NETWORK ANOMALY DETECTION USING THE MARKOV CHAIN MODEL

Pedrum Jalali
[Email address]

Abstract

The purpose of this paper is to detect network anomalies using the markov chain model. The paper aims at targeting anomalies where there is a significant change in the pattern of connections made to a particular server. The data used in this paper is from the DARPA intrusion detection evaluation program in the year 1998.

Traffic Sampling

During a 7 week period the traffic of a network system was monitored. Various controlled attacks where performed on the network at different times during the monitoring period. In this research we analyze the incoming connections to one of the servers.

1- Generating Clean Traffic Model

In the first stage of the research the clean traffic behavior was modeled using the markov chain model. The traffic was split into one minute periods. A state was defined using the properties below:

Connection Count: The number of connections that were initiated with the server during that one minute period.

Repeat: The number of previous intervals that the connection count property remained the same.

Each state has a series of actions. The actions show the probability that the state went from its current state to the next state. Therefore the markov chain model would look something like Figure 1. In order to avoid clutter in the figure below only the actions propagating from the states in the middle row have been drawn:

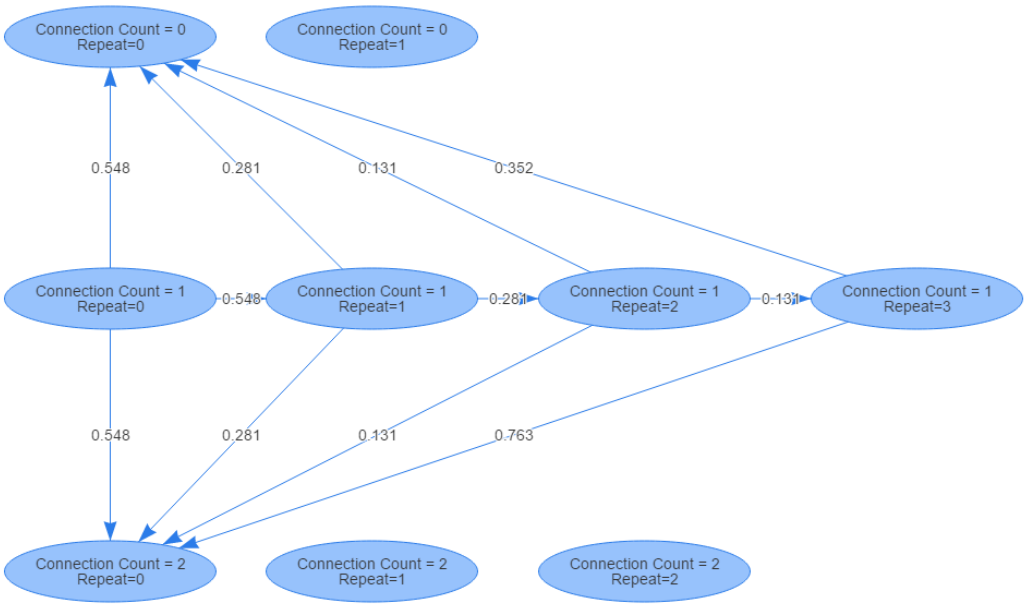


Figure 1: Markov chain model used in analysis. Some actions have been omitted for clarity

The algorithm for determining the next state is as follows:

Step 1: S = S(0, 0) and $T_i = T_0$

Step 2: $T_i = T_{i+1}$

Step 3: Find the number of connections initiated during T_i.

If $C_i = C_{i-1}$ go to step 4.

If $C_i \neq C_{i-1}$ go to step 5.

Step 4: $S_i = S(C_i, 0)$. Go to step 2.

Step 5: $R_i = R_{i-1} + 1. Go\ to\ step\ 6.$

Step 6: $S_i = S(C_i, R_i). Go\ to\ step\ 2$

Where

$\boldsymbol{S(C, R)}$: The state with connection count C and repeat R
$\boldsymbol{T_i}$: The I'th one minute time interval.
$\boldsymbol{C_i}$: The number of connections initiated with server during time interval i.
$\boldsymbol{R_i}$: The number of consecutive times this connection count has been repeated up to interval i.

2- Generating The 20 Minute Probability Distribution

Once the markov chain model for the clean traffic has been generated, the following distribution probability is calculated:

$$P_i = \prod_{j=1}^{20} A(S_j, S_{j+1})$$

Where:

P_i: $Probability\ obtained\ by\ passing\ clean\ traffic\ through\ the\ markov\ model\ during\ a\ 20\ minute\ period\ starting\ at\ minute\ i$

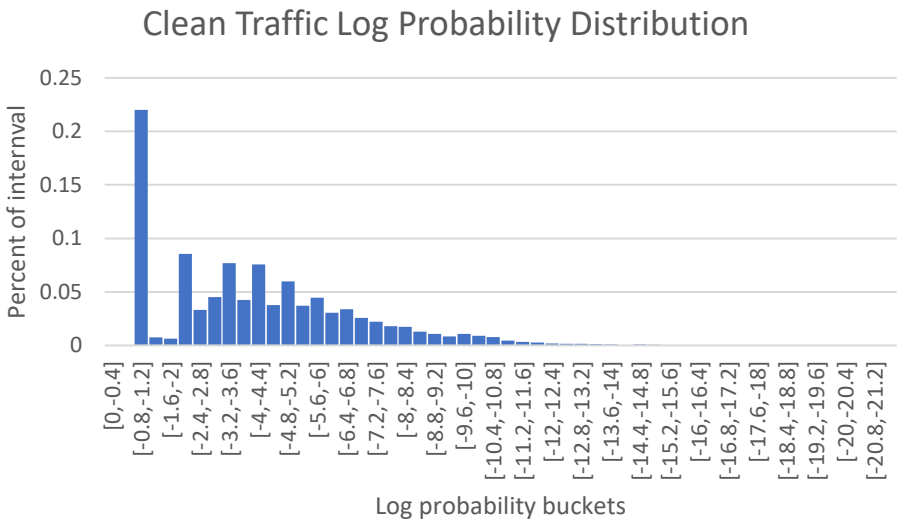
S_j: $The\ server\ state\ during\ time\ interval\ (j + i)$

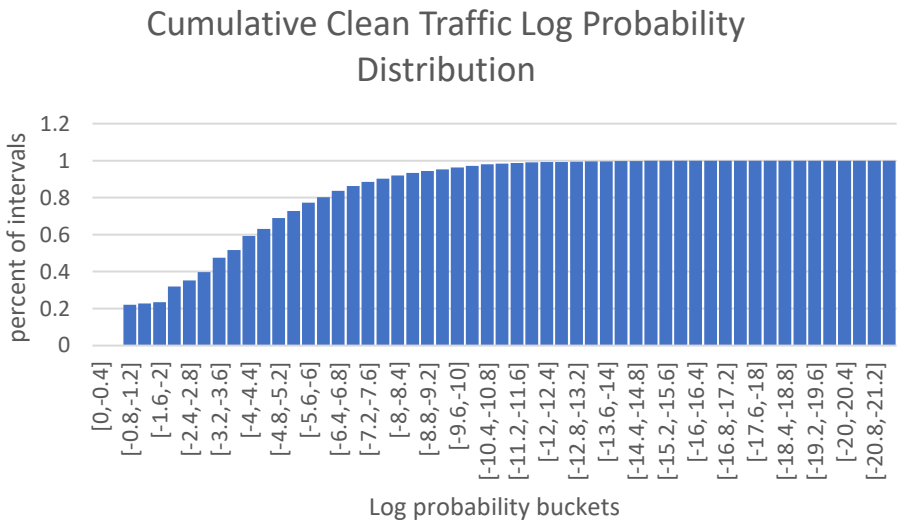
S_{j+1}: $The\ server\ state\ during\ time\ interval\ (j + 1 + i)$

$A(S_j, S_{j+1})$: $The\ action\ probablity\ for\ clean\ traffic\ to\ traverse\ from\ state\ j\ to\ j + 1$

The result of generating the probability distribution for the clean traffic are shown in the plots below. The 95% cutoff point can be obtained as follows:

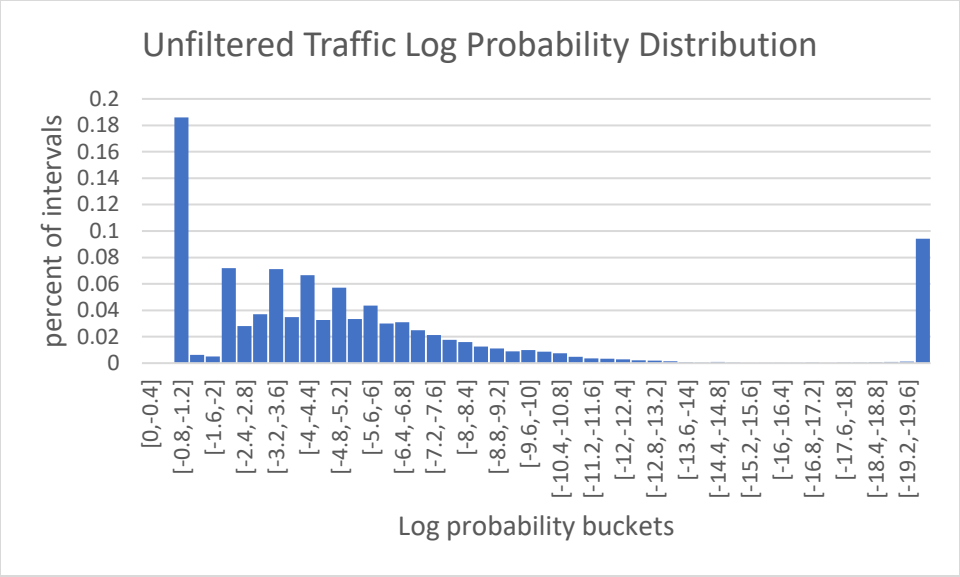
$$Prob(Log(P_i) > -9.2) = 0.95 \Rightarrow Log(P_i) > -9.2 \Rightarrow P_i > 0.000101$$



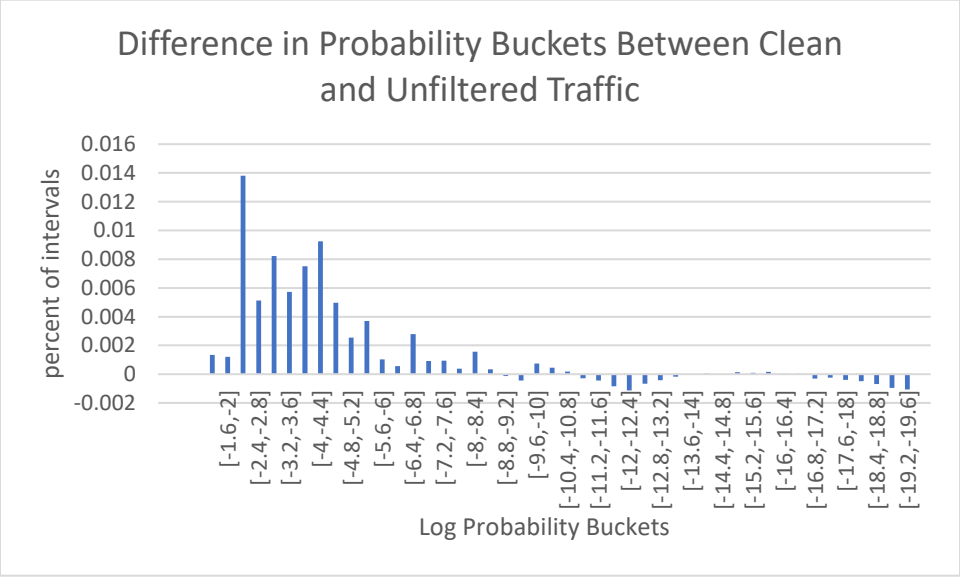


3- Unfiltered Traffic Distribution:

Using the markov model for clean traffic, the unfiltered traffic was passed through the model and the 20 min probability distribution was calculated. The results have been plotted below. The large number of observations in the right most bucket was due to observations with zero probability.



In the figure below the difference in log probability distribution between the 2 traffics is plotted.The really low and high end buckets have been removed for better visualization. It can be seen that the left hand buckets are positive which shows that clean traffic has a better correlation with the markov model. As we move to the right the values become negative indicating that the uncorrelation is more in the unfiltered traffic.



4- Attack Detection

In this paper our main goal was to find attacks where there is a significant change in incoming connections. Therefore our main targets where the following type of attacks:

- Network mapping

- Illegal upload of copyright content using Warez
- Illegal download of copyright content using Warez
- Syn flood denial of service
- Port sweep
- Network probing tools
- DOS attack using misfragmented UDP packets.
- DOS using ping of death

5- Results

We considered Log(P)<-18 as our cutoff point for detecting attacks. The results are based on a 7 week period of monitoring:

	True Detections	Missed Attacks
Count	20	4
Accuracy	83%	16%

	False Alarms	Total Connections
Count	7	25181
Accuracy	0.027%	

6- Further Improvements

While the markov model was able to determine attacks that had a significant change in connection count, it was not able to determine other types of attacks such as buffer overflow attacks.

The model also was also not able to detect attacks where the connections are made to multiple machines. Improvements can be made by considering connections made to all machines during the analyses.

Another issue with the model is that it is built based on normal network flow conditions. If there is a surge in regular traffic the model would most probably consider this an anomaly. More advanced simulations could increase the traffic by using the provided clean traffic and generating synthetic traffic to simulate a surge in traffic.

7- References

https://www.ll.mit.edu/ideval/data/1998data.html. (1998).