The Risks of the Blockchain
A Review on Current Vulnerabilities and Attacks

Lukas König*, Stefan Unger, Peter Kieseberg, and Simon Tjoa
Josef Ressel Center BLOCKCHAINS
St. Pölten University of Applied Sciences, Lower Austria, St. Pölten 3100 Austria
{is191836, is191807, peter.kieseberg, simon.tjoa}@fhstp.ac.at

Abstract

Although the first hype of blockchains is over, subject matter experts are still convinced, that this technology has potential to enable more groundbreaking innovations in multiple business domains. However, to develop the full potential of this emerging technology, it is necessary to consider and address the associated risks. This survey aims at supporting researchers and practitioners to design, implement and improve their blockchain security and resilience by providing a concise overview on the subject. In this article, we describe 24 risks, which we structured into the four domains "Blockchain Structure Vulnerabilities", "Attacks on the Consensus Mechanism", "Application Oriented Attacks" and "Attacks on the Peer-to-Peer System". For each entry, we outline a precise description of the vulnerability or attack, the complexity and its prerequisites.

Keywords: Blockchain, Vulnerabilities, Risks, Security

1 Introduction

The blockchain is associated with the attribute of being one of the most disruptive technologies of our time. Thus, it is no big surprise, that in the recent past the interest in blockchain and distributed ledger technologies has significantly increased in a number of business areas. Starting from the financial sector, especially in the area of crypto-currencies (e.g. Bitcoin, Ethereum), more and more sectors discovered the advantages and opportunities arising from this emerging technology over time.

The beginning of the modern blockchain technology was heavily driven by Bitcoin, which led to many misunderstandings and myths spread across both, the general populace and professionals alike. It is often believed and propagated that blockchains are the technological solution capable of solving all our future problems, while ensuring security and maintaining privacy [4].

Obviously this is a strong overstatement about what this technology is actually capable of doing. Like any other development in IT, blockchain systems face certain security threats and problems. As it appeared to be a trend to use blockchains, many companies started using blockchains without fully grasping their capabilities, limitations and risks. However, especially information security and assurance are a critical components to ensure compliance to laws and regulations.

For that reason, in this article we focus on the following central research question: *What are the threats, vulnerabilities and attack surfaces of modern blockchain systems?* In order to provide an answer to this question and identify blockchain risks, we surveyed relevant literature using an extensive search in widely-used online libraries (e.g. ACM digital library, IEEE Xplore) and academic search engines (e.g. Google Scholar). For relevant attacks or vulnerabilities, we selected publications to further describe the

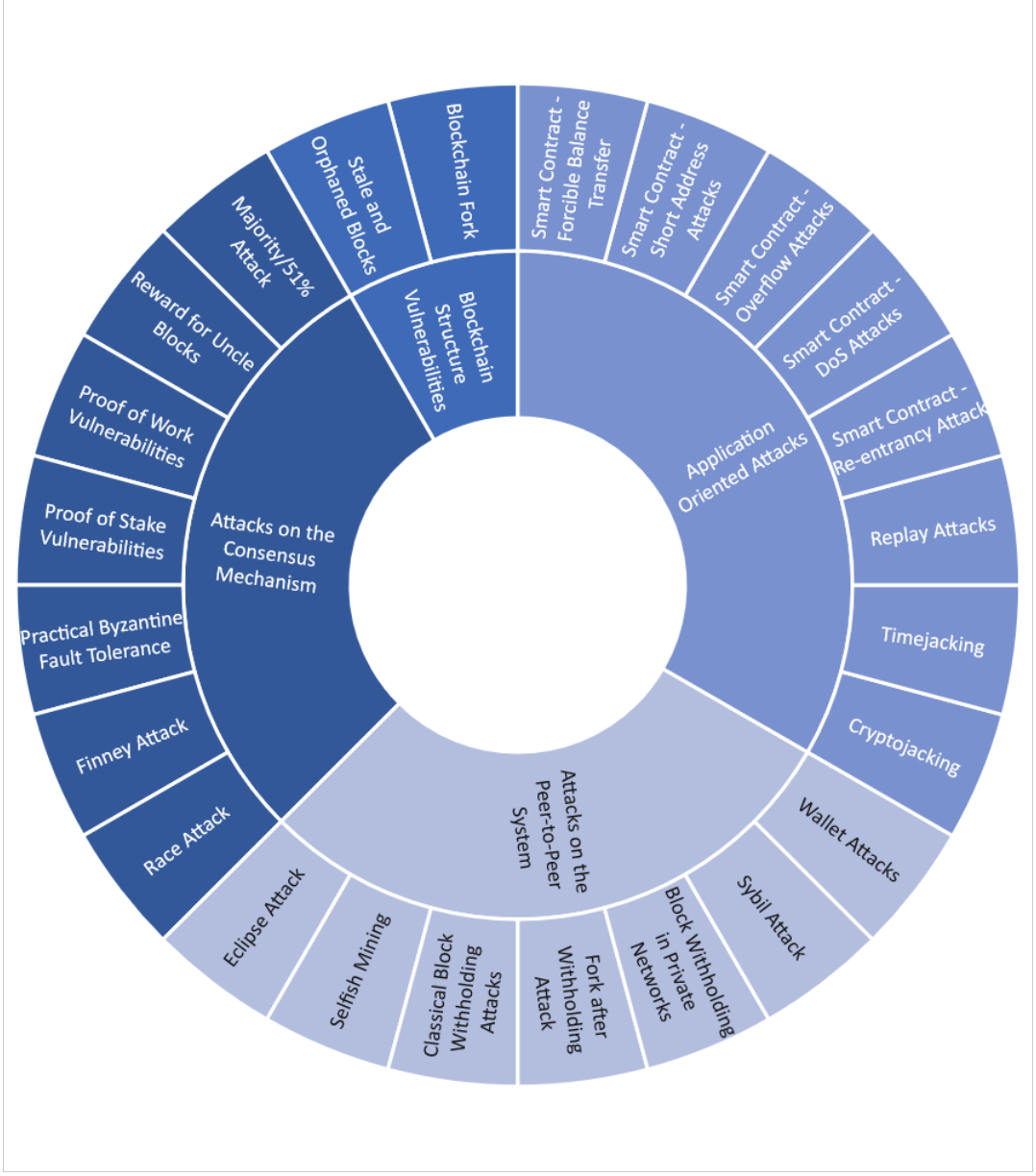
Journal of Internet Services and Information Security (JISIS), volume: 10, number: 3, pp. 110-127
DOI: *10.22667/JISIS.2020.08.31.110*

*Corresponding author: Institute of IT Security Research, St. Pölten University of Applied Sciences, Lower Austria, St. Pölten, 3100, Austria, Tel: +43-680-2382-190

issue. Other surveys with a similar research direction by Li et al. [22], Saad et al. [30] and Dasgupta et al. [8] show profound work already. However, each has its own focus point. For example, in [30] the authors explain different attacks and threats affecting vulnerabilities and weaknesses, [22] shows technical problems and execution processes, [8] states a number of challenges and issues that need to be overcome. *The main contribution of this article is the comprehensive review and analysis on current vulnerabilities and threats of blockchains (see Fig. 1) and a set of prerequisites for each to deepen the understanding even further.*

The remainder of this article is as follows: Section 2 outlines vulnerabilities, which arise from the blockchain structure. Section 3 highlights attacks, which are related to the consensus mechanism. Section 4 is dedicated to application-orient attacks. Section 5 presents attacks on peer-to-peer systems. In Section 6 we summarize and discuss the presented results, before we conclude the article in Section 7.

Figure 1: Overview of the different attacks covered in this article



2 Blockchain Structure Vulnerabilities

This section presents one of the most fundamental risks that come with blockchain technology. Although the risk can vary by different implementations, the herein problems are universal and a byproduct of the technology and its operating principles.

2.1 Blockchain Forks

A Blockchain fork constitutes a split of the current chain into different parts. The resulting forks all trace back to the so called *Genesis Block*, which indicates the last common block. There are many reasons for performing forks of a blockchain, such as an upgrade that is spread over the network and adopted by nodes at varying speeds, or malfunction.

While most forking attempts are not malicious, Blockchain forks represent an inconsistent state that can be exploited by adversaries to cause confusion, fraudulent transactions, and distrust within network.

The *Fork Problem* describes a situation, where nodes of a network can reach different consensus leading to a split of the chain. When a fork happens, this opens a window of opportunity for a variety of malicious behavior or attacks as long as its state is not consistent.

Probably one of the most prominent forks took place after the *DAO Hack* of Ethereum, which yielded more than a third of the total amount of the cryptocurrency. In this case Ethereum used a hard fork as a rollback measure to regain the money [23], [30], [34].

There are two types of forks, Hard Forks and Soft Forks. A **hard fork** is what we generally understand as fork. One blockchain separates and becomes incompatible with the other strain, leading to two completely separate new blockchains. This happens when the old blocks, thinking that their validation is the correct one, can't come to an agreement with the new blocks. Even if they get overruled by the new blocks during consensus, they will maintain the old chain with the previous consensus, even if outperformed by the new blocks. A Hard Fork becomes reality when node verification of the older version is more restrictive than the new system [23], [30], [33], [34].

A **soft fork** happens in a similar way, but instead of having the old blocks forcing a split of the chain, they remain on the same chain. The two different blocks are still incompatible to each other, but the computation power of the new blocks is much higher and therefore the old blocks get overruled and their validation discarded. In other words, the blockchain is forked, but in the end the nodes focus on just one strain again. If a soft fork happens, old nodes can still be gradually updated, until all nodes eventually reach the same state again. Soft forks happen if the newer node requirements are more restrictive than the old ones [23], [30].

Prerequisites: To allow forks to happen in regular blockchain operations it is most likely required that the blockchain forges new blocks in a probabilistic manner where multiple nodes can bring forth a valid block at the same time, resulting in a conflicting state. Systems where there is just a single entity to produce and append new blocks will not suffer from the forking problem under normal circumstances. Furthermore, systems using different methods for block arrangements and storage like a DAG (Directed Acyclic Graph) can not really suffer from the forking problem, as there is no single chain.

2.2 Stale and Orphaned Blocks

Stale and Orphan Blocks are validated blocks that end up outside of the main blockchain. They usually are a phenomenon of public blockchains, where miners try to outperform others in generating new blocks.

Stale blocks are the result of a race condition, which happens when blocks are created by two miners at once. As only one of them can be added to the chain, the other one ends as a stale extension. They

can be the product of malfunction or malicious behavior as well. A stale block is usually connected to the main chain directly.

An **orphaned block** has its parent in an already stale block and therefore has no direct next block connection to the main blockchain. It is entirely possible that an orphaned block started as part of the regular blockchain, but consensus pushed its parent out to be stale, making the block itself an orphaned block. Computation power spent on blocks that get stale or orphaned is essentially lost and usually not rewarded [15], [30].

Prerequisites: Orphaned blocks require a preliminary stale block, which they can append to. Stale blocks are the result of a probabilistic race between miners to be the one to append the block. This means that it only applies to systems having a single chain to which multiple parties are able to append blocks to. Stale blocks can be the product of malfunction or malicious behaviour as well.

3 Attacks on the Consensus Mechanism

A core component of every blockchain is the consensus mechanism. It is responsible for the decision if and what blocks are added to the blockchain and therefore for the direction the chain. Especially in public permissionless blockchains, where no form of centralised authority for checks and balances exists, the consensus mechanism is portrayed interchangeable with ”trust”. If a system relies on one component, the security of this component gets increasingly important. This section will therefore provide detailed insights on this essential matter.

3.1 51%-Attack

The *51%-Attack*, constitutes an attack on the consensus algorithm of a blockchain application. The adversary attempts to take over the network by controlling the majority of nodes, or more than 50% of the total computation power in the network. This unlocks the ability to overthrow the consensus algorithm, reject otherwise valid blocks from being added to the chain, or add malicious content. It is also known as *consensus hijacking*.

The specific characteristics of such an attack highly depend on the applied consensus algorithm and type of blockchain. The majority takeover of a blockchain network, even if just for a short period, could be used by an attacker to achieve more sophisticated goals and enables a variety of other attacks, such as *double spending* 3.4. The 51%-Attack can pose a critical threat to integrity and availability. [16], [1], [18] [12], [8], [23].

In permissioned blockchains, the threat situation with regards to 51%-Attacks is different [9], as there consists an increased level of trust between the members based on the authentication hurdle. With a central authority granting permissions, this type of attack is more realistic to be an *insider threat*, where an administrative entity could take over and control the blockchain. Others see this as a prime example of the vulnerable state of blockchains [18].

The main threat of 51%-Attacks stems from public, permissionless blockchains that use Proof-of-X consensus mechanisms [6], as there is no regulating or controlling authority and each adversary is unknown to each other.

[32] state that a 51%-Attack might be possible with less than that amount of computation power. It is possible to take other nodes out of action by disturbing the transmission of the current state of the blockchain. Therefore the power of these uninformed nodes would be lost and the overall cost for the attack decreases by partitioning the network.

Nonetheless, there are possible measures to mitigate majority attacks [1]. For example, if the blockchain network is based on a Proof-of-Work algorithm, it is possible to arbitrarily make the compu-

tation step harder, so an attacker would need exorbitant amounts of computation power to be successful [23].

However, this results in increased power consumption for the entire blockchain. One way to further mitigate such an attack is by spreading the adoption of blockchain technology. Having a large amount of nodes and miners increases the threshold of computation power required to possess more than 50%.

The author in [18] argues, that majority attacks are harder to achieve in Proof-of-Stake networks, because their validation and consensus operating principles. It is however still a viable option and should not be discarded entirely. One way of mitigating 51%-Attacks is a proposed Proof-of-Work that is running in two phases [6].

Prerequisites: The prerequisites of a 51%-Attack vary based on the affected blockchain structure and the used consensus algorithms and protocols:

- Proof of Work: more than 51% of the raw computational power of the network.
- Proof of Stake: more than 51% of the committed stake.
- Practical Byzantine Fault Tolerance: more than 33% of all replicas in theory to influcence the network, in practice it is sufficient to take over the single primary node.

3.2 Reward for Uncle Blocks

Rewards for uncle blocks refers to a vulnerability of the reward mechanism of Ethereum that incited to selfish mining [6]. Ethereum has an additional reward feature on top of regular block rewards, called uncle and nephew rewards.

An **uncle block** is a stale block outside of the main chain, which directly references to a regular block in the chain. In other words, stale blocks of the first order, closest to the main chain, can become uncle blocks. If a future regular block down the line starts referencing this stale uncle block, the new block is then called a **nephew block** and the creator of the uncle block gets the *uncle reward*. The rewarded amount is set by the distance between uncle and nephew. An uncle can gain up to $\frac{7}{8}$ of the amount of a regular block reward, nephews will always gain $\frac{1}{32}$ [27].

As uncle blocks are unique to Ethereum, there is no threat to other blockchain solutions not using this reward system. On Bitcoin, there is no reward mechanism for stale blocks and the stale block rate is at *0.41%*. Ethereum on the other hand with the uncle mechanism has a stale block rate of *6.8%*. This mechanism reduces the overall security of the blockchain as it loses stability because of the amount of stale blocks it has to handle [14], [29].

Prerequisites: Uncle Blocks and their reward mechanism appear to be a exclusive problem to Ethereum. Any selfish miner can produce an abundance of blocks in the hope for profit.

3.3 Vulnerabilities of Consensus Mechanisms

3.3.1 Vulnerabilities of Proof of Work

Proof-of-Work is a commonly used consensus mechanisms in public blockchains. Miners have to put effort (work) into providing mathematical proof during the creation of a new valid block.

Blockchains that exist for a considerable time, experience an increase of cumulative hash power in the network. This means that computing hashes gets increasingly more expensive to the individual miner as the level of difficulty increases.

Furthermore, hash computation usually has a narrow time-window until it has to be completed, which could be hard to get into. Since rewards for these efforts are usually just granted to those who are actually creating the new validated block, it means that all other miners will not be rewarded for their spent efforts.

To counter this problem, miners started to increase their hash rates by using specialized hardware. All this leads to large amounts of energy being wasted for no result. Bitcoin alone wastes more than *71 Terawatt-hours* yearly just for the Proof-of-Work mechanism, according to estimates.

Another measure to counter the expensive hashing process are mining pools. Miners put together their power which leads to them ”owning” certain parts of the network. If these pools are getting too large, attacks, such as double-spending or majority attacks, could get possible. Additionally, control over the entire blockchain could be at stake [30], [14].

3.3.2 Vulnerabilities of Proof of Stake

Proof-of-Stake (PoS) was introduced to overcome obvious problems of Proof-of-Work (PoW). Energy consumption is reduced, which makes it a more sustainable (”green”) solution. Additionally, thresholds for majority attacks get increased.

While Proof-of-Work uses a probabilistic approach, where miners get approved ”randomly”, Proof-of-Stake uses a deterministic approach. Validators are selected using a process called *bidding*. There they have to bid their stake, which is their balance, and the candidate with the highest stake are selected to be the next validator.

Malicious behaviour will lead to the loss of the stake. PoS is more resistant to majority attacks than PoW. It is no longer the computational power of the network, but rather the entire balance that an attacker needs to own half of beforehand.

On the other hand, the downside of Proof-of-Stake is that wealthy validators keep on winning the bid, which in turn grants them the block reward, leading to a few validators becoming richer each time while new candidates have barely any chance of winning a bid. In turn this leads to a pseudo-centralized system and defeats the purpose of a decentralized public blockchain [30], [21], [13].

3.3.3 Vulnerabilities of Practical Byzantine Fault Tolerance

The Practical Byzantine Fault Tolerance (PBFT) protocol is used in private permissioned blockchains the most. In this trusted environment, the network is grouped into active and passive replicas.

From all active replicas, one is selected to be the primary node. This node is the one who receives all transactions and pushes them out to the active replicas so they can sign the transactions and share it with all other replicas. The results then get transmitted back to the primary node who collects them and forges a new block with all signed transactions. This new block is then broadcasted over the network.

The whole protocol relies on the fact that the primary node is trustworthy and not compromised. There is a number of actions a compromised primary replica could take, such as discard correct approvals and stop transaction execution, meddling with the transaction order to actively create delay in the entire block generation process, withholding blocks and transactions and falsifying transaction approvals.

Actors in a private blockchain network are usually known and therefore it is relatively easy to pinpoint malicious activities to certain members. However, if damage is caused, locating the malicious member is just a reactive or corrective measure.

Another weakness of PBFT is the scalability. Larger networks would suffer from communication overhead between the replicas and performance decreases. One of the key weaknesses of Practical Byzantine Fault Tolerance is that it has a comparatively small tolerance to malicious nodes in the network with just 33% compared to the 50% of PoW and PoS. This low fault tolerance is a major problem, especially since private networks usually are tremendously smaller in size than public networks, which additionally reduces the actual number of nodes needed [30], [5].

Prerequisites: An attacker has to know the system and the implemented consensus mechanism and protocols to choose the right attacks and attack vectors. For each there are slight deviations.

- Proof of Work: Hash computation usually has a narrow time window until it has to be completed, which could be hard to get into. Since rewards for these efforts are usually just granted to those who are creating the new validated block, all other miners will not be rewarded for their spent efforts.
- Proof of Stake: The downside of Proof-of-Stake is that wealthy validators keep on winning the bid, which in turn grants them the block reward, leading to a few validators becoming richer each time while new candidates have barely any chance of winning a bid. This leads to a pseudo-centralized system and defeats the purpose of a decentralized public block-chain.
- Practical Byzantine Fault Tolerance: The protocol relies on the fact that the primary node is trustworthy and not compromised. There are a number of actions compromised primary replica could take, namely discard correct approvals and stop transaction execution, meddling with the transaction order to actively create delay in the entire block generation process, withholding blocks and transactions and falsifying transaction approvals.

3.4 Double-Spending

A general problem of digital currencies is that data can easily be copied. A bitcoin is basically a set of data which can be copied infinitely. This is the digital equivalent of counterfeit money. As with physical currencies, such copies will increase the pool of available coins massively and in turn decrease their value.

To address this problem Bitcoin verifies every transaction similar to a bank, but without a centralized component. **Double-Spending** is the term for an attacker spending the same funds twice by circumventing the verification mechanism. There are different techniques, which can be used by an adversary.

The main issue arises from the fact that confirmations of transactions take time. This circumstance is not ideal for payments, which have to be processed fast, in order to immediately sell goods (e.g. digital product downloads). If another transaction invalidates the payment, the already released goods are lost to the attacker [22], [17].

There are two prominent variants of double-spending:

Race Attack: In this attack, the malicious actor starts two different transactions at the same time, referring to the same funds that are only sufficient for one transaction. Only one of these transactions can be validated. This enables the attacker to retrieve double the amount of goods for the same amount of money, if both receivers don't validate first [22].

Finney Attack: During a Finney Attack, funds are used in a transaction, but the attacker is withholding a pre-prepared block with the same transaction to one of his own accounts. When the shop releases the goods, the attacker broadcasts his block which then invalidates the initial transaction by making the network believe that the actual transaction is the one that has been pre-prepared [17].

Prerequisites: For a successful double spending attack under normal circumstances, it is required to complete two successful separate transactions faster than it takes to verify one of them and realising the problem. However, there is also the possibility of double spending attacks exploiting a fork of the blockchain.

4 Application Oriented Attacks

Blockchain systems are far from being uniform. Blockchain technology is used by broad variety of applications. Each of these applications comes with its own strengths and weaknesses, as well as different

fields of use. Some of the attacks listed in this section could as well be seen fit for other categories. However, due to the fact that execution, requirements and outcome heavily depend on the targeted system, we group them as application-oriented.

4.1 Cryptojacking

Due to the rising demand for computing power mining is becoming less profitable to the individual. This lead to cases of **cryptojacking**, where an attacker utilizes the victim's infrastructure to covertly mine new blocks.

The two most common types are cloud-based and web-based cryptojacking. Initially cloud services were used to run mining operations in virtual machines, but the same principle applies to any hijacked system.

Web-based cryptojacking injects malicious JavaScript code into websites that sends mining tokens to all visitors without their consent. They then compute and send back the hashes. This negatively impacts the performance of the hosts and servers targeted by this attack and increases their CPU usage and battery drain [30], [10].

Prerequisites: An attacker has to inject the malicious JavaScript code that runs on the victim's machine, increasing the CPU usage and battery drain. Usually this requires the user to stay on a specific page for a prolonged time (e.g. streaming sites).

4.2 Timejacking

Timejacking can be seen both as a network attack and an application-oriented attack, based on the implementation of the blockchain system.

As an example, Bitcoin regularly receives a network time from neighboring peers. If the median time of all these nodes exceeds 70 minutes, the program falls back to system time of the node. If the neighbouring nodes are controlled by an attacker, they can be used to send different timestamps resulting in a median higher than 70 minutes. Additionally a block gets rejected if the network time varies by 120 minutes so this type of attack can help create other attack vectors [30], [35].

Prerequisites: An attacker needs control over a large number of neighbouring nodes to be successful with this attack. Ideally, the victim is isolated by a flood of malicious nodes, which hints at a Sybil Attack, further discussed in section 5.6.

4.3 Replay Attacks

A replay attack can take place, if a hard fork of a blockchain exist. This means that two separate blockchains originated from a common original chain, further described in 2.1.

Users will be granted an equal amount of assets on both of these new strains, if they possessed some before the split. Transactions can now happen on both chains separately. Since the ledgers are public an attacker is able to see a transaction on one fork and replay it on the other, if no countermeasures are taken. This results in equal loss of assets on both forked strains instead of one.

This was possible in Ethereum before they introduced *chainID* to identify the intended blockchain for a given transaction, but this feature is not enabled by default and some users are still vulnerable.

Whenever a fork takes place it is the responsibility of the developers to include appropriate mitigations for replay attacks. Other than the blockchain itself, smart contracts can contain similar vulnerabilities [28], [30].

Prerequisites: For a successful replay attack a malicious actor needs to find the blockchain in a state enabling such an attack, such as a forking situation or a badly written smart contracts. If the blockchain

is forked, an attacker requires access to a transaction on one strain to redo it on the other, which will result in funds moved on both of them.

4.4 Smart Contracts

4.4.1 Reentrancy Attacks

In a case specific to Ethereum, it is possible for an attacker to claim the entire balance of a user by recursively calling a function of the ERC20 token, if the user did not update his balance before sending his ether [30].

Attacks could be conducted by specifically crafted contracts that call withdrawing functions, leading to the possibility of being able to withdraw more times than it should be possible to do so [36].

4.4.2 Redirect and DoS Attacks

Another Ethereum-specific vulnerability in smart contracts arises when an attacker exhausts all available gas by enlarging the number of addresses that require refunds. That way the gas limit can be reached before the transaction is completed, thereby failing and cancelling it [30].

Similarly, a contract could be reliant on other contracts for it to fulfill its function. If these external contracts get invalidated, taken over by malicious players, or simply are unavailable by network exhaustion, it poses a threat to the original contract calling them [36].

4.4.3 Overflow Attacks

In Ethereum, the maximum amount of Ether in a variable can be 2^{256}. The usage of this vulnerability is extremely unlikely however, because it requires the sender to send an amount larger than the maximum, which would then be reset to zero [30].

Although overflows are not a new phenomena in computer science, in the world of (Ethereum) blockchain, such an overflow will start an infinite loop which makes it impossible to generate a new block out of the transaction [36].

4.4.4 Short Address Attacks

The Ethereum ERC20 tokens are affected the most by this attack by exploiting a token creating bug in the EVM. While purchasing tokens with a wallet ending on a 0, the attacker simply removes the last digit, which gets appended by the EVM automatically. This results in a grant of multiple times the tokens than originally bought [30].

4.4.5 Forcible Balance Transfer

This vulnerability depends on the Ethereum gas limit as well, but can work in other similar environments too. If there is badly written code or purposely created vulnerabilities that transfer balance without fallback functions, it can lead to failing transactions. The money spent in such a way is therefore lost [30].

Prerequisites: For most of the smart contract related vulnerabilities, it is obvious that coding errors and code weaknesses are a major problem. Insecure coding practices or mishandling of parameters and methods can result in vulnerabilities, which pave the way for new exploits.

5 Attacks on the Peer-to-Peer System

A distributed network is the backbone of a public blockchain. Classical network attacks are nothing new in IT. However, a fully distributed network behaves differently and possesses new threats.

Unsurprisingly, the introduction of blockchain systems enables a number of possible attack vectors in addition to valid traditional network attacks. In this section we take a closer look at attacks that are focused on vulnerabilities of peer-to-peer networks used for blockchain systems.

5.1 Eclipse Attack

An Eclipse attack focuses on the peer-to-peer component of blockchains. The goal is to isolate a victim from a legitimate network by making them connect only to nodes that are under the attacker's control until the maximum connection limit is reached. Afterwards it is much easier to start further attacks on the now shielded victim. The exact procedure differs depending on the implementation of the P2P network. Important factors are include, amongst others, the connection limit and the number of instances that can be started simultaneously on a device under an IP [15], [24].

Bitcoin The Bitcoin Client uses a random protocol to search for 8 peers. On top of that, 117 incoming connections are accepted. An attacker could now try to block all 117 connections to send garbage traffic to the victim, hoping to restart the client. When restarting, the existing peers are deleted and new ones are established. The likelihood that the new peers will be controlled by the attacker is now very high. However, since many nodes are required for this, it is usually necessary to use a botnet for this type of attack.

If a client is successfully eclipsed, this can lead to some problems and further attacks, as the victim is no longer able to tell the difference between a valid transaction and malicious acts [15].

Ethereum Ethereum is another crypto currency that has been vulnerable to Eclipse attacks. The maximum for outgoing connections is 13 instead of the 8 that Bitcoin has, which makes the attack more difficult, but an ECDSA public key is used as the ID for a client and not the IP. This means that any number of instances could be executed on one computer (before version 1.8). This eliminated the need for a botnet and allowed attacks with a fraction of the resources. Furthermore, the selection of the peers using the Kademlia algorithm was very predictable and thus facilitated the transfer of the peer connections.[24]

Follow-Up Attacks If an attacker has succeeded in isolating a victim, he can use various techniques to cause further damage. If, for example, two clients want to fulfill and transmit a task at the same time, a block race is created. In this case, only one result is accepted and the other is discarded. The attacker could now artificially create this problem by holding back results.

This causes a large computing effort without benefit. An isolated client is also no longer part of the network and does not contribute to the completion of tasks. Since some participants can be excluded from the network in this way, this can facilitate 51% attacks (see chapter 3.1). Many attacks also target crypto currencies specifically. They attempt to manipulate the system in such a way that profits are gained by withholding the true state of the transactions from the victim [15], [24], [30].

Prerequisites: An Eclipse attack will be of concern, if the connection pattern and behaviours of the nodes to its neighbours can be predicted. Knowing this, an attacker can set up his own nodes in preparation and then force a re-connect of his victim, via a DDoS (Distributed Denial of Service) for example.

5.2 Selfish Mining Attack

A blockchain in which new blocks are created by miners is vulnerable to *selfish mining attacks*. For the regular mining process, a new block is appended as soon as it is generated.

A selfish miner however wants to generate as much reward revenue as possible by mining new blocks without informing the main chain. These silently crafted blocks in the selfish miner's personal "micro blockchain" then get released all at once, which makes the selfish miner win over the network by overall length of the blockchain, which would result in his blocks getting accepted over the ones that other miners create and possibly fork the blockchain, resulting in a loss of previously mined blocks by other miners for that duration. This state is also described as *block race* in which the two parties, honest and selfish miners, try to be the one to get new valid blocks.

The success of a selfish miner can be greatly improved by increasing the used computation power. Honest miners can be deprived of what would be their reward. Additionally, the computation power spent is lost and the number of stale blocks, another blockchain problem, increases. If two or more selfish miner compete with their privately accumulated blocks, it will probably lead to a larger fork, which would delay the entire network and opens the door for a number of further attacks [30], [3].

Prerequisites: To conduct selfish mining, an attacker needs to be able to forge new blocks at a faster rate than the other nodes. This is usually done by combining hashing power in a mining pool. The blockchain needs to be probabilistic for selfish mining attacks in the traditional sense. For private blockchains, an attacker usually needs to take over or influence the main/primary node of the network to be successful.

5.3 Classical Block Withholding Attacks

The main target for classical block withholding attacks are mining pools. There are two main options described in [30]. One option is that there are two mining pools and one of them withholds a block until the other pool releases its own new block, so both of them can be released at the same time, confusing the network and possibly splitting the blockchain or resulting in a discard and waste of energy for both of them. The second and more practical option is what one miner inside a mining pool acts maliciously against its own pool by keeping a valid block secret to publish it as an independent miner and take all the reward alone, which again leads to a waste of computation power [31], [30].

Prerequisites: One option is that there are two mining pools and one of them withholds a block until the other pool releases its own new block, the second and more practical option is when one miner inside a mining pool acts maliciously against its own pool by keeping a valid block secret.

5.4 Fork After Withholding Attack

The *fork after withholding attack* (FAW) is similar to the first option of classical block withholding attacks. A miner that is member of two different mining pools and has a valid Proof of Work block withholds it until both pools can propagate a valid block at the same time.

The malicious miner will get the reward, no matter which one of the two blocks will be accepted in the main chain. This race condition can also be the case if mining pools are in conflict with each other and actively try to sabotage the other pool. The larger pool with more computation power will usually win this battle. Overall, it is more profitable to do a fork after withholding attack than regular selfish mining or block withholding attacks [20], [30].

Prerequisites: A miner that is a member of two different mining pools and in control of a new valid block.

5.5 Block Withholding in Private Blockchains

Blocks can be withheld in private blockchains as well [30], for example in PBFT networks when the primary replica node acts maliciously, or the attacker has control over a number of replicas already. A primary replica can cause harm on four different ways that are all cause by withholding and result in delay or compromise of the network:

1. Withholding issued transactions
2. Limiting the recipients of transactions in the network
3. Discard received signatures from the replicas
4. Withholding a generated block from the network

Prerequisites: Blocks can be withheld in private blockchains as well, for example in PBFT networks when the bad actor is in control of the primary replica, or the attacker has control over a large enough number of regular replicas to halt the overall consensus process.

5.6 Sybil Attack

As shown in [11, 8], a Sybil Attack is an identity-based Attack and threatens the peer-to-peer network. Communication amongst the peers in a blockchain network works so that a node gets his information from other surrounding nodes. An adversary can create many fake identities so that all connections from the victim will be established with the attacker [26]. When the victim is finally surrounded and cut off from other nodes, an attacker can feed the victim misleading information. This can potentially enable double-spending attacks [2], amongst others.

The Bitcoin protocol is considered to be Sybil resistance as it has countermeasures. There a node just needs one single honest node that provides the information to the true Bitcoin network. Then it will ignore the false information from the attacker [26]. Therefore, Bitcoin considers itself to be Sybil resistant. This of course is no guaranteed resistance since it still relies on a functioning connection to the network. It merely serves as a measure to reduce the risk.

Prerequisites: Although the Sybil Attack shares many similarities with the Eclipse attack, there are still differences, especially since the latter only focuses on identity and not the network in general. An attacker needs the ability to control or create enough Sybil nodes to overrule the voting outcome of honest nodes to slowly block them off from the real network.

5.7 Wallet Theft

In the world of cryptocurrencies, wallets are used to store cryptographic keys of the users. Research shows that there has been a multitude of attacks directed at wallets and a lot of money has been stolen by stealing or deleting keys.

The problem is that when an attacker has access to the victim's private key he can generate new transactions and spend the money of the victim, similar to a stolen credit card. There are many ways for an adversary to steal the keys:

One method is to compromise the Client Software. Bitcoin Core v0.15 contained a vulnerability that allowed an attacker to take over the wallet. Open-source code can be a problem as much as it is helpful, as it is likely that there are vulnerabilities in the code, in the implementation of cryptographic operations or any other level which can be found and exploited by a malicious actor.

Another method to get unauthorised access to a wallet are Man-in-the-middle attacks [30],[25].

Prerequisites: Wallet Attacks are usually found in public and permissionless monetary blockchain systems. There are different approaches on how an attacker might get access to the wallet. One is exploiting vulnerabilities in the client software or the blockchain system itself which in turn requires that there are in fact vulnerabilities present and able to be exploited, the other approach is by using preliminary man-in-the-middle attacks.

5.8 Classical Network Attacks

We are aware that a blockchain network can be subject to a variety of traditional network attacks and attack vectors. Two examples for such attacks are DNS and DDoS. Corruption of the DNS chache or spoofing, as well as man-in-the-middle attacks are still possible with blockchains [30], [9], [19].

DDoS attacks are even more present in blockchain peer-to-peer networks. Especially mining pools and currency exchange services are valuable targets. There have been over 140 reported incidents of DDoS attacks on a selection of Bitcoin services over a duration of two years [7], [2]. While classical network attacks like these examples here are still a threat to blockchains, these affect a broader spectrum and not solely the blockchain world, which we focus on.

6 Discussion

In this chapter we briefly summarise analyzed attacks and vulnerabilities. A quick and comprehensive overview of the condensed results is outlined in Table (1).

While the structure of a blockchain can have major implications on the finalised system, it is important to highlight the variety of such structures. In most articles concerning attacks, it is assumed that the used blockchain system is either used in a public and permissionless setting, or for cryptocurrencies. Most likely, this trend stems from the history of modern blockchains going back to Bitcoin, a public and permissionless cryptocurrency. Blockchain systems with such specifics may be relevant for individuals, but for companies and organisations a private blockchain is certainly the correct way, as they handle sensitive or personally identifiable data. Structures of private blockchains differ from public blockchains, most an foremost in scalability and communication. Vulnerabilities of the structure of such blockchains need to be identified by the operators.

The consensus mechanism of a blockchain system is essential for its operation. Overall, there is a multitude of varying consensus mechanisms of which some are practically just minor deviations of the mechanisms we covered in this article (i.e. Delegated Proof of Stake, Simplified Byzantine Fault Tolerance, etc.). Each of them comes with different advantages and disadvantages which the operators and users need to be aware of. Since the consensus mechanism decides the way of the chain, a foreign takeover is a valid concern. As it is with public blockchains, such consensus hijacks in the form of 51% Attacks is highly unlikely to be performed by individuals if the blockchain has amassed a certain size, as the required power to execute such an attack would be exorbitant. That's not to say that is entirely impossible. For private blockchains, consensus hijack is not particularly the correct term. As there is usually a governing central authority in such chains there is no need to hijack the consensus as the central machine is most likely all it takes.

The capabilities of blockchains have improved tremendously over the past decade. Especially with the introduction of smart contracts, the possibilities for blockchain system reached new heights. Leaving aside the legally binding aspect for lawyers, with smart contracts a blockchain can be used for virtually any market sector, as long as the programmers of such contracts are able to provide the functionality. Therein lies one of the problems. As soon as a smart contract is deployed to a public blockchain, changing the code becomes next to impossible. This means that the programmers of such contracts don't have

a margin of error to play with. Additionally, it can be expected that the number of smart contracts, distributed apps, chaincode and whichever name is used to describe them will certainly increase with the number of blockchain systems and user base growth of blockchains. Another aspect of applications is the user application, of course. Users of a blockchain need to have a way of executing their actions on a blockchain, such applications need to be secure, as it could lead to possible exploits otherwise.

Network attacks and threats are a common sight in the field of IT. Unsurprisingly, blockchains are not isolated from such problems. While some of the issues overlap for both traditional IT and blockchains, there is a number of threats specific to distributed systems and especially, blockchains. Most of these problems occur in public and permissionless blockchain systems, as there is no central authority that could be consulted in the case of an attack. Emerging and existing threats alike need to be addressed in the form of appropriate mitigations to guarantee the security of the whole blockchain system network.

7 Conclusion and Future Work

The blockchain technology has the potential to be trailblazing in contributing to future challenges across various sectors. This applies in particular to those challenges where trust and transparency are of central importance. As no technological progress comes without risks, it is key to be aware of current security challenges. In this paper, we support researchers and practitioners by presenting an overview of a number of highly relevant attacks on and vulnerabilities of blockchain systems. The plethora of source material that is readily available highlights the vulnerable state of blockchain systems and points out that blockchains are not invincible or entirely secure, as some enthusiasts like to propagate. On top of that, even after years of blockchain activities around the globe, security needs are often neglected, which is mirrored in the lack of countermeasures. Another finding of our research on current threats and vulnerabilities is that most of the current research efforts solely focus on the technical perspective of blockchain security, especially consensus and network problems. However, there is a severe disparity to organisational security, which often gets neglected. As compliance to legal and regulatory requirements (e.g. EU General Data Protection Regulation) is getting increasingly important, we are convinced that organizational security controls for blockchains are a critical component to ensure the success of blockchain systems. Therefore, we plan to analyze organizational problems arising when using blockchain technologies. We further aim at elaborating a set of organizational control objectives and recommendations to ensure security assurance.

Acknowledgments

This research was funded by the Josef Ressel Center for Blockchain Technologies & Security Management (BLOCKCHAINS). The financial support by the Christian Doppler Research Association is gratefully acknowledged.

Table 1: Summary of Attacks and Vulnerabilities of Blockchains

Category	Attack/Vulnerability	Underlying System	Affected Blockchains	Maturity Level
Blockchain Structure	Blockchain Fork	Blockchain with mining	Bitcoin, Ethereum, multiple	Executed
Blockchain Structure	Stale and Orphaned Blocks	Blockchain with mining	Bitcoin, Ethereum, multiple	Executed
Attacks on the Consensus Mechanism	Majority/51% Attack	Blockchains without central authority	multiple	High-Effort
Attacks on the Consensus Mechanism	Reward for Uncle Blocks	Blockchain with stale block rewards	Ethereum	Executed
Attacks on the Consensus Mechanism	Vulnerabilities in Proof of Work	Blockchain using Proof of Work	multiple	Vulnerable
Attacks on the Consensus Mechanism	Vulnerabilities in Proof of Stake	Blockchain using Proof of Stake	multiple	Vulnerable
Attacks on the Consensus Mechanism	Vulnerabilities in Practical Byzantine Fault Tolerance	Blockchain using Practical Byzantine Fault Tolerance	multiple	Vulnerable
Attacks on the Consensus Mechanism	Finney Attack	Blockchain with delayed verification	multiple	Possible
Attacks on the Consensus Mechanism	Race Attack	Blockchains with delayed verification	multiple, Bitcoin	Possible
Application Oriented Attacks	Cryptojacking	Blockchain with mining	multiple	Executed
Application Oriented Attacks	Timejacking	Blockchain with multiple writing nodes	Bitcoin, multiple	Possible
Application Oriented Attacks	Replay Attacks	Blockchain without protective measures	Ethereum, multiple	Possible
Application Oriented Attacks	Attacks on Smart Contracts	Blockchain with Smart Contracts enabled	Ethereum, multiple	Possible
Attacks on the Peer to Peer System	Eclipse Attack	Blockchain without authority	Bitcoin, Ethereum, multiple	Executed
Attacks on the Peer to Peer System	Selfish Mining	Blockchain with mining	multiple	Executed
Attacks on the Peer to Peer System	Classical Block Withholding Attacks	Blockchain with mining	multiple	Executed
Attacks on the Peer to Peer System	Fork after Withholding Attack	Blockchain with mining	multiple	Executed
Attacks on the Peer to Peer System	Block Withholding in Private Networks	Blockchain with authorities	multiple	Possible
Attacks on the Peer to Peer System	Sybil Attack	Blockchain with identity mechanism	Bitcoin, multiple	Possible
Attacks on the Peer to Peer System	Wallet Attacks	Blockchain with a wallet system	Bitcoin, multiple	Executed

References

[1] A. Alketbi, Q. Nasir, and M. A. Talib. Blockchain for government services—use cases, security benefits and challenges. In *Proc. of the 15th Learning and Technology Conference (L&T'18), Jeddah, Saudi Arabia*, pages 112–119. IEEE, February 2018.

[2] A. Alkhalifah, A. Ng, A. Kayes, J. Chowdhury, M. Alazab, and P. Watters. A taxonomy of blockchain threats and vulnerabilities. *Preprints*, pages 1–3, September 2019.

[3] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong. A deep dive into blockchain selfish mining. In *Proc. of the 53rd IEEE International Conference on Communications (ICC'19), Shanghai, China*, pages 1–6.

IEEE, May 2019.

[4] B. Carson, G. Romanelli, P. Walsh, and A. Zhumaev. Blockchain beyond the hype: What is the strategic business value. https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Blockchain%20beyond%20the%20hype%20What%20is%20the%20strategic%20business%20value/Blockchain-beyond-the-hype-What-is-the-strategic-business-value.pdf [Online; Accessed on July 25, 2020], June 2018.

[5] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, 20(4):398–461, November 2002.

[6] H. Chen, M. Pendleton, L. Njilla, and S. Xu. A survey on ethereum systems security: Vulnerabilities, attacks and defenses. *ACM Computing Surveys*, 53:1–43, June 2019.

[7] M. Conti, E. S. Kumar, C. Lal, and S. Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, May 2018.

[8] D. Dasgupta, J. M. Shrein, and K. D. Gupta. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology*, 3(1):1–17, April 2019.

[9] A. Davenport, S. Shetty, and X. Liang. Attack surface analysis of permissioned blockchain platforms for smart cities. In *Proc. of the the 4th IEEE Annual International Smart Cities Conference (ISC2'18), Kansas City, Missouri, USA*, pages 1–6. IEEE, September 2018.

[10] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark. A first look at browser-based cryptojacking. In *Proc. of the third IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, UK*, pages 58–66. IEEE, April 2018.

[11] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal*, 6(2):2188–2204, April 2018.

[12] P. Fraga-Lamas and T. M. Fernández-Caramés. A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE Access*, 7:17578–17598, January 2019.

[13] P. Gaži, A. Kiayias, and A. Russell. Stake-bleeding attacks on proof-of-stake blockchains. In *Proc. of the first Crypto Valley Conference on Blockchain Technology (CVCBT'18), Zug, Switzerland*, pages 85–92. IEEE, June 2018.

[14] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proc. of the 23rd ACM Conference on Computer and Communications Security (SIGSAC'16), Vienna, Austria*, pages 3–16. ACM, October 2016.

[15] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *Proc. of the 24th USENIX Security Symposium (USENIX Security'15), Washington, D.C., USA*, pages 129–144. USENIX, August 2015.

[16] K. M. Hossein, M. E. Esmaeili, T. Dargahi, et al. Blockchain-based privacy-preserving healthcare architecture. In *Proc. of the 32nd Canadian Conference of Electrical and Computer Engineering (CCECE'19), Edmonton, Alberta, Canada*, pages 1–4. IEEE, May 2019.

[17] A. Kaushik, A. Choudhary, C. Ektare, D. Thomas, and S. Akram. Blockchain—literature survey. In *Proc. of the second IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT'17), Bangalore, India*, pages 2145–2148. IEEE, May 2017.

[18] T. P. Keenan. Alice in blockchains: surprising security pitfalls in pow and pos blockchain systems. In *Proc. of the 15th Annual Conference on Privacy, Security and Trust (PST'17), Calgary, Alberta, Canada*, pages 400–402. IEEE, August 2017.

[19] R. Khatoun and S. Zeadally. Cybersecurity and privacy solutions in smart cities. *IEEE Communications Magazine*, 55(3):51–59, March 2017.

[20] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In *Proc. of the 24th ACM Conference on Computer and Communications Security (SIGSAC'17), Dallas, Texas, USA*, pages 195–209. ACM, October 2017.

[21] W. Li, S. Andreina, J.-M. Bohli, and G. Karame. Securing proof-of-stake blockchain protocols. In *Proc. of the 12th International Workshop on Data Privacy Management (DPM'17), on conjunction with the 22nd*

European Symposium on Research in computer Security (ESORICS'17) and the first International Workshop on Cryprocurrencies and Blockchain Technology (CBT'17), Oslo, Norway, volume 10436 of *Lecture Notes in Computer Science*, pages 297–315. Springer, September 2017.

[22] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, June 2020.

[23] I.-C. Lin and T.-C. Liao. A survey of blockchain security issues and challenges. *International Journal of Network Security*, 19(5):653–659, September 2017.

[24] Y. Marcus, E. Heilman, and S. Goldberg. Low-resource eclipse attacks on ethereum's peer-to-peer network. *International Association for Cryptologic Research (IACR) ePrint Archive*, 2018(236):1–15, February 2018.

[25] G. Morganti, E. Schiavone, and A. Bondavalli. Risk assessment of blockchain technology. In *Proc. in the 8th Latin-American Symposium on Dependable Computing (LADC'18), Foz do Iguaçu, Brazil*, pages 87–96. IEEE, October 2018.

[26] J. H. Mosakheil. Security threats classification in blockchains. `https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1093&context=msia_etds` [Online; Accessed on July 25, 2020], May 2018.

[27] J. Niu and C. Feng. Selfish mining in ethereum. In *Proc. in the 39th International Conference on Distributed Computing Systems (ICDCS'19), Dallas, Texas, USA*, pages 1–14. IEEE, July 2019.

[28] P. Otte, M. de Vos, and J. Pouwelse. Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107:770–780, June 2020.

[29] F. Ritz and A. Zugenmaier. The impact of uncle rewards on selfish mining in ethereum. In *Proc. of the 3rd IEEE European Symposium on Security and Privacy Workshops (EuroS&PW'18), London, UK*, pages 50–57. IEEE, April 2018.

[30] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and A. Mohaisen. Exploring the attack surface of blockchain: A systematic overview. *arXiv preprint arXiv:1904.03487*, pages 1–30, April 2019.

[31] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla. Security implications of blockchain cloud with analysis of block withholding attack. In *Proc. of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID'17), Madrid, Spain*, pages 458–467. IEEE, May 2017.

[32] M. Tran, I. Choi, G. J. Moon, A. V. Vu, and M. S. Kang. A stealthier partitioning attack against bitcoin peer-to-peer network. In *Proc. of the 41st IEEE Symposium on Security and Privacy (S&P'20), San Francisco, California, USA*, pages 1–16. IEEE, May 2020.

[33] N. Webb. A fork in the blockchain: income tax and the bitcoin/bitcoin cash hard fork. *North Carolina Journal of Law & Technology*, 19(4):283–313, October 2018.

[34] D. Yaga, P. Mell, N. Roby, and K. Scarfone. Blockchain technology overview. *arXiv preprint arXiv:1906.11078*, pages 1–68, June 2019.

[35] E. Zamani, Y. He, and M. Phillips. On the security risks of the blockchain. *Journal of Computer Information Systems*, pages 1–12, December 2018.

[36] P. Zhang, F. Xiao, and X. Luo. Solidity check: Quickly detecting smart contract problems through regular expressions. *arXiv preprint arXiv:1911.09425*, pages 1–21, November 2019.

Author Biography

Lukas König received his BSc degree in IT-Security from St. Pölten University of Applied Sciences in 2019 and is currently pursuing his master's degree in Information Security at that very place.
His research interests include information security management, information security awareness and information security education. He is currently part of the "Josef Ressel Center for Blockchain Technologies & Security Management", researching on the interdependencies of Blockchain technologies and security management.

Stefan Unger is a student of the St. Pölten University of Applied Sciences who received his BSc degree there in IT-Security in 2019. Now he is working on his master's degree in Information Security. The topic of his first thesis was the security of Kerberos in active directory systems and the potential use of systems like honeypots to protect them. He continued to show in interest in network perimeter defense and intrusion detection/prevention. He is currently working as IT-Security Officer at Prinzhorn Holding.

Simon Tjoa deputy head of the department "Computer Science and Security" at St. Pölten University of Applied Sciences and academic director of the master programs "Information Security" and "Cyber Security and Resilience".
He received his doctoral degree in informatics from the University of Vienna and has been working for more than 15 years in the information security domain. He holds various information security certifications, such as Certified Information Systems Auditor (CISA) or Certified Information Security Manager(CISM).
His research interests include cyber resilience, digital forensics and information security management. He currently researches on the interdependencies of Blockchain technologies and security management at the "Josef Ressel Center for Blockchain Technologies & Security Management". Furthermore, he serves as program committee member for several international conferences. He is a member of the IEEE and secretary of the Austrian IEEE SMC chapter.

Peter Kieseberg heads the "Josef Ressel Center for Blockchain Technologies & Security Management", as well as the "Institute of IT Security Research" at the St. Pölten University of Applied Sciences, Austria, and has worked for more than 10 years in the field of IT Security Research.
He is co-organizer of the Cross Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE) and founder and chair of the International Workshop on Security of Mobile Applications (IWSMA), which is taking place for the ninth time in 2020. Peter's research interests mainly focus on issues surrounding the foundations of blockchains regarding non-cryptocurrency applications, as well as privacy and data protection in data driven environments. He is a senior member of the IEEE and chair of the Austrian IEEE SMC chapter, as well as member of the ACM.