Detecting wire transfer fraud

Applicability

- Product: Splunk Enterprise or Splunk Cloud Platform
- Function: Monitoring, alerting, reporting
- Data: Business service data of wire transfer data
- A CSV or KV lookup file of data on suspicious countries

Scenario

You work for a financial services company that conducts a lot of wire transfers. You know that your customers are at risk for many common scams, such as fake rental deposit requests, lottery winnings that supposedly require a tax prepayment, and overpayments on bad checks. You want to do everything you can to protect your customers from scammers and protect the reputation of your business.

Some commands, parameters, and field names in the searches below may need to be adjusted to match your environment. In addition, to optimize the searches shown below, you should specify an index and a time range when appropriate.

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.

Detect large wire transfer immediately after account activation

One of your concerns is new accounts being used to transfer bad checks. You want to search for large sums of money transferred when the account is less than one day old, which is a common indicator of a fraudulent account.

```
|sourcetype=<wire transfer data source>
|eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S")
|sort - _time
|where _time<relative_time(OpenDate, "+1d") and amount>10000
|eval AccountOpenDate=strftime(OpenDate,"%Y-%m-%d %H:%M:%S")
|iplocation destIP
|rename Country AS DestCountry City AS DestCity
|table _time,AccountOpenDate, customer, amount, FromAccount, ToAccount, DestCountry, DestCity
|eval amount=tostring(round(amount, 2),"commas")
```
Detect wire transfers into suspicious or banned countries

Large amounts of money being transferred into certain countries can indicate fraud. You want to correlate user activity with your list of banned and suspicious countries.

```
sourcetype=<wire transfer data source>
| eval _time=strptime(_time, "%Y/%m/%d %H:%M:%S")
| sort - _time
| eval is_suspicious="no"
| lookup <name of lookup file of suspicious countries> ip AS destIP OUTPUT is_suspicious
| where is_suspicious="yes" AND amount>10000
| iplocation destIP
| rename Country AS destCountry
| table _time, customer, FromAccount, ToAccount, action, amount, destCountry, destIP
| eval amount= tostring(round(amount, 2), "commas")
```

Detect wire transfers from multiple client IP addresses

In order to look for signs of compromise and help protect your users, you want to report regularly on the IP addresses users are transferring money from.

```
sourcetype=<wire transfer data source>
| convert timeformat="%Y/%m/%d %H:%M:%S" ctime(epoch) AS c_time
| sort - _time
| eval amount= tostring(round(amount, 2), "commas")
| streamstats time_window=1m count(eval(action="authorized")) AS authorized_count list(FromAccount) AS FromAccount list(ToAccount) AS ToAccount list(amount) AS amount list(action) AS action list(c_time) AS time
| where clientIP_count>=2
| fields - OpenDate epoch destIP _time, clientIP_count c_time authorized_count
```

Detect when a number of wire transfers exceeds threshold

In order to help protect your users, you want to report on the number of large transfers that clients have completed in a one hour time span. Excessive transfers might indicate fraud.

```
sourcetype=<wire transfer data source>
| sort - _time
| eval amount= tostring(round(amount, 2), "commas")
| streamstats time_window=1h count(eval(action="authorized")) AS authorized_count list(FromAccount) AS FromAccount list(ToAccount) AS ToAccount list(amount) AS amount list(action) AS action list(_time) AS time
| where authorized_count>=7
| fields - OpenDate epoch clientIP destIP _time
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Detect multiple account login denials followed by authorization

Brute force password attacks are a concern in your financial services organization. In order to help protect your users, you want to report hourly on accounts that have multiple login denials followed by an authorization, which might indicate cracked credentials.

|source type=<wire transfer data source> |convert timeformat="%Y/%m/%d %H:%M:%S" ctime(epoch) AS c_time |sort - _time
|streamstats time_window=1h count(eval(action="denied")) AS denied_count count(eval(action="authorized")) AS "authorized_count" latest(action) AS latest_action BY customer,FromAccount,ToAccount |where denied_count>=5 and authorized_count>=1 and latest_action="authorized" |table _time, customer, FromAccount, ToAccount, amount, denied_count, authorized_count
|eval amount=tostring(round(amount, 2),"commas")

Results

Use the results of these searches to make recommendations to the rest of the security team about which users should be investigated for potentially fraudulent activity. Create reports based on these searches and schedule them to run at a regular cadence as needed. Be sure to follow any industry policies and regulations that are required for compliance.

Additional resources

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 fraud detection solutions for financial services:

- Banking fraud
- ATM fraud
- Credit card fraud

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

- .Conf Talk: 40 ways to use Splunk in financial services
- Blog: Using maths to fight financial crime