Network Partitioning Effects on Ripple Transactions

Yoan Martin
Today’s menu

• What is Ripple?

• Why is it interesting?

• Attacks

• Analysis
What is Ripple?

- Global Payments Network
- RippleNet vs XRP
- Gateway
  - Entry Point
- Ripple Bank
Why is it interesting?

• More than 200 financial institutions

• ~20’000’000 USD sent by hour

• Take place on internet
What is the network?
Network
What is the network?
Attacks

• What if an AS is malicious?
• What can it do?
  • Dropping the traffic
  • BGP Hijacking
Traffic dropped
BGP Hijacking
BGP Hijacking
I know B!
BGP Hijacking
How to measure the effect?

- Build the Ripple Network
  - Ripple API
- Caida
- Use previous transactions
- Replay transactions when an attack occurs
Build RippleNet

Ripple API, Gateways data

AS relationships

AS
AS with a Gateway
Map Result
Transactions

- Account A sends 100 XRP to account B
- Some transactions have gateways data
  - Account A sends 100 XRP using Gateway G to B
  - Account B receives 100 XRP using Gateway H from A
- Keep only transactions with matching Gateways
Simulation: traffic dropped

A sends 100 XRP to B
D sends 10 USD to A
C sends 4 EUR to B

...
Simulation: traffic dropped

- If \( == \), transaction is complete
- If \( != \), transaction is rerouted
- If no, transaction is lost

- If \( \text{green} == \text{red} \), transaction is complete
- If \( \text{green} != \text{red} \), transaction is rerouted
- If no \( \text{red} \), transaction is lost
## Example of results

<table>
<thead>
<tr>
<th></th>
<th>Completed</th>
<th>Rerouted</th>
<th>Lost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>10%</td>
<td>10%</td>
<td>80%</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>30%</td>
<td>20%</td>
<td>50%</td>
</tr>
<tr>
<td>China Telecom</td>
<td>60%</td>
<td>30%</td>
<td>10%</td>
</tr>
<tr>
<td>Swisscom</td>
<td>30%</td>
<td>30%</td>
<td>40%</td>
</tr>
</tbody>
</table>
Simulation: BGP Hijacking

A sends 100 XRP to B
D sends 10 USD to A
C sends 4 EUR to B

...
Simulation: BGP Hijacking

- If \( \text{green} == \text{red} \), transaction is complete
- If \( \text{green} != \text{red} \), transaction is rerouted
## Example of results

<table>
<thead>
<tr>
<th></th>
<th>Completed</th>
<th>Rerouted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>60%</td>
<td>40%</td>
</tr>
<tr>
<td>China Telecom</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>Swisscom</td>
<td>30%</td>
<td>70%</td>
</tr>
</tbody>
</table>
Real Results

• Transactions analysis
• Which ASes are the most dangerous?
• What is the effect on the Ripple network?
Transactions analysis

- % of transactions with AS as sender or receiver
- 13335 is Cloudflare (US)
- 19551 is Incapsula (US)
Which ASes are dangerous?

Traffic dropped

- % transactions lost corresponds to transactions distribution
- Lost if gateways in corrupted node
- Never lost if intermediaries
- Always possible to find a path
Which ASes are dangerous?

Traffic dropped

- Little % of rerouted transactions
- Certainly due to transactions distribution
- 553 is Belwue (DE)
- Connections with 680 ISP
- Switch, Swisscom
Which ASes are dangerous?
BGP Hijacking

- Many ASes can corrupt the network
- Long list of ASes reach almost 40% of rerouted transactions
What is the effect on Ripple?

- Time analysis
- On average low effect
Conclusion

- Most of the transactions go through 2 ASes
  - Big impact if one of them is corrupted
- BGP Hijacking has more effect than traffic dropped
- Limitations of this analysis
  - Network only considers Gateways
  - Hence, only a few transactions are considered
Thank you for your attention
White gap?

In [36]:
   for i in range(len(graphs_list)):
       print('16265 in {} : {}'.format(months[i], '16265 in graphs_list[i].nodes))

16265 in January : True
16265 in February : True
16265 in March : True
16265 in April : True
16265 in May : True
16265 in June : True
16265 in July : True
16265 in August : False
In [43]:
   1  for i in range(len(graphs_list)):
   2     print('Direct link for {} : {}'.format(months[i], graphs_list[i].has_edge('13335', '19551')))  

Direct link for January : True
Direct link for February : True
Direct link for March : True
Direct link for April : True
Direct link for May : True
Direct link for June : True
Direct link for July : True
Direct link for August : False