDA ile asıl matrisin alçak dereceden (low rank) en iyi doğrusal yakınsamasını sağlamaktadır [17].

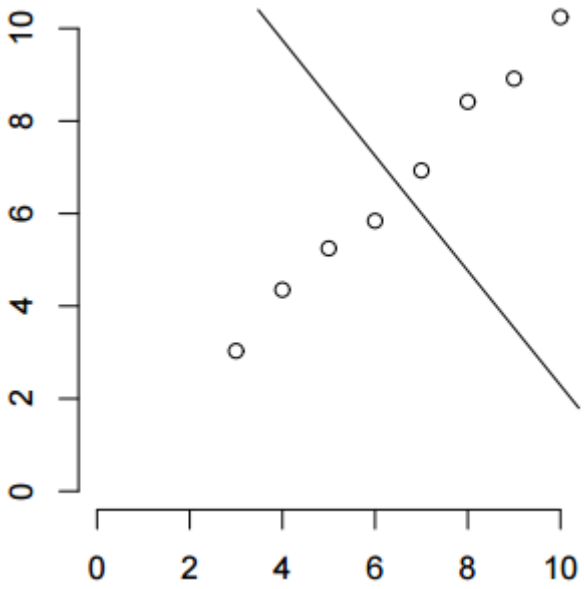
İki boyutlu uzayda düşünülecek olursa, Şekil 3.1'de yer alan doğrusal çizgi dağıtık durumda bulunan noktaları temsil eden en iyi çizgidir. Noktalardan dik olarak doğruya indirilen dik çizgilerin toplam uzunluğu, bu doğru için en küçüktür. Tekil değer ayrıştırmasının arkasındaki basit anlayış, çok boyutlu veri kümelerini azaltarak daha düşük boyutta aynı veri kümesini en iyi şekilde temsil edecek veri kümelerini hesaplamaktadır.

Şekil 3.1'de yer alan doğruya dik olarak çizilen diğer doğrular ise noktaları temsil konusunda başarısız olacaktır. Bu durumda noktalara ilişki açısından uzak olan doğrular da ortaya çıkmaktadır. TDA yakın ilişkileri ortaya çıkardığı gibi uzak ilişkileri de ortaya çıkarmaktadır.



Şekil 3.1. Noktaları temsil eden en iyi doğrusal çizgi

Şekil 3.2'de yer alan doğru, noktaları temsil konusunda uzak olan doğrulardan biridir. Temsil noktasında en uzak olan doğru, temsil noktasında en iyi olan doğruya her zaman diktir.



Şekil 3.2. Noktaları temsilde uzak olan doğrusal çizgi

3.1. Tekil değer ayrıştırması (TDA) matematiksel tanımı

Lineer cebirde gerçek veya karmaşık bir matrisin çarpanlara ayrıştırılması işlemine tekil değer ayrıştırması denir. Tekil değer ayrıştırması ile *mxn A* matrisi 3 parçaya ayrılır. Eşitlik 3.1'de gösterildiği gibi ifade edilir.

$$A_{mxn}=U_{mxm}\ S_{mxn}\ V^{T}_{nxn} \tag{3.1}$$

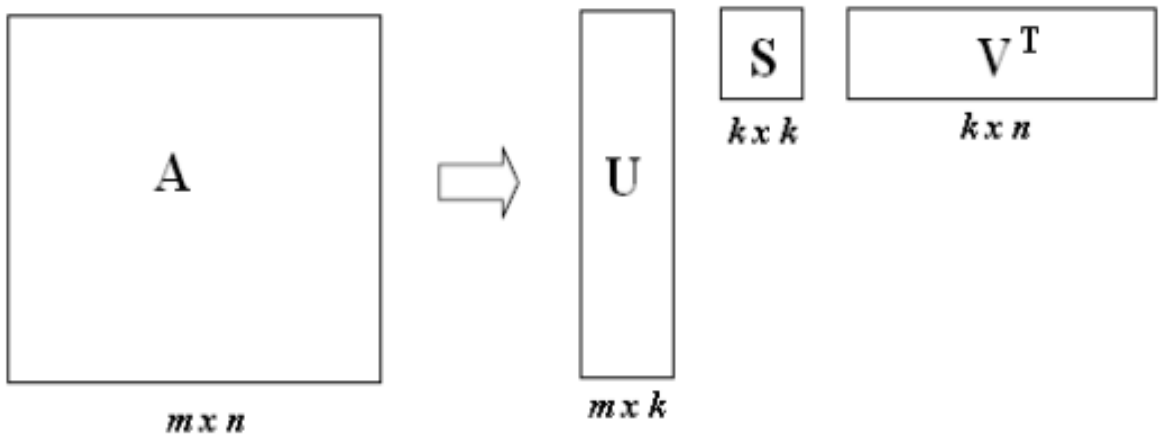
Eşitlikteki *U* ve *V* matrisleri ortogonal matrislerdir ve $\mathrm{U^TU} = \mathrm{V^TV} = \mathrm{I_n}$'dir.
S matrisi köşegen (diagonal) matris olup, tekil değerleri içermektedir ve
$S = \mathrm{diag}(\alpha_1, \ldots., \alpha_n)$, $\alpha_i > 0$ ve $\alpha_1 > \alpha_2 > \cdots > \alpha_n > 0$ 'dır.

α_i, A matrisinin tekil değerleridir. U ve V matrisleri sol ve sağ tekil değerlerdir.

$$Av_i = \alpha_i u_i \quad \text{ve} \quad Av_i = \alpha_i u_i \tag{3.2}$$

Eşitlik 3.2'deki u_i ve v_i vektörleri sırasıyla U ve V matrislerinin i^{th} sütunlarıdır.

U matrisinin sütunları sol tekil vektörleri, V matrisinin sütunları ise sağ tekil vektörleri içermektedir. Şekil 3.3'te A_{mxn} matrisi için tekil değer ayrışımının görsel olarak gösterimi bulunmaktadır.



Şekil 3.3. TDA'nın görsel gösterimi

S matrisinin köşegenleri üzerinde yer alan sayılar tekil değerleri oluşturmaktadır ve sıfırdan büyük sayılardır. S matrisi içerisinde yer alan diğer sayıların değerleri sıfırdır. S matrisi içinden r satırına kadar olan değerler alındığı takdirde aşağıdaki eşitlikte gösterildiği gibi A'_{mxn} matrisi elde edilecektir.

$$A'_{mxn} = U_{mxr}\ S_{rxr}\ V^T_{rxn} \tag{3.3}$$

Yeni A'_{mxn} matrisi *U* matrisindeki r adet sütun değerleri ile *V* matrisindeki r adet satır değerleri ile oluşturulur. Oluşan bu yeni matris önceki A matrisine yakındır. Hatta bazı araştırmalara göre üzerindeki gürültüden arındırılmış olması nedeni ile daha sağlıklıdır. Bu işlem ile boyut azaltma gerçekleştirilmiş olmaktadır [65, 66].

3.2. Tekil değer ayrıştırması örneği

A matrisi m=5, n=5;

$$A_{5x5} = \begin{matrix} 1 & 0 & 2 & 1 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 2 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 0 & 2 & 3 & 1 & 1 \end{matrix}$$

şeklinde olsun. Tekil değer ayrıştırması uygulandığı takdirde U, S, V matrisleri aşağıda gösterildiği gibi hesaplanacaktır.

$$U = \begin{matrix} -0{,}610 & -0{,}068 & 0{,}690 & 0{,}317 & -0{,}216 \\ -0{,}033 & -0{,}150 & -0{,}184 & -0{,}319 & -0{,}917 \\ -0{,}386 & 0{,}385 & -0{,}603 & 0{,}568 & -0{,}125 \\ -0{,}436 & 0{,}602 & 0{,}019 & -0{,}654 & 0{,}141 \\ -0{,}537 & -0{,}679 & -0{,}354 & -0{,}219 & 0{,}278 \end{matrix}$$

$$S = \begin{matrix} 5{,}773 & 0 & 0 & 0 & 0 \\ 0 & 3{,}172 & 0 & 0 & 0 \\ 0 & 0 & 2{,}202 & 0 & 0 \\ 0 & 0 & 0 & 1{,}540 & 0 \\ 0 & 0 & 0 & 0 & 0{,}628 \end{matrix}$$

$$V^T = \begin{matrix} -0{,}466 & -0{,}192 & -0{,}633 & -0{,}332 & -0{,}485 \\ 0{,}791 & -0{,}476 & -0{,}374 & 0{,}007 & -0{,}089 \\ -0{,}209 & -0{,}405 & -0{,}121 & -0{,}395 & 0{,}788 \\ -0{,}330 & -0{,}491 & -0{,}069 & 0{,}802 & 0{,}052 \\ -0{,}070 & -0{,}576 & 0{,}664 & -0{,}301 & -0{,}364 \end{matrix}$$

3.3. TDA kullanım alanları

Google'ın PageRank algoritmasından insan yüzlerini modellemeye, gen analizinden bilgi getirimine ve çıkarımına, boyut azaltmadan metin sınıflandırmaya, temel bileşenler analizinden sinyal işlemeye, veri sıkıştırmadan yüz tanımaya kadar uzanan geniş bir yelpazede kullanılan temel bir adımdır [65, 66]**.**

Tekil değer ayrıştırma yöntemi tamamen matematiksel bir yöntem olmasına rağmen doğal dil işleme uygulamalarında, çoğu kelimenin birden çok anlamı (sense) bulunduğundan, çok anlamlı bir kelimenin içinde geçtiği bağlamdan (context) faydalanarak ifade edilmek istenilen anlamın bulunması problemine belli oranda çözüm getirmektedir. Doğal dil işlemede kullanılan doküman-terim matrisinde TDA uygulandığı takdirde eş anlamlı kelimeler arasındaki ilişki ortaya çıkabilmektedir.

Veri kaybının önemli olmadığı veri ve resim sıkıştırma algoritmalarında da TDA kullanılabilmektedir. TDA'nın boyut azaltma özelliğinden faydalanılarak veri sıkıştırma işlemi yapılabilmektedir.

4. LİTERATÜR TARAMASI

Geleneksel siber saldırı örnekleri 90'lı yıllardan günümüze kadar var olmasına rağmen GKT siber saldırıları son yıllarda ortaya çıkmakta ve kurumlar tarafından yeni yeni fark edilmeye başlamaktadır. GKT saldırı tespitine yönelik literatürdeki çalışmalar konunun yeni olmasından dolayı fazla olmamakla birlikte, literatürde yer alan bazı çalışmalar aşağıda açıklanacaktır.

GKT saldırılarının doğası gereği, gerektiğinde sıfırıncı gün açıklığı kullanılması nedeniyle tespit edilmeleri veya bir kutu çözümle GKT saldırı tespiti yapılması zor gözükmektedir. Ancak GKT saldırısı esnasında, normal kullanıcı davranışından farklı olarak, anomali denebilecek ağ trafik davranışı mutlaka ortaya çıkacaktır. Bu yüzden saldırı tespitine yönelik olarak imza tabanlı tespit yapan sistemler GKT saldırılarına karşı çaresiz kalmaktadır. İmza tabanlı sistemler sadece bilinen saldırılara karşı etkili olabilmektedirler. GKT saldırı tespiti yapabilecek olan sistemlerin, ağ trafiğinde meydana gelen anomalileri ve aykırılıkları belirleyebiliyor olması gerekmektedir. GKT saldırıları çok aşamalı saldırılar (sosyal mühendislik, istihbari bilgi toplama ve keşif, komuta kontrol haberleşmesi, bilgi sızdırma aşamaları) olduğundan dolayı tespit edildiği aşamada bertaraf edilmesi gerekmektedir. Literatürde ağ anomali tespitine yönelik olarak, çeşitli istatistiksel ve davranışsal özelliklere bakan çalışmalar bulunmaktadır [1-9, 50, 52-55, 59]. Ponemon Enstitüsü tarafından yapılan bir araştırmaya göre [10] GKT saldırılarının %91'i imza tabanlı saldırı tespit ve anti-virüs güvenlik çözümleri tarafından fark edilememiştir.

Saldırı tespit sistemleri ilk olarak Denning tarafından 1987 yılında [5] ortaya çıkmıştır. Sabahi ve Movaghar 2008 yılında [6] saldırı tespit sistemlerini host tabanlı, ağ tabanlı ve hibrit sistemler olmak üzere üç gruba ayırmıştır. Host tabanlı sistemler saldırı tespitini yaparken sadece kullanıcı makine üzerindeki olaylara bakarak karar vermektedirler. Ağ tabanlı saldırı tespit sistemleri, sistemdeki çeşitli ağ kısımlarından gelen trafiği analiz ederek saldırı tespitini yapmaktadırlar. Hibrit sistemler, host ve ağ tabanlı olmak üzere her iki saldırı tespit sistemini kullanmaktadır.

Axelsson 2000 yılında yaptığı çalışmada [7], saldırı tespit sistemlerini iki gruba ayırmıştır. Birincisi yanlış kullanım, ikincisi ise anomali saldırı tespit sistemleridir. Birincisi kural

bazlı, imza bazlı olarak alt gruplara ayrılabilmektedir. Anomali saldırı tespit sistemleri, davranışı öğrenilen sistemden sapmalara göre saldırı tespitini yapmaktadır. Sabahi ve Movaghar'a [6] göre anomali saldırı tespit sistemleri, istatistik tabanlı, uzaklık tabanlı, kural tabanlı, profil tabanlı ve model tabanlı olarak beş gruba ayrılmaktadır. Yu 2012 yılında yaptığı çalışmada [8] anomali tabanlı saldırı tespit sistemlerini, istatistiksel teknikler, makine öğrenmesi teknikleri, yapay sinir ağları teknikleri, veri madenciliği teknikleri ve bilgisayar bağışıklık (immunology) teknikleri olarak 5 gruba ayırmıştır.

Chandola ve arkadaşları, 2009 yılında yaptığı çalışmada [9], ağ trafiği saldırı tespit sistemlerini altı farklı gruba ayırmıştır. Birinci grup, sınıflandırma (normal, anormal) tabanlı saldırı tespit sistemidir. İkinci grup en yakın komşu, üçüncü grup kümeleme tabanlı, dördüncü grup istatistik tabanlı, beşinci grup bilgi teoriği tabanlı saldırı tespit sistemleridir.

Andrew Vance 2014 yılındaki çalışmasında [1], ağ akış verilerinin istatistiksel olarak incelenmesi suretiyle GKT saldırılarının tespitine yönelik bir araştırma gerçekleştirmiştir. Normal olarak değerlendirilen ağ trafiği ile mevcut ağ trafiğinin istatistiksel karşılaştırılması ile saldırı tespiti yapılmıştır. Ağ akış verisi içerisinde yer alan IP, port ve paket boyutu bilgileri kullanılarak istatistiki ve entropi (dağınım) değerleri ile GKT saldırı tespiti yapılmaya çalışılmıştır. Tier 1 ve Tier 2 seviyesindeki firmalardan bulut bilişim veri merkezi trafik verileri toplanarak testler gerçekleştirilmiştir. İmza tabanlı saldırı tespit sistemlerine göre toplamda %429 oranında daha fazla doğru veya yanlış pozitif alarm ürettiği görülmüştür. Yine entropi değerlerini kullanarak yapılan başka bir çalışma da P. Bereziński tarafından 2014 yılında yapılmıştır [61].

S.M. Bridges ve R.B. Vaughn 2000 yılındaki çalışmalarında [2], bulanık veri madenciliği metodunu kullanarak, saldırı tespiti yapabilmek için normal ağ trafik akışının paternlerini çıkarmışlardır. Veri madenciliği bulanık ilişki kuralları vasıtası ile normal olarak belirlenen trafik ile anomali içeren trafik arasında benzerlik karşılaştırması yapılarak trafik akışındaki aykırılıklar tespit edilmeye çalışılmıştır.

D. Smallwood ve A. Vance 2011 yılındaki çalışmalarında [3], derin paket incelemesi (deep packet inspection (DPI)) ile gerçek zamanlı saldırı tespitinde kullanılabilecek Blitzdump

yazılımı vasıtasıyla testler gerçekleştirmiştir. Avrupa içerisinde yer alan iki büyük bulut bilişim servis sağlayıcı veri merkezinden alınan gerçek zamanlı veriler ile testler gerçekleştirilmiştir. Büyük hacimli veri trafiği üreten sistemler üzerinde DPI incelemesinin yapılabilmesi için çok hızlı sonuçlar üretecek yazılımlara ihtiyaç bulunmaktadır. Blitzdump ile tcpdump'a göre %6000 oranında daha yüksek hızlarda testler gerçekleştirilmiştir. Blitzdump ile yüksek veri üreten merkezlerde, gerçek zamanlı DPI incelemesi yapılarak ağda meydana gelen sapmalar ve saldırılar tespit edilebilecektir.

I. Friedberg, ve arkadaşları 2015 yılında yaptıkları çalışmada [4], ağda meydana gelen anomali davranışları tespit etmek için, sistem donanımları ve yazılımları tarafından üretilen log kayıtlarını kullanmışlardır. Sistem tarafından üretilen log kayıtlarının üzerinde belli örnekler ile arama yaparak (search patterns), ilgili alanların dolu veya boş olmasına göre 0 veya 1 değerlerini kullanarak her bir log kaydının parmak izlerini çıkarmışlardır. Sistemde oluşan log kaydı parmak izlerini daha sonra sınıflandırarak kural setlerini meydana getirmişlerdir. Kural setleri oluşturulup sistem log kayıtlarından sistemin davranışı ve olay ilişkileri öğrenildikten sonra, anomali tespiti öğrenilen kural setleri üzerinden yapılmaktadır. Daha sonra sistem tarafından üretilen log kayıtları, daha önce öğrenilmiş olan kural setlerine göre karşılaştırılacaktır. Hiçbir kural setine uymayan davranışlar anomali olarak değerlendirilecektir.

5. AĞ AKIŞ VERİSİ

Ağ akış verileri ilk olarak Cisco yönlendirici cihazları ile gelen bir özellik olarak kullanılmaya başlanmıştır. Daha sonra diğer markaların ağ cihazları da bu verileri oluşturmaya başlamışlardır. Cisco IOS işletim sistemi dışında Linux, BSD ve Solaris işletim sistemleri tarafından da ağ akış verileri desteklenmektedir. Ağ üzerinde bulunan güvenlik duvarı, saldırı tespit/önleme sistemleri, yönlendiriciler, anahtarlama cihazlarından vb. ağ akış verisi toplanabilir [18].

5.1. Ağ akış verisi tanımı

Ağ akış verisi, ağ trafiğinde paketlerin veri içeren kısımları işlenmeden, özel algoritmalar kullanılarak paketteki başlık bilgileri, protokol bilgileri gibi belirli kısımların kaydedilip bunlardan istatiksel analizler yapmaya olanak sağlayan standart bir teknolojidir [57, 58].

2 GB kullanıcı ağ trafiği ortalama olarak 2 MB ağ akış verisi oluşturmaktadır. Ayrıca genel olarak UDP protokolü ile gönderiliyor olması nedeniyle ağ trafiğini fazla yormamaktadır. Çoğu ağ cihazı üzerinde ayarlandığı takdirde işlemci gücü olarak fazla kaynak tüketmemektedir.

Kaydedilen ağ akışlarında kullanıcı mahremiyetini ve gizliliği ihlal edecek verilerin bulunmaması teknolojinin kurumlarda, şirketlerde, üniversitelerde yaygın olarak kullanılmasını sağlamıştır. Örneğin Cisco NetFlow v5 ağ akış standardına göre içerik olarak aşağıdaki çizelgede gösterildiği gibi 7 temel değer bulunmaktadır [41, 47, 49];

Çizelge 5.1. NetFlow v5 içerik bilgileri

S.Nu.	İçerik
1.	Kaynak IP
2.	Hedef IP
3.	Kaynak Port (TCP, UDP, ya da 0)
4.	Hedef Port (TCP, UDP, ya da 0)
5.	IP Protokolü
6.	IP Servis Tipi
7.	Giriş Ara yüzü

Bu temel değerlerin yanında paket sayısı ve trafik boyut bilgileri de NetFlow verisinin içerisinde yer almaktadır. Netflow ağ akış verisinin, komut satırı aracı olan nfdump yazılımından alınan çıktı aşağıdaki resimde gösterilmiştir. Komut satırından girilen komut aşağıda gösterilmektedir. İstenen tarih ve zaman aralığından dosyalar belirtilerek, kaynak IP, hedef IP ve hedef port bilgilerine göre gruplar halinde toplam ağ akış verisi alınmıştır.

```
nfdump -M /data/nfsen/profiles-data/live/upstream1 -T -R
2015/06/21/nfcapd.201506210250:2015/06/21/nfcapd.201506210645 -a -A
srcip,dstip,dstport -c 20
```

Date flow start	Duration	Src IP Addr	Src Pt	Dst IP Addr	Dst Pt	Packets	Bytes	bps	Bpp	Flows
2002-06-06 13:31:21.082	10.000	10.0.0.1	60866	10.0.0.2	443	0	2.9 M	2.3 M	0	3
2002-06-06 13:31:21.082	10.000	10.0.0.1	38891	10.0.0.2	443	0	573405	458724	0	1
2002-06-06 13:31:21.082	10.000	10.0.0.1	6099	10.0.0.2	80	0	5.0 M	4.0 M	0	1
2002-06-06 13:31:21.082	10.000	10.0.0.1	51198	10.0.0.2	80	0	25.6 M	20.5 M	0	2
2002-06-06 16:21:12.082	60.000	10.0.0.16	1000	4.4.4.169	22	0	6.2 M	832256	0	7
2002-06-06 13:31:21.082	10.000	10.0.0.1	54318	10.0.0.2	443	0	1.3 M	1.1 M	0	1
2002-06-06 13:31:21.082	10.000	10.0.0.1	60605	10.0.0.2	443	0	1.4 M	1.2 M	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.128	1000	4.4.4.88	80	0	638937	85191	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.205	1000	4.4.4.159	22	0	9.8 M	1.3 M	0	11
2002-06-06 13:31:21.082	10.000	10.0.0.1	2328	10.0.0.2	443	0	4.8 M	3.9 M	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.210	1000	4.4.4.156	22	0	7.6 M	1.0 M	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.58	1000	4.4.4.207	22	0	4.1 M	552653	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.215	1000	4.4.4.212	22	0	7.4 M	980795	0	9
2002-06-06 16:21:12.082	60.000	10.0.0.33	1000	4.4.4.207	22	0	8.0 M	1.1 M	0	10
2002-06-06 16:21:12.082	60.000	10.0.0.131	1000	4.4.4.203	22	0	7.7 M	1.0 M	0	7
2002-06-06 16:21:12.082	60.000	10.0.0.0	1000	4.4.4.140	22	0	6.2 M	821334	0	9
2002-06-06 16:21:12.082	60.000	10.0.0.61	1000	4.4.4.107	80	0	901065	120142	0	1
2002-06-06 13:31:21.082	10.000	10.0.0.1	7856	10.0.0.2	443	0	425958	340766	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.32	1000	4.4.4.86	22	0	6.6 M	884682	0	6
2002-06-06 16:21:12.082	60.000	10.0.0.42	1000	4.4.4.130	22	0	9.2 M	1.2 M	0	10

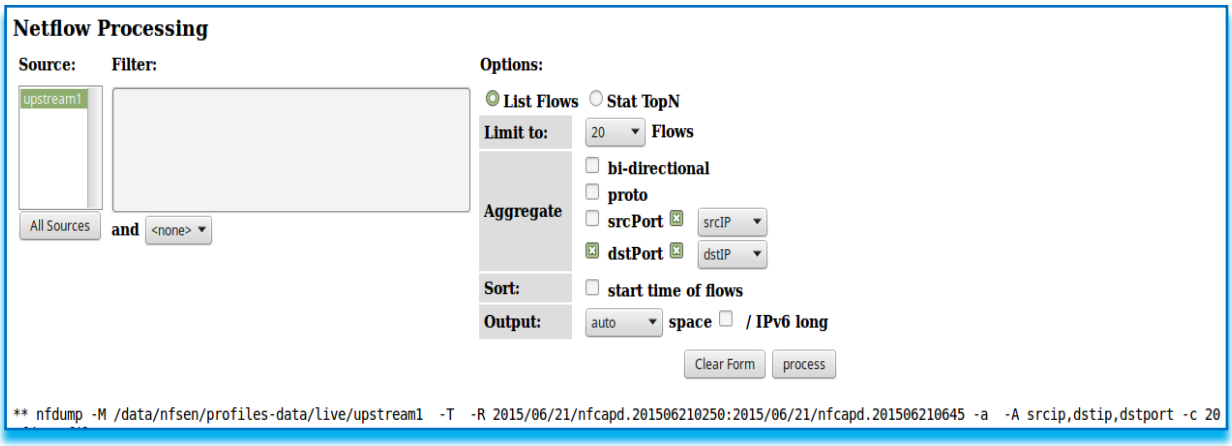
Summary: total flows: 7323378, total bytes: 2.7 T, total packets: 6.7 M, avg bps: 51889, avg pps: 0, avg bpp: 396875

Resim 5.1. Nfdump çıktı örneği

Kaynak adresten hedef adrese doğru olan ağ trafiği akış bilgileri nfdump yazılımı ile kolaylıkla alınabilmektedir. Ekran çıktısının alt bölümündeki özet kısmında toplam ağ akış trafiği, toplam paket ve boyut bilgileri görülebilmektedir.

Hedef port sütununda kullanıcıların 22 (SSH), 80 (HTTP) ve 443 (HTTPS) portlarını kullandıkları görülmektedir.

Ağ akış verisini incelemekte kullanılan bir başka yazılım NfSen yazılımıdır. NfSen yazılımının sağladığı arayüz sayesinde ağ akış verileri kolaylıkla incelenebilmektedir. NfSen yazılımının [13, 14] Linux makine üzerinde çalışması için Apache, PHP, RRD gibi uygulamaların kurulu olması gerekmektedir. NfSen yazılımı ağ trafiğini incelerken arka planda nfdump komut satırı aracını kullanmaktadır. Arayüzden seçilen opsiyonlara göre nfdump yazılımının kullanacağı parametreleri hazırlayıp sorguyu nfdump'a göndermektedir. Aşağıdaki örnekte de görülebileceği gibi GUI üzerinde seçilen srcIP, dstPort, dstIP seçenekleri doğrultusunda, resmin altında gözüken nfdump komut satırı parametreleri değişmiştir. Seçilen özelliklere göre *–A srcip,dstip,dstport* parametreleri komutun sonuna eklenmiştir.



Resim 5.2. NfSen yazılımı sorgulama örneği

5.2. Ağ akış verisi kullanım alanları

Ağ trafiğinin tamamı işlenerek yani DPI metodu ile anomali tespiti yapmak, ağı izlemek ve başarı elde etmek çok zor gözükmektedir [56]. Özellikle yüksek bant genişliğine sahip

ağlarda DPI yapmak daha da zorlaşmaktadır. Ağ akış verisinin sağladığı özet bilgiler ile DPI kullanılarak yapılamayacak gerçek zamanlı analizler, kurumlar tarafından yaygın olarak kullanılmaya başlanmıştır. Ağ akış trafik bilgileri fazla yer kaplamadığından dolayı, kurumlar geriye dönük olarak uzun süreli (aylık, hatta yıllar bazında) trafik bilgilerini saklayabilmektedir. Kurum ağında neler olduğuna dair özet bir resim ağ akış trafik bilgisi içinde bulunmaktadır. Böyle bir resmin çekilebiliyor olması kurum için durumsal farkındalığı büyük oranda artıracaktır. Bilgisayar olaylarına müdahale edebilmek için ağda meydana gelen trafik bilgilerinin görülebiliyor olması belirsizliğin azaltılmasında yardımcı olacaktır. Ağda yaşanan aykırılıkların ve aşırılıkların görülebilmesi ağ akış bilgileri sayesinde mümkün olmaktadır. Ağ akış verileri güvenlik, ağ izleme, ağ anomali tespiti, kullanıcı kota yönetimi ve faturalandırma, botnet tespiti, SSH atak tespiti gibi çeşitli alanlarda kullanılmaktadır. Ağ akış bilgileri ile botnet saldırı tespiti daha önceden botnet olduğu bilinen saldırılar için yapılabilmektedir. Botnet komuta kontrol sunucusu olduğu daha önceden yayınlanmış olan sunucular için alarm üretilebilmektedir. SSH atak saldırılarının tespiti ise iki nokta arasındaki bağlantı sayısı ve bağlantı başına düşen paket sayısı değerleri dikkate alınarak yapılabilmektedir [48].

5.3. Ağ akış verisi standartları

Ağ akış trafiği için kullanılan birden çok standart bulunmaktadır. Endüstri standardı olan aşağıdaki çizelgede de görüldüğü gibi sFlow'dur. Ancak ağ cihazlarında Cisco çok fazla kullanıldığı için kendi geliştirdiği standart olan NetFlow yaygın bir şekilde kullanılmaktadır. sFlow ağ akış standardı 2. ve 3. OSI katmanlarında çalışabildiği için 2. katman protokolleri olan IPX, Appletalk, XNS gibi IP protokolü tabanlı olmayan trafiğin de bilgisini sunabilmektedir. NetFlow ağ akış standardı sadece 3. katman protokolü olan IP protokolü ile ilgili ağ trafik bilgilerini yakalayabilmektedir. IP dışındaki diğer protokollerin ağ trafik bilgileri NetFlow ile tutulamamaktadır.

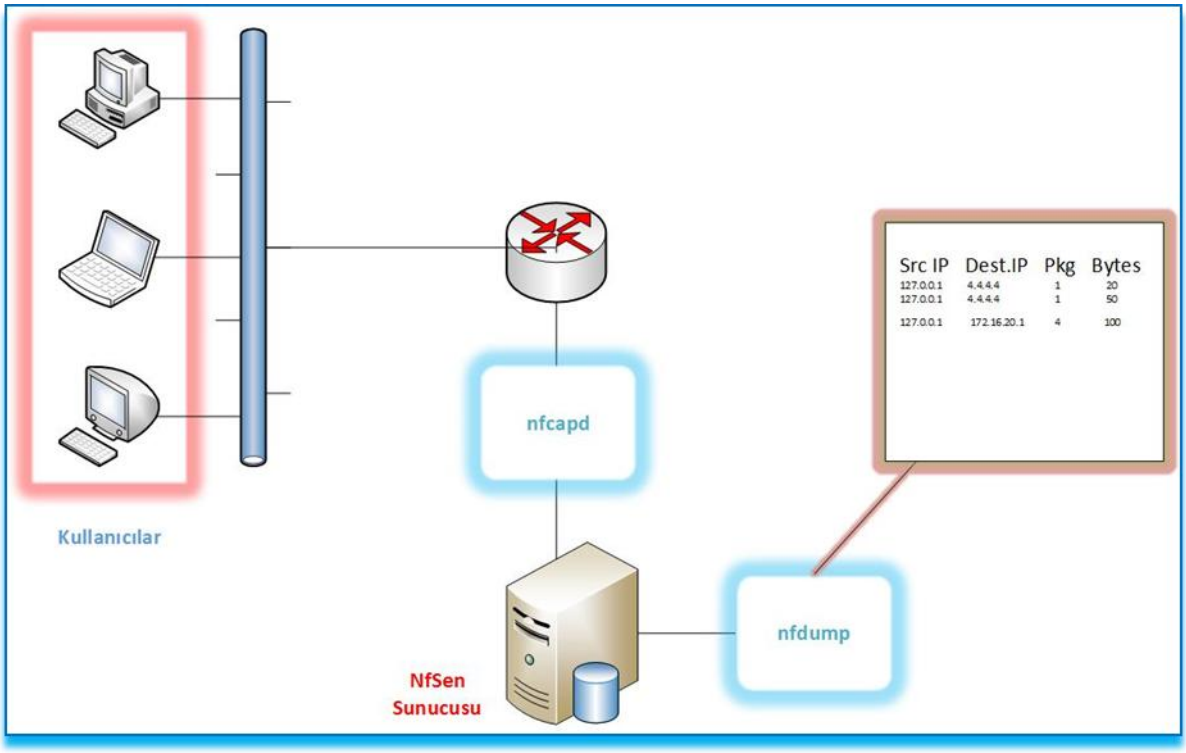
Çizelge 5.2. Ağ akış verisi standartları

Sıra Nu.	Ağ Akış Verisi Standardı	Kullanan Donanımlar
1	NetFlow	Cisco
2	jflow ve cflow	Juniper Networks
3	NetStream	Huawei Technologies, 3Com, HP
4	AppFlow	Citrix
5	sFlow	Alaxala, Alcatel Lucent, Allied Telesis, Arista Networks, Brocade, Cisco, Dell, D-Link, Enterasys, Extreme, Fortinet, HP, Hitachi, Huawei, IBM, Juniper, LG-Ericsson, Mellanow, MRV, NEC, Netgear, Proxim Wireless, Quanta Computer, Vyatta, ZTE, ZyXEL

5.4. Otomatik ağ akış trafiği üreticisi ile oluşturulan ağ akış verisinin toplanması

NfSen ve NetFlow Traffic Generator yazılımları kullanılarak, nasıl ağ trafik verisinin toplanacağı örnek olarak bu bölümde anlatılacaktır. Şekil 5.1'de yer alan örnekte gösterildiği gibi ağ akış verisi bilgileri, kullanıcı bilgisayarlarının bağlı olduğu yönlendirici cihazı üzerinden *ncapd* yazılımı vasıtası ile alınmaya başlanmıştır. Yönlendirici cihazının üretmiş olduğu ağ trafik bilgilerini, NfSen sunucusu üzerine kayıt eden yazılım *ncapd* yazılımıdır. Ncapd yazılımı NfSen sunucusu üzerinde çalışmaktadır.

Daha önce de bahsedildiği gibi NfSen yazılımı bir RRD aracı olup ağ akış trafiğinin grafiksel olarak gösterimini ve veri üzerinde istenen sorguların, filtrelemelerin yapılmasına olanak sağlar. Şekil 5.1'de NfSen sunucusu üzerinde çalışan *nfdump* yazılımının trafik bilgilerini sorgulayabildiği gösterilmiştir.



Şekil 5.1. NfSen yazılımı ile ağ trafiğinin analizi

NfSen sunucuları aynı anda birden fazla kaynaktan, ağın farklı bölümlerinden ve ağda bulunan çeşitli cihazlardan, ağ trafiği akış bilgilerini alabilirler. Şekil 5.1'deki örnekte tek bir cihazdan gelen ağ trafiği gösterilmiştir.

NetFlow Traffic Generator yazılımı kullanılarak Resim 5.3'te gösterildiği gibi 10.0.0.0-10.0.0.255 kaynak IP'lerinden 4.4.4.0-4.4.4.255 hedef IP'lerine 22 ve 80 hedef portlarını kullanacak şekilde NetFlow trafiği ürettirilmiştir.

Date flow start	Duration	Src IP Addr	Dst IP Addr	Dst Pt	Packets	Bytes	bps	Bpp	Flows
2002-06-06 16:21:12.082	60.000	10.0.0.10	4.4.4.53	22	0	3.1 M	408482	0	3
2002-06-06 16:21:12.082	60.000	10.0.0.210	4.4.4.128	22	0	5.0 M	664057	0	6
2002-06-06 16:21:12.082	60.000	10.0.0.115	4.4.4.61	22	0	9.7 M	1.3 M	0	12
2002-06-06 16:21:12.082	60.000	10.0.0.220	4.4.4.75	22	0	5.1 M	674979	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.106	4.4.4.246	22	0	6.8 M	913079	0	7
2002-06-06 16:21:12.082	60.000	10.0.0.187	4.4.4.227	22	0	3.0 M	399745	0	3
2002-06-06 16:21:12.082	60.000	10.0.0.77	4.4.4.112	22	0	4.3 M	572312	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.159	4.4.4.1	22	0	13.8 M	1.8 M	0	12
2002-06-06 16:21:12.082	60.000	10.0.0.224	4.4.4.33	22	0	4.9 M	659688	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.69	4.4.4.7	22	0	6.6 M	884682	0	6
2002-06-06 16:21:12.082	60.000	10.0.0.176	4.4.4.93	22	0	7.6 M	1.0 M	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.189	4.4.4.46	22	0	3.5 M	460908	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.201	4.4.4.18	22	0	7.8 M	1.0 M	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.92	4.4.4.22	22	0	4.9 M	653135	0	6
2002-06-06 16:21:12.082	60.000	10.0.0.197	4.4.4.182	22	0	3.9 M	526440	0	4
2002-06-06 16:21:12.082	60.000	10.0.0.187	4.4.4.87	22	0	5.6 M	749249	0	7
2002-06-06 16:21:12.082	60.000	10.0.0.236	4.4.4.102	22	0	12.3 M	1.6 M	0	12
2002-06-06 16:21:12.082	60.000	10.0.0.2	4.4.4.113	22	0	8.8 M	1.2 M	0	12
2002-06-06 16:21:12.082	60.000	10.0.0.46	4.4.4.196	22	0	3.4 M	454355	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.16	4.4.4.207	22	0	4.6 M	611632	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.212	4.4.4.184	22	0	11.8 M	1.6 M	0	11
2002-06-06 16:21:12.082	60.000	10.0.0.47	4.4.4.220	22	0	6.8 M	910894	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.241	4.4.4.67	22	0	4.7 M	622554	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.182	4.4.4.227	80	0	573405	76454	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.113	4.4.4.25	22	0	4.5 M	600710	0	5
2002-06-06 16:21:12.082	60.000	10.0.0.214	4.4.4.215	22	0	6.0 M	795121	0	8
2002-06-06 16:21:12.082	60.000	10.0.0.232	4.4.4.101	22	0	12.2 M	1.6 M	0	14
2002-06-06 16:21:12.082	60.000	10.0.0.245	4.4.4.57	80	0	1.6 M	214071	0	1
2002-06-06 16:21:12.082	60.000	10.0.0.24	4.4.4.62	22	0	4.6 M	611632	0	7
2002-06-06 16:21:12.082	60.000	10.0.0.244	4.4.4.163	22	0	2.2 M	288340	0	2

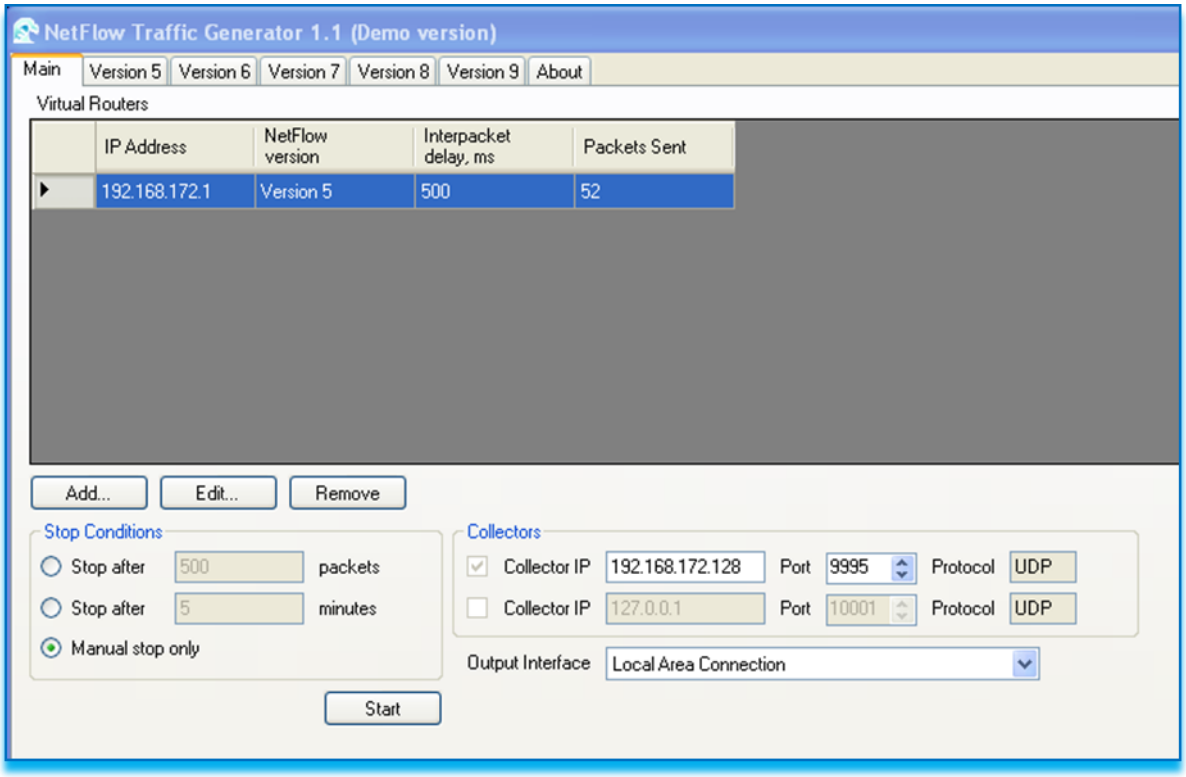
Summary: total flows: 7323378, total bytes: 2.7 T, total packets: 6.7 M, avg bps: 51889, avg pps: 0, avg bpp: 396875

Resim 5.3. NetFlow Traffic Generator tarafından üretilen ağ akış trafiği verisi

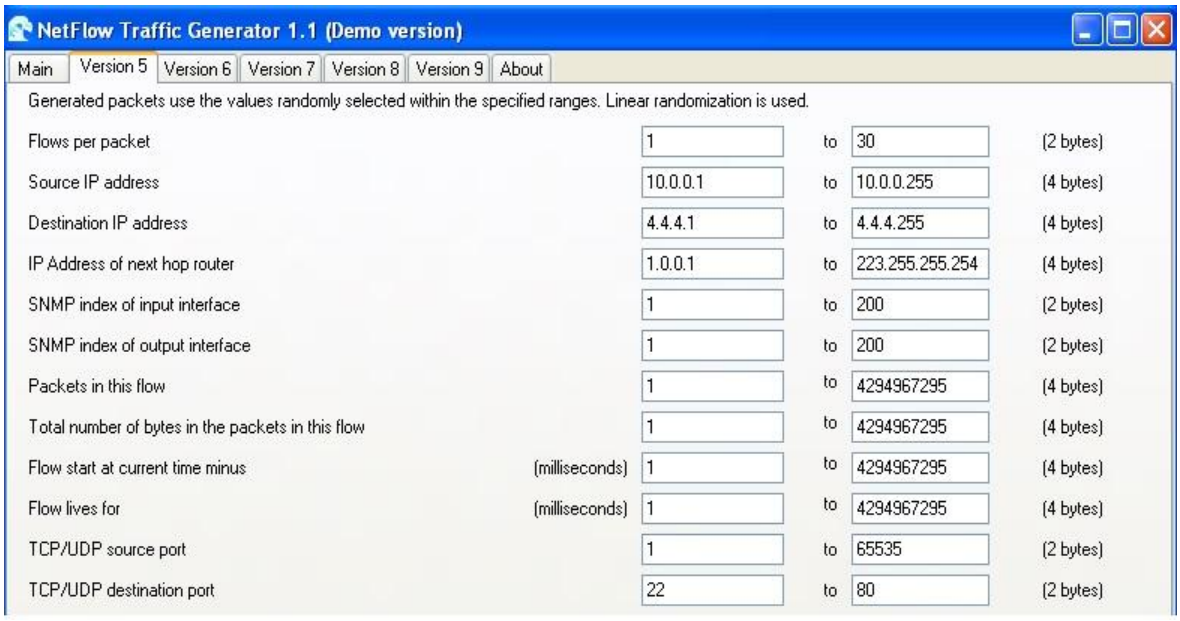
NetFlow Traffic Generator ile kaynak IP ve port ile hedef IP ve port aralık bilgileri girilerek test maksatlı olarak istenildiği gibi NetFlow verisi üretilebilmektedir. Bu araç ile üretilen ağ trafik verisi gerçek yönlendirici ve anahtarlama cihazlarından çıkıyormuş gibi değerlendirilebilir.

Resim 5.4'te gösterildiği gibi sanal yönlendirici IP bilgisi 192.168.172.1 olup, üretilen ağ akış trafik bilgileri bu yönlendiriciden üretiliyor gibi olacaktır. Eğer aynı anda farklı yönlendirici cihazlarından ağ akış trafik verisi üretilmek isteniyor ise 5000 adede kadar sanal yönlendirici IP bilgisi eklenebilir.

NetFlow Traffic Generator yazılımı, Cisco NetFlow versiyon 5, 6, 7, 8 ve 9 formatında ağ akış trafik bilgisi üretebilmektedir. Resim 5.5'te NetFlow versiyon 5 ile ağ akış trafik verilerinin nasıl üretildiği gösterilmiştir.



Resim 5.4. NetFlow Traffic Generator 1.1 ekran görüntüsü



Resim 5.5. NetFlow Traffic Generator 1.1 ayrıntılı özellikler

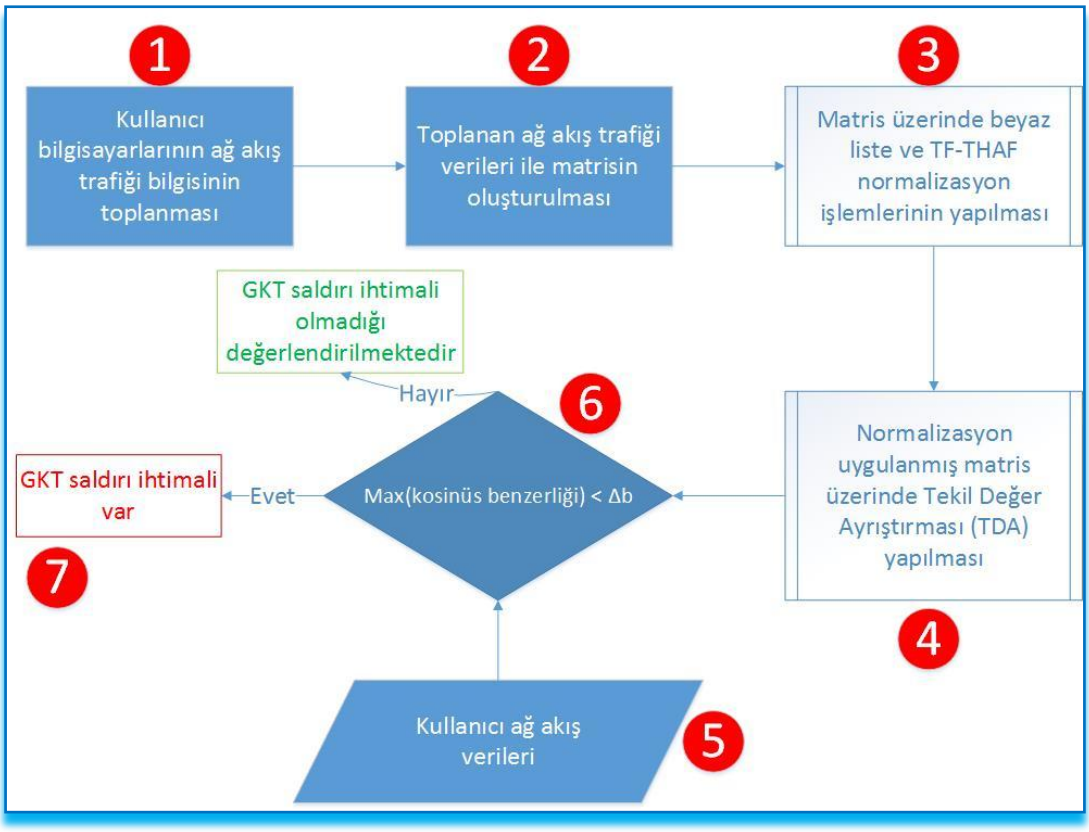
Otomatik ağ akış verisi üreten NetFlow Traffic Generator yazılımı kullanılarak, *nfcapd* yazılımı marifetiyle ağ akış verileri NfSen yazılımı yüklü olan bilgisayara

kaydedilebilmektedir. Bu tez çalışmasında gerçek sistemlerden alınan ağ akış verileri kullanılmamıştır. Ancak bu verilerin nasıl kaydedileceği, nasıl sorgulama ve filtreleme yapılabileceği, grafiksel olarak nasıl izlenebileceği anlatılmıştır. İleriki çalışmalarda gerçek veriler üzerinde testler gerçekleştirilmesi durumunda bu bölümde bahsedilen araçlar kullanılacaktır.

6. AĞ AKIŞ VERİSİ İLE TDA KULLANARAK GELİŞMİŞ KALICI TEHDİT SALDIRISININ TESPİTİ

Gelişmiş kalıcı tehdit saldırılarında ilk hedef, güvenlik duvarı ve saldırı önleme sistemlerinin arkasında bulunan nispeten güvenli alanlardaki sunucu bilgisayarlar değil, öncelikli olarak kurumun son kullanıcı bilgisayarlarıdır. Çoğu saldırıda kullanılan zararlı yazılım, sıfırıncı gün açıklığından faydalanabilmek için kullanıcı ilk hareketine ihtiyaç duymaktadır. Kullanıcı hareketinden kastedilen zararlı bir e-posta eklentisinin açılması, zararlı yazılım içeren bir sitenin ziyaret edilmesi gibi kullanıcı tarafından yapılması gereken eylemlerdir. Bu gibi eylemler güvenlik bilincine sahip kurumlarda sunucu bilgisayarlar üzerinde yapılmamaktadır. İnternet ortamında hizmet veren sunucu sistemler (e-posta, web, uygulama sunucuları vb.) normal şartlarda sadece gelen taleplere hizmet vermektedirler, doğrudan internet çıkışları bulunmamaktadır. Bu yüzden DMZ bölgesinde hizmet veren bu sunucular, GKT aktörleri tarafından, GKT saldırılarında ilk hedef olarak kullanılmamaktadırlar.

Şekil 6.1'de önerilen modele ilişkin akış diyagramı gösterilmiştir. Birinci adım, GKT saldırısı olmadığı değerlendirilen bir zaman aralığında, kullanıcı bilgisayar ağ akış trafiği verilerinin toplanmasıdır. İkinci adım, toplanan bu veriler ile kullanıcı bilgisayar-ağ akış trafiği matrisinin oluşturulmasıdır. Daha sonra üçüncü adımda matris üzerinde beyaz liste ve trafik frekansı – ters hedef adres frekansı (TF-THAF) normalizasyon işlemleri yapılacaktır [60]. Dördüncü adımda normalizasyon yapılmış matris üzerinde tekil değer ayrıştırması (TDA) işlemi yapılarak daha düşük derecede orijinal matrise benzeyen başka bir matris elde edilecektir. Bu sayede kurum içerisindeki kullanıcı bilgisayarlarının çok boyutlu uzayda yerini gösteren matris oluşturulmuş olacaktır. Kullanıcı bilgisayarlarının bundan sonraki ağ akış verileri oluşturulan bu matrisin satırları ile tek tek karşılaştırılarak kosinüs benzerliğine bakılacaktır. Altıncı adımda Kosinüs benzerlik değeri Δb'den küçük olan bilgisayarlar için GKT saldırısı ihtimalinin bulunduğu tespiti yapılmış olacaktır. Δb değeri ağ veya sistem yöneticisi tarafından belirlenen bir değerdir. 0 ile 1 arasında olan Δb değerinin hesaplanması hususunda uygulama simülasyonunda yapılan testlerde ortalama 0.6 değeri bulunmuş ve yedinci adımda bu değerin üzerindeki kosinüs benzerlikleri saldırı ihtimalinin olmadığını, aşağısındaki benzerliklerin ise GKT saldırısı ihtimali olduğunu göstermektedir.



Şekil 6.1. Önerilen modele ait akış diyagramı

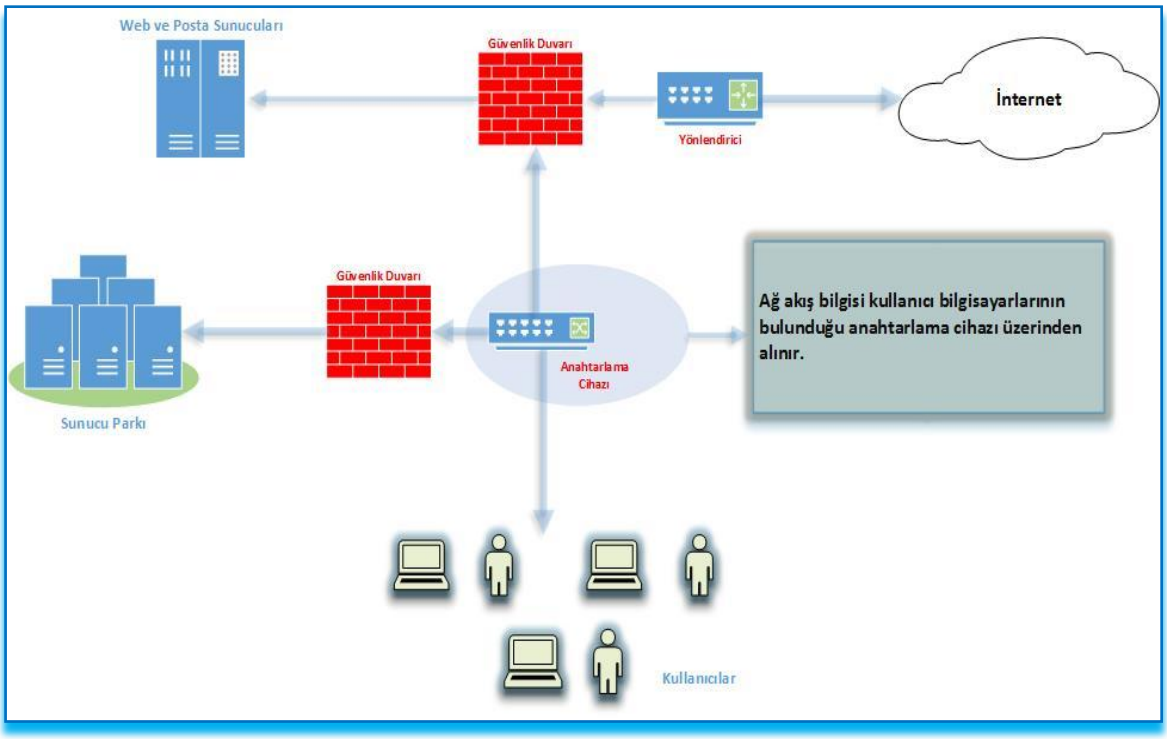
6.1. Ağ akış verisinin toplanması

Kurumların DMZ bölgesinde yer alan sunuculardan ayrı olarak internet ortamına hizmet vermeyen iç ağda kullandıkları sunucular bulunmaktadır. Bu sunucular dosya sunucuları, yazıcı sunucuları, kaynak kod deposu sunucuları, veri tabanı sunucuları, uygulama sunucuları gibi iç ağa hizmet veren donanımlardır. Kurumun değerli varlıklarını oluşturan bilgiler bu sunucularda bulunmaktadır. Kurumun değerli varlıklarını barındıran bu sunuculara, kullanıcı bilgisayarlarının genel olarak doğrudan erişimi bulunmaktadır. Güvenlik algısı az olan bazı kurumlarda, arada güvenlik duvarı olmadan kullanıcı bilgisayarlarının bu sunuculara erişimi vardır. GKT saldırılarının tespitinde kullanıcı bilgisayarlarının eylemlerinin büyük önemi bulunmaktadır.

GKT aktörleri her ne kadar kurum ağı içerisinde fark edilmemek için elinden gelen çabayı gösterecek olsa da mutlaka normal trafikten farklı bir trafik oluşturacaklardır. Eğer saldırı aktörlerinin oluşturduğu bütün trafik normale yakın ise fark etmek gerçekten zorlaşacaktır.

GKT aktörleri eğer sabotaj amacı ile (Stuxnet saldırısında olduğu gibi) sisteme sızmadılar ise mutlaka kurumun değerli verisini dışarıya sızdırmaya çalışacaklardır. Bu durum mutlaka normal trafikten farklı olarak anomali olarak tarif ettiğimiz ağ trafiğinde bir aykırılık oluşturacaktır. Bu anomaliyi tespit etmek maksadıyla kullanıcı bilgisayarların bağlı olduğu anahtarlama cihazı üzerinde, ağ akış bilgilerinin toplanabilmesi için ağ akış verisinin açılması ve ayarlarının yapılması gerekmektedir.

Örnek olarak Şekil 6.1'de kullanıcı bilgisayarlarının bulunduğu ağda yer alan anahtarlama cihazı üzerinde ağ akış ayarlarının yapılabileceği gösterilmiştir. Bu tez çalışmasında GKT saldırılarının tespiti için sadece kullanıcı bilgisayarlarının ağ akış trafiği kullanılmıştır. Kuruma ait kullanıcıların ağ akış bilgilerini toplamak amacıyla seçilen anahtarlama cihazı aşağıdaki şekilde gösterilmiştir. Ağ akış verisi, kurumun katlarında bulunan anahtarlama cihazlarından alınabileceği gibi, ağ omurgasını oluşturan merkezi anahtarlama veya yönlendirici cihazı üzerinden de alınabilir.



Şekil 6.2. Ağ akış verilerinin toplanması

6.1.1. Ağ akış verisi ve kullanıcı makine bilgileriyle matrisin oluşturulması

Anahtarlama veya yönlendirici cihazları üzerinden geçen ağ akışı bilgilerinin kaydedilmesi için *nfcapd* yazılımı, ağ akış bilgilerinin sorgulanması ve filtreleme yapılabilmesi için *nfdump* yazılımları kullanılmaktadır. Ağ akış trafiğinin grafiksel görünümü için RRD veri tabanlarını okuyabilen *NfSen* yazılımı kullanılmaktadır. NfSen, nfcapd ve nfdump yazılımları Linux işletim sistemleri üzerine kurulabilmektedir. Bu tez çalışmasında gerçek ağ trafik bilgileri kullanılmamıştır. Tezin dördüncü bölümünde ağ akış verilerinin nasıl toplanacağı, NfSen, nfdump ve ncapd yazılımlarının nasıl kullanılacağı anlatılmıştır. Ayrıca NetFlow Traffic Generator yazılımı ile üretilen ağ akış trafiğinin NfSen ve nfdump ile analiz edilmesi ayrıntılı olarak gösterilmiştir. Matris içeriği oluşturulurken Microsoft Office Excel yazılımının *randbetween* fonksiyonu kullanılmıştır. Matrisin oluşturulması sırasında izlenen yollar aşağıdaki bölümlerde açıklanmıştır.

Kullanılacak matrisin satır bölümünde kullanıcı bilgisayarlarının IP bilgileri yer almaktadır. Her bir sütun ise o satırdaki bilgisayarın gitmeye çalıştığı hedef IP ve port bilgilerini göstermektedir. IP ve port bilgisi beraber soket bilgisini oluşturmaktadır. Örneğin *173.194.67.94:443* soketi ile *173.194.67.94:80* soketi aynı IP adreslerini içermelerine rağmen portları farklı olduğu için oluşturulan matriste ayrı bir sütun olarak gösterilir. Kullanıcı bilgisayarlarının hangi IP'ye hangi port numarasından gittiği GKT saldırısının tespiti için önem arz etmektedir. Örnek olarak oluşturulan matris bilgileri aşağıda gösterilmiştir.

Çizelge 6.1. Örnek matris içeriği

Kullanıcı Bilgisayar IP'si	Hedef Adres Bilgisi (Soket olarak)				
	4.4.4.4:53	173.194.67.94:443	85.111.27.167:80	31.13.64.1:443	
10.0.0.1	12	234	0	212	
10.0.0.2	34	0	321	35	.....
10.0.0.3	2	78	15	20	
10.0.0.4	5	89	18	32	
10.0.0.5	23	155	0	0	
.....	.....	.....	.....	.....	

Matris içerisinde yer alan sayılar, MB cinsinden olup, kullanıcı bilgisayarın hedef adrese olan toplam ağ akış verisini göstermektedir. Örneğin 10.0.0.4 IP adresli kullanıcı bilgisayarının 31.13.64.1:443 (https://www.facebook.com) hedef adresle olan ağ akış trafiği toplamı 32 MB'tır. Tüm değerler megabyte olarak matris içerisine yazılmıştır. 100 KB değerinde bir trafik varsa bu değer yukarıya yuvarlanarak 1 MB olarak matrise yazılır. Aynı şekilde 2.1 MB değerindeki bir trafik de 3 MB olarak yukarıya yuvarlanır. Bu değer Facebook'tan 10.0.0.4 IP'li kullanıcı bilgisayarına gelen trafik değildir. Sadece kaynak adres olan kullanıcı bilgisayardan Facebook'a olan ağ akış trafiği toplamıdır. Facebook'tan kullanıcı bilgisayara gelen trafik bilgisi matriste yer almaz. Kısaca internet ortamından içeriye doğru olan ağ akış trafiği matriste yer almaz. Matriste her zaman sıfır veya artı değerler bulunmaktadır. Satır ve sütunlara karşılık gelen yerlerde sıfır değerinin bulunması son kullanıcı bilgisayarından, karşılık gelen hedef adrese hiç ağ trafiğinin bulunmadığını göstermektedir.

6.2. Matris normalizasyon işlemleri

Son kullanıcı bilgisayarlarının ağ akış verilerinden oluşturulan matris üzerinde, normalizasyon için bazı ayarlamaların yapılması gerekmektedir. Örneğin tüm son kullanıcı bilgisayarlarının Google arama motorlarının IP adreslerine gitmesi, bu hedef IP adres sütununun uzaysal olarak matriste anlam ifade etmemesi anlamına gelmektedir. Tam tersinden ele alacak olursak, bir son kullanıcı bilgisayarının daha önce hiçbir son kullanıcı bilgisayarının gitmediği bir hedef IP adresine gidiyor olması uzaysal olarak önem derecesi yüksek bir boyutu matrisin içine katacaktır. Bu yüzden matris içerisindeki sayı dağılımının normalizasyonu, daha sağlıklı sonuçların alınabilmesi için gereklidir.

6.2.1. TF-THAF normalizasyonunun uygulanması

Saklı anlam indeksi (LSI – Latent semantic index) uygulamalarında sıklıkla kullanılan terim frekansı - ters metin frekansı (TF-IDF Term Frequency-Inverse Document Frequency) normalizasyon yönteminin bir benzeri, matris içinde yer alan ağ akış bilgilerinin dağılımını düzenlemek için kullanılmıştır.

Ağ akış trafiği matrisinin satırları, son kullanıcı bilgisayarlarının IP bilgilerini, matrisin sütunları ise bilgisayarların ziyaret etmiş olduğu hedef adresleri belirtmektedir. Trafik

frekansı (TF) ve ters hedef adres frekansı (THAF) normalizasyonları kullanılarak, bunlardan çıkan sonucun çarpılmasıyla üçüncü bir normalizasyon olan TF-THAF değeri elde edilir.

TF normalizasyonunun hesaplanması;

$$TF(t) = \frac{tf_{m,n}}{Bilgisayar\ toplam\ a\breve{g}\ trafi\breve{g}i} \quad (6.1)$$

şeklindedir. Formülde yer alan bilgisayar toplam ağ trafiği, matris satırında yer alan bilgisayar için toplam ağ trafiğini ifade etmektedir. $tf_{m,n}$ değeri A matrisi içerisindeki $[a_{ij}]$ ağ akış trafiği değeridir.

THAF normalizasyonunun hesaplanması;

$$THAF(t) = \log\frac{Toplam\ bilgisayar\ sayısı}{df_i} \quad (6.2)$$

şeklindedir. Formülde yer alan toplam bilgisayar sayısı, A matrisinin toplam satır sayısıdır. df_i değeri sütunda yer alan sıfırdan farklı pozitif değerlerin toplam sayısıdır.

Son olarak TF-THAF normalizasyonunun hesaplanması;

$$TF\text{-}THAF(t) = TF \times THAF \quad (6.3)$$

TF ve THAF normalizasyon değerlerinin çarpılmasıyla yapılır.

6.2.2. Beyaz liste normalizasyon uygulanması

Gürültünün ve matris boyutunun azaltılması için beyaz liste normalizasyonunun uygulanması hem hesaplama maliyetini azaltacak hem de daha sağlıklı sonuçların alınmasını sağlayacaktır. GKT saldırı aktörleri tarafından kullanıldığı düşünülmeyen, kesin güvenli olduğu değerlendirilen hedef IP adresleri oluşturulacak matris içine dahil edilmez. Beyaz liste normalizasyonunun dezavantajı, uygulandığı takdirde eğer GKT aktörleri tarafından kullanılan bir IP ise saldırı tespiti şansı azalacaktır. Bu tez çalışmasında beyaz

liste normalizasyonu uygulanarak, çok fazla miktarda içeriden dışarıya trafiğin bulunduğu, uzaysal olarak matrisi farklı boyutlara çeken adresler, güvenli olduğu değerlendirilerek, matris içerisine dâhil edilmemiştir.

6.3. Tekil değer ayrıştırması uygulayarak boyut küçültme

Tekil değer ayrıştırması için Java yazılım dilinde hazırlanmış olan *Apache commons-math3-3.5.jar* kütüphanesi kullanılmıştır [15]. Bu tez çalışmasında yazılan yazılım, NetBeans IDE 8.0.2 kullanılarak kodlanmıştır. Matris üzerinde tekil değer ayrıştırması yapıldıktan sonra, tekil değerleri barındıran S matrisine göre boyut küçültme uygulanmıştır. Örneğin S matrisindeki tekil değer sayısı 22 ise 22 sayısına göre A matrisi U, S, V^T matrisleri çarpılarak tekrar oluşturulmuştur. Sadece tekil değer sayısı kadar boyut küçültme işlemi yapılmıştır.

6.4. Kosinüs benzerliği kullanımı

Kosinüs benzerliği aşağıdaki Eşitlik 6.4'te gösterildiği gibi;

$$\text{Benzerlik} = \cos(\partial) = \frac{A.B}{A \quad B} = \frac{\sum_{i=1}^{n} A_i \; x \; B_i}{\sqrt{\sum_{i=1}^{n} (A)_i^2} \; x \; \sqrt{\sum_{i=1}^{n} (B)_i^2}} \tag{6.4}$$

şeklinde hesaplanır.

Tamamen aynı yönü gösteren vektörler için kosinüs benzerlik değeri büyüklükleri ne olursa olsun 1 olacaktır. Kosinüs benzerliğinin 0 olması vektörlerin dik olduğunu ve birbirine benzemediğini göstermektedir. Kosinüs benzerliği büyüklük ile ilgili bir kavram olmayıp, oryantasyon yani açı ile ilgili bir kavramdır. Bu durum şu şekilde açıklanır;

$$A = \begin{matrix} 1 & 2 & 3 \end{matrix} \qquad \text{ve} \qquad B = \begin{matrix} 2 & 4 & 6 \end{matrix}$$

A ve B iki ayrı vektör için kosinüs benzerliği;

$$\text{Benzerlik(A ile B)} = \cos(\partial) = \frac{1x2+2x4+3x6}{\sqrt{1^2+2^2+3^2} \; x \; \sqrt{2^2+4^2+6^2}} = \frac{28}{3{,}7416 \; x \; 7{,}4863} = 1$$

olarak hesaplanır. Bu hesaplamada da görüldüğü gibi kosinüs benzerliği büyüklükle ilgili değildir. A ve B vektörlerinin kosinüs değerlerinin 1 çıkmasının nedeni aynı oryantasyona sahip olmalarıdır. Bu durumda bu iki vektör birbirine benzerdir fakat büyüklükleri farklıdır. Genel olarak pozitif değerlere sahip olan matrislerde ve vektörlerde kosinüs benzerliği hesaplaması kullanılmaktadır.

6.5. Kosinüs benzerliği kullanarak kullanıcı bilgisayar ağ akış bilgileri ile GKT saldırısı tespitinin yapılması

A_{mxn} matrisinin herbir $a=(a_1, \ldots, a_k)$ vektörleri ile B_{mxn} matrisinin herbir $b=(b_1, \ldots, b_r)$ vektörleri için Eşitlik 6.5'te gösterildiği gibi matris satırı bazında;

$$Min.Benzerlik(A,B) = \Delta b = min \quad max \quad \cos\_sim \begin{matrix} a_k, k = 1, \ldots, m \\ b_r, r = 1, \ldots, m \end{matrix} \tag{6.5}$$

Δb minimum benzerliği bulmak için karşılaştırılacaktır. B_{mxn} matrisinde bulunan herbir satır vektör için b_r, A_{mxn} matrisinde yer alan her bir satır vektör a_k ile kosinüs benzerlikleri karşılaştırılacak ve maksimum benzerliklerin minimum olan değeri Δb minimum benzerlik olarak bulunacaktır.

Δb minimum benzerliği, Eşitlik 6.5'te gösterildiği gibi hesaplanabileceği gibi ağ veya sistem yöneticisi tarafından belirlenecek bir değer ile de Δb minimum benzerliğinden küçük olan kosinüs benzerlikleri için GKT saldırısı ihtimali vardır denebilir.

A_{mxn} matrisi ve $\alpha=(\alpha_1, \ldots, \alpha_n)$ vektörü için GKT saldırı tespiti;

$$\text{GKT saldırısı} = \max(\text{cos\_similarity} \begin{matrix} \alpha \\ a_k, k = 1, \ldots, m \end{matrix}) < \Delta b \tag{6.6}$$

formülüyle hesaplanır.

6.6. Uygulama simülasyonu

Bu tez çalışmasında önerilen modelin uygulama simülasyonu için üç ayrı test yapılmıştır. Testlerde kullanılan son kullanıcı bilgisayar sayısı, hedef adres sayısı (soket bazında) ve toplam matris boyutu aşağıdaki çizelgede gösterilmiştir.

Çizelge 6.2. Test tertibi ve düzeni

Simülasyon Nu.	Son Kullanıcı Bilgisayar Sayısı	Hedef Adres Sayısı (Socket olarak)	Matris Boyutu / Açıklama
Test #1	30	100	30x100
Test #2	100	1000	100x1000
Test #3	100	1000	100x1000

6.6.1. Simülasyon düzeni ve parametreleri

Bir numaralı testte 30 adet bilgisayar kullanıcısı, toplamda 100 adet hedef adresle veri alışverişinde bulunmuştur. İki numaralı testte 100 adet bilgisayar kullanıcısı, toplamda 1000 adet hedef adresle veri alışverişinde bulunmuştur. Bu durumda sırasıyla 30x100 ve 100x1000 boyutlarında iki adet matris test tertibinde kullanılmıştır.

Web siteleri barındırdığı içeriğe göre çeşitli kategorilere ayrılmaktadır. Bu kategorilerden bazıları alışveriş, spor, seyahat, bahis, arama motoru, haber, sosyal ağlar, teknoloji, dini içerik, yetişkin içerik, politika, iş vb.'dir. Ağ akış verisi toplanır iken toplam hedef adres sayısı 6 farklı kategoriye ayrılmıştır. Örneğin oluşturulan bu 6 farklı web kategorilerindeki site sayısı, toplam hedef adres sayısı 100 ise 15 adet haber (kategori 1), 15 adet sosyal ağ (kategori 2), 15 adet teknoloji (kategori 3), 15 adet spor (kategori 4), 15 adet bahis (kategori 5) ve 25 adet alışveriş sitesi (kategori 6) olarak belirlenmiştir. Son kullanıcı bilgisayarlarının kategorideki web sitelerine dağılımı aşağıdaki çizelgede gösterildiği gibi dağıtılmıştır. Her iki farklı test tertibi için de farklı kullanıcı dağılımları oluşturulmaya çalışılmıştır.

Çizelge 6.3. Ağ akış trafiğinin kategori ve son kullanıcı bazında dağılımı

Simülasyon Nu.	Kategori						Son kullanıcı dağılımı
	1	2	3	4	5	6	
Test #1	15	15	15	15	15	25	1/3 kullanıcı 2 kategori, 1/3 kullanıcı 3 kategori, 1/3 kullanıcı 4 kategori, 3 kullanıcı tüm kategoriler
Test #2	250	200	100	150	150	150	1/3 kullanıcı 3 kategori, 1/3 kullanıcı 4 kategori, 1/3 kullanıcı 5 kategori, 10 kullanıcı tüm kategoriler

Her bir test tertibi için 4 adet GKT saldırı simülasyonu hazırlanmış olup aşağıdaki çizelgede ayrıntılı olarak açıklanmıştır. Medya raporlarına göre Operation Aurora GKT saldırısında 6,5 TB veri kurum dışına sızdırılmıştır. Sızdırılan verinin çok büyük olması bu tezde önerilen modelde saldırının tespit edilmesini kolaylaştırmaktadır. Bu yüzden sızdırılan veri maksimum 100 GB (100 MB, 1 GB, 10 GB, 100 GB) olarak sınırlandırılmıştır. 100 GB gibi küçük boyutlardaki GKT aktörleri tarafından sızdırılan veri ile önerilen modelin testleri yapılmıştır. Kurum içerisinden büyük boyutta veri sızdırıldığı takdirde uzaysal olarak matris içinde diğer bilgisayarlardan çok farklı yeri göstereceği için saldırı tespiti çok kolay yapılabilmektedir. Küçük boyutlardaki verilerde önerilen modelin başarısı ölçülmeye çalışılmıştır.

Çizelge 6.4. GKT saldırısı simülasyonu

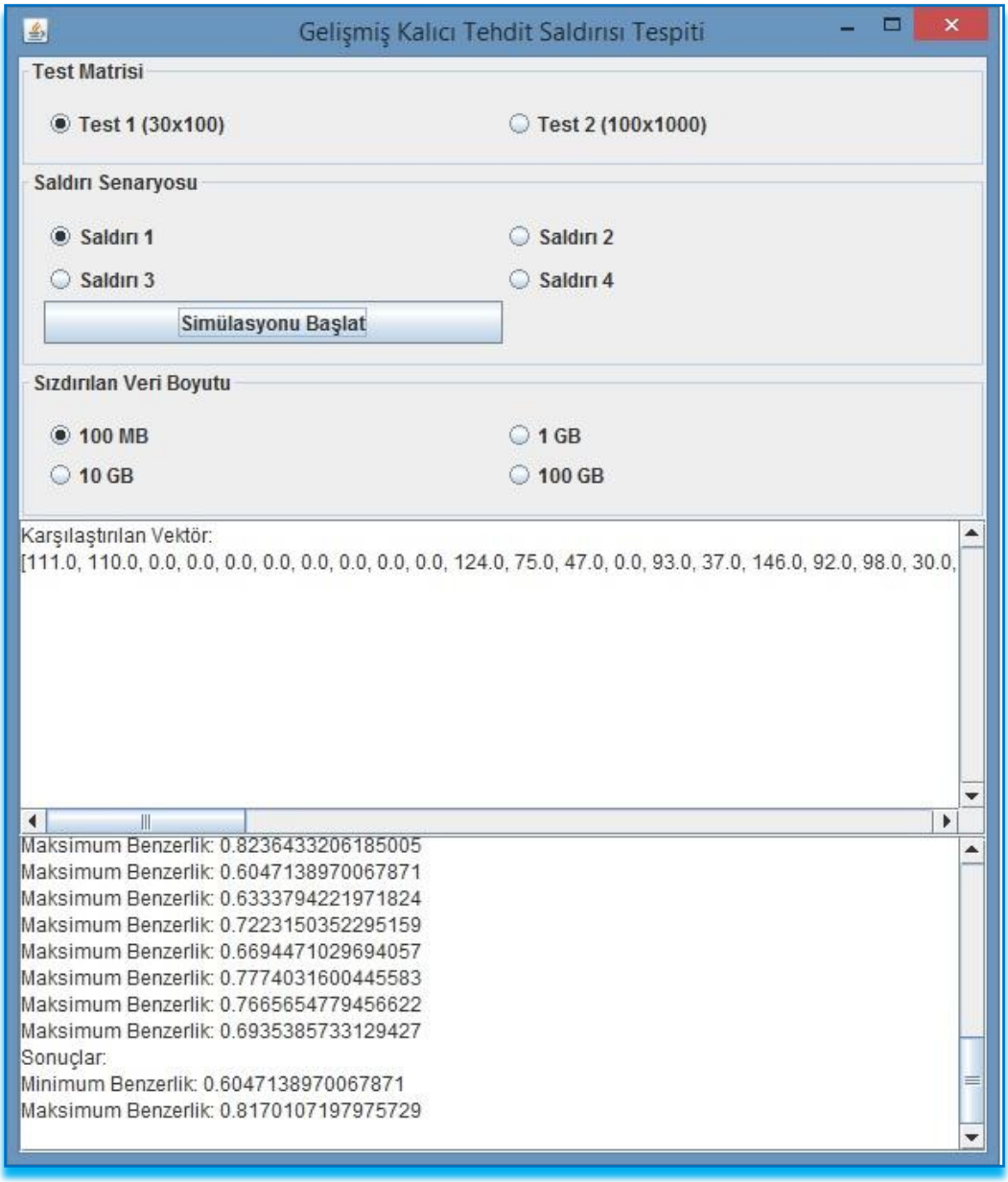
Saldırı Senaryosu	Açıklama
#1	Tek bir bilgisayardan ayrı ayrı 100 MB, 1 GB, 10 GB, 100 GB verinin tek bir hedef adrese (kullanıcılar tarafından hiç ziyaret edilmemiş) sızdırılması
#2	Tek bir bilgisayardan ayrı ayrı 100 MB, 1 GB, 10 GB, 100 GB verinin tek bir hedef adrese (kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması
#3	Tek bir bilgisayardan ayrı ayrı 100 MB, 1 GB, 10 GB, 100 GB verinin üç adet farklı hedef adrese (kullanıcılar tarafından hiç ziyaret edilmemiş) sızdırılması
#4	Tek bir bilgisayardan ayrı ayrı 100 MB, 1 GB, 10 GB, 100 GB verinin üç adet farklı hedef adrese (kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması

Test senaryosu için kullanılan Java yazılımının ekran görüntüsü Resim 5.1'de gösterilmiştir. Test #1 (A_{30x100} matrisi) ve Test #2 ($A_{100x1000}$ matrisi) senaryolarından biri seçilerek test işlemi yapılabilmektedir. Saldırı senaryosu bölümünden Çizelge 5.4'te belirtilen saldırı 1, 2, 3 ve 4 çeşitlerinden biri seçilebilmektedir. Sızdırılan veri boyutu kullanılarak kurumdan kaçırılan değerli veri boyutu belirlenebilmektedir. 100 MB, 1 GB, 10 GB ve 100 GB büyüklüğündeki değerler seçilebilmektedir. "Simülasyonu Başlat" düğmesine basıldığı takdirde seçilen saldırı senaryosu ve sızdırılan veri büyüklüğüne göre test işlemi gerçekleşmektedir.

Yazılımın alt kısmında iki adet metin alanı bulunmaktadır. Üstteki metin alanı veri sızdırma işlemini gerçekleştiren kullanıcı veya kullanıcıları temsil etmektedir. Sızdırma işlemini gerçekleştiren kullanıcı bilgisayarın ağ akış verileri, vektör gösterimi şeklinde metin alanda bulunmaktadır. Sızdırılan veri boyutu ve saldırı türü seçimine göre bu vektörün içeriği değişmektedir.

İkinci metin alanı simülasyon işlemi başlatıldıktan sonra sonuçları göstermektedir. Yazılımın hafızasında iki adet matris bulunmaktadır. Birinci matris daha önce öğretilmiş

olan, GKT saldırısının bulunmadığı, kurumun kullanıcı bilgisayarlarının davranışlarını gösteren matristir. İkinci matris yine birinci matrise benzemekte fakat farklı bir zamana ait kurumun kullanıcı davranışlarını göstermektedir. Bu ikinci matris içerisinde de GKT saldırısı bulunmamaktadır.



Resim 6.1. Yazılımın ekran görüntüsü

Sonuçlar bölümünde iki ayrı sonuç gösterilmektedir. Simülasyon başladıktan sonra yazılım, daha önceden öğrenilen matris ile ikinci matrisi karşılaştırmaktadır. İkinci matrisin satırlarında yer alan tüm kurum kullanıcı bilgisayarları, birinci matrisin satırları

ile karşılaştırılarak kosinüs benzerlik değerleri hesaplanmaktadır. Her satırın maksimum kosinüs benzerliklerinin minimum olanı *"Minimum Benzerlik"* olarak sonuçlar bölümünde gösterilmektedir. En alt kısımda yer alan *"Maksimum Benzerlik"* değeri *"Karşılaştırılan vektör"* olarak gösterilen saldırı örneğinin benzerlik sonuçlarını göstermektedir.

6.6.2. Simülasyon sonuçları

Test #1 : 30x100 matris

İlk test simülasyonu 30x100 boyutundaki matris ile yapılmıştır. Test işlemi iki bölümden oluşmaktadır. Öncelikle TDA uygulanan matris, GKT saldırısı olmayan 5 başka matris ile karşılaştırılacaktır. Eşitlik 6.5'te verilen formüle göre Δb değerleri hesaplanarak, 5 matrisin ortalaması alınacaktır. Daha sonraki bölümde TDA uygulanan matris, rastgele seçilmiş olan 5 ayrı kullanıcı bilgisayarın ağ akış verilerine göre kosinüs benzerlikleri hesaplanarak karşılaştırma yapılacaktır.

1. Bölüm:

TDA uygulanan matris ile 5 ayrı matris karşılaştırılarak minimum kosinüs benzerlikleri Çizelge 6.5'te sunulmuştur. Δb değeri 0,5922 olarak hesaplanmıştır.

Çizelge 6.5. Test #1 için 5 matrisin minimum benzerlik sonuçları

Minimum Benzerlik Sonuçları	Matrisler					
	1	2	3	4	5	**Ortalama**
	0,6047	0,5953	0,5299	0,6243	0,6072	**0,5922**

2. Bölüm:

Test için rastgele 5 kullanıcı bilgisayarın ağ akış trafiği verisi seçilmiş olup, saldırı çeşidi ve sızdırılan veri boyutuna göre kosinüs benzerlik sonuçları Çizelge 6.6'da gösterilmiştir. Toplamda 85 adet test sonucu sunulmuştur.

Çizelge 6.6. Test #1 için GKT saldırısı simülasyonu sonuçları

Saldırı Senaryosu	Sonuçlar (Minimum Benzerlik Ortalaması (Δb): 0,5922)				
	Saldırı olmadığında Kosinüs Benzerliği	100 MB veri sızdırma için Kosinüs Benzerliği	1 GB veri sızdırma için Kosinüs Benzerliği	10 GB veri sızdırma için Kosinüs Benzerliği	100 GB veri sızdırma için Kosinüs Benzerliği
Saldırı #1	0,8170 0,8593 0,7456 0,7133 0,8241	0,7945 0,8359 0,6955 0,6719 0,8098	0,3149 0,3512 0,3069 0,3045 0,3466	0,0341 0,0407 0,0327 0,0308 0,0389	0,0034 0,0039 0,0033 0,0031 0,0038
Saldırı #2		0,7944 0,8273 0,6989 0,6891 0,8060	0,3692 0,3901 0,3522 0,3417 0,3712	0,3796 0,3811 0,3417 0,3309 0,3805	0,3779 0,3807 0,3349 0,2944 0,3781
Saldırı #3		0,8094 0,8512 0,7498 0,7096 0,8137	0,4792 0,5001 0,4595 0,4462 0,4891	0,0589 0,0593 0,0432 0,0401 0,0573	0,0059 0,0063 0,0051 0,0049 0,0061
Saldırı #4		0,8093 0,8271 0,7206 0,7113 0,8182	0,4792 0,5122 0,4501 0,4461 0,4827	0,4139 0,4277 0,4032 0,3921 0,4187	0,4115 0,4183 0,4004 0,3932 0,4152

30 kullanıcı bilgisayarının ağ akış verisinin bulunduğu bu simülasyonda minimum ortalama kosinüs benzerliği 0,5922 olarak test hesaplanmıştır. Bu değer öğrenilen matris ile kullanıcıların davranışlarını gösteren farklı zamanlara ait olan matrisin satır satır karşılaştırılması ile elde edilen maksimum benzerliğin minimum olan değeridir. Bu değerin altında çıkan kosinüs benzerlik oranları için GKT saldırısı tespiti yapıldığı söylenebilir.

Sızdırılan veri boyutunun 100 MB olduğu durumlarda her saldırı senaryosu için, minimum benzerlikten daha büyük kosinüs benzerlikleri ortaya çıkmıştır. Küçük boyutlardaki veri sızıntılarının tespiti normal ağ trafik akışına benzediği için kosinüs benzerlik oranları minimumun üzerinde çıkmıştır.

Sızdırılan veri boyutu 1 GB olarak değiştirildiğinde, simülasyon içerisinde kullanılan her dört saldırı türü için de GKT saldırı tespiti doğru olarak yapılmıştır. Her saldırı için kosinüs benzerliklerinin hepsi, minimum benzerlik 0,6047 oranının altında kalmıştır. Sızdırılan veri boyutu 10 GB ve 100 GB boyutlarına göre değerlendirildiğinde bu tez çalışmasında önerilen model daha iyi sonuçlar vermiştir.

Test #2 : 100x1000 matris

İkinci test simülasyonu 100x1000 boyutundaki matris ile yapılmıştır. Test işlemi iki bölümden oluşmaktadır. İlk test işleminde olduğu gibi öncelikle TDA uygulanan matris, GKT saldırısı olmayan 5 başka matris ile karşılaştırılacaktır. Eşitlik 6.5'te verilen formüle göre Δb değerleri hesaplanarak, 5 matrisin ortalaması alınacaktır. Daha sonraki bölümde TDA uygulanan matris, rastgele seçilmiş olan 5 ayrı kullanıcı bilgisayarın ağ akış verilerine göre kosinüs benzerlikleri hesaplanarak karşılaştırma yapılacaktır.

1. Bölüm:

TDA uygulanan matris ile 5 ayrı matris karşılaştırılarak minimum kosinüs benzerlikleri Çizelge 6.5'te sunulmuştur. Δb değeri 0,5654 olarak hesaplanmıştır. Test #1'de hesaplanan Δb değerinden biraz daha düşük benzerlik oranı ortaya çıkmıştır.

Çizelge 6.7. Test #2 için 5 matrisin minimum benzerlik sonuçları

Minimum Benzerlik Sonuçları	Matrisler					
	1	2	3	4	5	**Ortalama**
	0,6068	0,5949	0,6118	0,4077	0,6058	**0,5654**

2. Bölüm:

İkinci test simülasyonu 100x1000 boyutundaki matris ile yapılmıştır. Saldırı çeşidi ve sızdırılan veri boyutuna göre kosinüs benzerlik sonuçları Çizelge 6.8'de gösterilmiştir.

Çizelge 6.8. Test #2 için GKT saldırısı simülasyonu sonuçları

Saldırı Senaryosu	Sonuçlar (Minimum Benzerlik Ortalaması (Δb): 0.5654)				
	Saldırı olmadığında Kosinüs Benzerliği	100 MB veri sızdırma için Kosinüs Benzerliği	1 GB veri sızdırma için Kosinüs Benzerliği	10 GB veri sızdırma için Kosinüs Benzerliği	100 GB veri sızdırma için Kosinüs Benzerliği
Saldırı #1	0,7558 0,6571 0,7234 0,8054 0,8173	0,7442 0,6445 0,6912 0,7954 0,8041	0,3716 0,3349 0,3544 0,4049 0,4103	0,0426 0,0331 0,0380 0,0440 0,0453	0,0042 0,0033 0,0038 0,0044 0,0045
Saldırı #2		0,7442 0,6539 0,6914 0,7631 0,7746	0,3716 0,3351 0,3547 0,4135 0,4268	0,2517 0,2289 0,2305 0,0447 0,0449	0,2485 0,2095 0,2218 0,2517 0,2632
Saldırı #3		0,7519 0,6528 0,6958 0,7702 0,7981	0,5284 0,4558 0,4855 0,4987 0,5018	0,0735 0,0651 0,0693 0,0453 0,0511	0,0073 0,0069 0,0071 0,0087 0,0089
Saldırı #4		0,7519 0,6626 0,6965 0,7702 0,7912	0,5284 0,4617 0,4974 0,4992 0,5123	0,2561 0,2132 0,2318 0,2788 0,2817	0,2510 0,2123 0,2311 0,2649 0,2805

100 kullanıcı bilgisayarının ve 1000 farklı hedef adresin ağ akış verisinin bulunduğu bu simülasyonda minimum benzerlik 0,5654 olarak hesaplanmıştır. Test #1 simülasyon denemesinde olduğu gibi Test #2'de de sızdırılan veri boyutunun 100 MB olduğu durumlarda her bir saldırı senaryosu için, minimum benzerlikten daha büyük kosinüs benzerlikleri ortaya çıkmıştır. Küçük boyutlardaki veri sızıntılarının tespiti normal ağ trafik akışına benzediği için kosinüs benzerlik oranları minimumun üzerinde çıkmıştır.

Sızdırılan veri boyutu 1 GB olarak değiştirildiğinde, Test #1 simülasyonda olduğu gibi her dört saldırı türü için de GKT saldırı tespiti doğru olarak yapılmıştır. Her saldırı için kosinüs benzerliklerinin hepsi, minimum benzerlik 0,6068 oranının altında kalmıştır. Sızdırılan veri boyutu 10 GB ve 100 GB olduğunda bu tez çalışmasında önerilen model daha iyi sonuçlar vermiştir.

Test #3 : 100x1000 matris

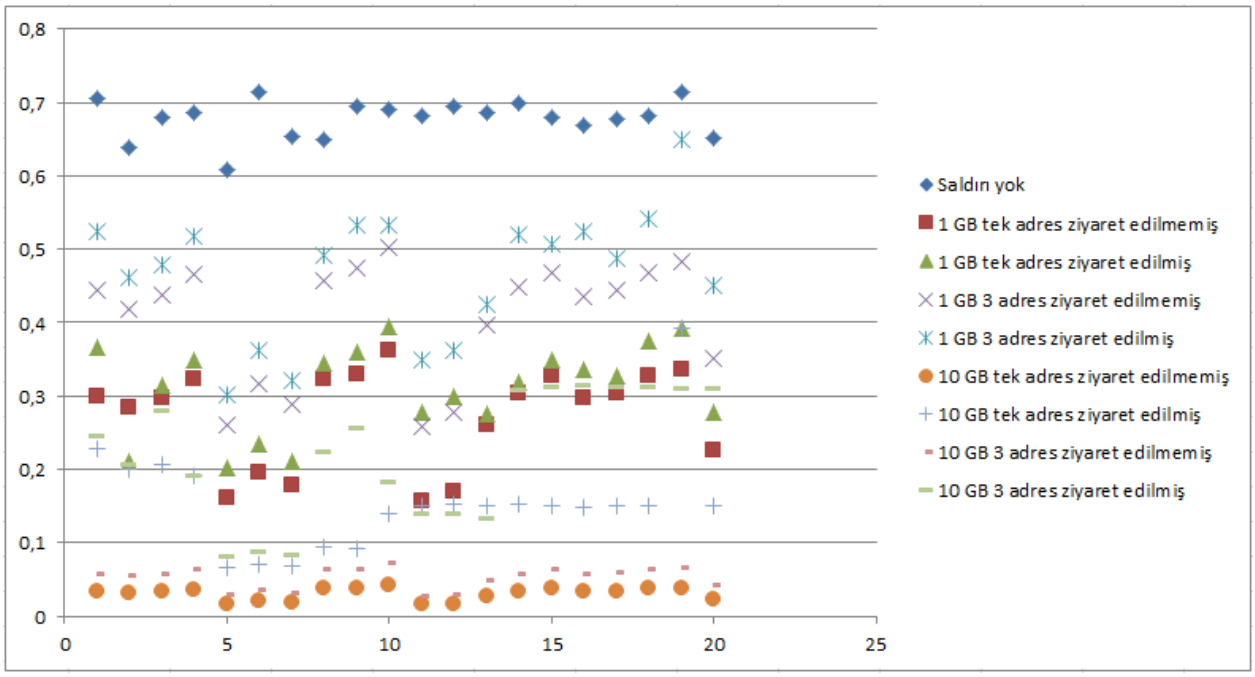
Test #3 simülasyonu diğer Test #1 ve Test #2 farklıdır. Bu testte, 100 MB ve 100 GB büyüklüğündeki veri sızdırmaları kullanılmamıştır. 1 GB ve 10 GB veri büyüklükleri kullanılmıştır. Ayrıca diğer iki test işleminde, sadece 5 kullanıcı bilgisayar için test yapılmasına rağmen Test #3'te ise 20 kullanıcı bilgisayar için test işlemi yapılmıştır.

Rastgele seçilen 20 kullanıcı bilgisayar ağ akış verisi için maksimum benzerlik değerleri Çizelge 6.7'de gösterildiği gibi elde edilmiştir. Test #2 simülasyonunda Δb değeri 0,6068 olarak hesaplanmıştı. Test #3'te gerçekleştirilen saldırı simülasyonlarında, Δb değerini geçen sadece 19 numarılı bilgisayar için "1 GB verinin üç adet farklı hedef adrese (kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması" saldırı tipindeki işlemdir. Δb değeri 0,6501 olarak hesaplanmıştır.

Çizelge 6.9. Test #3 için GKT saldırısı simülasyonu sonuçları

Saldırı Nu.		#1	#2	#3	#4	#1	#2	#3	#4
Kull. Bilg.	Saldırı olmadığında Kosinüs Benzerliği	1 GB veri sızdırma için Kosinüs Benzerliği				10 GB veri sızdırma için Kosinüs Benzerliği			
1	0,7059	0,2985	0,3659	0,4437	0,5239	0,0329	0,2286	0,0568	0,2446
2	0,6378	0,2853	0,2099	0,4176	0,4623	0,0318	0,1991	0,0550	0,2062
3	0,6798	0,2966	0,3139	0,4372	0,4780	0,0329	0,2073	0,0568	0,2803
4	0,6869	0,3228	0,3484	0,4657	0,5168	0,0365	0,1915	0,0630	0,1924
5	0,6074	0,1610	0,2022	0,2612	0,3025	0,0167	0,0670	0,0289	0,0816
6	0,7133	0,1966	0,2351	0,3173	0,3613	0,0204	0,0710	0,0353	0,0887
7	0,6535	0,1793	0,2098	0,2895	0,3203	0,0186	0,0681	0,0322	0,0837
8	0,6500	0,3233	0,3448	0,4580	0,4914	0,0372	0,0950	0,0642	0,2247
9	0,6958	0,3301	0,3604	0,4748	0,5321	0,0374	0,0930	0,0646	0,2555
10	0,6912	0,3616	0,3952	0,5034	0,5323	0,0423	0,1402	0,0730	0,1832
11	0,6812	0,1565	0,2782	0,2578	0,3502	0,0160	0,1496	0,0278	0,1387
12	0,6958	0,1697	0,2984	0,2778	0,3622	0,0174	0,1515	0,0302	0,1391
13	0,6872	0,2598	0,2754	0,3969	0,4244	0,0280	0,1507	0,0484	0,1332
14	0,6986	0,3034	0,3188	0,4478	0,5195	0,0336	0,1518	0,0581	0,3073
15	0,6805	0,3273	0,3481	0,4688	0,5061	0,0372	0,1512	0,0643	0,3129
16	0,6681	0,2967	0,3361	0,4352	0,5231	0,0330	0,1481	0,0571	0,3154
17	0,6765	0,3044	0,3277	0,4449	0,4884	0,0340	0,1494	0,0588	0,3117
18	0,6816	0,3271	0,3753	0,4688	0,5408	0,0372	0,1499	0,0643	0,3129
19	0,7141	0,3353	0,3932	0,4838	0,6501	0,0379	0,3932	0,0654	0,3094
20	0,6522	0,2251	0,2781	0,3504	0,4509	0,0239	0,1505	0,0414	0,3111

Şekil 6.3'te Test #3 simülasyonunun kosinüs benzerliklerinin dağılımı gösterilmektedir. Saldırı olmadığında benzerlik oranları Δb değerinin üzerinde iken saldırı gerçekleştiğinde benzerlik oranlarının çok düştüğü görülmektedir. 20 adet kullanıcı bilgisayar için yapılan 8 ayrı test işleminde (toplam 20x8=160) başarı oranı % 99 olarak gerçekleşmiştir.



Şekil 6.3. Test #3 kosinüs benzerlik dağılımı

7. SONUÇ VE DEĞERLENDİRME

Kurumlar ve devletler faaliyetlerinde ne kadar çok bilgi teknolojilerini kullanırlarsa, GKT saldırılarına karşı alınacak güvenlik tedbirlerinin önemi de o derecede artmaktadır. GKT saldırılarında genellikle sıfırıncı gün açıklığı kullanıldığı için gerçek zamanlı olarak tespit edilebilmesi ve kutusundan çıkarıp kurduğunuz tek bir donanımla GKT saldırısının önlenebilmesi mümkün gözükmemektedir.

GKT saldırılarını gerçekleştiren aktörler, otomotize edilmiş bilgisayar yazılımlarından ziyade her zaman saldırı metadolojisini değiştirebilen insanlardır. Asıl tehdit unsurunun zararlı yazılımdan ziyade insan olmasından dolayı hedefte olan kurum veya devletin bilgi sistemleri mutlaka bir şekilde ele geçirilecektir. Önümüzdeki senelerde GKT saldırıları devletler için çok büyük baş ağrılarına neden olacak olup, kurumları ve devletleri, siber güvenlik açısından stabil ve sakin bir gelecek beklememektedir.

Büyük finans desteği olan ve teknik kapasitesi çok yüksek GKT aktörlerine karşı alınacak tedbirlerden en önemlisi, bilgi sistemleri altyapısının sürekli olarak izlenmesidir. Bilgi sistemlerinin izlenebilmesi ve GKT saldırı tespitinin yapılabilmesi için iz bilgilerinin ve log kayıtlarının geniş bir yelpazede tutuluyor olması gerekmektedir. Ağ trafiğinin devamlı olarak izlenmesi ve çeşitli sistemlerin ürettiği log kayıtlarının tutulmasının yanında yorumlanarak anlam çıkarılıyor olması en önemli unsurlardan biridir [42].

Bu tez çalışmasında GKT saldırısı aktörlerinin, normal kullanıcı davranışından mutlaka farklı bir davranış sergileyeceklerinden yola çıkarak, bu farklılığın tespitine yönelik bir çalışma yapılmıştır. Kullanıcıların ağ akış trafiği bilgileri kullanılarak, farklı davranış sergileyen kullanıcıların uzaysal olarak da normal kullanıcılardan farklı bir yerde olacağı düşünülmüştür. Bu doğrultuda satır bölümünde kullanıcı bilgisayarlarının, sütun bölümünde ise kullanıcı bilgisayarlarının ziyaret ettiği hedef adres-port bilgilerinin olduğu bir matris oluşturulmuştur. Matris içeriğini ise megabyte cinsinden kullanıcıların ağ akış trafik miktarları oluşturmaktadır. Oluşturulan bu matris üzerinde çeşitli normalizasyon (TF-THAF, beyaz liste) işlemleri yapıldıktan sonra tekil değer ayrıştırması ile boyut küçültme işlemi yapılmıştır. Kosinüs benzerliği kullanılarak kullanıcı bilgisayar

davranışlarının uzaysal olarak oryantasyonunun farklılığı ölçülmüştür. Önerilen yaklaşımın başarılı olduğunu test edip göstermek için ayrıca bir yazılım geliştirilmiştir.

Bu tez çalışmasında GKT saldırılarının tespiti için, kurum üzerine herhangi bir maliyet getirmeyen, mevcut bilgi sistem altyapısını kullanan bir model sunulmuştur. Oluşturulan bu sistemde kullanılan tek veri, ağ akış trafiğidir. Sistem altyapısını izlemek ve saldırı tespitini yapmak için, kullanıcı bilgisayarlarına program, ajan vb. yazılımların kurulmasına gerek kalmadan, sadece ağ cihazları arasında uygun görülenlerin üzerinde ağ akış trafiği etkin hale getirilerek sistem kullanılabilmektedir. Önerilen model, gerçek dünya sistemlerine kolaylıkla uyarlanabilir.

Bu tez çalışmasında 3 farklı test gerçekleştirilmiştir. Birinci test, 30 kullanıcı bilgisayarı, 100 farklı hedef adres için yapılmıştır. İkinci test ise 100 kullanıcı bilgisayarı, 1000 farklı hedef adres için yapılmıştır. 100 MB, 1 GB, 10 GB ve 100 GB büyüklüğünde veri sızdırma işlemi gerçekleşen saldırılarda 100 MB veri büyüklüğü için saldırı tespiti yapılamamıştır. Her iki testte de kosinüs benzerlikleri $\Delta b'$den büyük çıkmıştır. 1 GB, 10 GB ve 100 GB veri büyüklüklerinde ise başarılı sonuçlar alınmıştır.

Test #3 simülasyonunda, Test #2'de olduğu gibi 100x1000 matrisi kullanılmıştır. Sızdırılan veri büyüklüğü için sadece 1 GB ve 10 GB'lık saldırılar seçilmiştir. 20 kullanıcı bilgisayarı için ayrı ayrı 8 test yapılmış ve %99 oranında başarı elde edilmiştir. Sadece "1 GB verinin üç adet farklı hedef adrese (kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması" saldırı tipindeki işlemde kosinüs benzerliği $\Delta b'$den büyük çıkmıştır.

Saldırı tespiti için sadece iç ağdaki kullanıcı bilgisayarlarından dışarıya giden trafik kullanılmıştır. Sistemi daha da geliştirmek adına, kullanıcı bilgisayarlarından gelen uygulama, güvenlik, sistem vb. log kayıtlarının bazıları süzülerek, oluşturulan matrise ayrı birer boyut olarak eklenmek suretiyle daha etkin sonuçlar elde edilebilir. Sistemdeki ağ trafiği ve log kayıtlarının beraber oluşturacağı uzay üzerinden yapılan değerlendirmenin false pozitif sayısını azaltabileceği değerlendirilmektedir.

Bu tez çalışmasında yoğun olarak üzerinde durulan ağ akış bilgileri ile ileriki çalışmalar için önerilen kullanıcı bilgisayar log kayıtlarından farklı olarak daha başka bilgiler ile de

matrisin boyutları artırılabilir. Eklenen her yeni boyut ile yanlış güvenlik alarmlarının azalacağı bir sistem tesis edilebilir. Ayrıca tekil değer ayrıştırmasının uygulandığı matris tek bir matris olarak kullanılabileceği gibi, ağ akış trafiği ve sistem log kayıtları için ayrı ayrı oluşturulabilir.

Simülasyon ortamında gerçekleştirilen bu test çalışmalarının gerçek veriler ile de yapılması gerekmektedir. Test ortamında ümit verici sonuçlar elde edilmiştir. Gerçek ortamda da test edilip karşılaşılacak problemlerin çözümü ile çok esneklik sağlayan bu model, bilgi sistemleri dünyasında yaşanan hızlı değişimlere çok çabuk uyum sağlayarak, gerçek dünya problemlerine hizmet edecek kapasiteye sahiptir.

Geliştirilen sisteme ait elde edilen çıkarımlar aşağıda belirtilmiştir;

- Metin sınıflandırma, arama motoru optimizasyonu uygulamarında başarılı şekilde kullanılan tekil değer ayrıştırması işleminin, güvenlik üzerine GKT tespitinde de kullanılabileceği,

- Beyaz liste ve TF-THAF normalizasyon işlemlerinin sistemin başarısına doğrudan etkisi olduğu,

- Ağ akış trafiğinin GKT saldırısı tespitinde başarılı şekilde kullanılabileceği,

- Geliştirilen sistemin simülasyon ortamında yapılan testlerde başarılı olduğu, gerçek veriler ile test edilmesi gerektiği,

- Kurumlarda bulunan ağ ve sistem yöneticilerinin, bilgi sistem altyapısında meydana gelen aykırılılıkları daha kolay fark edebilmesini sağlayacağı,

- Hatalı GKT saldırısı bildirimlerinin matrise eklenecek başka boyutlar ile azaltılabileceği,

- Mevcut bilgi sistem altyapısına hiçbir yük getirmeden, bilgisayarlara ayrı bir program, ajan vb. kuruluma gerek kalmadan, önerilen modelin saldırı tespitinde kullanılabileceği,

- Yapılan araştırmalarda, ülkemizde ve dünyada GKT saldırılarının tespitine yönelik bu tez çalışmasında önerilen modelin bulunmadığı ve ilk olma niteliği taşıdığı,

- Bu çalışmanın bu alanda yapılacak diğer çalışmaların önünü açacağı değerlendirilmektedir.

KAYNAKLAR

1. Vance A. (2014), “Flow based analysis of advanced persistent threats, detecting targeted attacks in cloud computing”, Infocommunication Science and Technology.

2. Bridges S.M., Vaughn R.B. (2000), “Fuzzy data mining and genetic algorithms applied to intrusion detection”, Proceedings of the National Information Systems Security Conference, Baltimore, MD.

3. Smallwood D., Vance A. (2011), “Intrusion Analysis with Deep Packet Inspection: Increasing Efficiency of Packet Based Investigations”, International Conference on Cloud and Service Computing.

4. Friedberg I., Skopik F., Settanni G., Fiedler R. (2015), "Combating advanced persistent threats: from network event correlation to incident detection," Computers & Security, 48, 35-57.

5. Denning D.E. (1987), “An intrusion-detection model”, Software Engineering IEEE Trans. 2, 222, e32.

6. Sabahi F., Movaghar A. (2008), “Intrusion detection: a survey”, In: Systems and Networks Communications, ICSNC'08. 3rd International Conference on. IEEE; 23e6.

7. Axelsson S. (2000), “Intrusion detection systems: A survey and taxonomy”, Technical Report, Chalmers University of Technology.

8. Yu Y. (2012), “A survey of anomaly intrusion detection techniques”, J Comput.Sci. Coll;28(1):9e17.

9. Chandola V., Banerjee A., Kumar V. (2009), “Anomaly detection: a survey”, ACM Comput Surv (CSUR);41(3):15.

10. Ponemon Institute LLC, “The State of Advanced Persistent Threats,” Ponemon Research Report, 2013.

11. İnternet: FireEye Threat Intelligence, “Advanced Threat Report: 2013”, URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.fireeye.com%2Fblog%2Fthreat-research%2F2014%2F02%2Fthe-2013-fireeye-advanced-threat-report.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

12. İnternet: McAfee Security, “Combating Advanced Persistent Threats”, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.mcafee.com%2Fus%2Fresources%2Fwhite-papers%2Fwp-combat-advanced-persist-threats.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

13. İnternet: Nfsen ağ analiz yazılımı, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fnfsen.sourceforge.net%2F&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

14. İnternet: Tcpdump ağ yazılımı, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.tcpdump.org%2F&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

15. İnternet: Apache lineer cebir kütüphanesi, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fcommons.apache.org%2Fproper%2Fcommons-math%2Fuserguide%2Flinear.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

16. Cole E. (2013), "Advanced persistent threat: understanding the danger and how to protect your organization", Waltham, Mass., Syngress.

17. İnternet: Baker K. (2015), "Singular Value Decomposition Tutorial", URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.ling.ohio-state.edu%2F%7Ekbaker%2Fpubs%2FSingular_Value_Decomposition_Tutorial.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

18. İnternet: NetFlow, URL: http://www.webcitation.org/query?url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FNetFlow&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

19. İnternet: Mandiant Intelligence Center, "APT1: Exposing one of China's cyber espionage units", mandiant.com, 2013, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fintelreport.mandiant.com%2FMandiant_APT1_Report.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

20. Shakarian P., Shakarian J., Ruef A. (2013), "Introduction to cyber-warfare: a multidisciplinary approach," Newnes.

21. Bhatt P., Yano E.T., Gustavsson M. (2014), "Towards a framework to detect multi-stage advanced persistent threats attacks," In: Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on. IEEE, 390-395.

22. Brewer R. (2014), "Advanced persistent threats: minimising the damage," Network Security, 5-9.

23. Tankard C. (2011), "Advanced persistent threats and how to monitor and deter them," Network security, 2011, 8, 16-19.

24. D. Bradbury, "Shadows in the cloud: Chinese involvement in advanced persistent threats," Network Security, 2010, 2010.5: p. 16-19.

25. Virvilis N., Gritzalis D., Apostopoulos T. (2013), "Trusted computing vs. advanced persistent threats: can a defender win this game?" In: Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), IEEE, 396-403.

26. İnternet: Hale B., "Estimating log generation for security information event and log management", URL:

http://www.webcitation.org/query?url=http%3A%2F%2Fcontent.solarwinds.com%2Fcreative%2Fpdf%2FWhitepapers%2Festimating_log_generation_white_paper.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

27. MacDonald N. (2012), "Information security is becoming a big data analytic problem", Gartner.

28. Oppliger R., Rytz R. (2005), "Does trusted computing remedy computer security problems?", Security & Privacy, IEEE, 3(2), 16-19.

29. Thomson G. (2011), "APTs: a poorly understood challenge," Network Security, 11, 9-11.

30. İnternet: Falliere N., Murchu L.O., Chien E., "W32.Stuxnet dossier", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.symantec.com%2Fcontent%2Fen%2Fus%2Fenterprise%2Fmedia%2Fsecurity_response%2Fwhitepapers%2Fw32_stuxnet_dossier.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

31. İnternet: Kaspersky Lab Global Research and Analysis Team, "Gauss: abnormal distribution", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fkasperskycontenthub.com%2Fwp-content%2Fuploads%2Fsites%2F43%2Fvlpdfs%2Fkaspersky-lab-gauss.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

32. İnternet: Kaspersky Lab Report, "The Regin platform nation-state ownage of GSM networks", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fsecurelist.com%2Ffiles%2F2014%2F11%2FKaspersky_Lab_whitepaper_Regin_platform_eng.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

33. İnternet: Kaspersky Labs Global Research & Analysis Team, "A 2014 update on one of the world's most unusual APT operations", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fsecurelist.com%2Fblog%2Fincidents%2F64107%2Fminiduke-is-back-nemesis-gemina-and-the-botgen-studio&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

34. İnternet: Kaspersky Lab, "Red October diplomatic cyber attacks investigation", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fsecurelist.com%2Fanalysis%2Fpublications%2F36740%2Fred-october-diplomatic-cyber-attacks-investigation%2F&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

35. İnternet: Symantec Security Response, "W32.Duqu the precursor to the next Stuxnet", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.symantec.com%2Fcontent%2Fen%2Fus%2Fenterprise%2Fmedia%2Fsecurity_response%2Fwhitepapers%2Fw32_duqu_the_precursor_to_the_next_stuxnet_research.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

36. İnternet: Bejtlich R. (2010), "Understanding the advanced persistent threat", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fsearchsecurity.techtarget.co

m%2FmagazineContent%2FUnderstanding-the-advanced-persistent-threat&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

37. İnternet: The Washington Post, "Google China cyberattack part of vast espionage campaign, experts say", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.washingtonpost.com%2Fwp-dyn%2Fcontent%2Farticle%2F2010%2F01%2F13%2FAR2010011300359.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

38. Shakarian P., Shakarian J., Ruef A. (2013), "Duqu, Flame, Gauss, the next generation of cyber exploitation", Syngress, 159-170.

39. Ewaida B. (2010), "Pass-the-hash attacks: tools and mitigation", SANS Institude.

40. İnternet: Verizon, "2013 data breach investigations report", 2013, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.verizonenterprise.com%2FDBIR%2F2013%2F&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

41. Cisco, Cisco IOS NetFlow Technology Data Sheet.

42. Kent K., Souppay M. (2006), "Guide to computer security log management", National Institude of Standards and Technology.

43. İnternet: Symantec security response team, "Regin: top tier espionage tool enables stealthy surveillance", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.symantec.com%2Fcontent%2Fen%2Fus%2Fenterprise%2Fmedia%2Fsecurity_response%2Fwhitepapers%2Fregin-analysis.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

44. İnternet: Early S., "German government denies falling victim to cyber attack", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.dw.de%2Fgerman-government-denies-falling-victim-to-cyber-attack%2Fa-18158951&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

45. İnternet: Kelley M. (2013), "The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous Than Previously Thought", Business Insider, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.businessinsider.com%2Fstuxnet-was-far-more-dangerous-than-previous-thought-2013-11&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

46. İnternet: Tan J., Tan A., "Business Under Threat, Technology Under Attack, Ethics Under Fire: The Experience of Google in China" URL: http://www.webcitation.org/query?url=https%3A%2F%2Fbotfl.nd.edu%2Fpdf%2Fsecurity%2Fgoogle.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

47. İnternet: Cisco, "Introduction to Cisco IOS NetFlow - A Technical Overview", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.cisco.com%2Fc%2Fen%2Fus%2Fproducts%2Fcollateral%2Fios-nx-os-software%2Fios-

netflow%2Fprod_white_paper0900aecd80406232.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

48. İnternet: Erdem Ö., "Ağ Akış Kayıtlarının Siber Güvenlikte Kullanımı", URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.bilgiguvenligi.gov.tr%2Fag-guvenligi%2Fag-akis-kayitlarinin-siber-guvenlikte-kullanimi.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

49. İnternet: Cisco Systems NetFlow Services Export Version 9, URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.rfc-base.org%2Frfc-3954.html&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

50. Tavallaee M., Stakhanova N., Ghorbani A.A. (2010), "Toward credible evaluation of anomaly-based intrusion-detection methods," IEEE Trans. Syst. Man Cybern. C Appl. Rev., 40, 5, 516–524.

51. İnternet: Neumann B., "Knowledge Management and Assistance Systems", URL: http://www.webcitation.org/query?url=http%3A%2F%2Fkogs-www.informatik.uni-hamburg.de%2F%7Eneumann%2FWMA-WS-2007%2FWMA-1.pdf&date=2015-08-11 Son Erişim Tarihi: 11 Ağustos 2015.

52. Bhuyan M., Bhattacharyya D., Kalita J. (2014). "Network anomaly detection: methods, systems and tools." Communications Surveys & Tutorials, IEEE 16.1, 303-336.

53. Liao Y., Vemuri V.R. (2002), "Use of K-nearest neighbor classifier for intrusion detection", Computers & Security, 21, 439–448.

54. Garcia-Teodoro P. (2009), "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security, 28.1, 18-28.

55. Lazarevic A., Kumar V., Srivastava J. (2005), "Intrusion detection: a survey", Managing cyber threats: issues, approaches, and challenges Springer Verlag, 330.

56. Androulidakis G., Vassilis C., Symeon P. (2009), "Network anomaly detection and classification via opportunistic sampling", Network, IEEE 23.1, 6-12.

57. Barford P., Plonka D. (2001), "Characteristics of Network Traffic Flow Anomalies," Proc. 1st ACM SIGCOMM Internet Measurement Wksp., San Francisco, CA, 69–74.

58. Duffield N., Lund C., Thorup M. (2005), "Estimating Flow Distributions From Sampled Flow Statistics," IEEE/ACM Trans. Net., 13, 5, 933–46.

59. Bhuyan M., Bhattacharyya D.K., Kalita K. (2012), "An effective unsupervised network anomaly detection method." Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ACM.

60. Davis J., Clark A.J. (2011), "Data preprocessing for anomaly based network intrusion detection: A review." Computers & Security 30.6, 353-375.

61. Bereziński P. (2014), "Network anomaly detection using parameterized entropy." Computer Information Systems and Industrial Management. Springer Berlin Heidelberg, 465-478.

62. Adamova K. (2014), "Network anomaly detection in the cloud: The challenges of virtual service migration", Communications (ICC), IEEE International Conference on. IEEE.

63. Akoglu L., Faloutsos C. (2013), "Anomaly, event, and fraud detection in large network datasets." Proceedings of the sixth ACM international conference on Web search and data mining. ACM.

64. Frei A., Rennhard M. (2007), ” Histogram Matrix: Log File Visualization for Anomaly Detection”, IEEE.

65. Rufai A.M., Anbarjafari G., Demirel H. (2014), "Lossy image compression using singular value decomposition and wavelet difference reduction." Digital Signal Processing, 24, 117-123.

66. Yoshizawa T. (2014), "Singular value decomposition of multiarray data and its applications", Recent Developments in Clustering and Data Analysis: Développements Récents en Classification Automatique et Analyse des Données: Proceedings of the Japanese-French Scientific Seminar, Academic Press.

EKLER

EK-1. GKT saldırı tespiti Java kaynak kodları

```
package advancedpt;

import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.*;
import java.io.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.BorderFactory;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import org.apache.commons.math3.linear.Array2DRowRealMatrix;
import org.apache.commons.math3.linear.RealMatrix;
import org.apache.commons.math3.linear.SingularValueDecomposition;

/**
 *
 * @author bayrak
 */
public class AdvancedPTapache extends JFrame {
    JTextArea sonuclarTextArea = new JTextArea();
    JTextArea vectorTextArea = new JTextArea();

    JRadioButton test1, test2/*, test3*/;
    JRadioButton saldiri1, saldiri2, saldiri3, saldiri4/*, saldiri5, saldiri6*/;

    //matris dosyaları ogrenilen
    public static String OTUZLUKogrenilen = "src\\30luk.txt";
    public static String YUZLUKogrenilen = "src\\100luk.txt";
    public static String BINLIKogrenilen = "src\\1000lik.txt";

    public static String OTUZLUKtestedilecek = "src\\30luktest.txt";
    public static String YUZLUKtestedilecek = "src\\100luktest.txt";
    public static String BINLIKtestedilecek = "src\\1000liktest.txt";

    private String testDosyasiOgrenilen = AdvancedPTapache.OTUZLUKogrenilen;
    private String testDosyasiTestEdilecek = AdvancedPTapache.OTUZLUKtestedilecek;

    //saldırı ornekleri
    public static double[] SALDIRI11 = new double[]{};
    public static double[] SALDIRI12 = new double[]{};
```

(EK-1'in devamı)

```
    public static double[] SALDIRI13 = new double[]{};
    public static double[] SALDIRI14 = new double[]{};

    public static double[] SALDIRI21 = new double[]{};
    public static double[] SALDIRI22 = new double[]{};
    public static double[] SALDIRI23 = new double[]{};
    public static double[] SALDIRI24 = new double[]{};

    public static double[] SALDIRI31 = new double[]{};
    public static double[] SALDIRI32 = new double[]{};
    public static double[] SALDIRI33 = new double[]{};
    public static double[] SALDIRI34 = new double[]{};

    private double[] saldiriOrnegi = AdvancedPTapache.SALDIRI11;



    AdvancedPTapache() {
        super("Gelişmiş Kalıcı Tehdit Saldırısı Tespiti");

        //northPanel
        JPanel northPanel = new JPanel(new BorderLayout());
        JPanel testPanel = new JPanel(new GridLayout(1, 3));
        testPanel.setBorder(BorderFactory.createCompoundBorder(
                    BorderFactory.createTitledBorder("Test Matrisi"),
                    BorderFactory.createEmptyBorder(10,10,10,10)));
        JPanel saldiriPanel = new JPanel(new GridLayout(3, 3));
        saldiriPanel.setBorder(BorderFactory.createCompoundBorder(
                    BorderFactory.createTitledBorder("Saldırı Senaryosu"),
                    BorderFactory.createEmptyBorder(10,10,10,10)));
        JPanel sizdirilanVeriPanel = new JPanel(new GridLayout(2,2));
        sizdirilanVeriPanel.setBorder(BorderFactory.createCompoundBorder(
                    BorderFactory.createTitledBorder("Sızdırılan Veri Boyutu"),
                    BorderFactory.createEmptyBorder(10,10,10,10)));
        northPanel.add(testPanel, BorderLayout.NORTH);
        northPanel.add(saldiriPanel, BorderLayout.CENTER);
        northPanel.add(sizdirilanVeriPanel, BorderLayout.SOUTH);


        test1 = new JRadioButton("Test 1 (30x100)");
        test1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                testDosyasiOgrenilen = AdvancedPTapache.OTUZLUKogrenilen;
                testDosyasiTestEdilecek = AdvancedPTapache.OTUZLUKtestedilecek;

                if(saldiri1.isSelected())
                    saldiriOrnegi = AdvancedPTapache.SALDIRI11;
```

(EK-1'in devamı)

```
            if(saldiri2.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI12;
            if(saldiri3.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI13;
            if(saldiri4.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI14;

            vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
         }
      });
      test2 = new JRadioButton("Test 2 (100x1000)");
      test2.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            testDosyasiOgrenilen = AdvancedPTapache.YUZLUKogrenilen;
            testDosyasiTestEdilecek = AdvancedPTapache.YUZLUKtestedilecek;

            if(saldiri1.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI21;
            if(saldiri2.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI22;
            if(saldiri3.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI23;
            if(saldiri4.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI24;

            vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
         }
      });
      /*test3 = new JRadioButton("Test 3 (1000x2000)");
      test3.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            testDosyasiOgrenilen = AdvancedPTapache.BINLIKogrenilen;
            testDosyasiTestEdilecek = AdvancedPTapache.BINLIKtestedilecek;
         }
      });*/
      ButtonGroup bG1 = new ButtonGroup();
      bG1.add(test1);
      bG1.add(test2);
      //bG1.add(test3);
      ButtonGroup bG2 = new ButtonGroup();
      saldiri1 = new JRadioButton("Saldırı 1");
      saldiri1.setToolTipText("Tek bir bilgisayardan 100 GB verinin tek bir hedef adrese
(kullanıcılar tarafından hiç ziyaret edilmemiş) sızdırılması");
      saldiri1.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
```

(EK-1'in devamı)

```
            if(test1.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI11;
            if(test2.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI21;
            /*if(test3.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI31;*/

            vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
         }
      });
      saldiri2 = new JRadioButton("Saldırı 2");
      saldiri2.setToolTipText("Tek bir bilgisayardan 100 GB verinin tek bir hedef adrese
(kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması");
      saldiri2.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            if(test1.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI12;
            if(test2.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI22;
            /*if(test3.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI32;*/

            vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
         }
      });
      saldiri3 = new JRadioButton("Saldırı 3");
      saldiri3.setToolTipText("Tek bir bilgisayardan 100 GB verinin üç adet farklı hedef adrese
(kullanıcılar tarafından hiç ziyaret edilmemiş) sızdırılması");
      saldiri3.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            if(test1.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI13;
            if(test2.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI23;
            /*if(test3.isSelected())
               saldiriOrnegi = AdvancedPTapache.SALDIRI33;*/

            vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
         }
      });
      saldiri4 = new JRadioButton("Saldırı 4");
      saldiri4.setToolTipText("Tek bir bilgisayardan 100 GB verinin üç adet farklı hedef adrese
(kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması");
      saldiri4.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
```

(EK-1'in devamı)

```
            if(test1.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI14;
              if(test2.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI24;
              /*if(test3.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI34;*/

              vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
            }
        });
        //saldiri5 = new JRadioButton("Saldırı 5");
        //saldiri5.setToolTipText("Üç bilgisayardan 100 GB verinin tek bir hedef adrese
(kullanıcılar tarafından hiç ziyaret edilmemiş) sızdırılması");
        /*saldiri5.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
              if(test1.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI15;
              if(test2.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI25;
              if(test3.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI35;

              vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
            }
        });*/
        //saldiri6 = new JRadioButton("Saldırı 6");
        //saldiri6.setToolTipText("Üç bilgisayardan 100 GB verinin tek bir hedef adrese
(kullanıcılar tarafından daha önce ziyaret edilmiş) sızdırılması");
        /*saldiri6.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
              if(test1.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI16;
              if(test2.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI26;
              if(test3.isSelected())
                saldiriOrnegi = AdvancedPTapache.SALDIRI36;

              vectorTextArea.setText("Karşılaştırılan Vektör: \n" + Arrays.toString(saldiriOrnegi));
            }
        });*/

        bG2.add(saldiri1);bG2.add(saldiri2);bG2.add(saldiri3);
        bG2.add(saldiri4);//bG2.add(saldiri5);bG2.add(saldiri6);
        JButton simBaslat = new JButton("Simülasyonu Başlat");
        simBaslat.addActionListener(new ActionListener()
        {
```

(EK-1'in devamı)

```
        @Override
        public void actionPerformed(ActionEvent e)
        {
           sonuclarTextArea.setText("Sonuçlar: \n");//sonuclar alanını temizlemek icin
           try {
              simulasyonu_Baslat();
           } catch (FileNotFoundException ex) {
              Logger.getLogger(AdvancedPTapache.class.getName()).log(Level.SEVERE, null,
ex);
           }
        }
      });

      ButtonGroup bG3 = new ButtonGroup();
      JRadioButton sizdirilanVeri100MB = new JRadioButton("100 MB");
      JRadioButton sizdirilanVeri1GB = new JRadioButton("1 GB");
      JRadioButton sizdirilanVeri10GB = new JRadioButton("10 GB");
      JRadioButton sizdirilanVeri100GB = new JRadioButton("100 GB");
      bG3.add(sizdirilanVeri100MB);bG3.add(sizdirilanVeri1GB);
      bG3.add(sizdirilanVeri10GB);bG3.add(sizdirilanVeri100GB);

      testPanel.add(test1);
      testPanel.add(test2);
      //testPanel.add(test3);
      saldiriPanel.add(saldiri1);
      saldiriPanel.add(saldiri2);
      saldiriPanel.add(saldiri3);
      saldiriPanel.add(saldiri4);
      //saldiriPanel.add(saldiri5);
      //saldiriPanel.add(saldiri6);
      saldiriPanel.add(simBaslat);
      sizdirilanVeriPanel.add(sizdirilanVeri100MB);
      sizdirilanVeriPanel.add(sizdirilanVeri1GB);
      sizdirilanVeriPanel.add(sizdirilanVeri10GB);
      sizdirilanVeriPanel.add(sizdirilanVeri100GB);
      test1.setSelected(true);
      saldiri1.setSelected(true);
      sizdirilanVeri100MB.setSelected(true);
      //center Panel
      JPanel centerPanel = new JPanel(new GridLayout(2, 1));
      //JLabel vectorLabel = new JLabel("Karşılaştırılan Vektör");
      //centerPanel.add(vectorLabel, BorderLayout.NORTH);
      vectorTextArea = new JTextArea();
      vectorTextArea.setText("Karşılaştırılan Vektör: \n" +
Arrays.toString(AdvancedPTapache.SALDIRI11));
      vectorTextArea.setWrapStyleWord(true);
      JScrollPane vectorScroll = new JScrollPane(vectorTextArea);
```

(EK-1'in devamı)

```
vectorScroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
        centerPanel.add(vectorScroll);
        //JLabel sonucLabel = new JLabel("Sonuçlar");
        //centerPanel.add(sonucLabel, BorderLayout.SOUTH);
        sonuclarTextArea.setText("Sonuçlar: \n");
        JScrollPane sonuclarScroll = new JScrollPane(sonuclarTextArea);

sonuclarScroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS
);
        centerPanel.add(sonuclarScroll);


        this.setSize(600,700);
        this.setLayout(new BorderLayout());
        this.add(northPanel, BorderLayout.NORTH);
        this.add(centerPanel, BorderLayout.CENTER);
        this.setVisible(true);
    }

    /**
     * @param args the command line arguments
     * @throws java.io.FileNotFoundException
     */
    public static void main(String[] args) throws FileNotFoundException {

        AdvancedPTapache ekran = new AdvancedPTapache();
        ekran.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //System.exit(0);
    }

    public void simulasyonu_Baslat() throws FileNotFoundException {

        Scanner input;
        input = new Scanner(new File(testDosyasiOgrenilen));

        int rows = 0;
        int columns = 0;
        while (input.hasNextLine()) {
            ++rows;
            Scanner colReader = new Scanner(input.nextLine());
            columns = 0;
            while (colReader.hasNextInt()) {
                ++columns;
                colReader.nextInt();
            }
        }
        double[][] a, aTestEdilecek;
```

(EK-1'in devamı)

```
        a = new double[rows][columns];
        aTestEdilecek = new double[rows][columns];

        input.close();

        // read in the data
        System.out.println("Text dosyası   :");
        input = new Scanner(new File(testDosyasiOgrenilen));
        for (int i = 0; i < rows; ++i) {
           for (int j = 0; j < columns; ++j) {
              if (input.hasNextInt()) {
                 a[i][j] = input.nextInt();
                 System.out.print(" " + a[i][j]);
              }
           }
           System.out.println();
        }
        input.close();
        System.out.println("rows: " + rows);
        System.out.println("columns: " + columns);

        // read in the data
        System.out.println("Test edilecek text dosyası   :");
        input = new Scanner(new File(testDosyasiTestEdilecek));
        for (int i = 0; i < rows; ++i) {
           for (int j = 0; j < columns; ++j) {
              if (input.hasNextInt()) {
                 aTestEdilecek[i][j] = input.nextInt();
                 System.out.print(" " + aTestEdilecek[i][j]);
              }
           }
           System.out.println();
        }
        input.close();
        System.out.println("rows: " + rows);
        System.out.println("columns: " + columns);

        double[][] b = tfidfCalculate(a, rows, columns);
        System.out.println("Tf-idf uygulanan matris   :");
        printMatrix(b, rows);

        RealMatrix mA = new Array2DRowRealMatrix(b);

        SingularValueDecomposition sVD = new SingularValueDecomposition(mA); //A = U S
V^T
        RealMatrix u = sVD.getU();
        int rank = sVD.getS().getRowDimension();
        System.out.println("Rank   :" + rank);
```

(EK-1'in devamı)

```
        //System.out.println("Singular values   :" + Arrays.toString(sVD.getSingularValues()));
        //rank = rank /3;
        RealMatrix U = sVD.getU().getSubMatrix(0, rows-1, 0, rank-1);
        RealMatrix S = sVD.getS().getSubMatrix(0, rank-1, 0, rank-1);
        RealMatrix V = sVD.getV().transpose().getSubMatrix(0, rank-1, 0, columns-1);

        RealMatrix x = U.multiply(S).multiply(V);
        printMatrix(x.getData(), rows);
        //System.out.println("u.rows: " + u.getRowDimension());
        //System.out.println("u.columns: " + u.getColumnDimension());
        //System.out.println("U matrisi   :");
        //printMatrix(u.getArray(), rows);

        System.out.println("Cosine similarity   :");
        double[] x2 = saldiriOrnegi;
        cosineSimilarityMatrix(x.getData(), rows, aTestEdilecek);

        cosineSimilarity(x.getData(), rows, x2);
    }

    private static void printMatrix(double m[][], int rows) {

        for (int i = 0; i < rows; ++i) {
            for (int j = 0; j < m[i].length; ++j) {
                System.out.print(" " + m[i][j]);
            }
            System.out.println();
        }
    }

    private static double[][] tfidfCalculate(double m[][], int rows, int columns) {

        // idf calculation basladi.
        double[] idf = new double[columns];
        int n = 0;
        System.out.println(" idf degerleri     :");
        for (int j = 0; j < columns; j++) {
            for (int i = 0; i < rows; i++) {
                double tf = m[i][j];
                if ( tf != 0)
                    n++;
            }
            if (n != 0) {
                idf[j] = Math.log10((double)rows/n);
                System.out.print(" " + idf[j]);
            }
            n=0;
        }
```

(EK-1'in devamı)

```
        System.out.println("");
        // idf calculation bitti


        // tf calculation basladi
        double[] tf = new double[rows];
        double fr = 0;
        System.out.println(" tf degerleri     :");
        for (int i = 0; i < rows; i++) {
            fr = 0;
            for (int j = 0; j < columns; j++) {
                fr += m[i][j];
            }
            tf[i] = fr;
            System.out.print(" " + tf[i]);
        }
        System.out.println("");
        // tf calculation bitti.

        // tf-idf calculation basladi.
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                m[i][j] = m[i][j]/tf[i] * idf[j];
            }
        }
        // tf-idf calculation bitti.
        return m;
    }

    public void cosineSimilarity(double m[][], int rows, double docVector2[]) {
        double[] docVector1;
        double dotProduct, magnitude1, magnitude2, cosineSimilarity;
        double maxSimilarity = 0;
        double ortalamaSimilarity = 0;

        for (int k=0; k<rows; k++) {
            dotProduct = 0.0;
            magnitude1 = 0.0;
            magnitude2 = 0.0;
            cosineSimilarity = 0.0;
            docVector1 = m[k];

            for (int i = 0; i < docVector2.length; i++) //docVector1 and s must be of same length
            {
                double a1 = 0.0;
                if (i<docVector1.length)
                    a1 = docVector1[i];
                dotProduct += a1 * docVector2[i];  //a.b
```

(EK-1'in devamı)

```
            magnitude1 += Math.pow(a1, 2);  //(a^2)
            magnitude2 += Math.pow(docVector2[i], 2); //(b^2)
         }

         magnitude1 = Math.sqrt(magnitude1);//sqrt(a^2)
         magnitude2 = Math.sqrt(magnitude2);//sqrt(b^2)

         if (magnitude1 != 0.0 | magnitude2 != 0.0) {
            cosineSimilarity = dotProduct / (magnitude1 * magnitude2);
            ortalamaSimilarity += cosineSimilarity;
            if(cosineSimilarity > maxSimilarity)//maxsimilarity bulmak icin
               maxSimilarity = cosineSimilarity;
         } else {
            System.out.println("row : " + k + " cosine similarity : " + 0);
            //return 0.0;
         }
         System.out.println("row : " + k  + " cosine similarity : " +
               String.format("%1$,.100f", cosineSimilarity));
      }
      ortalamaSimilarity = ortalamaSimilarity / rows;

      sonuclarTextArea.append("Maksimum Benzerlik: " + maxSimilarity + "\n" //+
            /*"Ortalama Benzerlik: " + ortalamaSimilarity*/);
      //return cosineSimilarity;
   }

   public void cosineSimilarityMatrix(double m[][], int rows, double mTestEdilecek[][]) {

      double maximumlarinMinimumBenzerligi = 1;
      double maxSimilarity = 0;

      for(int z=0; z<rows; z++){
         if (maximumlarinMinimumBenzerligi > maxSimilarity && maxSimilarity != 0)
            maximumlarinMinimumBenzerligi = maxSimilarity;
         double[] docVector2 = mTestEdilecek[z];
         double[] docVector1;
         double dotProduct, magnitude1, magnitude2, cosineSimilarity;
         maxSimilarity = 0;
         double ortalamaSimilarity = 0;

         for (int k=0; k<rows; k++) {
            dotProduct = 0.0;
            magnitude1 = 0.0;
            magnitude2 = 0.0;
            cosineSimilarity = 0.0;
            docVector1 = m[k];

            for (int i = 0; i < docVector2.length; i++) //docVector1 and s must be of same length
```

(EK-1'in devamı)

```
            {
               double a1 = 0.0;
               if (i<docVector1.length)
                  a1 = docVector1[i];
               dotProduct += a1 * docVector2[i];  //a.b
               magnitude1 += Math.pow(a1, 2);  //(a^2)
               magnitude2 += Math.pow(docVector2[i], 2); //(b^2)
            }

            magnitude1 = Math.sqrt(magnitude1);//sqrt(a^2)
            magnitude2 = Math.sqrt(magnitude2);//sqrt(b^2)

            if (magnitude1 != 0.0 | magnitude2 != 0.0) {
               cosineSimilarity = dotProduct / (magnitude1 * magnitude2);
               ortalamaSimilarity += cosineSimilarity;
               if(cosineSimilarity > maxSimilarity)//maxsimilarity bulmak icin
                  maxSimilarity = cosineSimilarity;
            } else {
               System.out.println("row : " + k + " cosine similarity : " + 0);
               //return 0.0;
            }
            System.out.println("row : " + k  + " cosine similarity : " +
                  String.format("%1$,.100f", cosineSimilarity));
         }
         ortalamaSimilarity = ortalamaSimilarity / rows;

         sonuclarTextArea.append("Maksimum Benzerlik: " + maxSimilarity + "\n" //+
               /*"Ortalama Benzerlik: " + ortalamaSimilarity*/);
      }
      if (maximumlarinMinimumBenzerligi > maxSimilarity && maxSimilarity != 0)
            maximumlarinMinimumBenzerligi = maxSimilarity;
      sonuclarTextArea.append("Sonuçlar: \nMinimum Benzerlik: " +
maximumlarinMinimumBenzerligi + "\n" );
      //return cosineSimilarity;
   }

   //internal cosine similarity
   public static void cosineSimilarity2(double m[][], int rows) {
      double[] docVector1;
      double[] docVector2;
      double dotProduct = 0.0;
      double magnitude1 = 0.0;
      double magnitude2 = 0.0;
      double cosineSimilarity = 0.0;

      for (int k=0; k<rows; k++) {
         dotProduct = 0.0;
         magnitude1 = 0.0;
```

(EK-1'in devamı)

```
        magnitude2 = 0.0;
        cosineSimilarity = 0.0;
        docVector1 = m[k];
        for (int j = k; j<rows; j++){
           if (j+1 < rows) {
              docVector2 = m[j+1];
              for (int i = 0; i < docVector2.length; i++) //docVector1 and docVector2 must be of
same length
              {
                 dotProduct += docVector1[i] * docVector2[i];  //a.b
                 magnitude1 += Math.pow(docVector1[i], 2);  //(a^2)
                 magnitude2 += Math.pow(docVector2[i], 2); //(b^2)
              }

              magnitude1 = Math.sqrt(magnitude1);//sqrt(a^2)
              magnitude2 = Math.sqrt(magnitude2);//sqrt(b^2)

              if (magnitude1 != 0.0 | magnitude2 != 0.0) {
                 cosineSimilarity = dotProduct / (magnitude1 * magnitude2);
              } else {
                 System.out.println("row : " + k + " ve " + (j+1) + " cosine similarity : " + 0);
                 //return 0.0;
              }
              System.out.println("row : " + k + " ve " + (j+1) + " cosine similarity : " +
                    String.format("%1$,.100f", cosineSimilarity));
           }
        }
     }
     //return cosineSimilarity;
  }

  /*public static double euclideanSimilarity(double m[][], int rows)
  {
     double Sum = 0.0;
     for(int i=0;i<array1.length;i++) {
       Sum = Sum + Math.pow((array1[i]-array2[i]),2.0);
     }
     return Math.sqrt(Sum);
  }*/
}
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BAYRAK, Mehmet Emin
Uyruğu : T.C.
Doğum tarihi ve yeri : 28.01.1980, Ödemiş
Medeni hali : Evli
Telefon : 0 (312) 456 25 55
e-mail : mehmet.bayrak@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üni. /Fen Bil.Ens.Bilg.Müh.A.B.D.	2015
Lisans	Marmara Üni./Müh.Fak.Bilg.Müh.	2003
Lise	Ödemiş Süper Lisesi	1998

İş Deneyimi

Yıl	Yer	Görev
2003-2005	Havelsan A.Ş.	Yazılım Mühendisi
2005-Halen	Jandarma Genel Komutanlığı	Subay

Yabancı Dil

İngilizce

Yayınlar

-

Hobiler

Linux, masa tenisi, seyahat.

GAZİ GELECEKTİR...