Typical vulnerabilities in Lightning Apps ⚡

Ололо пыщь пыщь

Igor Korsakov / bluewallet.io / Berlin / October, 2019

blue wallet

Plan

1. Double spends with hold-invoices; anatomy of sendPayment
2. Stealing free fees
3. Race-condition attacks
4. Negative amounts
5. ...
6. Profit!
7. Best practices

blue wallet

What is a Lapp?

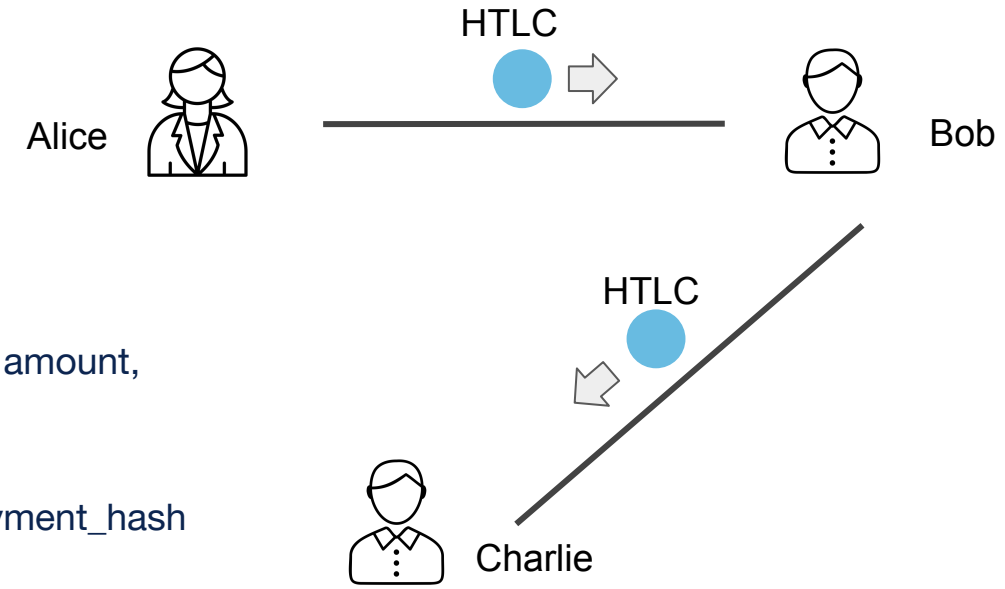
A web app that can interact with the Lightning network:

- Receive payment
- Send payment
- Authenticate (sign with node’s key)

blue wallet

Hold-invoice attack

Anatomy of SendPayment



1. Create **preimage**
2. **payment_hash** = hash(preimage)
3. Create **bolt11 invoice** with payment_hash, amount, description, signature
4. Give **invoice** to Alice
5. Alice sends **HTLC to Bob** protected by payment_hash promising that Charlie has solution to hash
6. Bob sends **HTLC to Charlie** protected by same hash
7. Charlie **reveals preimage** in order to get the payment

blue wallet

Hold-invoice attack

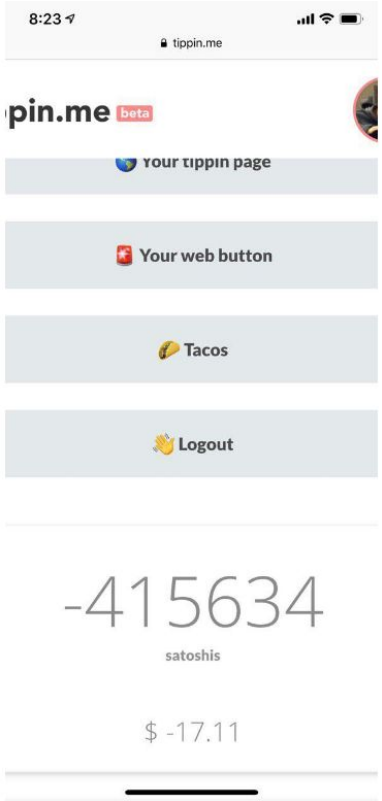
```
router.post('/payinvoice', function(req, res) {

    if (userBalance >= num_satoshis) {
      // got enough balance

      lightning.sendPayment(invoice, function(err, result) {
        // callback with result of sent payment
        reduceUserBalance(num_satoshis);
      });
    }
```

blue wallet

Hold-invoice attack. Execution



blue wallet

Hold-invoice attack. Execution

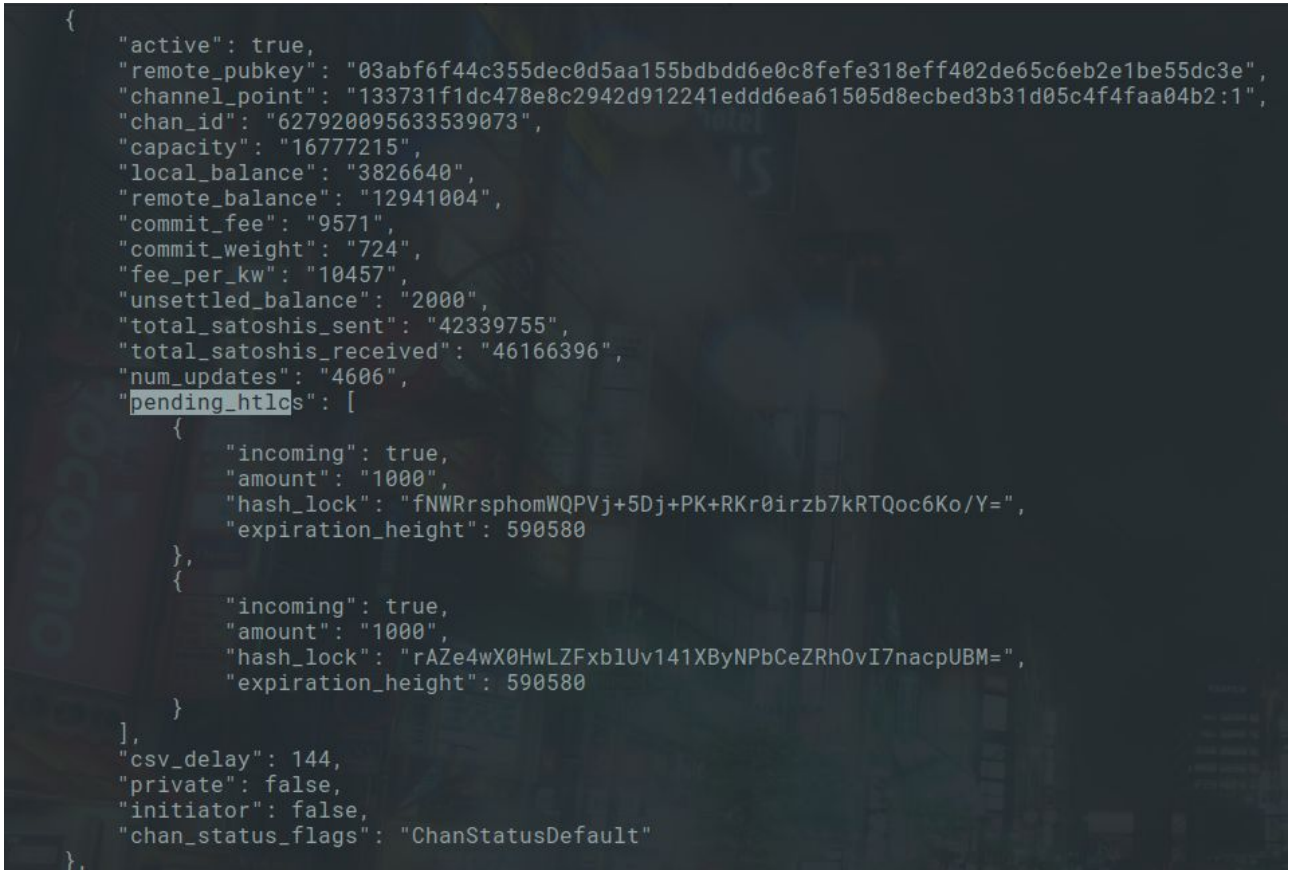
```
$ # in lnd folder
$ make tags="invoicesrpc" && make install tags="invoicesrpc"
```

```
$ cat invoices.sh
PREIMAGE=$(cat /dev/urandom | tr -dc 'a-f0-9' | fold -w 64 | head -n 1)
HASH=`node -e "console.log(require('crypto').createHash('sha256').update(Buffer.from('$PREIMAGE', 'hex')).digest('hex'))"`

echo "lncli settleinvoice $PREIMAGE" >> settle.sh
INV=`lncli addholdinvoice $HASH --expiry 600  --amt 99`
INV2=`echo $INV | awk '{print $3}' | sed "s/[^a-zA-Z0-9']//g"`
echo "pre = $PREIMAGE hash = $HASH"
echo $INV2
echo $INV2  | qrcode-terminal
```

blue wallet

Hold-invoice attack. Execution



blue wallet

Hold-invoice attack. Protection

- Atomically lock out full withdrawal amount (with fees) before doing anything else
- Lock should not auto-expire. Release lock only when payment is in determined state (either failed or went through)
- Check stuck payments periodically (usually up to ~1day):

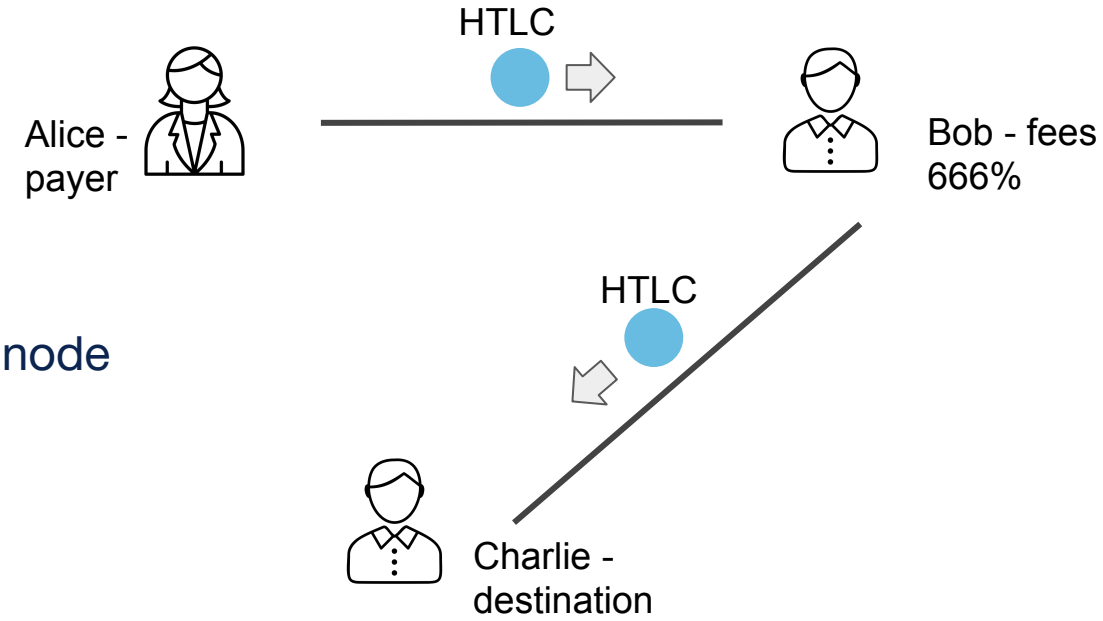
  $ lncli listpayments

  or smth like that

- Disregard invoice expiry

blue wallet

Stealing free fees



1. Offer free withdrawals
2. Someone sets intermediate node with high fees
3. ...
4. Profit!

FEE LIMIT won't help!

blue wallet

Stealing free fees. Protection

1. Don’t giveaway fees: feelimit - lock payment amount + feelimit
2. OR calculate route’s fees and add them to amount when charging user

blue wallet

Probe route example

```
igor@vmi224865:~$ lncli queryroutes --dest=03e2b5fc50b765a9126e027b2220580ce80cd1e81d592746cd581273b06fb6605e  --amt=666 | lncli
sendtoroute  --payment_hash=ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff  -
{
        "payment_error": "UnknownPaymentHash(amt=666000 mSAT)",
        "payment_preimage": "",
        "payment_route": null
}
igor@vmi224865:~$
```

blue wallet

Race-condition attacks

```
router.post('/payinvoice', function(req, res) {

    if (userBalance >= num_satoshis) {
      // got enough balance

      lightning.sendPayment(invoice, function(err, result) {
        // callback with result of sent payment
        reduceUserBalance(num_satoshis);
      });
    }

});
```

blue wallet

Race-condition attacks. Protection

```
router.post('/payinvoice', function(req, res) {

    if (!(await lock.obtainLock())) {
      return errorTryAgainLater(res);
    }

    if (userBalance >= num_satoshis) {
      // got enough balance

      lightning.sendPayment(invoice, function(err, result) {
        // callback with result of sent payment
        reduceUserBalance(num_satoshis);
      });
    }
```

blue wallet

Negative amounts 😂

```
router.post('/addinvoice', function(req, res) {
  if (req.body.amt < 0) return errorBadArguments(res);

  ...

});
```

blue wallet

Negative amounts. Protection

Write tests!

Worth nothing! Other risks

- Unsafe zero-amount invoices
- Unsafely-opened channels
- DDOS to prevent you from issuing retaliate tx
- Observing counterparty offline/online patterns to choose best timing to issue old state tx
- All web-app vulnerabilities apply to you! XSS, injections, fuzzing, etc. Study OWASP!

As LN economy grows, be sure. Black hats will come. Tooling will be made, exploits will be written.

blue wallet

Best practices

1. Don’t store user balance as single variable. It should be a sum of all transactions
2. Don’t store amounts as float, only as int. Signed int is ok, no point to enforce unsigned int everywhere
3. RDBMS and Transactional databases are nice to have
4. Log everything, and keep all logs
5. Do regular accounting. At least daily, and investigate if actual values differ from expected
6. Don’t be obsessed with MVP

blue wallet

Acknowledgements

1. Justin Camarena from Bitrefill
2. Andrey Samokhvalov from Bitlum/Zigzag
3. Great guys from https://ion.radar.tech

blue wallet

“Not great, just ok”

“Only one coin”

“Just another wallet”

blue wallet

“Could be better”

i@bluewallet.io

“Meh”

“Some features”

← even Roger is not impressed