Failed Login Anomaly Detection

In this use case, we track the behavior of failed logins. An entity is represented by [User_ID, Source_IP]. For each client, we obtain a history of events of all failed logins going back to *x* days. See example below:

History			
Entity			Daily Counts
User ID	Source IP	Destination IP	Failed Login Count
U1	S1	D1	10
	S2	D2	2
	S3	D2	1
U2	S2	D3	15
	S7	D4	4

Step 1: Outlier Threshold Calculation

The first step after obtaining events, we need to identify what quantities of *failed login counts* would be considered outliers. Various statistical methods can be used here but we prefer non-parametric methods which do not depend on the distribution of the events. We will be using *Chebyshev Inequality* in this use case based on this research paper [1]. In this model, all the failed logins are treated the same irrespective of the entity they are associated with. Once we have identified the outlier quantity thresholds from historical data, we can then use them to identify future entities whose failed logins are likely to be anomalous if they are greater than the threshold.

Step 2: Validation of anomalous events

After identifying entities that could be anomalous, the second step involves extra validation in order to minimize false alarms. It turns out that users forget their passwords/usernames all the time and sometimes systems are reconfigured which leads to lots of failed logins especially by administrators who are likely to run automated authentications which quickly add up to huge numbers of failed logins. To minimize the level of non-malicious failed logins, we carry out an extra check by finding out if a seemingly anomalous entity had ever previously successfully logged into that specific destination computer. If this is the case, all those 'anomalous' entities are ignored.

Step 3: Aggregation of failed logins at the user level

While we can flag anomalous failed logins at the entity level, we might want to aggregate this anomaly at the user level in order to capture repeat offenders. Usually, systemic hacks involve a widespread campaign of attempting to break into many machines within a network either as part of lateral movement, credential stuffing, password re-use or brute-force. The only way to detect this is by aggregating all the entity failed logins at the user level. This is done by using the Fisher method [2] which is a probabilistic fusion and scoring function.

References

1. https://kyrcha.info/2019/11/26/data-outlier-detection-using-the-chebyshev-theorem-paper-review-and-online-adaptation
2. https://en.wikipedia.org/wiki/Fisher%27s_method