1 Código Network Threat model and Security Assumptions

To be able to tackle the complex problems that arise in the design of decentralized applications, the categories of possible adversaries must be identified. This project has many commonalities with the OpenBazaar e-commerce platform [1]. Código network has a more narrow focus on peer-to-peer (P2P) distribution of firmware for embedded devices. The combination of the Código's narrow focus and the lack of monetary transactions makes it a much less vulnerable system. This comes from the fact that agents using the system, to receive firmware updates have no immediate financial incentives should they deviate from a proper execution of the protocol (This does not apply to OpenBazaar). Contrary, in the design of the network, the users need to protect themselves from malicious developers that want to infect embedded devices with malicious software.

A malicious developer is an agent in the network that will upload a "firmware" to the network knowing that the firmware is either not working or is exploitable in some way. The benefit of polluting the network with unusable software is to grief the users of the system and cause denial of service. Contrary polluting the network with exploitable firmware comes with many financial advantages as described in the introduction. A subtle observation is that in both cases the reward the malicious agent is either non-monetary or if it is monetary it comes from a source outside of the network. As a result, it is not possible to model the users of the system as rational and rely on a game-theoretic approach to protect the system against malicious developers. Another subtle observation, is that the adversary is a favorable position in decentralized environments. In the decentralized setting the attacker can generate multiple identities and use them collectively to attack the system. Additionally, the pseudo-anonymity that comes in every decentralized applications makes it harder for users to trust other agents of the network.

To protect Código Network for the aforementioned adversaries it assumed that the following hold true:

1. Cryptographic primitives such as Digital Signatures, RSA encryption, AES, SHA-256, SHA-3 are secure

2. The frameworks used (IPFS, Ethereum) are theoretically secured and implemented correctly

3. The libraries used to develop the system are secure (Namely: ...)

[1] OpenBazaar is decentralized marketplace similar to Ebay, that uses blockchain and cryptocurrencies to perform transactions.

4. Nodes downloading a firmware are able to detect malicious firmware using hardware virtualization tools [Cite Tom Spink Paper] or malware classfication software [Cite David Aspinall].

5. The embedded device used is secure.

An importance observation should be made on the 4th security assumption. This assumption is both theoretically reasonable and solves an important issue for Código network. In particular with this assumption the system is able to address the content curation issue that arises in decentralized systems. With this assumption nodes we assume that nodes are able to distinguish on their own whether a firmware is malicious or honest.

2 Modeling Trust in Decentralized Environments

Trust in decentralized networks is an open research area that lays outside the primary scope of this project. However, as it plays a crucial role in the system, it would be a big omission not to address it, even partially. To do so various solutions, that are widely used in practice and have some theoretical background have been considered. Before delving into the available solutions, the trust-sensitive operations of Código Network must be identified.

A requirement specific to Código Network is that, naturally, the manufacturer of the IoT system is a trusted party. As a result, as long as the manufacturer is actively maintaining the system, there exists a firmware that is infinitely trusted by nodes. Nodes are still free to choose any firmware they want but they acknowledge the risk involved in such action. When the manufacturer stops maintaining the firmware, the users of the system will be incentivized to seek alternative firmware developed by the community. This is the point when nodes lose their infinitely trusted firmware and need shift their trust to an unknown developer.

A naive solution to the problem is to use a star based rating system for each developer. This solution is inadequate as a malicious developer could easily create multiple accounts (Sybil Attack) and rate himself with the highest rating thus making the rating system useless. Various heuristics, such as not counting the votes of new users, could be employed to alleviate Sybil Attack. Such heuristics lack are easy to deploy but hard if possible at all to optimize with respect to the user experience.

In general, people tend to trust less people who are not committed in their actions. This the problem a traveling salesman faces in her quest to sell her merchandise. The customers trust her less, because if she sells them a bad quality product then she has practically nothing to lose. A common workaround for this problem is providing the system with a form of commitment (Note that commitment here does not mean a cryptographic commitment scheme). The commitment can come in numerous ways such as

This paragraph may be moved to security section

providing the system with valid computational PoW or by burning/donating some of his wealth. This kind of system, is also used in Código Network with dual benefit. Firstly, it increases the trust of user in the developers and also increases the cost of Sybil attacks. Additionally, to make the system more resistant to Sybil Attack, the cost of uploading a new firmware exponentially increases for uploading new firmware within the same day.

From the literature the following approaches were identified for addressing trust issues in decentralized systems.

1. Web of trust: The Web of Trust is an old idea that is currently used in the PGP project. Intuitively, its a mechanism to generate a graph with vertices's representing users and edges representing the existence of trust between to users. Then trust is quantified as the number of edges required to reach from one user to an other. With this approach trust is relative. In the Código Network setting this means that different users will trust different firmware. As a result, trust should be calculated every time a node queries the smart-contract.

2. Stake as trust: Stake as trust is novel idea that on an intuitive level shares some commonalities with the Proof of Stake concept. The idea is based on the fact that a user who allocates a lot of stake as an initial investment to upload her firmware into the network is incentivized to have developed a trustworthy firmware. An advantage of this approach is that the trust users allocate to a firmware is uniquely defined for all users.

3. Trust is Risk [Cite relevant paper]:

Add description