Tezos Foundation Comments on the Least Authority Protocol Security Audit Report

DRAFT: 26 February 2019

Compiled/Maintained by Ryan Lackey (Tezos Foundation) <ryan.lackey@tezos.com>
Answers from Nomadic Labs and others

Overview

The Tezos Foundation engaged Least Authority to perform an independent third-party audit of the Tezos project codebase in 2018. We are pleased with both the process and the result, and are appreciative of the depth and quality of review performed. Based on the report provided to us, we have the following response and comments on the report and our remediation efforts.

This response addresses the general security audit report.

Findings:

1) Language Selection: We agree that OCaml has many technical advantages, particularly for security-critical domains such as cryptocurrency, and that its less widespread adoption is a concern. We have undertaken many efforts to increase both our visibility within the OCaml community and the number of developers and auditors familiar with OCaml. We also intend to continue to engage with auditors and the security community, particularly in the more novel and/or security-critical aspects of the system.
2) Code Quality: In general, we have been working on these improvement areas since the betanet release (June 2018) as part of the general lifecycle of software, now that the team size has increased. We will periodically refactor code to make it more readable and maintainable. Increasing test coverage is a particular area of effort. Documentation is recognized as an area which needs to be expanded, and we've engaged multiple teams to improve the available documentation and examples both in the source code and externally.

Response to issues:

Overall, identified issues were primarily denial-of-service weak points of varying degrees of severity, and we have addressed these through updates to the codebase. Due to network protocol updates, the Tezos community is using the updated software.

Issue A: Peer authentication vulnerable to replay attacks.

Following the potential denial of service issue found by Least Authority about bogus handshake, the negotiation code has been changed, thereby eliminating this as a potential attack.

Particularly, a fix to the predictable nonce was committed quickly, and random nonce is generated for each connection attempt.

We discussed the idea of ephemeral session key, but it didn't seem useful in our context where the future secrecy of the exchanges is not critical.

A very simplified view of the p2p handshake protocol is described by the following pseudocode. Potential errors in this protocol could be caused by the simplification process, please also refer to the authenticate method in the actual code.

```
authenticate(id, fd) {
    /* random seed for each connection attempt */
    local_nonce_seed = random_nonce ();

    /* prepare a nonce generated from public key and a fresh nonce */
    sent_msg = { id.pk; local_nonce_seed }

    /* sends this nonce */
    socket_send (fd, sent_msg)

    /* receives the message from peer */
    recv_msg = socket_recv(fd);

    /* generates nonce from the received and generated messages */
    local_nonce, remote_nonce = generate_nonces(sent_msg, recv_msg);

    /* stripped: checks identity */
    /* stripped: check proof of work difficulty */

    channel_key = dh(id.sk, );
    box = (channel_key, local_nonce, remote_nonce);
    return (fd, box)
}

send(fd, box, msg) {
    c = enc(box.remote_nonce, box.channel_key, msg);
    socket_send(fd, c);
}

recv(fd, box, peer) {
    c = socket_recv(fd);
```

```
    msg = decrypt(box.local_nonce, box.channel_key, c);
    return msg;
}
```

Furthermore, this handshake protocol was formally verified to be correct against known attacks with the proverif tool, in a joint work with Bruno Blanchet (Prosecco team).

Issue B: Misauthenticated peers can replay message packets

We have corrected the handshake protocol, eliminating this potential denial of service attack vector.

Issue C: Proof of work is weak mechanism to authenticate a peer.

This potential denial of service is addressed by increasing the level of difficulty as warranted by network conditions. We are planning to replace the existing algorithm with a more cpu+ram difficult challenge (such as scrypt) on the next update of the p2p layer.

We may investigate other forms of DoS resistance for peer connections at various layers of the system. Requiring proof of stake for connections presents some usability issues, but this is an area of active exploration.

Issue D: Tezos Server is Vulnerable to DNS-Rebinding

This issue is remediated by changes to the Tezos RPC server. While the usual deployment model of tezos nodes is isolated and thus protected from this kind of attack, it is a concern with systems used in development or by some end users. We implemented host header/cors checks and are investigating options for local RPC authentication.

Suggestion A: Explicitly disallow cross origin RPC access

While this issue is not a current problem (due to the other identified factors preventing exploitation), we will explicitly protect against it to be defensive in case of future changes in the codebase. We are preparing a local RPC pairing to explicitly allow RPC only with authorized clients and html5 local applications.

Suggestion B: Use OS system calls for random number generation

This issue has been addressed by switching to HACL. We will make a note of this issue as something to check when using random numbers generally.

Areas for Further Discussion

1) Fuzzing: We are discussing various fuzzing and testing strategies and potential implementation changes to make ongoing automated security analysis more feasible.
2) Tezos Self-Compiler: We agree that the system is biased toward flexibility in this area, and that security analysis of proposed protocol changes must be thorough. The suggestion of sponsoring an “underhanded OCaml competition” is very interesting and may be something we do in the near future, both for security reasons and to increase the visibility of OCaml in the broader community.
3) Secure handshakes: We are familiar with these handshake protocols and improving handshake procedures globally in the Tezos codebase is an area of interest.

Conclusion

We appreciated the opportunity to work with Least Authority on an independent security review of the Tezos software, and feel the process has resulted in identified issues which have been remediated, as well as better understanding of areas for future development effort and particular security concern within the system. We look forward to future security reviews by third parties as the system evolves.