**Detecting changes in user connection access behavior**

In this use case, we track entity login patterns over a duration of time and use those patterns to detect when an entity access pattern has significantly deviated from its expected behavior based on connections opened. Table [1] below shows a history of successful logins by entities [User_ID, Source_IP].

| Entity | | |
|---|---|---|
| **User ID** | **Source IP** | **Destination IP** |
| U1 | S1 | D1 |
| U1 | S2 | D2 |
| U1 | S3 | D3 |
| U2 | S2 | D4 |
| U3 | S5 | D7 |
| U3 | S5 | D6 |
| U4 | S1 | D6 |
| U4 | S2 | D7 |
| U4 | S3 | D5 |

Table[2] below shows new events by the same entities tracked from historical logs.

| Entity | | |
|---|---|---|
| **User ID** | **Source IP** | **Destination IP** |
| U1 | S1 | D1 |
| U1 | S2 | D3 |
| U2 | S2 | D4 |
| U3 | S5 | D7 |
| U3 | S5 | D5 |
| U3 | S4 | D8 |
| U4 | S1 | D9 |
| U4 | S2 | D10 |
| U4 | S3 | D11 |

The history is made up of unique entries and all repetitions can be dropped in order to improve computation and minimize disk space requirements. For all new observations, we can compute the deviation of each entity using two main metrics:

**Jaccard Index**

Jaccard index is based on an entity which could be for example a user ID or a combination of a user ID and Source IP etc. In the example below, the former is used.

We use a modified form of Jaccard Index as follows:

Modified Jaccard Index = $\frac{new\ event\ counts}{counts\ of\ \ all\ new\ unique\ events}$

| User ID | History (Dst Machines) | New (Dst Machines) | Count New NOT in history | Count ALL New | Modified Jaccard score |
|---|---|---|---|---|---|
| U1 | D1,D2,D3 | D1,D3 | 0 | 2 | 0.00 |
| U2 | D4 | D4 | 0 | 1 | 0.00 |
| U3 | D5,D6 | D5,D7,D8 | 2 | 3 | 0.67 |
| U4 | D5,D6,D7 | D9,D10,D11 | 3 | 3 | 1.00 |

**Login Probability**
The second metric we consider is the probability of a certain number of connections being opened by an entity. Ordinarily, in any network, there is an underlying distribution of connections opened by all the entities. Whenever an entity opens extremely high connections in comparison with others then this is usually treated as a suspicious occurrence. We calculate the rarity of connections opened by computing the probability of a given number of connections. Table [3] below shows likelihood table of connections opened during a set time-period for instance a day, hour etc

| Entities | Connection Counts |
|---|---|
| U1, S1 | 3 |
| U2, S2 | 2 |
| U3, S3 | 1 |
| U4, S3 | 1 |

We calculate Likelihood (Probability of Connections) based on events in table[3] above:

| Connection Counts | Probability (Connection Count) |
|---|---|
| 3 | 1/4 |
| 2 | 1/4 |
| 1 | 2/4 |

Lastly, we combine the two metrics (Jaccard Index and Connection likelihood) to come up with a score of estimated anomalousness of the logins. We combine the two metrics using Fisher probability combination [1] to come up with a final score. Ordinarily the fisher combination uses p-values as inputs but in this case, we use it heuristically as a logarithmic function to sum the probabilities instead of multiplying them out. Instead of using real probabilities you can also use z-scores of values in order to standardize the fisher scores even more.

$$fisher\ score = \ -2\sum_{k=1}^{k} log[p(k)]$$

Table [4] below shows the combined Fisher scores from the combined Jaccard and Likelihood measures. Higher Fisher scores signifies rarer and more anomalous events.

| Entity | Connection Counts | 1.0 – modified Jaccard Index | Probability Of Connections | Fisher Score |
|---|---|---|---|---|
| U4, S3 | 3 | 0.00 | 0.25 | 21.20 |
| U3, S3 | 2 | 0.67 | 0.25 | 1.55 |
| U2, S2 | 1 | 1.00 | 0.5 | 0.60 |
| U1, S1 | 1 | 1.00 | 0.5 | 0.60 |

**Aggregation**

The final aggregation of fisher scores can also be performed at the user level by combining all the entity scores in order to identify the likelihood of malicious behavior at the user level. One simple way to do this is by summing fisher scores at the user entity level.

References

1. https://en.wikipedia.org/wiki/Fisher%27s_method