

**KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ  
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**



**YAPAY ZEKA DERSİ VİZE RAPORU**

Sistem LOG dosyalarını inceleyerek, sistemde aktif ya da öncesinde  
var olmuş yetkisiz girişlerin tespit edilmesi

Barış AZAR

2118121004

Anahtar kelimeler: LOG Analizi, LSTM, Yapay Zeka, Access.log,  
LOG, Terminal.

**Abstract**

Bu çalışmada, yetkisiz girişleri tespit etmek amacıyla yapay zeka teknolojilerinden yararlanılarak bir log analiz sistemi geliştirilmiştir. Çalışmanın temel amacı, hem içeriden hem de dışarıdan yapılan yetkisiz erişim girişimlerini, sunucu erişim logları ve terminal komutları üzerinden analiz ederek belirlemektir. Farklı senaryolar ve yapay zeka modelleriyle, bu verilerin yapay zeka modelleriyle işlenmesi sonucu yetkisiz erişim girişimlerinin tespit edilip edilemeyeceği araştırılmıştır. Sonuç olarak, bu çalışma, log dosyalarının yapay zeka yardımıyla analiz edilerek sistem güvenliğinin nasıl artırılacağına dair önemli bilgiler sunmaktadır.

# 1 Giriş

Bu raporda vize gününe kadar yapılan çalışmalar ele alınmıştır. Raporda çalışmada yapılan adımlar karşılaşılan sorunlar ve çözümler, veri seti, Yapay Zeka modelleri yer almaktadır.

Güvenlik, günümüzde dijital sistemler için kritik bir öneme sahiptir. Yapılan araştırmaya göre, 2023 yılında küresel çapta 2.365 siber saldırı yaşanmış ve bu saldırılar sonucunda 343.338.964 kişi etkilenmiştir. Saldırlardan kaynaklanan veri ihlallerinin ortalama maliyeti 4.45 milyon USD olmuştur. Saldırıların etkisi ve maliyeti düşünülünce siber güvenlik alanında yatırımlar ve veri sızmasını engellemek için çalışmaların olması hayattır.[1]

Çalışmanın amacı yüksek güvenli bir sunucu ve kritik sistemleri ve o sunuculara içeriden erişebilen çalışan yönetici vb. kişilerin şahsi ve kurumsal bilgisayarlarında keşif yapmayı amaçlamaktadır. Çalışmanın önemi araştırmacının Yapay Zeka, log analizi konusunda tecrübe sağlaması ve bu tecrübelerin temel aldığı farklı çalışmalar yapmasına katkı sağlamaktır. Literatür açısından Yapay zeka ile log analizi konusunda farklı Yapay Zeka modellerini karşılaştırılması önemli bir katkıdır. Çalışmada, sistem güvenliğini artırmak amacıyla yapay zeka kullanarak sistemde aktif olarak işlem yapan ya da geçmişte işlem yapmış izinsiz girişleri tespit etmek amacıyla geliştirilmektedir. Sistem erişim (access) log dosyaları ve terminal üzerinden çalıştırılan komutların analiz edilerek çıktı üretilmesi üzerine çalışılmaktadır.

## 2 Veri Seti

Çalışma veri seti 2 parçadır. İlk parça access.log dosyasının analizi için, ikinci parça terminal üzerinden çalıştırılan komutların tespiti için oluşturulmuştur. İlk parça veri setinin detayları şöyledir;

Veri seti, "Kirli" ve "Temiz" olarak 2 adet kategori mevcuttur. Kirli veri seti çalışma için özel olarak elde edilmiştir. Kirli veri setini hazırlamak için yerel ağ üzerinden apache sunucu sistemi üzerine saldırı gerçekleştirilmiştir. Saldırı türü wordlist'dir. Wordlist olarak [2] kullanılmıştır. Saldırı sonucunda oluşan access.log dosyası kirli verileri oluşturmaktadır. Karşıt olarak temiz verisi için [3] verileri kullanılmıştır. Çalışmada farklı yapay zeka modellerinin çıktıları inceleneceği için senaryolar oluşturulmuştur. Senaryo hakkında detaylar şöyledir;

Senaryo 1, veri setinde eşit ve az miktarda veriyi.

Senaryo 2, veri setinde eşit ve büyük miktarda veriyi.

Senaryo 3, veri setinde toplam miktarın 1/4'ü kadar kirli 3/4'ü kadar temiz veriyi. Senaryo 4, veri setinde toplam miktarın 1/4'ü kadar temiz 3/4'ü kadar kirli veriyi.

Table 1: Senaryo veri durumu

(a) Senaryo 1

Veri Tipi	Veri Adedi	Yüzde
Kirli	250 Bin	% 50
Temiz	250 Bin	% 50

(b) Senaryo 2

Veri Tipi	Veri Adedi	Yüzde
Kirli	Bir milyon	% 50
Temiz	Bir milyon	% 50

(c) Senaryo 3

Veri Tipi	Veri adedi	Yüzde
Kirli	250 Bin	% 25
Temiz	Bir milyon	% 75

(d) Senaryo 4

Veri Tipi	Veri Adedi	Yüzde
Kirli	Bir milyon	% 75
Temiz	250 Bin	% 25

Terminal üzerinde çalıştırılan komutların analizi için veri seti içerisinde normal kullanıcının çok sık uğramayacağı dizinler, yetki yükseltme saldırıları için komutlar gibi hacker'lar tarafından kullanılan komutlar yer almaktadır. Yetki yükseltme komutları [4] alınmıştır. Veri seti içerisinde normal kullanıcıların çalıştırabileceği komutlar da mevcuttur. Komutların tamamına 0-10 arasında (0 ve 10 dahil değil) 1 normal 9 şüpheli sıklasında puan verilmiştir. Veri setinde bulunan bazı örnek komutlar ve puan karşılıkları şöyledir;

Table 2: Komut puan tablosu

Komut	Puan
pwd	5
ssh	9
setxkmap	6
telnet	9
ls	1
cd	1
rm	2
rm -rf	4
sudo	1
sudo su	1
cd /var/log	9
cd /var/log/	9
search	3
sudo install -m =xs \$(which python) . ./python -c 'import os; os.execl("/bin/sh", "sh", "-p")' python -c 'open("file_to_write","w+").write("DATA")'	9
SUID	9
GUID	9
chmod -x	5
msfconsole	9
whoami	9
rmdir	1
mkdir	1
operators	1
grep	1
help	1
netcat	6
tcpdump	6
Chroot	6
find	4
sudo install -m =xs \$(which mv) . LFILE=file_to_write TF=\$(mktemp) echo "DATA" > \$TF ./mv \$TF \$LFILE sudo install -m =xs \$(which chmod) . LFILE=file_to_change	9

## 3 Yöntem

### 3.1 Hazırlanan scriptler

Veri setinin düzenlenmesi için Python dilinde scriptler yazılmıştır. Scriptler hakkında detaylar şöyledir;

#### 3.1.1 Format değiştirmek ve satır düzenlemek amacıyla hazırlanan script

Yazılan script ihtiyaç duyulmayan sütunları, sütun içerisindeki verileri siler, veri formatını txt veri formatından xlsx veri formatına çevirir ve xlsx sütunlarına belirlenen sütun adlarını yazar.

```
1 import os
2 import pandas as pd
3
4 def process_file(input_file, output_file):
5     df = pd.read_csv(input_file, sep='\\s+', header=None, names=["IP","X","Y","Z","Url and parametre","HTTP CODE","Length","-","Agent","q"], on_bad_lines='skip')
6     df = df.drop(columns=["IP", "X", "Y", "Z", "q"])
7
8     # "HTTP/1.1" kısmını "Url and parametre" sütunundan kaldır
9     df["Url and parametre"] = df["Url and parametre"].str.replace(" HTTP/1.1", "", regex=False)
10
11     df.to_csv(output_file, index=False)
12
13 # İşlem yapılacak dizin
14 directory = r"C:\Users\user\Desktop\Yeni klasör"
15
16 # Dizin içindeki dosyaları sırayla işle
17 for filename in os.listdir(directory):
18     if filename.endswith(".txt"): # Sadece txt dosyalarını işle
19         input_file = os.path.join(directory, filename)
20         output_file = os.path.join(directory, "deneme_" + os.path.splitext(filename)[0] + ".xlsx") # Temizlenmiş dosya adı ve uzantısı
21         process_file(input_file, output_file)
22         print("Dosya işlendi:", filename, "=>", output_file)
```

Şekil 1: Senaryo 1 çıktı görüntüsü

#### 3.1.2 Verileri bölmek için script

Access.log veri boyutu 12 milyon satır sayısı içermesi ve verilerin tamamının kullanılamayacak olması sebebiyle bölünmesi gerekiyordu. Bu yüzden veri 1 milyon satır olacak şekilde 12 dosyaya bölündü.

[illegible]

### 3.2 Yapay Zeka modeli

### 3.2.1 LSTM

### 3.2.2 Ölçüm Metriği

### 3.3 Karşılaşılan hata ve sorunlar

### 3.3.1 LSTM çıktı hakkında

6

```

1 def dosyayı_böl(dosya_yolu, çıktı_dizin, satır_sayısı=1000000):
2     with open(dosya_yolu, 'r') as dosya:
3         dosya_adı = 1
4         satır = dosya.readline()
5         while satır:
6             with open(f"{çıktı_dizin}/bölünmüş_dosya_{dosya_adı}.txt", 'w') as çıktı_dosya:
7                 for _ in range(satır_sayısı):
8                     if not satır:
9                         break
10                    çıktı_dosya.write(satır)
11                    satır = dosya.readline()
12                dosya_adı += 1
13
14 dosya_yolu = r"C:\Users\user\Desktop\new\access.log"
15
16 çıktı_dizin = r"C:\Users\user\Desktop\Yeni klasör"
17
18 dosyayı_böl(dosya_yolu, çıktı_dizin)

```

Şekil 3: Veri bölmek için kullanılan script

LSTM'nin zamana bağlı analizde iyi olması ancak eğitim verilerinde zaman damgası olmaması başarımı düşüren bir sebep olabilir.

### 3.3.2 Eğitim sırasında yaşanan sorunlar

Eğitim sırasında model kodunun 1 satırını uzun süre çalıştığını fark edildi. Hata olabileceği düşünüldü ve eğitim durdurulup tekrar başlatıldı. Aynı durumla karşılaşılnca sorunu çözmek için araştırma yapıldı. Araştırma sonucunda bu durumun verinin büyüklüğü ile alakalı bir durum olabileceğini sonucuna varıldı. Bu yüzden model Colab sistemi yerine yerel bilgisayar üzerinde tekrar denendi ve yaklaşık 2 saniye 1 satır işlendiği farkedildi. Bu durumu çözmek için veri seti azaltıldı ve tekrar denendi ve eğitim süreci sorunsuz şekilde başladı.

## 4 Bulgu ve Tartışma

### 4.1 Kullanılan donanımlar

#### 4.1.1 Colab

Tesla T4 GPU[7]

#### 4.1.2 Yerel Bilgisayar

GTX 1650 GPU

16 GB RAM

Intel Core i5-11300H CPU

## 5 Sonuç

Çalışma henüz devam etmektedir. Çalışma bitirilince ilerleyen tarihte eklenecektir.

### 5.1 İleride yapılması planlanan çalışmalar

SIEM ürün entegrasyonu: SIEM ürünleri ile entegre bir şekilde "alert" üretilip SOC analistine bildirim gönderilmesi. LLM sistemleri kullanılarak alert'in detaylandırılması sağlanabilir.

Farklı log türleri ve işletim sistemleri için çalışmanın gerçekleştirilmesi: Çalışma Linux sistemler üzerine odaklanmaktadır. Gelecekte farklı işletim sistemleri (Windows, macOS vb.) üzerine çalışma gerçekleştirilmesi

## Kaynakça

- [1] M. S. John, "Cybersecurity stats: Facts and figures you should know." <https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/>. Erişim Tarihi: 21 Nisan 2024.
- [2] HNK7, "wordlists." <https://github.com/v0re/dirb/blob/master/wordlists/big.txt>. Erişim Tarihi: 28 Mart 2024.
- [3] E. Dabbas, "Web server access logs." <https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs>. Erişim Tarihi: 15 Nisan 2024.
- [4] Epinna and A. Cardaci, "Gtfobins." <https://gtfobins.github.io>. Erişim Tarihi: 15 Nisan 2024.
- [5] E. Güngör, "Emre güngör." <https://avesis.ksbu.edu.tr/emre.gungor>. Erişim Tarihi: 22 Nisan 2024.
- [6] "Google colab." <https://colab.google>. Erişim Tarihi: 22 Nisan 2024.
- [7] NVIDIA, "Nvidia turing gpu architecture." <https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf>. Erişim Tarihi: 25 Nisan 2024.