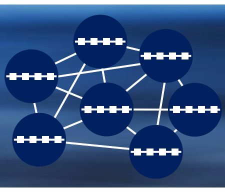

Smart Contract Security – Expect and Deal with the Attack

Blockchain as disruptive innovation, a flood of Initial Coin Offerings, aka ICOs, and corresponding software implementations called Smart Contracts – those intrigued by technological progress and societal advancements have a lot to digest these days. Nevertheless, the traditional laws and dynamics of software development still apply: Any source code is prone to errors. In a world where Smart Contracts are executed by a network of mutually distrusting nodes without central authoritative entity, potentially handling large amounts of value units, transformable into "real" money, security plays a pivotal role. However, security matters are often neglected – with dire consequences, as seen in cases like DAO and Parity Wallet Breach. This whitepaper provides some insight and recommendations how to expect and deal with attack vectors.

Keywords: Blockchain | Smart Contracts | Security Review | Penetration Testing | DAO | Parity | Ethereum | Solidity | ICO – Initial Coin Offering | Token Sale

Smart Contracts & Decentralized Apps

Blockchain technology has come a long way since Bitcoin and its simple scripting language. Today advanced architecture and functionality allows for Smart Contracts as the Blockchain-based back end of so called decentralized applications (DApps).

A lot of emerging platforms for decentralized applications like the global players Ethereum and Hyperledger Fabric as well as others like Lisk, Emercoin, etc. promise the next digital revolution, expanding the benefits of Blockchain far beyond cryptocurrencies.

Relevance of Smart Contract Security

However, the vast differences between conventional and Blockchain development have to be taken into account. Just following intuition instead of best practices may only lead to unwanted loopholes.

The well-known DAO (Decentralized Autonomous Organisation) hack can be considered a prime example for overlooked side effects in Smart Contracts. The Ethereum-based DAO began as the world's largest crowd-funding project, yet quickly turned into one of the most memorable failures when the exploit of a recursive call pattern resulted in sudden removal of DAO funds. The following hard fork of the Ethereum Blockchain to revert the damage of over $50 million[(1)] only fueled controversy.

Just as severe with a damage of $30 million[(2)] is the more recent Parity wallet breach related to Ethereum in July 2017. A careless use of a delegatecall command in a critical library enabled the attacker to gain ownership of a multisignature wallet and to redirect all of its funds.

Even though developers are usually quick to react when it comes to minimizing damage, prevention is far more important while working on an immutable ledger where "code is law". Navigating through this complex world of decentralized consensus, cryptography, transparent code and possible hard forks may prove difficult at first and user-friendliness is still not quite in reach. And with much of its potential yet to be revealed, adapting to the Blockchain environment is an ongoing process.

So even though the underlying technology can be considered secure, and the correct operation of the biggest Blockchains has yet to be challenged, the development process of DApps is still prone to human error and calls for its own security approach.

A Security Assurance Approach

A rough blueprint for a security assurance approach is sketched in the following. This is intended to introduce the high-level scheme required for drawing and applying a comprehensive, detailed picture, which is mandatorily applied in practical, real-world implementation scenarios.

(1) http://www.zeit.de/digital/internet/2016-06/the-dao-blockchain-ether-hack
(2) https://www.coindesk.com/30-million-ether-reported-stolen-parity-wallet-breach

Technology stack analysis. Before starting with a development based on a Blockchain, it is vital to accept the fact that Smart Contract code is not an isolated piece of technology. There is a lot more happening behind the scenes and many layers below the actual code have to be considered. Like which Blockchain to use. Every Blockchain consists of many nodes and it has to be evaluated who these are. It is certainly not possible to get this kind of information in detail for every node but an overview should be obtained. Conducting such research helps finding the most suitable Blockchain and allows an estimation how big the danger of a 51%-attack is.

Penetration testing. As soon as the solution is initially available, even as a prototype, start penetration testing. Make it a habit – this is not something that has to be just done once, it is an ongoing process. Penetration testing focuses on testing the whole system and checking for security issues. The goals which one would like to achieve with penetration testing are the identification of vulnerabilities, the identification of potential errors resulting of misuse and the increase of security on technical and organizational level. So even if a public permissionless Blockchain is used, where it is not possible to test all clients involved in the Blockchain, penetration testing is in fact very helpful. Potential issues with the deployed DApp and the used Blockchain node can occur and early intervention can prevent possible damage.

Source code review. As the examples shown above indicate, it is necessary to review the source code of DApps thoroughly before deploying them on the Blockchain. As one of the main Blockchain characteristics, immutability does not only ensure data integrity, it also prevents code published on the ledger from being modified in any way, making further updates to a contract impossible. This and the accessibility of code for everyone (in case of a public ledger) can be a dangerous combination as attackers may quickly find and exploit vulnerabilities in the irrevocably published code. Therefore **esatus** AG is convinced that source code review – as it is important for any application – is highly important for all Blockchain-based applications. More true than ever: Better be safe than sorry!

Consider any Attack Vectors

To embrace a security attitude, thinking like an attacker is crucial – always expect the worst malicious intent of highly skilled renegades. Even if the code of the Smart Contract is not faulty, the nowadays classic problem of a Denial of Service attack can be applied to a Smart Contract. However, this means spamming the whole Blockchain it is operated on. Still, it can be done. Additionally, consider "dumb" users: As long-term practical experience shows, anything that can be done with your software – including Smart Contracts – will be done. To go beyond that, not-so-skilled users may be attacked by traditional phishing attacks, meaning that their private keys get compromised. This causes real damage to the user and reputational damage to the Smart Contract, even if it is perfectly secure. All in all, diving into attack vectors requires acceptance of real bugs and design flaws allowing undesired functions, as well as general technology risks and user errors. Furthermore, a Smart Contract developer always has to remember that own faults and those of he relies on have to be taken into account.

Discussion and Outlook

While the disruptive potential of Blockchain and distributed ledger technology (DLT) offer a lot of chances and opportunities, breaking new ground still comes with risks. Growing and ever-changing environments – like with decentralized apps – also attract people with malicious intent. Constant review of code and proper analysis of possible attack vectors become a necessity – and one of the most important aspects of keeping up with Blockchain evolution.

This whitepaper provided a brief introduction into the challenging realm of Smart Contract Security. There is obviously much more to it than what can be transported in two pages. In case you need your project security-reviewed: **esatus** AG will happily get involved.

Stay tuned for useful upcoming esatus Academy publications, diving deeper into these Blockchain topics:

- Smart Contract Development – Best Practice
- The Quantum Computing Risk
- Self-Sovereign Identity meets Hacker Ethic

esatus AG is a medium-sized IT consulting company. True to its vision "Enforcing Information Security" **esatus** AG is the qualified, experienced and flexible partner for any information security project. For its demanding customers, **esatus** AG implements optimal, tailor-made solutions addressing challenges in Identity & Access Governance (IAG), IT Security, and Governance, Risk und Compliance (GRC).

More information on "Smart Contract Security – Expect and Deal with the Attack" of **esatus** AG is accessible under: https://esatus.com.