



Exercise 1 Security Model and Auditing

1) Extend the constraints (using $L(O)/L(S)$ and $C(O)/C(S)$) and the auditing requirements (what information to record and how to find a violation) for ss-property and *-property to the full BLP model. 

For ss-property, S reads O if and only if $L(S) \geq L(O) \ \&\& \ C(O) \subseteq C(S)$

For *-property, S wrote O if and only if $L(S) \leq L(O) \ \&\& \ C(S) \subseteq C(O)$

 2) Let's consider BSM audit records. Shown in a lecture, one BSM audit record example has the following attributes: {event, auid, ruid, euid, egid, rgid, pid, sid, remoteIP, time, error_message, process_error, retval}. Select those attributes that can be used for the information required for full BLP auditing. Which attributes of these are necessary and which optional? (Note that it might be the case that some required information is not available in this record.)

~~Time~~, event, error_message, euid, egid are necessary since audit records has to record what event happens, when does the event happens, ~~error information about this event for detection~~, who causes this event and which group that subject belongs to. And the others are optional for audit records.

Exercise 2 Basics of Analysis

Please use IBL (in its original form, not clusters) to design anomaly detection and misuse detection engines respectively using the above data records as instances. Specifically, describe the following:

- (a) In training, how do you generate the profile/signatures to be used for detection?
- (b) In detection, how do you classify a new data point/event to be normal or intrusive?
- (c) How to set up control threshold(s) used in (b).


For Anomaly detection:

First, we can transform the observed sequential data into the format of the given data record(events that are known as **normal** behaviors with other information such as login time and session duration from logs or other files) as feature vectors. A user profile is built to contain a collection of well-formatted data records/events selected from a user's observed actions. Then we apply the K-NN model to the profiles to train the model.

To classify whether a new data point/event, \mathbf{X} , to be normal or intrusive, we can calculate the similarity between the new data record and the profile, i.e. the similarity between \mathbf{X} and a vector in the profile that is most similar to \mathbf{X} . To measure similarity in this case, we use distance between \mathbf{X} and the most similar vector as the metric. Then a threshold ϵ is chosen. If the similarity between \mathbf{X} and profile is greater than ϵ , \mathbf{X} is considered normal; otherwise, \mathbf{X} is considered abnormal.

I think the threshold can be an empirical parameter. Therefore, to obtain the optimal threshold, we can do a couple of experiments based on the data we have. We can initialize a range of potential thresholds. Every time we pick up a potential value from the range and use the K-NN algorithm to classify the data point and evaluate the precision and accuracy. We reduce the range gradually according to precision and accuracy. Generally, the higher the accuracy is, the better the threshold used in this prediction is. At last, we can get a good enough threshold ϵ for detection. In addition, we can keep updating the threshold ϵ according to new predictions performed by our model later.

For Misuse detection:

First, we can transform the observed sequential data into the format of the given data record(events that are known as **intrusions** with other information such as login time and session duration from logs or other files) as feature vectors. And we maintain a database of these well-formatted data records of misuses(signatures). To classify whether a new data point/event, \mathbf{X} , to be normal or intrusive, we can calculate the similarity between the new data record and the signature database, i.e. the similarity between \mathbf{X} and a vector in the signature that is most similar to \mathbf{X} . To measure similarity in this case, we use distance between \mathbf{X} and the most similar vector as the metric. Then a  threshold ϵ is chosen. If the similarity between \mathbf{X} and profile is greater than ϵ , \mathbf{X} is considered intrusive; otherwise, \mathbf{X} is considered normal.

The way of Setting up control thresholds is the same as in anomaly detection.

.