EP2420 *Intrusion Detection - Task 1*

Federico Giarrè

December 1, 2023

1 Task 1

The task is useful in order to get familiar with the content of the datasets of both the attacker action sequences and the alert values detected.

1.1 Datasets composition

There are six possible attack actions: *Continue*, *Ping Scan*, *DVWA SQL Injection Exploit*, *Install tools*, *Network service login*, *Sambacry Explolit*.

Each action may or may not produce an alert value. The following Table 1 describes the relation between each type of attack and the alert values produced

Attack Action	Count	Mean	STD	Min	25%	50%	75%	Max
Continue	5e+03	1.8e-03	1.3e-01	0	0	0	0	1e+01
DVWA SQL Injection Exploit	1.6e+03	0	0	0	0	0	0	0
Install tools	3.1e+03	3.2e-03	1.7e-01	0	0	0	0	1e+01
Network service login	3.4e+03	0	0	0	0	0	0	0
Ping Scan	4.5e+03	7.8e+02	2.5e+02	0	5.1e+02	1e+03	1e+03	1.02e+03
Sambacry Explolit	1.9e+03	1e-02	3.1e-01	0	0	0	0	1e+01

Table 1: Distribution of alert values with respect to the type of attack action

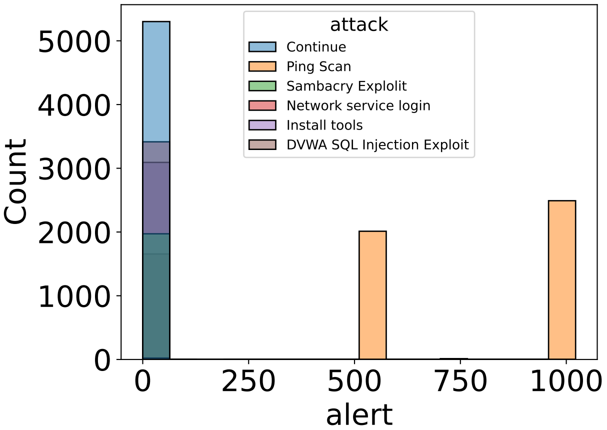

Figure 1: Distribution of alert values with respect to the attack action performed

An histogram is plotted to show the distribution of alert values with respect to the attack action performed is shown in Fig. 1. As it is possible to see, only the attack Ping Scan really impact the alert vales, with alert values for other attacks varying between 0 and 10. Notably, action such as the DVWA SQL Injection Exploit and the Network Service Login do not produce any alert value ever. In consideration of this, density plots for the remaining actions can be found in the appendix[1].

1.2 Background on HMM

Hidden Markov Models are models that allow to predict a sequence of variables starting from observations.

1.2.1 Forward-Backward algorithm

This algorithm is used to perform inference using the model. In particular we want to compute for each timestep the forward probabilities (the probability of ending up in a certain state given the first t observation) and the backward probabilities (the probability of visiting observation given any timestep t). Putting together the two sets of probabilities, what we obtain is the probability of visiting any observation at any timestap, enabling the inference process.

1.2.2 Viterbi algorithm

The Viterbi algorithm is a dynamic programming algorithm is used to determine the most probable sequence of actions that lead to a set of observation under an Hidden Markov Model.

1.2.3 Supervised Training Process

An HMM is defined by the tuple $\lambda = (A, B, \pi)$ where A is the transition matrix between the hidden states S, B is the emission matrix that relate states to the observation that can be made and π is the distribution of the initial probability of each hidden state. If the hidden states are available, then the learning can be completed by just counting the frequencies of the events of: state transition to estimate A, observations emitted to estimate B and initial probability for each state π.

1.3 Observations

Rows of the alert dataset are composed by the observation that relates to certain action performed on the system. When plotting a sample, we obtain something similar to Figure 2. One clear thing is that the value at each timestep has a baseline of 0, but can spike with respect to the action performed. In this case, the sequence of action that generated the observation is: *Continue*, *Ping Scan*, *Sambacry Exploit*, *Network service login*, *Install tools*, *Ping Scan*, *DVWA SQL Injection Exploit*, *Network service login*, *Install tools*, *Ping Scan*. As expected, the action that produced the only spikes is the Ping Scan.

[1] Due to problems with seaborn, an ensamble plot was not possible.

Action	Mapped Value
Continue	0
Ping Scan	1
Install tools	2
DVWA SQL Injection Exploit	3
Network Service Login	4
Sambacry Explolit	5

Table 2: Mapping between action and integer representation

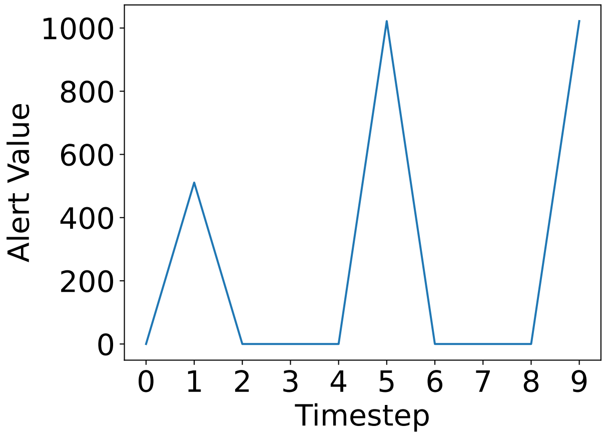

Figure 2: Alert value in a sample observation

1.4 Modelling the problem for HMM

In order to use the HMM to perform a classification task, we need to model the data we possess in states and the observations. An hidden state q is a succession of actions q_t where t is the timestep at which each action is executed. For simplicity, we map the actions to integers, so that an hidden state can become a touple of integers. More specifically, the mapping is expressed as for the Table 2

The 38 unique alert values, that range from 0 to 1022, can already be used as symbols for the observation space, hence they don't need to be remapped.

A Density plots

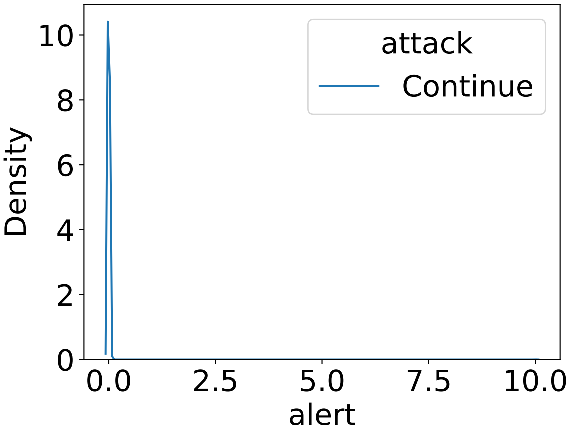

Figure 3: Density distribution of alert values for the ”Continue” action

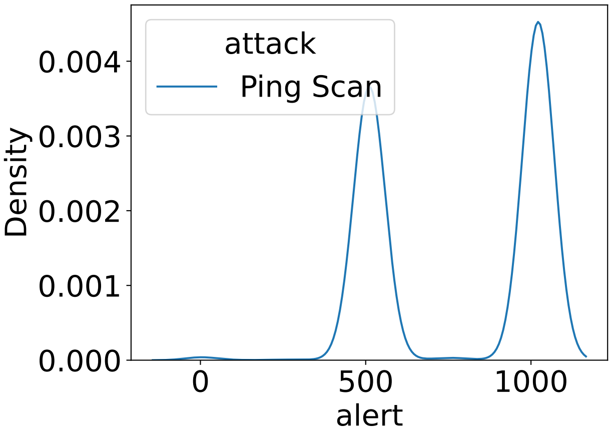

Figure 4: Density distribution of alert values for the ”Ping Scan” action

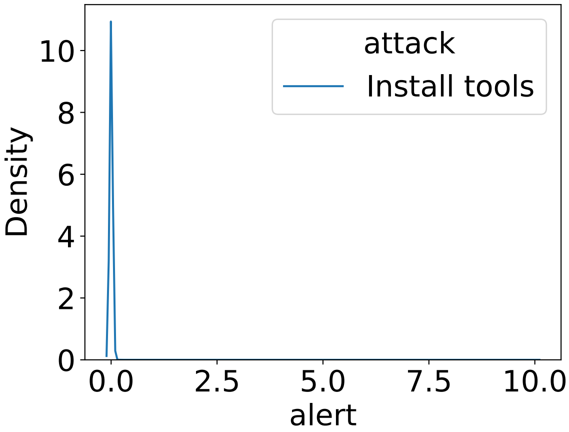

Figure 5: Density distribution of alert values for the ”Install Tools” action

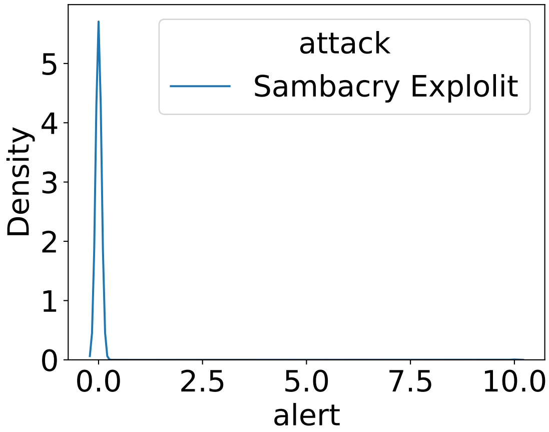

Figure 6: Density distribution of alert values for the ”Sambacry Exploit” action