Setting up a GNU Taler Auditor

Version 1.0 from August 4, 2021

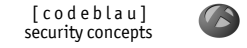
Code Blau GmbH
Klemkestr. 39
13409 Berlin

Contents

Abstract

Code Blau has setup an instance of the GNU Taler Auditor software for a test-environment. This is the log of the taken steps and the resulting audit report.

contact:	Code Blau GmbH
email:	contact@codeblau.de
fon:	+49.30.65004524
fax:	+49.30.55145804
address:	Klemkestr. 39
city:	13409 Berlin
url:	http://www.codeblau.de



1 Introduction

In June and July 2021, Code Blau has setup an instance of the GNU Taler Auditor system to audit the GNU Taler Exchange operated by Taler Systems SA, at `https://exchange.chf.taler.net`.

The preparation and setup was done by Özgür Kesim from Code Blau GmbH in cooperation with Christian Grothoff from Taler Systems SA.

2 Setup steps

We follow the instructions given in section 11, *GNU Taler Auditor Operator Manual* of the GNU Taler documentation

2.1 Preparation Debian 10

After installing Debian 10 on a server, we follow the instructions on 11.2.2. and prepare the package manager *apt* for the use of download-URL's provided by Taler Systems SA.

After a call to `apt update` we install the GNU Taler packages for the auditor and postgres with a call to `apt install taler-auditor postgres-11`.

2.2 Creating an offline signing-key

On a separate, isolated system, also with package `taler-auditor` installed, the offline key for signing denominations of the exchange is created:

```
root@offline# taler-config -s auditor -o BASE_URL -V https://bfh.auditor.codeblau.de
root@offline# taler-config -s exchange -o MASTER_PUBLIC_KEY -V DF0DFM8BRBAFCYGCF4E6KBZTXSJR19K0YJ0P2AQ739ZJ629HSZN0
root@offline# taler-auditor-offline setup
EWY86W5DG6QP80YKY72R4J0NQE7K9JN1DFNMFYN58DPAN1Z8MWXG
```

The highlighted result is the freshly generated public key that is than sent to the exchange operator over a secure communication channel.

The signing system is then taken offline.

2.3 Initial setup

The actual auditor system needs to be prepared, too:

```
root@auditor# taler-config -s auditor -o BASE_URL -V https://bfh.auditor.codeblau.de
```

2.4 Technical user accounts

We continue with section 11.2.3 to create technical users for various operational tasks:

ingress – maintains a network connection to the exchange and replicates the database of the exchange into the database `taler-ingress`.

talersync – has read-access to the replica of the exchange database and maintains a copy of that replica with a locally defined schema into the database `talersync`.

auditor – has read-access to the local copy of the exchange database and performs the actual evaluation/auditing of the contents.

```
root@auditor:/etc/taler-auditor# adduser --disabled-password auditor
root@auditor:/etc/taler-auditor# adduser --disabled-password ingress
root@auditor:/etc/taler-auditor# adduser --disabled-password talersync

root@auditor:/etc/taler-auditor# su - postgres
postgres@auditor:~$ createuser auditor
postgres@auditor:~$ createuser ingress
postgres@auditor:~$ createuser talersync
postgres@auditor:~$ createdb -O ingress taler-ingress
postgres@auditor:~$ createdb -O talersync taler-sync
postgres@auditor:~$ createdb -O auditor taler-auditor
```

2.5 Database creation

We then create the databases `taler-ingress` for the replication of the exchange database, `taler-sync` for a internal copy of the replica, both with approriate rights to the appropriate technical users.

```
root@auditor:~# su - ingress
ingress@auditor:~$ psql -d taler-ingress
taler-ingress=> GRANT SELECT ON ALL TABLES IN SCHEMA public TO talersync;

root@auditor:~# su - talersync
talersync@auditor:~$ psql -d taler-sync
taler-sync=> GRANT SELECT ON ALL TABLES IN SCHEMA public TO auditor;

auditor@auditor:~$ taler-config -s auditordb-postgres -o CONFIG -V postgres:///taler-auditor
auditor@auditor:~$ taler-auditor-dbinit
```

2.6 Setting up a tunnel to the exchange

We create a SSH keypair for the `ingress` user and send the public key to the exchange operator.

```
ingress@auditor:~$ cat .ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPYIhuWnJgSmiCMjoPX/NlkAAkMYfvAzgRMCuD2SGUQQ ingress@codeblau.de
```

After sending the public key to the exchange operator, we create a SSH-tunnel to the exchange and use the port-forwarding feature of ssh in order to access the postgres instance at the exchange.

```
ingress@auditor:~$ ssh egress@exchange.chf.taler.net  -L5555:localhost:5432
```

2.7 Exchange database replication

With the provided tunnel to the exchange, we can now setup the replication of the exchange database.

```
postgres@auditor:~$ psql -d taler-ingress
psql (13.3 (Debian 13.3-1), server 11.12 (Debian 11.12-0+deb10u1))
Type "help" for help.

taler-ingress=# CREATE SUBSCRIPTION bhf CONNECTION
                'dbname=taler-exchange host=localhost user=egress password=XXXXXXX port=5555'
                PUBLICATION codeblau;
NOTICE:  created replication slot "bhf" on publisher
CREATE SUBSCRIPTION
```

On success, this provides us with a replica of the exchange database:

```
taler-ingress=# \d
                                List of relations
 Schema |                            Name                             |   Type   |  Owner
--------+-------------------------------------------------------------+----------+---------
 public | aggregation_tracking                                        | table    | ingress
 public | aggregation_tracking_aggregation_serial_id_seq              | sequence | ingress
 public | auditor_denom_sigs                                          | table    | ingress
 public | auditor_denom_sigs_auditor_denom_serial_seq                 | sequence | ingress
 public | auditors                                                    | table    | ingress
 public | auditors_auditor_uuid_seq                                   | sequence | ingress
 public | denomination_revocations                                    | table    | ingress
 public | denomination_revocations_denom_revocations_serial_id_seq    | sequence | ingress
 public | denominations                                               | table    | ingress
 public | denominations_denominations_serial_seq                      | sequence | ingress
 public | deposits                                                    | table    | ingress
 public | deposits_deposit_serial_id_seq                              | sequence | ingress
 public | exchange_sign_keys                                          | table    | ingress
 public | exchange_sign_keys_esk_serial_seq                           | sequence | ingress
 public | known_coins                                                 | table    | ingress
 public | known_coins_known_coin_id_seq                               | sequence | ingress
 public | prewire                                                     | table    | ingress
 public | prewire_prewire_uuid_seq                                    | sequence | ingress
 public | recoup                                                      | table    | ingress
 public | recoup_recoup_uuid_seq                                      | sequence | ingress
 public | recoup_refresh                                              | table    | ingress
 public | recoup_refresh_recoup_refresh_uuid_seq                      | sequence | ingress
 public | refresh_commitments                                         | table    | ingress
 ...
(47 rows)
```

And data is actually arriving:

```
taler-ingress=# SELECT * from auditors;
 auditor_uuid |                          auditor_pub                           |  auditor_name  | ...
--------------+--------------------------------------------------------------------+----------------+ ...
            1 | \xce70a19091b2172ff0cc40f93a19a97db84704444d92b4905e476eadf0ededa2 | BFH Auditor    | ...
            2 | \x773c8370ad81af6403d3f1c5824815bb8f34caa16beb47faa5436caa87e8a73b | Code Blau GmbH | ...
(2 rows)
```

2.8 Copy of the Replica

After the creation of the replica, we need to grant access to the table.

```
ingress@auditor:~$ psql -d taler-ingress
psql (13.3 (Debian 13.3-1), server 11.12 (Debian 11.12-0+deb10u1))
Type "help" for help.

taler-ingress=> GRANT SELECT ON ALL TABLES IN SCHEMA public TO talersync;
GRANT
```

We can now create a controlled copy of the ingress database:

```
talersync@auditor:~$ taler-config -c .config/src.conf -s exchange -o DB -V "postgres"
talersync@auditor:~$ taler-config -c .config/src.conf -s exchangedb-postgres -o CONFIG -V "postgres:///taler-ingress"
talersync@auditor:~$ taler-config -c .config/dst.conf -s exchange -o DB -V "postgres"
talersync@auditor:~$ taler-config -c .config/dst.conf -s exchangedb-postgres -o CONFIG -V "postgres:///taler-sync"
talersync@auditor:~$ taler-exchange-dbinit -c .config/dst.conf
talersync@auditor:~$ taler-auditor-sync -s .config/src.conf -d .config/dst.conf -t
```

2.9 Exchange configuration

Following section 11.5.1, we setup the auditor with the necessary parameters provided by the exchange.

```
auditor@auditor:~$ taler-auditor-exchange -u https://chf.exchange.taler.net/
                                -m df0dfm8brbafcygcf4e6kbztxsjr19k0YJ0P2AQ739ZJ629HSZN0
auditor@auditor:~$ taler-config -s exchange -o master_public_key
                                -v df0dfm8brbafcygcf4e6kbztxsjr19k0YJ0P2AQ739ZJ629HSZN0
auditor@auditor:~$ taler-config -s exchange -o base_url -v https://exchange.chf.taler.net/
```

2.10 Signing and uploading denominations

We bring the signing system online again and prepare it for the signing operation with the same commands as in 2.9:

```
root@offline:~$ taler-auditor-exchange -u https://chf.exchange.taler.net/
                                -m df0dfm8brbafcygcf4e6kbztxsjr19k0YJ0P2AQ739ZJ629HSZN0
root@offline:~$ taler-config -s exchange -o master_public_key
                                -v df0dfm8brbafcygcf4e6kbztxsjr19k0YJ0P2AQ739ZJ629HSZN0
root@offline:~$ taler-config -s exchange -o base_url -v https://exchange.chf.taler.net/
```

We then download the denomination public keys from the exchange, sign

and upload them again:

```
root@offline:~$ taler-auditor-offline -L debug download > input.json
   Jun 22 17:13:49-425973 taler-auditor-offline-15095
        INFO Received keys from URL `https://exchange.chf.taler.net/keys' with status 200.
   Jun 22 17:13:49-559136 taler-auditor-offline-15095
        INFO Successfully downloaded exchange's keys
   Jun 22 17:13:49-559222 taler-auditor-offline-15095
        INFO Connecting to auditor at URL `https://auditor.chf.taler.net/service/' (0x55e00a1b8d20).
   Jun 22 17:13:49-562612 taler-auditor-offline-15095
        INFO Disconnecting from auditor at URL `https://auditor.chf.taler.net/service/' (0x55e00a1b8d20)
root@offline:~# taler-auditor-offline sign < input.json > output.json
root@offline:~# taler-auditor-offline upload < output.json
```

After these calls, the signing system is taken offline again. The auditor has now succesfully signed the denominations of the exchange, as can be verified with the data provided by the URL https://exchange.chf.taler.net/keys.

2.11 Generating an audit report

The setup is now complete for the auditor account to run the actual evaluation/auditing of the exchange:

```
auditor@auditor:~$ taler-auditor
```

This results in a report as pictured in appendix A.

A The produced auditor report

Taler Auditor Report

Code Blau GmbH

August 4, 2021

The report was generated by the auditors at the following times:

Auditor	Start	End
Aggregation	Sun Aug 01 13:17:38 2021	Sun Aug 01 13:17:38 2021
Coins	Sun Aug 01 13:17:39 2021	Sun Aug 01 13:17:39 2021
Deposits	Sun Aug 01 13:17:39 2021	Sun Aug 01 13:17:39 2021
Reserves	Sun Aug 01 13:17:39 2021	Sun Aug 01 13:17:39 2021
Wire	Sat Jun 26 14:52:40 2021	Sat Jun 26 14:52:42 2021

In that time, the auditors processed the following table ranges:
In that time, the wire auditor processed the following table ranges:

Account	Table	Start	End
exchange-account-1	Reserves Incoming	0	7
	Outgoing wire transfers	0	4

Table 2: Range of account data processed by the wire auditor.

Table	Start	End
Reserves Incoming	7	7
Reserves Out (withdraw)	14	14
Reserves Recoup	0	0
Reserves Close	0	0
Aggregation	4	4
Coin withdraw	14	14
Coin deposit	7	7
Coin melt	9	9
Coin refund	2	2
Coin recoup	0	0
Coin recoup refresh	0	0

Table 1: Serial number ranges of the tables processed by the audit.

1 Operations

The balance of the escrow account should be **CHF:29** (coins) plus **CHF:31** (reserves).
The active operational risk stands at **CHF:30.66**.
Loss (actualized risk from recoups) is **CHF:0**.
Recoups of non-revoked coins are at **CHF:0** (coins) plus **CHF:0** (reserves).

2 Income

This section analyzes the income of the exchange operator from fees.

Table 3: Fee revenue summary

Category	Amount
Withdraw fees	CHF:0
Deposit fees	CHF:0
Melt fees	CHF:0
Refund fees	CHF:0
Aggregation fees	CHF:0

3 Lag

This section analyzes lag, which can be due to some component being behind in executing transactions. This is usually either the exchange's aggregator, the bank's wire transfer logic, or the synchronization of databases between exchange and auditor. Significant lag may be indicative of fraud, while moderate lag is indicative that the systems may be too slow to handle the load. Small amounts of lag can occur in normal operation.

3.1 Deposit lag

The total amount the exchange currently lags behind in deposits is **CHF:0**.

Note that some lag is perfectly normal, as tiny amounts that are too small to be wired are deferred beyond the due date, hoping that additional transfers will push them above the tiny threshold. Below, we report *non-tiny* wire transfers that are lagging behind.

No non-tiny wire transfers that are lagging behind detected.

3.2 Reserve closure lag

The total amount the exchange currently lags behind in reserve closures is **CHF:0**.

Note that some minimal lag may be normal as transactions may be in-flight.

No closure transfers that are lagging behind detected.

3.3 Deposit confirmation lag

This section analyzes the lag, which is by how much the exchange's database reporting is behind in providing us with information about deposit confirmations. Merchants probabilistically report deposit confirmations to the auditor directly, so if the exchange is slow at synchronizing its database with the auditor, some deposit confirmations may be known at the auditor only directly. However, any delta not accounted for by database synchronization delays is an indicator of a malicious exchange (or online signing key compromise) and should be answered by revoking the exchange's online signing keys.

The total amount the exchange currently lags behind is **CHF:0** from a total number of **0** deposit confirmations.

Note that some lag is perfectly normal. Below, we report *all* deposit confirmations that are lagging behind.

No deposit confirmations that are lagging behind detected.

4 Major irregularities

This section describes the possible major irregularities that the auditor has checked, and lists all of the actual irregularities encountered in detail.

4.1 Emergencies

Emergencies are errors where more coins were deposited than the exchange remembers issuing. This usually means that the private keys of the exchange were compromised (stolen or factored) and subsequently used to sign coins off the books. If this happens, all coins of the respective denomination that the exchange has redeemed so far may have been created by the attacker, and the exchange would have to refund all of the outstanding coins from ordinary users. Thus, the **risk exposure** is the amount of coins in circulation for a particular denomination and the maximum loss for the exchange from this type of compromise.

4.1.1 Emergencies by counting coins

No emergencies detected by counting coins.

4.1.2 Emergencies by value deposited

Note that emergencies by value deposited can *also* arise if the exchange fails to properly detect double spending (or simply fails to properly account for the remaining balance of a coin). Thus, if issues are listed here **in combination with** arithmetic problems (Section 4.2) issues, then they may not be a definitive indicator that the exchange's private signing key was compromised.

No emergencies by value detected.

4.2 Arithmetic problems

This section lists cases where the arithmetic of the exchange involving amounts disagrees with the arithmetic of the auditor. Disagreements imply that either the exchange made a loss (sending out too much money), or screwed a customer (and thus at least needs to fix the financial damage done to the customer).

Note that the deltas only sum up the issues where $P \neq 0$ as only then we can tell if the problem lead to a profit or loss.

The **P** column is set to '1' if the arithmetic problem was be determined to be profitable for the exchange, '-1' if the problem resulted in a net loss for the exchange, and '0' if this is unclear or at least the gain/loss is not easily determined from the amounts and thus not included in the totals.

4.2.1 For aggregation

No arithmetic problems detected.

4.2.2 For coins

No arithmetic problems detected.

4.2.3 For reserves

No arithmetic problems detected.

4.3 Reserve withdrawals exceeding balance

This section highlights cases where more coins were withdrawn from a reserve than the reserve contained funding for. This is a serious compromise resulting in proportional financial losses to the exchange.

All withdrawals were covered by sufficient reserve funding.

4.4 Claimed outgoing wire transfer inconsistencies

This section is about the exchange's database containing a justification to make an outgoing wire transfer for an aggregated amount for various deposits. It is reported as an inconsistency if the amount claimed for the wire transfer does not match up the deposits aggregated. This is about a *claimed* outgoing wire transfer as violations do not imply that the wire transfer was actually made (as that is a separate check). Note that not making the wire transfer would be reported separately in Section 4.9.

All aggregations matched up.

4.5 Coin history inconsistencies

This section lists cases where the exchange made arithmetic errors found when looking at the transaction history of a coin. The totals sum up the differences in amounts that matter for profit/loss calculations of the exchange. When an exchange merely shifted money from customers to merchants (or vice versa) without any effects on its own balance, those entries are excluded from the total.

All coin histories were unproblematic.

4.6 Operations with bad signatures

This section lists operations that the exchange performed, but for which the signatures provided are invalid. Hence the operations were invalid and the amount involved should be considered lost.

4.6.1 For aggregation

All signatures were valid.

4.6.2 For coins

All signatures were valid.

4.6.3 For reserves

The key given is always the key for which the signature verification step failed. This is the reserve public key for "withdraw" operations, the coin public key for "recoup" operations, and the master public key for "recoup-master" operations (where the master's signature on the revocation is invalid).

All signatures were valid.

4.7 Actual incoming wire transfers

This section highlights cases where the exchange's record about incoming wire transfers does not match with that of the bank.

All incoming wire transfer amounts and subjects matched up.

4.8 Missattributed incoming wire transfers

This section lists cases where the sender account record of an incoming wire transfer differs between the exchange and the bank. This will cause funds to be sent to the wrong account when the reserve is closed and the remaining balance is refunded to the original account.

All incoming wire transfer sender accounts matched up.

4.9 Actual outgoing wire transfers

This section highlights cases where the exchange misbehaved with respect to outgoing wire transfers.

All outgoing wire transfers matched up.

5 Minor irregularities

5.1 Denominations without auditor signature

This section highlights denomination keys that lack a proper signature from the `taler-auditor-offline` tool. This may be legitimate, say in case where the auditor's involvement in the exchange business is ending and a new auditor is responsible for future denominations. So this must be read with a keen eye on the business situation.

All denominations officially audited by this auditor.

5.2 Incorrect reserve balance summary in database

This section highlights cases where the reserve balance summary in the database does not match the calculations made by the auditor. Deltas may indicate a corrupt database, but do not necessarily translate into a financial loss (yet).

All balances matched up.

5.3 Wire table issues

This section describes issues found by the wire auditor that do not have a clear financial impact.

No wire row inconsistencies found.

5.4 Outgoing wire transfer subject issues

This section describes issues found by the wire auditor that relate to outgoing wire transfers subjects being duplicated.

No wire format inconsistencies found.

5.5 Wire fee structure inconsistencies

This section lists cases where the exchange's database may be ambiguous with respect to what wire fee it charges at what time.

No wire fee timing issues detected.

5.6 Other issues

This section describes issues found that do not have a clear financial impact.

5.6.1 For aggregation

No row inconsistencies found.

5.6.2 For coins

No row inconsistencies found.

5.6.3 For reserves

No row inconsistencies found.

6 Delays and timing

This section describes issues that are likely caused simply by some job process of the exchange not running properly or not having caught up with the work load yet.

6.1 Delayed closure of reserves

This section describes cases where the exchange did not close a reserve and wire back the remaining funds when the reserve expired.

All expired reserves were closed.

6.2 Hanging refresh operations

This section describes cases where the exchange booked a coin as spent from `/refresh/melt` but where the wallet did not yet complete `/refresh/reveal`. This may happen even if the exchange is correct.

All melted coins were refreshed.

6.3 Denomination key invalid at time of withdrawal

This section lists cases where a denomination key was not valid for withdrawal at the time when the exchange claims to have signed a coin with it. This would be irregular, but has no obvious financial implications.

All denomination keys were valid at the time of withdrawals.

6.4 Wire transfer timestamp issues

This section lists issues with wire transfers related to timestamps.

No timestamp issues detected.