Analyzing credit limit increase requests

Credit limit increase requests can be approved or denied quickly. For example, customers not in good standing would be denied. Limit approvals or denials can be mapped over time to see trends. Splunk can participate in granting approvals based on reference data and rules. Splunk software not only monitors machine data but also actively participates in the preliminary transactions.

Data required

- Business service data for credit limit increase requests
- A CSV or KV lookup file that shows customers' credit standing, income, and number of yearly transactions
- Splunk recommends that customers look into using data models, report acceleration, or summary indexing when searching across hundreds of GBs of events in a single search. The searches provided here are a good starting point, but depending on your data, search time range, and other factors, more can be done to ensure that they scale appropriately.
- You can optimize the following searches by specifying an index and adjusting the time range. You may also need to adjust field names (such as customer name/ID and desired credit limit increase) to match what is available in your application logs and lookup files.

Credit limit increase authorization and denial details

Customers often ask for a credit limit increase. Using reference data, you can correlate with the customer’s request and provide a list of customers that were preliminarily authorized. Customers must be in good standing, have enough income, and enough transactions.

Now, apply rules to authorize. For instance, if the customer is in good standing, has income that is ten times the credit limit amount, and at least 10 transactions per year, authorize the customer in a table. A separate process can review these reports to make a final decision. Run the reports on a hourly basis.

```
|source="<credit increase request logs>
|eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S")
|sort - _time
|eval action="authorized"
|lookup <CustomerData> customer
|eval action=if(good_standing="yes" AND year_transactions>10 AND income>(new_limit*10), "authorized", "denied")
|where action="authorized"
|eval income=tostring(round(income, 2),"commas"), new_limit=tostring(round(new_limit, 2),"commas")
|fields - epoch
```

Reasons for denial could be bad credit standing, low income, or not enough transactions. This search looks for...
customers who are in good standing but do not have enough yearly transactions, defined as less than or equal to 10 per year. The reason is output into a table. A separate process can review these reports to make a final decision.

|sourcetype=<credit increase request logs> |
eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S") |
sort - _time |
eval action="authorized" |
lookup <CustomerData> customer |
eval denied_reason="Number of yearly transactions too low" |
where good_standing="yes" AND year_transactions<=10 |
eval action="denied" |
eval income=tostring(round(income, 2),"commas"), new_limit=tostring(round(new_limit, 2),"commas") |
fields - epoch |

Make the following adjustments to the denial search above to output other reasons that customers might be denied a credit limit increase.

<table>
<thead>
<tr>
<th>Original search</th>
<th>Denial reason</th>
<th>Replacement search lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>eval denied_reason=&quot;Number of yearly transactions too low&quot; where good_standing=&quot;yes&quot; AND year_transactions&lt;=10</td>
<td>Denied because income is too low (less than 10 times credit limit)</td>
<td>eval denied_reason=&quot;Income too low compared to new credit limit&quot; where good_standing=&quot;yes&quot; AND income&lt;=(10*new_limit)</td>
</tr>
<tr>
<td>eval denied_reason=&quot;Number of yearly transactions too low&quot; where good_standing=&quot;yes&quot; AND year_transactions&lt;=10</td>
<td>Denied because customer is not in good standing (bad credit)</td>
<td>eval denied_reason=&quot;Customer is not in good standing&quot; eval action=&quot;denied&quot; where good_standing=&quot;no&quot;</td>
</tr>
</tbody>
</table>

Authorization versus denial over time

When customers ask for a credit limit increase, you can correlate reference data with the customer's request and provide a list of customers who were preliminarily authorized or denied. The timechart helps you compare counts of authorizations versus denials over time.

| sourcetype=<credit increase request logs> |
eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S") |
sort - _time |
eval action="authorized" |
lookup <CustomerData> customer |
eval action=if(good_standing="yes" AND year_transactions>10 AND income>(new_limit*10), "authorized", "denied") |
timechart span=10m fixedrange=F count BY action |
New credit limit requested amount over time

This search gives a look at how much credit the average customer is requesting at any given time, helping the business prepare for giving out future credit.

```
|sourcetype=<credit increase request logs>
|eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S")
|sort - _time
|timechart span=10m fixedrange=F avg(new_limit) AS requested_limit
```

Credit limit increase denial counts side-by-side

You want to correlate reference data with the customer's request and see side-by-side counts of reasons for denial over time. This search provides counts for the 3 reasons credit limit increases are denied. Save the visualization as a Trellis Chart to see over time reasons for denial side-by-side.

```
|sourcetype=<credit increase request logs>
|eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S")
|sort - _time
|eval action="authorized"
|lookup <CustomerData> customer
|timechart span=15m fixedrange=F count(eval(good_standing="no")) AS "denied not good standing"
        count(eval(year_transactions<=10)) AS "denied not enough yearly transactions"
        count(eval(income<(10*new_limit))) AS "denied not enough income"
```

Next steps

Check with your local government regulation to determine how you can use these searches to determine credit worthless for an individual or company. The results of these searches are not final determiners of risk. By using Splunk software to provide reports on customer credit limit increase requests, you can operationalize the preliminary decision from in coming log events, allowing for faster decision making and streamlining the business. You can also get insights into the why limit increases are denied over time for future planning and to troubleshoot rules that may be too strict. Too many denials, or authorizations, may mean the rules may need to be tweaked.

The Splunk Essentials for the Financial Services Industry app helps you automate the searches provided in this article. The app also provides more insight on how they can be applied in your environment, how they work, the difficulty level, and what data can be valuable to run them successfully. In addition, the Splunk Essentials for the Financial Services Industry app provides a number of other monitoring and reporting solutions for credit services:

- Credit fraud
- Credit monitoring

Finally, these additional Splunk resources might help you understand and implement this use case:

- Conf Talk: 40 ways to use Splunk in financial services