Logo

Description automatically generated

Capstone Project – Intrusion Detection using AI

Craig Perkins – [perkins.cr@northeastern.edu](mailto:perkins.cr@northeastern.edu)

Northeastern University

January 2023 – EAI6980

Capstone Project – Intrusion Detection using AI

For the capstone project, I will be using artificial intelligence to develop an intrusion detection system. Over the last few decades, more and more companies have been relying on technology to host mission critical applications that keep industry running. In the last decade, cloud computing has emerged as an enormous field where small companies can leverage the same infrastructure as major technology companies like Microsoft, Google and Amazon. In addition to cloud services, many large companies also host applications on-premise in a data center that is managed by the same company or co-located in a data center. When securing a physical location like a data center, companies can install cameras to monitor for unauthorized entry to restricted locations to detect intrusions. It’s also important to monitor the digital infrastructure inside the data center which is connected to the internet and accessible by the outside world. There are many methods for securing servers like protecting the server with username and password, restricting traffic to certain ports and adding a firewall to block traffic. Even with all of the security measures in place, hackers can still find methods to exploit to remotely takeover machines. There have been prominent examples of zero-day vulnerabilities that allow attackers to remotely execute code on a vulnerable application server. Examples include heartbleed and log4shell.

In this project, I seek to find a solution for automated monitoring for intrusion detection to determine anomalous behavior within a cluster that should trigger a security alert. I will be examining network traffic to determine if and when a machine was compromised.

**Dataset**

For this project I will be examining raw network packets from the UNSW-NB 15 (University of New South Wales) dataset. The dataset can be located here: <https://research.unsw.edu.au/projects/unsw-nb15-dataset>. The dataset has nine types of attacks:

1. Fuzzers
2. Analysis
3. Backdoors
4. DoS
5. Exploits
6. Generic
7. Reconnaissance
8. Shellcode
9. Worms

All data and python notebooks for this project can be found in a public repository for my Github account here: <https://github.com/cwperks/EAI6980>. Below is a data dictionary of data in the dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | srcip | nominal | Source IP address |
| 2 | sport | integer | Source port number |
| 3 | dstip | nominal | Destination IP address |
| 4 | dsport | integer | Destination port number |
| 5 | proto | nominal | Transaction protocol |
| 6 | state | nominal | Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state) |
| 7 | dur | Float | Record total duration |
| 8 | sbytes | Integer | Source to destination transaction bytes |
| 9 | dbytes | Integer | Destination to source transaction bytes |
| 10 | sttl | Integer | Source to destination time to live value |
| 11 | dttl | Integer | Destination to source time to live value |
| 12 | sloss | Integer | Source packets retransmitted or dropped |
| 13 | dloss | Integer | Destination packets retransmitted or dropped |
| 14 | service | nominal | http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service |
| 15 | Sload | Float | Source bits per second |
| 16 | Dload | Float | Destination bits per second |
| 17 | Spkts | integer | Source to destination packet count |
| 18 | Dpkts | integer | Destination to source packet count |
| 19 | swin | integer | Source TCP window advertisement value |
| 20 | dwin | integer | Destination TCP window advertisement value |
| 21 | stcpb | integer | Source TCP base sequence number |
| 22 | dtcpb | integer | Destination TCP base sequence number |
| 23 | smeansz | integer | Mean of the ?ow packet size transmitted by the src |
| 24 | dmeansz | integer | Mean of the ?ow packet size transmitted by the dst |
| 25 | trans\_depth | integer | Represents the pipelined depth into the connection of http request/response transaction |
| 26 | res\_bdy\_len | integer | Actual uncompressed content size of the data transferred from the server’s http service. |
| 27 | Sjit | Float | Source jitter (mSec) |
| 28 | Djit | Float | Destination jitter (mSec) |
| 29 | Stime | Timestamp | record start time |
| 30 | Ltime | Timestamp | record last time |
| 31 | Sintpkt | Float | Source interpacket arrival time (mSec) |
| 32 | Dintpkt | Float | Destination interpacket arrival time (mSec) |
| 33 | tcprtt | Float | TCP connection setup round-trip time, the sum of ’synack’ and ’ackdat’. |
| 34 | synack | Float | TCP connection setup time, the time between the SYN and the SYN\_ACK packets. |
| 35 | ackdat | Float | TCP connection setup time, the time between the SYN\_ACK and the ACK packets. |
| 36 | is\_sm\_ips\_ports | Binary | If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0 |
| 37 | ct\_state\_ttl | Integer | No. for each state (6) according to specific range of values for source/destination time to live (10) (11). |
| 38 | ct\_flw\_http\_mthd | Integer | No. of flows that has methods such as Get and Post in http service. |
| 39 | is\_ftp\_login | Binary | If the ftp session is accessed by user and password then 1 else 0. |
| 40 | ct\_ftp\_cmd | integer | No of flows that has a command in ftp session. |
| 41 | ct\_srv\_src | integer | No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26). |
| 42 | ct\_srv\_dst | integer | No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26). |
| 43 | ct\_dst\_ltm | integer | No. of connections of the same destination address (3) in 100 connections according to the last time (26). |
| 44 | ct\_src\_ ltm | integer | No. of connections of the same source address (1) in 100 connections according to the last time (26). |
| 45 | ct\_src\_dport\_ltm | integer | No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26). |
| 46 | ct\_dst\_sport\_ltm | integer | No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26). |
| 47 | ct\_dst\_src\_ltm | integer | No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26). |
| 48 | attack\_cat | nominal | The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms |
| 49 | Label | binary | 0 for normal and 1 for attack records |

The dataset is split into a train dataset with 175,341 and a test dataset with 82,332 records.

The dataset has 2 labels, 1 binary and 1 multi-classification. The binary value signifies if an entry is malicious or benign and the `attack\_cat` is the category for the attack. This analysis will build predictors for both the binary and multi-class cases.

References

1. The Heartbleed Bug - <https://heartbleed.com/>
2. Log4Shell a year on - <https://usa.kaspersky.com/blog/log4shell-still-active-2022/27531/>
3. UNSW-NB15 Dataset - <https://research.unsw.edu.au/projects/unsw-nb15-dataset>