Please note:
That document needs update to the current state of the project. Some information might not be valid anymore, though the basics have not changed. We are working on a new document with updated and better structured information.

Risk analysis

Version 0.9

This document analyzes risks, fraud opportunities, attack scenarios and possible protection against those.
It is just a first attempt for a more profound analysis and need more discussion, input and peer reviews.

Risks on the network layer are very short handled yet and need more input from experts in that area.

The basic use case is the one described in the paper and overview, where Alice is the BTC buyer who publishes the offer. The other use case with Alice as seller is similar, but for keeping things simple not mentioned here further.

Later it will be tested if that use case opens up different attack scenarios which are not covered here.

Use cases

Broadcast offer

Alice creates an offer and publish it to the P2P network.

Abuse cases

1. Create an fake offer

Protection

- Offer fee
- Report take offer rejections or timeouts to a reputation system

2. Spams the order book with many fake offers

Protection

- Offer fee
- Report take offer rejections or timeouts to a reputation system
- Limit number of possible offer broadcasts
- Adopt fee to number of broadcasts (hard to implement, not considered yet)
- Proof of work (not considered yet)
- Payment for broadcasts above a flat rate limit (hard to implement, not considered yet)

3. Fake offer with intention of price manipulation

If Eve wants to sell 1 BTC for 400 EUR but there are better offers in the orderbook, she could create fake buy offers with a price above market price, so the sell offers in the orderbook could be canceled with the intention to take advantage of those better buy offers. That way she could kick out competitors for a while and increase her position in the orderbook. The same mechanism could be used with taking an offer, but not continuing the trade, so the offer got removed from the orderbook for a while and need to be rebroadcasted after a timeout or rejection. That time span could be used as advantage in the orderbook.

The protection with fees highly depends on the actual situation. How many offers are there, how long is the timeout (the time she can win with that attack). So it is hard to protect it with fees only. Too high fees have negative impact to the overall exchange as well, too low fees will not serve as enough protection.

Protection

- Locally blacklist nodes who rejected take offer requests above a certain threshold

- Report take offer rejections or timeouts to a reputation system

4. Boycott storage and distribution of competing offers

A node could find out if competing offers are stored or relayed in his DHT realm and boykott the delivery to other nodes.

Protection

- A protection against misbehaving/unreliable nodes needs to be handled on the messaging layer.

TODO: Find out how TomP2P handles that.

5. Offer creation fee payment could be double spent as we dont wait for 1 confirmation

That will be accepted as a trade off to make the trade process more fluent.
It could be protected as well with local blacklisting and reputation, afterwards when the trading peer discover that the offer fee has been double spent. But in reality it will probably not be a big problem as double spends are not so easy to execute and the fees are very low.

6. Offer fee payment goes to a offerer controlled sockpuppet

The selection of the receiver of the fee will done by a verifiable but unswayable mechanism. A combination of the message ID and the hash of the offer (offer ID) might be a good root for the selection algorithm.

Protection solutions

Offer fee

Easy to implement and to verify by the trading peer.
Offers without fee payment will not accepted in a trade.
Only protects against economical motivated attackers.
It is intended to use the fee as payment to the project donators.
The fee should be in the range of 0.1 - 1 EUR.
One problem with a fixed fee is the BTC volatility which might increase the fee to an undesirable value. Versioning could help to solve that, so software updates changes the fee policy (like it was done in the bitcoin client).
Best would be a fee derived from the actual trading activity (e.g. 1% of the average of the last 1000 trades) but that will be too much effort for the initial version.
The protection with fees might not be enough for price manipulation attacks.

Network limitation

A general anti-spam mechanism need to be available on the messaging layer.
Message traffic limitation and local blacklists for IPs/message IDs sending too much messages should be a basic protection against that. The limitation can be dynamic so it will not decrease the speed for normal behaviour. E.g. up to 10 messages in 1 min. is tolerated as normal

behaviour. If that limit is exceeded, the node gets ignored for the next 5 minutes. If it is still flooding afterwards the node gets ignored for the next 1 hour, after that a continuous flooding will lead to local blacklisting of the IP address/message ID.

TODO: Find out which features TomP2P offer regarding spam protection.

The Bitcoin p2p network uses a 'banscore' system to blacklist peers that misbehave. Something like this could also be used. Perhaps a separate table keyed by registration account would also be useful for blacklisting accounts that misbehave by violating the message protocol.

Local Blacklist

To blacklist locally misbehaving nodes should be a basic protection against flooding/spam. After repeated take offer rejections from the same trader that node can be blacklisted, so it will not appear anymore in the orderbook. It might be not enough protection against market manipulator attacks.
TODO: Find out which features TomP2P offer regarding blacklisting.

Reputation system

Please note: The reputation system for traders is not planned to be implemented as it is not really necessary and has several weaknesses and complexity. For the arbitration system we will use a reputation system based on similar ideas described here.

A reputation system where rejections could be listed could help against several attacks.
The main problem with reputation systems is that they are vulnerable to Sybil attacks/sock puppets and fake reputation entries. To get a more qualitative reputation value we use a hierarchical model similar to web of trust models.

We store the reputation in the DHT with the Message ID as key and a map of reputation entries. The entry key is the reporters message ID and he need to sign the entry content. There is only one entry per peer possible but it can be updated, that way misuse is limited. If a user wants to damage the reputation of another peer he need to create a lot of trading accounts which produce registration costs and he needs different bank accounts. His own reputation can be looked up and that way a reputation tree with weightings could be build up.
The reputation entry can be updated to increase or decrease the given reputation.
Reputation entry contains:

- My message ID (used as key)
- Number of rejections (most legitime behaviour)
- Number of timeouts (legitime behaviour, but maybe a peer with poor connectivity)
- Number of trade discontinuations (resulted in a loss of the take offer fee)
- Number of disputes (arbitration was necessary)
- Number of bank chargebacks
- Number of crime involvements (stolen bank account)
- Number of incorrect reputation report (description needed)
- Number of other misbehaviour (description needed)

- Signature of the entry

The data structure looks like that:
DHTStorage{ key: peerID, reputationEntries: [reporterKey: myID, data: { reputationEntry: {rejections: 0, timeouts: 1, discontinuations:0, disputes:0, chargebacks:0, crime:0, other: [“hash to a report of the description”,...] }, signature: <signature of reputationEntry> },...] } }

The successful trades are not included to prevent privacy leaks (otherwise you could reconstruct the trading history from the reputation).
We might include the number of offer fee payments as a value for the trade activity. That way the privacy stays intact, but that is not very helpful, as you could create a lot of fake offers which you cancel later, just to increase that value. The offer fee will be not that high that this will be enough protection. So it is questionable if it makes sense to include that.
But there would be missing some information of the trade activity. A trader with a lot of trades have probably a lower reputation because sometimes something goes wrong. A new trader has the highest reputation as he had not yet the opportunity to lose any.
Maybe a time to life could be used for the non critical reputation attributes (timeout, rejections), so old negative reputation entries will fade out after a time.

The client calculates the weight of a reputation entry from the reputation score of the reporter. We can go a few levels deep, but probably due performance reasons 1 or 2 levels will be realistic.
The different reputation attributes have different influence to the score. E.g. rejections lowest impact, crime highest impact.
TODO: check out more details about web of trust and how reputation is designed in the Open Bazaar project

The reputation system will have low priority, as it is not essential for the basic protection and comes with too many unsolved problems.

For an extended version of the exchange to a generic P2P marketplace a reputation system will become mandatory as the trade cannot be reduced to a binary result anymore (success if money received, otherwise failed). With product trades there are different levels of satisfaction (article received, but does not fulfill the expectation, delivery time,...).
It is an open challenge how to prevent privacy leaks of the trade history for such a model.

Take offer

1. Fake take offer requests with intention of price manipulation

Same attack scenario and protection mechanisms like above.

Deposit transaction

Abuse cases

1. Offerer could extort taker as taker has put more fund into the deposit tx

That extortion risk is specially then a real risk when the offerer sends the taker a pre-signed payout tx with payment to his favor. In that case there is an atomically tx and the taker can publish that tx without any further interaction with the offerer (so the who pays first problem does not apply here which would make normal blackmail less successful).

2. Taker double spends his input

When passing the deposit tx to the peer the taker could spend his inputs to another tx and make the deposit tx invalid. When the offerer publishes the tx it will be rejected, so the offerer cannot lose his payment but will lose time.
Would be same type of price manipulation attacks like above with same protection mechanisms.

3. The take offer fee could be double spent

Same like above in the case of offer creation.

4. One of the trader does not continue with his step:

A timeout will protect every interaction, so after that timeout got triggered, the trade will be canceled and the offerers removed offer re-published. As long as the deposit tx is not published nobody lose more than the take offer fee and time. Repeated trade cancellations due timeouts will be treated like rejections (local blacklist).

5. Take advantage of price changes during the trade process

The time for the initial deposit tx is very short (a few seconds), so to use that period is not considered as a risk.
As soon the deposit tx is published any breach of the trade contract will lead to arbitration with the possible loss of the collateral or collateral+payment for the parties.
For Alice it might be an advantage to lose the collateral and not pay the fiat if the price change is higher than the collateral.
Example: If she made the offer to buy 1 BTC for 500 EUR with a collateral of 10%, and the price changes to less than 450 EUR she would have an advantage to not fulfil the trade contract and take into account the loss of the collateral (50 EUR). That price change must happen pretty fast, which is not very uncommon due Bitcoins high volatility, but she would risk also that the timeout window for the bank transfer will be reached if she waits too long and then lose her collateral maybe with a price not to her advantage.

For Bob there is no way to take advantage of a price change as he would only lose his collateral and even if BTC is close to 0, it would be economically unreasonable.

Protection solutions

An arbitrator system is the strongest protection against the extortion attack.

Fee double spending could be detected later during the trade process and be used for local blacklisting.
Same for trade interruption.
For the fraud with taking advantage form price changes, a high collateral would be the easiest protection. Further a reputation system and local blacklists could be used. The fraud report list seems a too harsh measure but could be considered if that kind of fraud would become a plague (like it is in localBitcoin).

Bank transfer

Abuse cases

1. Offerer does not transfer the money

After timeout taker can request arbitration.

2. Taker received the money but claims he did not

After timeout offerer can request arbitration.

3. Offerer has transferred the money but taker did never received it due a bank tx problem

After timeout both can request arbitration.

4. Bank blocks the tx

After timeout both can request arbitration.

5. Bank tx last longer as the defined max. tx duration

After timeout taker can request arbitration.

6. Bank transfer chargeback

The offerer could initiate a bank chargeback after he has received the BTC from the payout tx. In that case the arbitration system does not help anymore.

7. Bank transfer chargeback due stolen bank account

The bank might initiate a bank chargeback in case that the offerer has used a stolen bank account. In that case the arbitration system does not help anymore.

Protection

Arbitrator system

In the cases1-5 a arbitrator system will be the best protection. The arbitrator needs to find out the truth and decide who will get the payment. The arbitrator system has its own complexity and risks and will be handled separately in an dedicated document.

For the case of bank chargeback and stolen bank accounts we need additional protection.

Bank chargeback

Contract

When signing the deposit tx the traders sign as well a contract with all the trade details and the bank account details. That contract could be used for legal help outside the scope of the exchange. That might be helpful in some countries, in others not. As there are considerable costs engaging an lawer, that option gets even more limited. But it might be an open option in some circumstances and the sole possibility could serve as fraud prevention.

Fraud reports

A system wide fraud report can be used to list a trader who has done a chargeback. The fraud report can be edited by arbitrators only and the entry need to be signed by the arbitrator. A description with proofs attached need to be linked to the accusation. That way a unjustified fraud report listing could be removed by other arbitrators and the arbitrator himself could lose his reputation or get banned and listed in the fraud report.

Arbitrators have a high collateral to lose if they behave incorrect. More details are discussed in the dedicated paper.

In case of a fraud report the privacy of the scammer gets unveiled. There will be published the message ID, the trading account ID and the bank details (including holder name).

We can only protect against repeated frauds. To prevent repeated attacks with new trading accounts we use a registration system with a moderate fee and a tx with the bank account attached. That way we get 2 scarce resources which get lost in case of fraud report listing:

- The registration fee, as a fraud report listed trading account will become worthless.
- The bank account associated with the fraud get listed in the fraud report, so it cannot be used for repeated frauds with a newly registered trading account.

Stolen bank accounts

The above all applies to that case as well, but additionally it can be more critical as there might be accusations against the deceived trader and he might prove that he is not involved with the crime.

Also there might be legal requirements that the trader need to verify the other peers identity before doing the trade (AML). The legal environment will vary in different countries, might depend on the trade volume and probably there will be no clear guidelines available.

So that is an open question if that is an issue at all.

Bank account verification

A trader could make a bank transfer with a small fee to an arbitrator. When the arbitrator has received that bank transfer and it matches the data from the registration then the arbitrator could sign a certificate of the verification. That verification certificate could be used for higher trust level in trades. It could be added as trading constraint (only do trades with verified peers) or be used for higher trade limits.

Identity verification

Similar like above the trader could get verified by an arbitrator. The verification process need to be standardized so all arbitrators verifications have the same quality and value.
A ID verification could be done out of the system e.g. via a skype video chat demonstrating a few ID documents.
Problematic with all the above solutions is that the privacy get leaked to the arbitrator. An arbitrator collecting that data might misuse that.
It depends on the number of arbitrators how critical that risk would be.
Alternatively for real life ID verification there could be used online IDs like Facebook/G+/Twitter accounts or other IDs derived from reputation systems like BTC OTC, web of trust, PGP,...

One assumption to the above is that the arbitrator system use a selection mechanism which makes misuse inside the arbitrator system hard. The selection mechanism could use the 30% of arbitrators with the highest reputation and use verifiable but unbiased selection algorithm (use message ID of the trader to derive the selected arbitrator).

The ID verification will be optional. We could let it open to the users to use those measurements or not.
All in all it is not sure if its really necessary and might not be included in the first release.
It needs more discussion as it touches critical privacy issues.

Payout transaction

The case that the taker does not release the BTC is already covered above and can be protected with the arbitration system.

Infrastructure

Messaging system

Needs more research on protection mechanisms for existing DHT systems and TomP2P.

Peer discovery

At startup the node need to find other peers.
There will be a hardcoded list of trusted nodes (master nodes, servers). Furthermore the client stores his actual list of connected nodes, so for the next startup he can try to reactivate those connections, thus making the master nodes unnecessary.
TODO: Check out how it is done in btc

Message flooding

A node is flooding messages to other nodes.
Peers who exceeds a traffic limit will be disconnected and their Message ID/IP locally blacklisted.

Sybil attacks for DHT storage

A group of sock puppets could control the storage of certain offers and then go offline simultaneously, so the stored offer disappears. The offer creator always keep a copy of his offer and periodically (e.g. every 10-60 min.) checks if the offer is still available in the network, if not he republish the offer.

Bitcoin library

Peer discovery

BitcoinJ is dependent on DNS seeding servers. Nodes can be added manually. That should be extended to a more resistant model. Save a list of last connected peers locally to reuse that at the next connection attempt.

Privacy loss between traders

With address reuse, or coin merge, trades can be connected. Every trade should operate with its dedicated addresses/keys and different keys should be used for deposit and for payout. Coin mixing is outside the scope of the exchange, the user has to take care of that at the funding and withdraw level.

Anonymity

The privacy needs to be protected outside of the system. The btc payments for funding or withdrawal can be mixed to protect the anonymity.
Maybe a mixing protocol could be added later, but that is out of scope yet.

Wallet protection
HD wallet should be used (bitcoinJ does not support that in the current stable release version, but it is on the way). Password protection and automated backups will be included.
2 factor authentication would be a nice to have feature, but has no high priority for the first version.

Internet network layer

For the first version there is no Tor implementation planned. Later we might add that to get more privacy.
One limitation of the exchange is that it will not be recommended to use in countries where Bitcoin is illegal, because the risk of undercover agents (privacy leak to the trading partner). That cannot be prevented and would be a high risk for the trader in such countries. So the attack case of a nation state attacking bitcoin in general (declaring it illegal) is out of the scope for the protection measurements as the whole exchange concept will fail for such a case.

Type of attackers

Economic motivated attacker

Attacker is peer user

- Limited financial resources
- Fees will have some effect

Attacker is organisation (competitor)

- Higher financial resources
- Fees might have a very limited effect
- Financial loss can be acceptable in the strategy as it might have positive effect outside the system (kick clients out and move them to a centralized exchange,...)
- Financial loss can be acceptable in the strategy as it might have positive effect in the future (market manipulation)

Political motivated attacker

Same like above but higher or near unlimited financial resources (nation state).

Other risks

Technical issues

- Software bugs
- Problems with updating the protocol as it needs consensus in the network (fee policy,...)
- Forks which undermine the fee model
- Other clients with different implementations or policies splitting or confusing the network

Unintended breach of contract (illness, accident, death,...)

- A trader might breach the trading contract due unintended reasons. The reputation system need to be tolerant to not discredit such traders. The arbitrator can unlock the funds in such cases.

Breach of contract due awkwardness

- Will be treated like bad intention -> arbitration needed.

Breach of contract due carelessness or laziness

- Will be treated like bad intention -> arbitration needed.

Legal attack on software developer

- We need to check the legal risks for the project developers.
- As long as we are not involved with monetarisation from the project there should be no legal consequences for the developers.

- The funding model need to be checked by a lawyer

Legal attack on users

- As long as Bitcoin itself is not illegal, the use of the software should be legal as well

Bank blocking bank accounts due the usage of the software

- It will be hard for banks to prove that a user used that software. But even if there are no reasons a Bank might block a bank account without justification.
- There would be a high false positive rate if the banks start to ban potential users, thus creating high collateral damage for their customer reputation.
- It will be recommended to not use the primary bank account but a dedicated one, thus a blocking will not hurt that much.