CAMPUS CYBER

CRYPTO-ASSET ATTACKS CATALOG

GROUPE DE TRAVAIL

CRYPTO-ASSET

Table of contents

Disclaimer

INTRODUCTION

In 2008, a new electronic peer to peer payment cash system named Bitcoin was created by someone under the alias of Satoshi Nakamoto. We would discover that this system would solve one of the major problems in the digital payment system, which is the problem of trust between parties. Indeed, in the current commercial system, banks are the central agent enabling companies to perform financial transactions. In this case, banks play a role in trust/control systems.

However, in "The Bitcoin" system, it would be technically possible to ensure the validity of transactions without a centralized entity to validate the transaction.

In 2009, when Bitcoin went live, it paved the way for a new world of possibilities. After understanding technology's potential, new projects wanted to improve wanted to take the blockchain concept further by creating a solution that could address new use cases, besides the financial transactions enabled by the bitcoin system. In 2014, The Ethereum foundation started developing the Ethereum Blockchain, a new distributed network enabling not only the same capabilities as bitcoin, regarding peer-to-peer payment but it also created a more complex system based on smart contracts. Their breakthrough came after realizing that instead of only keeping track of financial transactions in a ledger, they could also keep the state of more complex structures, the smart contracts.

Under jurisdiction of the ISO/TC 307, blockchain normalization is under progress.

Because of the limitation of Bitcoin and Ethereum, scalability issues and high fees, other blockchain infrastructure projects have arisen. Each one of them is trying to win the race to blockchain technology mass adoption. Most projects promise to solve the different problems found in the previous blockchain projects. For instance, some projects focus on reducing the transaction fees, others are focusing on enabling a higher number of transactions per second, or focusing on reducing the environmental impact.

Under the jurisdiction of the ISO/TC 307, blockchain normalization is under progress. Currently, the Blockchain and distributed ledger technologies vocabulary known as ISO/22739:2020 has been published in July 2021 and is now under revision. A new standard named ISO/CD 22739 is under progress.

Apart these documents, twelve normalization projects on the blockchain are under progress and four others have been published related to the security management of digital asset custodians (ISO/TR 23576:2020), interactions between smart contracts in blockchain and distributed ledger technology systems (ISO/TR 23455:2019), Taxonomy and Ontology (ISO/TS 23258:2021) and privacy and personally identifiable information protection considerations (ISO/TR 23244:2020).

Before going into further depth on the different attacks observed in the fast-growing industry, we will firstly describe some blockchain concepts. Moreover, we present its architecture, its most relevant components and their interactions. Finally, we assess the different existing vulnerabilities based on theoretical analysis and practical exploits observed in the industry.

CONCEPTS

THE BLOCKCHAIN TECHNOLOGY

Blockchain technology is a distributed storage system. Unlike common centralized databases, they are distributed or decentralized. Blockchains are composed by an elementary subsystem called "nodes" that are part of a network. Their main responsibility is conserving, validating and sharing data in a secure and peer-to-peer manner. It's basically a database that encapsulates data blocks of defined size embedding cryptography means (hash and timestamp) to link them one after the other, therefore creating a chain of blocks, named the blockchain.

The blockchain was the first time described and implemented in the public network of Bitcoin. However, over time, new blockchain solutions emerged proposing different and flexible features more suitable for their usage in different ecosystems such as in private companies or in consortiums. Different than in a public blockchain, where data and control are decentralized, in some permissioned and private networks control can be centralized and assigned to some key members and participants of the network.

Blockchains stand out for their way of accessing their data, the control of their nodes, and the nature of validators.

BLOCKCHAIN NODES

A node is a device on a blockchain network, which is the foundation of Blockchain technology. The nodes are distributed over a wide area network and perform a variety of tasks. A node can be an active electronic device, such as a computer, a phone, or even a printer, as long as it is connected to the Internet and has an IP address. The role of a node is to support the network by managing a copy of a blockchain and, in some cases, to process transactions. Nodes are often arranged in the structure of trees, known as binary trees. Each blockchain network has its own nodes, which hold token transaction records.

Nodes also store network data related to other nodes, so they can connect and interact with each other. To conclude, nodes can request information on both network data as well as transaction data from other nodes in the network.

Depending on the blockchain, there exist several types of nodes: Full nodes, archival nodes, light clients and stateless clients.
Full nodes store all the blockchain data on disk and actively participate in securing the network. They perform tasks such as participating in block validation, receiving and verifying all transactions, and providing the network with data.

Archival nodes are full nodes with additional history data of accounts, state change in the network and it is mostly used by services of blocks explorers, data analytics or infrastructure provider.

Storing the whole blockchain can be resource intensive. Consequently, light clients were created. They synchronize a minimum amount of blockchain data from full nodes and are mostly for querying transactions, verifying that transactions have been validated. However, they cannot create any transaction.

The main goal of the ledger being replicated is to guarantee data integrity.

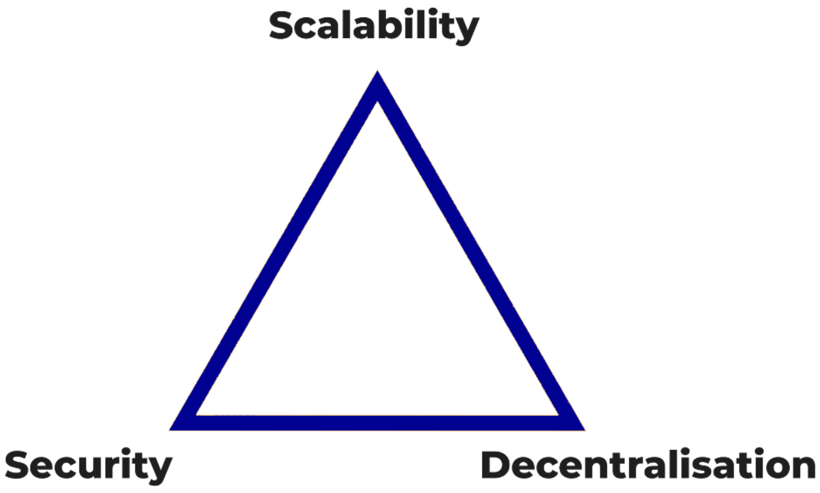
BLOCKCHAIN SECURITY

The main goal of the ledger being replicated is to guarantee data integrity. Each node holds a copy of the blockchain, making it tamper resistant due to the data redundancy and constant communication between nodes. When blocks are validated by a node, the updated version of the blockchain is forwarded to the node's neighbors. The latter proceeds to validate the new chain and propagate it further in a peer-to-peer way. The creation of a new block, the sharing it with the network for validation, is part of the consensus process.

To ensure an absolutely secure ledger, protected against data tampering, the network needs a defined number of independent nodes. (being "independent" in a way that nodes do not collude with one each other).

Thanks to robust consensus mechanisms and replication, the pieces of data managed by blockchain are hard to corrupt, erase or modify. If most of the network possesses similar information, then it is trustful. When the majority of nodes are independent, any attempt to tamper data is corrected by the other nodes of the network possessing the valid version of information.

THE BLOCKCHAIN TRILEMMA



In software architecture it is common to have to make tradeoffs between software applications properties. Indeed, in the case of blockchain it is important to note that the most important properties are the decentralization security and scalability. The decentralization is sometimes calculated as the Nakamoto coefficient. Also, the security of a blockchain ensures fine tuning between the secure consensus mechanism and the properties such as block difficulty, block and transaction sizes. Finally, it is important that blockchain systems scale with the increasing number of transactions. The number of transactions per second (throughput) is usually the metric used to compare the scalability of different blockchain systems.

In decentralized systems, it has been demonstrated that it is not possible to build a system with an elevated level of security, decentralization and throughput. All the different blockchains test several types of architecture, to solve the problem of throughput. For instance, the bitcoin network is very secure, decentralized but has a limited throughput: the network can sustain at most 7 transactions per second, on average. Similarly, other blockchains propose a lower degree of decentralization to be able to increase the number of transactions the blockchain can offer. But lower degree of decentralization can have an impact in the control of the blockchain by one or by a small group of entities.

To solve the problem of scalability, blockchain projects proposed new types of architecture and scaling solutions such as the side chain and layer 2 chains.

NETWORK TYPES[1]

PRIVATE NETWORK

Private blockchains have the particularity of being owned by a centralized entity, authority. In such a scenario, the owner defines participants' access, data visibility, and their roles in the system. Services are restricted to a limited and defined number of users. Their nodes only belong to a limited circle of actors. For example, we find private blockchains for applications in industry such as internal supply chain traceability. But whenever data must be validated and shared across different legal entities, companies tend to join forces in building a co-owned blockchain, in a consortium.

CONSORTIUM NETWORK

A consortium blockchain is a private blockchain administrated by several players with the same level of permissions. The diversity of administrators creates resiliency in the blockchain management, removing every single point of failure from the system as well as decentralizing the control of the blockchain. Usually, the changes in the blockchain network (such as the addition of new members, creation of new governance models, change of consensus algorithms) must be put up to a vote among the different entities of the consortium.

PUBLIC NETWORK

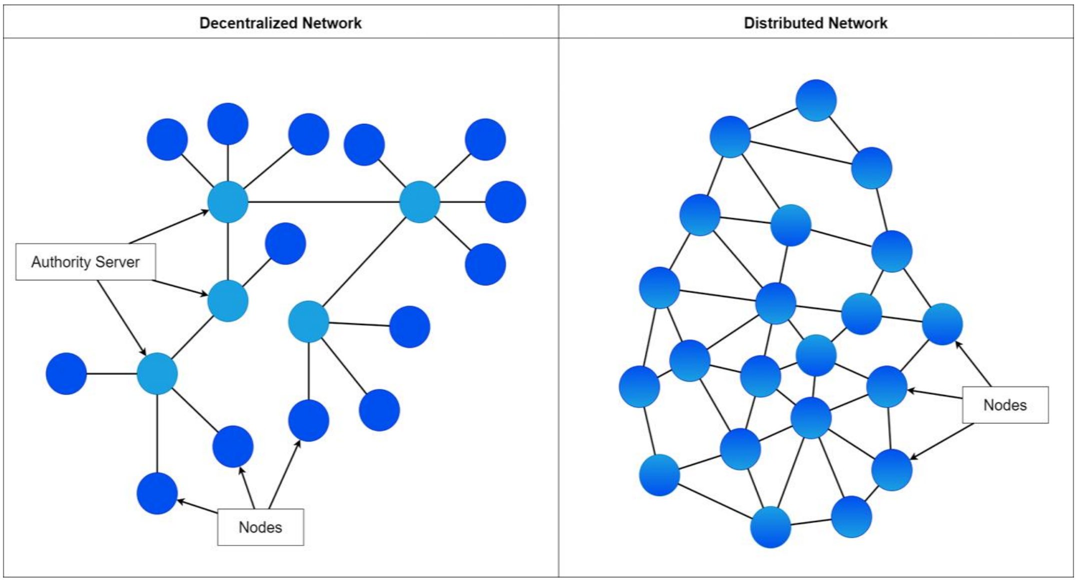
Different from private blockchains, a public blockchain is fully accessible. Anyone is able to access the blockchain ledger, to check or to send transactions, and to become a validator by running a network node. This is the case for the major known blockchain projects such as Bitcoin or Ethereum.

[1] Blockchain - Wikipedia

DIFFERENCE BETWEEN DECENTRALIZED AND DISTRIBUTED NETWORKS

Firstly, decentralized networks are interconnected computers and servers, sometimes geographically distant, working together to provide services based on shared data. In such systems, data synchronization is mandatory but challenging.

Decentralized network is a subset of a Distributed network.



Thanks to consensus mechanisms (Byzantine fault tolerant algorithms), the blockchain can address this problem. Furthermore, distributed networks are understood as decentralized networks where every node is a client and a server at the same time. Meaning that every machine connected to the network is sharing and requesting data in a peer-to-peer method, with every node they are connected to.

DECENTRALIZED NETWORK

In decentralized networks, distant servers are providing information or services to final users. Servers can belong to the same or different private or public institutions. For instance, companies can decide to host their services closer to their clients in specific countries around the globe. In such a scenario, the servers are hosted and monitored by the same company. This is the case of a Blockchain system.

As seen previously, lights clients rely on full nodes to synchronize data and do not participate in the creation or validation of transactions.

DISTRIBUTED NETWORK

An example of a distributed system is the BitTorrent protocol for peer-to-peer file sharing, where every node is a client and a server simultaneously. They are responsible for providing all the services expected by the system.

BLOCKCHAIN LEDGER

The main functionality of a blockchain is to store, share and manage a distributed ledger shared across all the participants in the network. Because nodes agree on the same ledger, it ensures the integrity of the ledger content. Whoever tries to alter the ledger and share it with the network would be prevented by the consensus mechanism.

DATA REPRESENTATION

Depending on the blockchain system, different types of data can be stored in different types of data structures. For instance, in the Bitcoin ledger stores UTXO (unspent transactions outputs), the remaining amount of a Bitcoin transaction. Bitcoin ledger doesn't keep a user balance in a single place. Rather, Bitcoin clients must go through the blockchain history to calculate one's balance. Ethereum approach is different, it keeps four different ledgers, one for the world state of the blockchain, a storage one for keeping track of each contract's state over time, a transaction and a receipt ledger for keeping track of all transactions validated as well as their receipts from previous transactions.

LEDGER DATA STRUCTURE

Nonetheless, the ledgers are known to be immutable, and this property is mainly due to a verification mechanism existing in the data structures used by blockchain: the Hash Trees, also known as Merkle Trees, a type of Binary Trees. This type of data structure has two advantages. Algorithms running on binary trees usually have better performance when compared to other data structures. In addition, Merkle trees guarantee integrity of their structure thanks to cryptographic hashing functions. Some blockchain use a combination of different data structures to improve the system's performance as a whole. For instance, Ethereum uses Merkle Patricia Tries, which is a combination of Merkle Trees and Radix Trees.

In general, a blockchain could be represented by the following diagram:

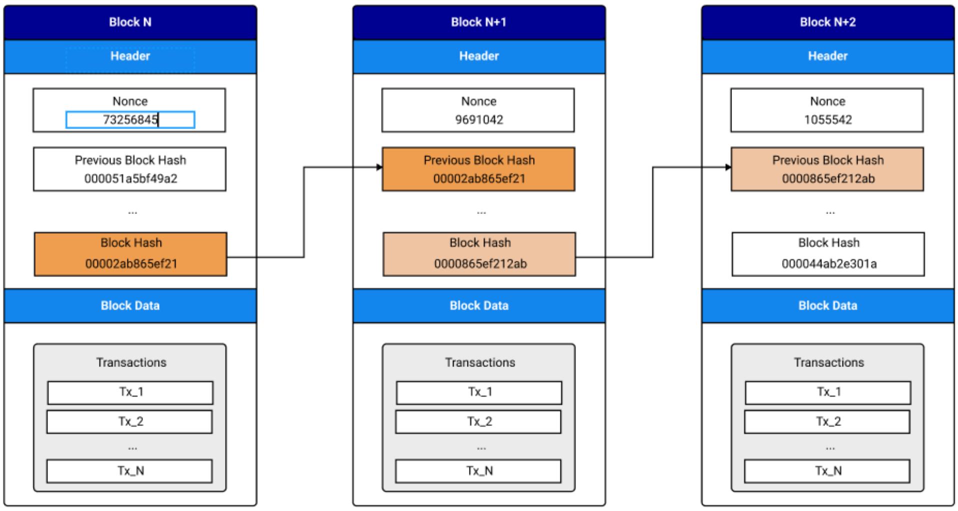


Figure 1 : Blockchain Ledger

CONSENSUS METHODS

A consensus mechanism is a manner for servers and systems to reach an agreement. Such concepts are mostly studied in decentralized and distributed systems. Even if many algorithms exist, the most effective ones are named the Byzantine Fault Tolerant (BFT) algorithms, allowing decentralized systems to connectivity issues leading to non-responsive nodes in the network. We will describe further the most common types of consensus methods used today in blockchain systems.

Consensus mechanism can be based on proof of work, proof of stake, Delegated proof of stake or proof of authority.

PROOF OF WORK

The proof of work was the first blockchain consensus used in the Bitcoin network. To incentivize network members to correctly validate blocks and their transactions, the first nodes capable of solving a cryptographic mathematical problem obtain a reward in a crypto coin. In the case of Bitcoin network, node validators received bitcoins. The proof of work is a fair challenge, the result of such calculation is random and can only be found by trial and error. Such property is possible thanks to the irreversibility of cryptographic hashing functions. As a results, all the participants have a probability of obtaining the reward proportionally to their computational power. The proof of work is considered to be the most secure consensus algorithm because every validator competes in the race to obtain the reward. However, the Proof of work has been heavily criticized because of its energy consumption.

PROOF OF STAKE

The Proof of Stake consensus algorithm imposes block validators to lock a certain number of tokens in the network. Such an amount is generally high and can be taken away in case of misbehavior. Instead of competing against each other, validators are chosen randomly to validate blocks based on the number of crypto coins they are staking in the protocol. To secure the system and prevent malicious actors, if any other validator sees malicious transactions, the protocol will punish (slash) the node by taking part or the total of their staked tokens. Slashing has a key role in dissuading malicious behaviors. This consensus algorithm is considered an alternative to Proof of Work because of its lower carbon footprint and higher throughput of transactions.

DELEGATED PROOF OF STAKE

Delegated proof of stake consensus allows users to participate in the protocol by delegating their tokens to a trusted third party. This is rather useful when a user wants to participate in the validation process but either doesn't have enough tokens to run their own node or when they don't want to manage a node themselves. Third party nodes will then be responsible for validating transactions on behalf of the token owners. As a consequence, if the validators misbehave or fail to comply with the network requirements, it will be slashed and user's tokens can be lost.

PROOF OF AUTHORITY

Proof of authority is a preferred consensus algorithm in networks where node's identities are known and therefore there is trust established. Usually, they rely on smaller networks composed of less nodes. Tens of validator nodes will be responsible for validating all the incoming transactions. Also, thanks to the lower numbers of servers, it enables a system with higher throughput, higher level of trust and lower transaction fees. In case of malicious attacks, it will be easy to identify malicious actors because of the identity of validators and thanks to the blockchain transparency.

BLOCKCHAIN COMPONENTS

This document will base its catalog approach on a scheme developed and shared by contributors. We will focus on component vulnerabilities. Next chapters will drill down to all components.

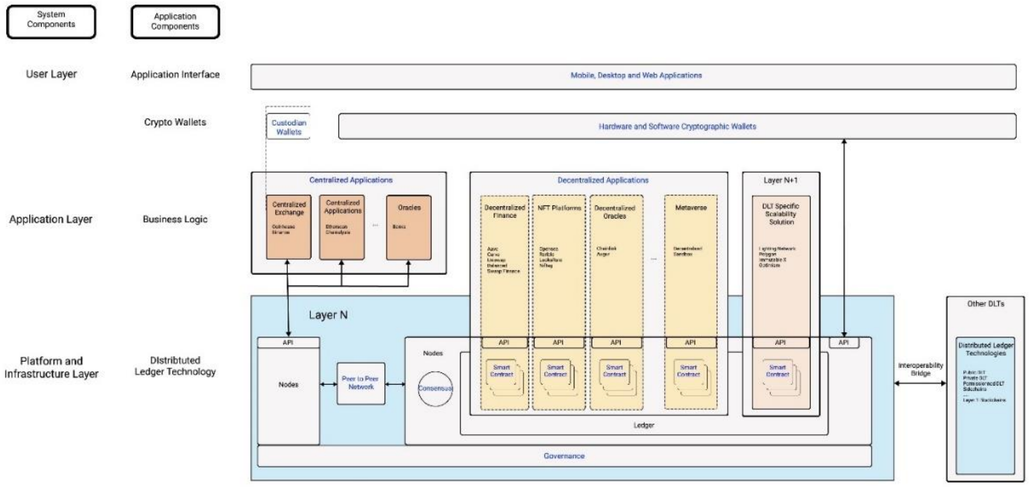


Figure 2 : Blockchain Components (see Appendix)

The blockchain is represented by the "Layer N" block in blue. It contains the nodes, consensus, smart-contracts and associated governance.

This Blockchain is consumed by centralized applications, decentralized applications and N+1 layers which all need a Wallet (two-key) to interact with the Blockchain

COMPONENT ATTACKS

DATA LAYER (SMART CONTRACTS)

DEFINITION

Smart contracts are software programs stored on blockchains ledger. Once they are integrated in the blockchain, they become immutable. No change can be made. And thanks to blockchain transparency, anyone can read it. Smart contract programming language and structure depend on the blockchain specific technology. For instance, on the one hand, private blockchains such as Hyperledger Fabric allow one to write smart contracts in Java and other programming languages. On the other hand, the public blockchain Ethereum, developed its own domain specific language named Solidity that should be executed on the Ethereum Virtual Machine (EVM). In public blockchains, the actions of deploying and making changes in smart contracts can engender fees. Note that this is not the case for some open source private blockchains. Similarly, executing smart contract code on the blockchain can have different financial costs, the more complex the code is to be executed, the higher the fees are. The cost of code execution is mostly deterministic and in the case of Ethereum, it is defined for every instruction at machine level. Fees are generally collected by the node responsible for executing the smart contract code and adding the generated transaction to the blockchain. In some private blockchains, no fees are required since there are no financial incentives existing in its consensus algorithm. For blockchain using tokens or crypto coins, it is possible to perform automated payments with smart contracts.

UTILITY

Smart contracts allow users to execute software in a deterministic way resulting in an immutable and definitive transaction stored in the blockchain ledger. For public blockchains, anyone willing to pay the transactions fees can leverage the blockchain execution and storage mechanism. The immutability and transparency of the transactions in the blockchain were one of the main reasons for its adoption. For instance, we have seen insurance adopting the technology to provide automatic reimbursement of loans or automatic insurance reimbursement.

Since smart contracts are a piece of software, their use cases have been evolving, addressing increasingly complex problems. As a result, some smart contract standards have been defined for the most common use cases. For instance, decentralized applications exposing services on the blockchain require payment in their specific utility token. Such tokens are smart contracts and standards have been defined for each blockchain ecosystem. For instance, in the Ethereum blockchain, the ERC20 standard is currently used. Furthermore, items or products that can be acquired were defined following another standard (ERC 721), named as non-fungible tokens (NFT).

THREAT SOURCE

This component is the favorite target of hackers since they are the heart of applications managing sometimes hundreds of millions of euros. In the case of public blockchains, smart contracts code is publicly available on the blockchain and anyone can interact with them. Malicious actors will try all possible edge cases with the intention of obtaining the funds stored in smart contracts, or to elevate their privilege leading to a financial profit at some point.

VULNERABILITIES

Regarding smart contract vulnerabilities, we can enumerate the following problems:

- A vulnerable implementation of smart contract logic.
- Flaws in the programming language execution and toolchain.
- Flaws in the smart contract execution environment.

IMPACTS

- Non-authorized code execution leading to changes in the smart contract. (Integrity).
- Deny service (Availability).

- Financial losses.
- Elevation of privileges.

EVENTS CATALOG

Numerous events are linked to wrong coding on smart contracts occurred with small or large squall impacts

- The DAO Attack
- The CoinDash ICO Hack
- The BitGo Hack

NETWORK LAYER (Peer to Peer Connection)

DEFINITION

Blockchain technologies leverage a peer-to-peer network to communicate with other participants. Depending on the type of clients' software, a node can download a full copy of the blockchain ledger. When a new node joins the network, it discovers its peers to whom they can connect and maintain the information internally in a dynamic routing table. Such a table contains the details of the nodes it is connected to: node ID, IP address and port.

The node discovery leverages specific protocols. In the case of the Ethereum blockchain it uses RLPx as well as the Ethereum Wire protocol to facilitate the data exchange between the nodes. In general, it is used for chain synchronization as well as exchanging transactions and blocks between nodes.

UTILITY

The network layer allows sharing block transactions information on a secure p2p communication between nodes using the Waku protocol (previously Whisper). It enables the synchronization of blockchain between nodes when a new node enters the network or when a node needs to catch up on the latest blocks generated.

THREAT SOURCE

The network layer is a privileged target because of:

- Only a few restrictions on the node creation process make it easier for anyone to create one of several nodes.
- Malicious nodes can try to control the information a node receives from its peers by eclipsing them. This usually happens with high-profile nodes such as miners or merchants.
- Network and routing configuration might not be secured or can be misconfigured, enabling actors.

VULNERABILITIES

Regarding vulnerabilities, we can mention the following problems:

- Conception and implementation of blockchain client software allowing connectivity between users and the blockchain.
- Misconfiguration of nodes and human flaws.

IMPACTS

- Potential for double spending attack.
- Leak of private keys (Confidentiality)

EVENT CATALOG

- Eclipse attack
- Account Hijacking Attack

CONSENSUS LAYER (CONSENSUS PROTOCOLS)

DEFINITION

Consensus mechanisms (also known as consensus protocols or consensus algorithms) allow distributed systems (computer networks) to work together and reach agreement on the current state of the network. The constant alignment of nodes on which is the trusted version of the blockchain provides security to the system.

UTILITY

For decades, these mechanisms have been used to build consensus between database nodes, application servers, and other computing infrastructures.

In recent years, new consensus mechanisms have been invented to allow crypto economic systems, such as Ethereum and Bitcoin networks, to agree on the current state of the network.

A consensus mechanism in a crypto economic system also helps prevent certain types of economic attacks. In theory, in blockchains using the proof of work consensus algorithms, an attacker can compromise consensus by controlling 51% of the network. Consensus mechanisms are designed to make this "51% attack" impractical. The different mechanisms are designed to solve this security problem in diverse ways.

For instance, proof of work and proof of stake as defined above.

THREAT SOURCE

A malicious validator can try to leverage the consensus in its advantage.

- A crafty miner can split solving tasks to externalize it.

A malicious validator can adopt a “selfish mining” behavior.

- Offer higher Gas Fees to foster the use malicious transaction

- Uncle-rewarding mechanism allowing use of obsolete blocks to gain rewards or double spending.

Honest mining (i.e., including the most valuable transactions in new blocks) is the most profitable strategy for each miner—it may not be true. This is because it can be more profitable to deviate from honest mining strategies, such as conducting selfish mining, accepting bribes, and reaping ordering optimization fees. This vulnerability is caused by the consensus protocol for not being incentive-compatible, due to the tradeoff between availability and consistency stated by the CAP theorem:

- When new transaction verification requires non-trivial computational effort, miners are exposed to attacks whether they choose to verify the transaction or not. If miners verify a computationally heavy transaction, they will spend a considerable amount of time and give malicious miners an advantage in the race for the next block; if the miners accept the transaction without verifying it, the blockchain may include an incorrect transaction.

VULNERABILITIES

Regarding vulnerabilities, we can mention the following problems:

- Design vulnerabilities
- Implementation vulnerabilities

IMPACT

- DDOS (Availability)
- Groundless transactions (Integrity)

EVENT CATALOG

- Fomo3D Attack
- ETC 51% Attack
- Selfish Mining Attack
- Resource Exhaustion Attack

GOVERNANCE

DEFINITION

Governance is the set of rules defining how the blockchain should work as well as the processes defining how decisions should be taken to change those rules. We are talking here about decisions regarding the functional and technical orientations of the system. The first distinction that can be made in terms of types of governance is whether the decision-making process involves all the stakeholders or only a central authority. We will talk of decentralized blockchain in the first case and of centralized blockchain in the second.

The other important distinction is about the process used to make decisions. For instance, some blockchain systems prefer to include a small portion of the community in discussions. In this case, agreements can be reached in a more centralized manner with less transparency. Some other systems incentivize the whole community to vote using the tokens associated with the project. Such voting process is done on chain, using smart contracts deployed on the blockchain. The main benefit of this approach is that the whole community can participate and the voting process is more transparent to everyone. For instance, some projects consider a token to be equal to a vote. Also, some projects create dedicated governance tokens with the sole purpose of enabling votes in the evolution of the system.

In the first case, we explained the off-chain governance. In the second, we mention the on-chain governance.

Among decentralized blockchains, there are two ways to make decisions about the orientations in term of project direction, types of updates to implement or extra functionalities to develop.

The first one is called off-chain governance. It is applied on famous blockchains like Bitcoin or Ethereum and is used by most of the Proof-Of-Work blockchains. The decision-making process involves all the stakeholders who are supposed to interact informally through conferences or online forums in order to reach a global agreement. In case of no consensus, a split in several chains may happen and the “child chain” with the biggest computational power ends up designated as the successor or the initial chain. E.g., Ethereum and Ethereum Classic

The second one is called on-chain governance and works according to defined algorithms that were previously validated by stakeholders chosen according to criterias that are transparent for all blockchain users. One example is the Proof-Of-Stake blockchains in which validators are chosen according to algorithms where the number of tokens owned plays an important part. On-chain governance is praised for enabling a faster and more transparent decision-making process than off-chain governance and limiting the risk of fork but suffers criticism due to a risk of sliding into a plutocratic mode of governance. Most of the time, on-chain governance has a part of off-chain governance where the involved parties (developers, stakeholders, delegates, etc.), or some of them, discuss and try to reach consensus on what evolution proposals to submit to the global community. In this case also, global consensus may not be reached. There are 2 rules in public blockchains: code is law (specially consensus code) and if you don't agree with this law, you can decide leave. A user activated hard fork can be understood as a revolution or a secession from the original sovereign community.

UTILITY

Governance is a fundamental part of every blockchain project. Indeed, projects tend to evolve over time because of internal requirements such as changes in the consensus protocols or in the technical parameters of the project.

For instance, an example of relevant change is Ethereum move from proof of work to proof of state. But also, it is important to consider processes to allow projects to react when unexpected events happen. For instance, in the case of major vulnerability of hack. Some guidelines could be associated with a more classical approach such as disaster recovery procedures.

THREAT SOURCE

Governance is an essential and complex part of a blockchain system that varies according to the project and community vision. It not only rules technical aspects of the blockchain but also regulates the business model, tokenomics, and evolution of the system. Thereby some malicious agents could be interested in the financial advantages of modifying the governance, similarly to the attack on Beanstalk Farms. Because the governance also describes the vision and the rules of the system, a group of actors could try to modify essential elements of the governance system to suit their interest or to support their vision, which could lead to blockchain forks such as the one happening to Ethereum leading to the creation of Ethereum Classic. Some of these actors could be hackers trying to obtain financial advantage, blockchain competitors trying to destabilize the trust on the project, governments or even politically engaged activists.

VULNERABILITY

As for on-chain governance, vulnerabilities by design (such as bad decentralization caused by unbalanced stake distribution) can be exploited by Threat Groups to take control of the blockchain. Regarding off-chain governance, the risk is to have forks because of the incapacity of stakeholders to reach a consensus.

IMPACT

As for the on-chain blockchains, hostile takeovers caused by design vulnerabilities can cause theft of funds as illustrated by the BeanStalk hack, or gain of control over the blockchain as illustrated by the Steem/Hive fork.

As for the off-chain blockchains, forks may happen due to a lack of consensus.

EVENT CATALOG

- ↗ Ethereum fork in 2016 and creation of Ethereum Classic
- ↗ Beanstalk Farms: Flash loan to obtain majority of decision chair
- ↗ Terra Blockchain Halted To ‘Prevent Attacks’
- ↗ Steem hostile takeover and creation of Hive

DECENTRALISED APPLICATIONS (DEFI, NFT, METAVERSE…)

DEFINITION

Decentralized applications, also known as Dapps, are applications where part or all their business logic relies on one or more smart contracts. Indeed, any application using software running on a distributed system, such as the blockchain, is considered a decentralized application. Their main difference with centralized applications comes from the fact that there is no central entity, holder of the services and the data used by them. Therefore, decentralized applications have the benefit of being constantly available, regardless of the will of the entity that created them. Indeed, the information present on the smart contracts is replicated on a blockchain system and cannot be controlled or deleted by the entity.

UTILITY

Decentralized applications make it possible to offer new types of services where the parties no longer need to trust each other. This involves commercial relationships without the need for intermediaries, allowing service providers to be directly connected to their customers.

Although distributed ledger technologies are already used in the context of private companies to automate tasks and payments, most of the more innovative use cases are mainly observed in public blockchains. We see the development of new industries such as decentralized finance, digital art, games on blockchain as well as the mixing of virtual reality technologies with blockchain to create the metaverse.

DEFI

Decentralized finance is one of the most promising applications for blockchain and DLT technologies. The tokenization of financial assets on the blockchain could not only provide the transparency and traceability desired by citizens, institutions and regulators but it could also put the citizen at the center of financial services economy. For instance, financial services such as lending and exchange of crypto coins could be provided from individual to individuals without financial institutions as a middle man. For instance, protocols such as AAVE and Uniswap provide such services.

NON-FUNGIBLE TOKENS

Non fungible tokens are a standardized manner to represent asset on a blockchain. They can represent both real world and digital assets. NFT can also be understood as a digital certificate of authenticity or digital certificate of ownership. Even if NFT has been mostly associated with digital art in form of images, gif, videos and music, it could be used to represent more complex and abstract concepts such as physical products, real estate properties, carbon footprint, domain names, membership services or even identity. Platforms such as Opensea and Rarible provide means for users to create and sell NFTs.

DECENTRALIZED ORACLES

Oracles are an important service because they provide a sole source of truth and are useful for many applications such as giving the price of a certain asset. However, oracles are usually controlled by one entity and it because of the risk associated, decentralized oracles were created. They rely on a group of entities to agree on the data that is provided in the blockchain. This is an important means to add trustworthy data to the blockchain to be used by other smart contracts. Projects such as Chainlink and Augur are paving the way for decentralized oracles.

METAVERSE

The metaverse is still in its infancy and it is a vision for a digital world where people can interact via avatars, attend events, university, courses, play games, consume products and services in general. The main characteristic of the Metaverse is that the virtual world is connected to the blockchain. Every asset represented in the metaverse should be associated to NFTs, making them the atoms of the metaverse. It therefore possible to own assets in the metaverse and develop a whole economy, using crypto coins as a form of payment. The French projects Sandbox and Decentraland can be considered as the most advanced projects.

GAMING

Similarly, to the metaverse, the gaming industry is also being disrupted by blockchain technology. Now, assets existing in the games can easily be traded, sold, thanks to the blockchain. Before, in-game assets only existed in the game. However now they can freely exist inside different games. Also, thanks to the underlying crypto coin economics, new business models for games have emerged such as "Play to Earn" (P2E) where gamers are able to obtain crypto coins or NFTs by playing a game. For instance, one of the famous P2E games is Axie Infinity.

DECENTRALIZED IDENTITY

Digital Identity is becoming central issue to be tackle as it is the trust anchor of any electronic transaction. Blockchain can be seen as one of the infrastructure to manage some dedicated Digital Identity registers and some time, for non critical application manage the Digital Identity itshelf.

In addition, decentralized applications also make it possible to create services and functionalities to improve and secure the blockchain ecosystem. For example, some decentralized applications create bridges between different blockchain ecosystems. In addition, multi-signature wallets are smart contract-based applications for securing digital wallets. Finally, any service available based on the blockchain is resistant to censorship and also benefits from the security and transparency of the transactions recorded on the blockchain.

THREAT SOURCE

Decentralized applications have the challenge of defining what part of the application should be on chain and which part should be off chain. From a functional perspective, applications try to provide as much visibility as possible to their users about the core logic of their applications. At the same time, they try to reduce the surface of attacks.

Malicious agents partly attack smart contracts under the control of decentralized applications in order to change protocols' behaviors, elevate privileges, among other techniques to ultimately steal crypto tokens.

Although it is difficult to distinguish attackers, some hacks could be associated with groups of cyber actors linked to nations. For example, the FBI was able to trace the funds that were stolen from the Axie Infinity game and certified the involvement of the Lazarus and APT38 groups. In addition, the American agency underlines the link of these groups with the North Korea.

VULNERABILTY

Poor implementation of smart contract functionality often puts decentralized applications at risk of cyberattacks. (e.g.: integer underflow, overflow or poor management of functionalities permissions). Since decentralized applications are the combination of different smart contracts functionalities, all the vulnerabilities targeting smart contracts can also impact decentralized applications. Because of the complex software architecture of some decentralized applications, it makes it harder for developers to identify vulnerable flows.

- Poor management of access control to smart contract methods allows attackers to gain access to features that are only accessible to a specific number of users. For example, adding the attacker's account to the list of accounts authorized to withdraw funds stored in the protocol.
- The order of smart contract code can create some unexpected and unsafe behavior. For instance, it is the case for Reentrancy attacks such as the DAO hack.
- In DeFi, oracle price manipulation.

IMPACT

Malicious agents can elevate their privileges to access restricted functionalities, to retrieve funds, to alter smart contract behavior or illegally change smart contract state (integrity).

EVENT CATALOG

- The Lazarus group has stolen $625 million in tokens belonging to the game Axie infinity.
- The Poly network protocol, which allows the interoperability of crypto coins between different blockchains, lost $600 Million in digital tokens.
- A flaw in the Wyvern Protocol has allowed hackers to recover free NFTs offered for sale on the Opensea platform.

CENTRALISED APPLICATIONS (INCLUDING EXCHANGE PLATFORMS)

DEFINITION

Within blockchain applications, decentralization and centralization naming refers to governance models. A centralized blockchain application is managed by a limited number of entities, or even a single actor. In contrast, the management of decentralized applications is more open to all their members (this concept is detailed in the previous point "DECENTRALISED APPLICATIONS (DEFI, NFT, METAVERSE…) ".

These two management models will not have the same impact on the choice of architecture for an application and its use. The vulnerabilities will be markedly different. Attackers thus adopt specific strategies to the degree of centralization/decentralization of the targeted applications.

UTILITY

Centralized management provides better control over applications, access and ease the regulation appliance. Centralized management can be applied equally on network nodes control or blockchain-based services.

For blockchain-based services we can cite the following examples :

CEX EXCHANGE PLATFORMS ("CENTRAL EXCHANGE")

These are purchase, sale and trading platforms on which digital assets can be obtained via an intermediary, the website or the APIs of the exchange platform. The majority of these players host the wallets of their clients.

CEFI ("CENTRALIZED FINANCE") SERVICES

CeFi services were created by companies to deal financial offers inspired by DeFi (financial services offered without intermediaries). However, access to CeFi services is only via the website or APIs of exchange platforms. The wallets of their users are mostly hosted by the CeFi service.

PART OF ORACLES

Oracles provide information to a blockchain application from external sources. For example, a smart contract will use an oracle to retrieve weather data or the real-time price of a token. Suppliers managing centralized oracles are linked to a limited number of information sources.

PART OF BRIDGES

Bridges between blockchain networks allow the transfer of digital assets from one blockchain to another. In a centralized bridge, a single organization is solely responsible for this service.

SOME OF THE CRYPTO-ASSET PORTFOLIO PROVIDERS

These players host and hold their clients' portfolios. Wallets providers are compatible with centralized (CeFi) or decentralized (DeFi) finance services.

THREAT SOURCES

Although major players are robust against attacks, centralized applications are more traditional and known by attackers than those of decentralized applications. Malicious actors may choose the easiest target.

As for centralized blockchain networks, their restricted numbers of nodes expose them to consensus attacks and DDOS attacks. An attacker will be more motivated to target a centralized blockchain network than a decentralized network that has similar node access management flaws.

On the other hand, another source of threats is the connection between centralized applications and decentralized services. An attacker can use a centralized application to impact another target. For example, a malicious user alters the information provided by a centralized oracle to destabilize the operation of a smart contract on a decentralized service.

VULNERABILITY

- DDOS
- Stronger exposure to "51% attacks"
- Oracle data feed poisoning
- Centralized bridge attacks: cross chain replay attack, token recovery without deposits
- Keys compromising of the wallets hosted by the centralized platform. The attacker takes control of the user's wallet

IMPACTS

- Service denial
- Network takeover
- Services malfunction connected to the centralized application
- Theft of fund

EVENT CATALOG

- Hot wallet attack: BitMart - 2022
- Backend vulnerability: OpenSea - 2022
- NFT's stolen in apparent phishing attack: OpenSea
- Oracle price manipulation Cream Finance – 2021
- DDOS attack on Bitfinex - 2017

WALLET (HARDWARE AND/OR SOFTWARE)

DEFINITION

A Wallet in the blockchain eco-system is the link between the natural crypto asset's owner and the crypto asset itself. Crypto wallets are simply defined as a pair of asymmetric cryptographic keys. Counterintuitively, wallets don't store or hold any crypto asset, instead, the ownership is done via the association of the crypto asset and the user public key (and therefore associated to his private key). Also, nodes use wallets to authenticate themselves in the network and the blocks they have validated.

Wallets can exist in multiple forms for different purposes. At first, we could make the difference between hot and cold wallets. These definitions come from the fact that hot wallets are connected to the internet and cold wallets are not. Cold wallets were created with the intention of reducing the risk associated with the wallet component being exposed to malicious attacks. On the one hand, hot wallets exist in different formats: they can be web-based, a desktop application installed in a computer or a server, or a mobile application. On the other hand, cold wallets can be certified hardware wallets, wallets stored on disk or even paper wallets.

It is also important to note that hot wallets can exist in two formats: custodial and non-custodial. Users might be interested in delegating the complexity of managing and securing their wallets to a trusting third party entity. Such entities will be responsible for securing the keys and their assets on the user's behalf. If third parties get hacked or create fake transactions, it would be mostly impossible for the final user to undo the malicious transactions. Therefore, some users prefer to hold and manage their keys on their own because as said in the crypto community "Not your keys, not your coins."

UTILITY

Wallets are a fundamental component in the blockchain ecosystem. They are used for authentication purposes and for enabling transactions of crypto assets between users. As mentioned previously, crypto assets are not stored in wallets. They are stored and represented in smart contrats and associated with the wallets via the wallet's address (based on the public key). For a user to be able to claim transfer assets, they have to cryptographically sign the transaction with their wallet private keys. Similarly to public key infrastructure, private keys must be protected at all costs. If they are compromised, an attacker could easily steal all their crypto assets stored on the blockchain associated with a specific wallet.

Also, for security concerns, multi-signature wallets have been created. Private keys can be considered as a single point of failure. If one loses them, it is impossible to recreate them, and crypto assets associated with that wallet are basically lost. If it gets compromised, there is no way for the user to prevent the attacker from stealing their assets. As a result, a specific type of smart contract was created to offer users to be able to associate more than one private key with a wallet. The goal is for every transaction to be validated by a specific number of those private keys, making it more resilient in case of loss or theft of private keys.

The Wallet and the User interface are strongly related and, in some cases, can represent the same subsystem.

THREAT SOURCE

Because wallets are the entry point components for managing assets, it is the most desired prize for hackers. With wallet access, malicious agents can steal crypto assets by stealing the wallet seed phrase of private key.

As the private key is stored in the Wallet and as the Wallet operates into a non-trusted environment, the attack surface is very large. The threats agents can have several profiles such as:

- Opportunists.
- Professional hackers, digital merceries.
- State funded espionage.

VULNERABILITIES

Such components could be vulnerable to known attack existing for each layer where the wallets and private key can be stored. In the case of web-based wallets, malicious users can leverage phishing attacks to persuade final users to share their credentials to access their wallets or even share their sensitive wallet data itself. In the case of custodial wallets or blockchain nodes, key management becomes a problem. Wallets are vulnerable to all the following actions:

- Social engineering attacks.
- Key logger activities to obtain login, password, passphrase.
- Bad wallet implementation (leakage, weak cryptography library),
- Code injection attacks.
- Hooking attacks.
- Brute force attacks.
- Dictionary attacks.
- Fuzzing attacks (hardware and software).
- Hardware Fault injection attacks.
- Hardware Side channel attacks.
- Adversarial attacks.

IMPACT

- Stealing of funds
- Misbehavior of blockchain nodes
- Validation of fraudulent transactions.

EVENT CATALOG

- Trezor vulnerable wallet.

- Horizon bridge hack.

WEB APPLICATION INTERFACE

DEFINITION

The user interface includes the tools used to access the Blockchain and wallets, while requiring a formal action from the user. This part includes mobile or heavy client applications, Web applications (Web 2 or Web 3), browser extensions, as well as the integrated functions of mobile OS or non-mobile OS.

UTILITY

User interface is the gateway to the Blockchain and all related services.

Ergonomic tools are essential to help towards mass adoption. The cryptographic concepts and the required security bring complexity for users, without mentioning key management (including the absence of a "usual" recovery mechanism in the event of loss of passwords or seed words).

It is therefore possible to make a simple differentiation between the centralized services in charge of key management (custodial services) which hide this complexity and all the other non-custodial services.

THREAT SOURCE

Malicious agents will seek to attack users first using generic and well-known attacks mechanisms. These are generally the same “usual” ones, (not dedicated to Blockchain) based on user credulity.

Another source of threat is blind signing: signing operations without understandable content.

VULNERABILITY

- Lack of awareness of risks and attacks (ex: phishing, fake sites)
- Lack of control over downloaded apps (ex: fake mobile apps)
- Lack of control for browser extensions (ex: fake extensions)
- Blind signing
- Misuse of security fallback functions (ex: simswap)
- Users' credulity (ex: Investissement scam)
- Bad investor behaviour (ex: rug pull, high profile doubler scam)
- Physical attacks over people

IMPACT

Main impact is the loss of funds or tokens

EVENT CATALOG

- Dec 2020: false Metamask extension advertised on $Whale community
- Nov 2020: rug pull from Defi Project SharkTron (around 10 M$)
- Feb 2021: Several SIM swap attacks (around100M$)
- 2018: Bitconnect investment scam

EVENT CATALOG

DATA LAYER ATTACKS

The DAO Attack

TECHNIC	Ethereum, Smart Contract attack		
HOLDER	Unknown	YEAR	2016

VICTIM	Slock.it	COUNTRY	Germany
IMPACT	Hard fork of Ethereum into Ethereum (ETH) and Ethereum Classic (ETHC) led to 3,6M$ stolen		

DESCRIPTION	On June 18th, 2016, an unidentified attacker managed to drain 3,6M$ onto a clone of “The DAO” using loophole in the coding of the smart contract named “Reentrancy”. This type of attack takes advantage of the smart contract using what was named “external contract”, which relies upon the smart contract and can be modified by attacker to take control on the transaction and make it act in an unexpected way. In our case, the attacker used two Reentrancy attacks: Reentrancy on a Single Function and Cross-function Reentrancy. Reentrancy on a Single Function consists in calling the same function repeatedly (here the withdrawal function was used), using a flaw in the contract conception which was that the withdrawal balance wasn’t set to 0 before calling an external contract, making it possible to create a loop to withdraw without limit the amount originally stated. Reentrancy on a Single function, similar on a build-based way, use two distinct functions that share the same state (here the transfer function), ultimately leading to a withdrawal of a large quantity of ETH on the smart contract, even if it is not own by the attacker. This attack led to the hard fork of Ethereum, respectively named “Ethereum” and “Ethereum Classic” to correct the issue and to the defunct of “The DAO”
RESSOURCES	David Siegel, “Understanding The DAO Attack”, June 25, 2016 consensy.github.io, “Ethereum Smart Contract Best Practices” Cryptopedia Staff, “What Was The DAO?”, April 27, 2021 Pawel Kurylowicz, “Reentrancy attack in smart contracts – is it still a problem?”, Sept 22, 2021

TECHNIC	Airdrop lottery exploited for a tiny profit		
HOLDER	Researcher for ETH Zurich	YEAR	2017

VICTIM	Fomo3D	COUNTRY	USA
IMPACT	Predict the randomness logic to win the race		

DESCRIPTION	The contract's airdrop lottery can be exploited for a tiny profit. This issue was discovered by Péter SzilágyiIt. Basically, this issue is a combination of two common mistakes: • Attempting to generate a random number in a fully deterministic system. • Making wrong assumptions about how an EVM command should work. The easiest way to predict random numbers based on block data is to call the randomization function from a contract. Every call within a particular transaction is guaranteed to be executed within the same block. So, an attacker can simply duplicate the randomness logic and pre-calculate any random values to check if they can win the race. If a transaction has no chance of winning, the contract can simply revert and let the attacker try again. In order to exploit airdrops in Fomo3D, we need to create a contract that will pre-calculate the “airdrop()” function result. If it has a value of true, we can call the airdrop function in the Fomo3D contract and either trigger an airdrop or revert. Plus, there are several ways we can increase our chances of winning. In particular, we can generate more addresses or make the contract create its own copy and try again with a different starting address instead of simply reverting.
RESSOURCES	Apriorit.com,“Blockchain Vulnerabilities: Fomo3D Exploit”, Aug 18th, 2018 medium.com, “How the winner got Fomo3D prize — A Detailed Explanation”, Aug 23th, 2018

Access control vulnerabilities

TECHNIC	Use coding weakness		
HOLDER	Emin Gün Sirer	YEAR	2020

VICTIM	BitGo	COUNTRY	N/A
IMPACT	Blocking a wallet		

DESCRIPTION	Emin Gün Sirer, a hacker, discovered and disclosed after patch a potential security breach due to error in the code conception of BitGo, a company offering Cold and Hot Wallet solution to people wishing to store their tokens. The flaw was the use of a default (public) identifier for the "tryInsertSequenceId()" method, making it callable by everyone. The problem is, by calling it and setting it close to the maximum value, the wallet will be stuck, unable to take transaction anymore, making the token stored inside stuck indefinitely. This problem was resolved by making the method private. After being notified, BitGo responded that they changed the identifier to perform test and forgot to switch it back. Two things could be remembered: the first one is that Ethereum language, Solidity, use a default-public identifier, making it risky without supplementary attention allocated on the conception phase. Instead of the default-public, Emin Gün Sirer suggested a default-private identifier, making it a lot more secure in case of forgetfulness. The second thing to remember is to have a precise procedure during testing to avoid deploying "test state" code into the public release.
RESSOURCES	Tayvano, "Unprotected function", Feb 20th, 2020 Emin Gün Sirer, "Parity's Wallet Bug is not Alone", Jul 20th, 2017 GitHub.com, "BitGo/eth-multisig-v2", Aug 29th, 2016

NETWORK LAYER ATTACKS

Eclipse attack

TECHNIC	Controlling enough IP addresses to monopolize all connections to and from a victim bitcoin node		
HOLDER	Boston Univ. & MSR Israel	YEAR	2015

VICTIM	N/A	COUNTRY	N/A
IMPACT	Monopolizes all of the victim's incoming and outgoing connections		

DESCRIPTION	The attacker can then filter the victim's view of the blockchain, force the victim to waste computing power on obsolete views of the blockchain, or coopt the victim's computing power for its own nefarious purposes. Eclipse attack uses extremely low-rate TCP connections, so the main challenge for the attacker is to obtain enough IP addresses. We consider two attack types: (1) infrastructure attacks, modeling the threat of an ISP, company, or nation-state that holds several contiguous IP address blocks and seeks to subvert bitcoin by attacking its peer-to-peer network, and (2) botnet attacks, launched by bots with addresses in diverse IP address ranges. Apart from disrupting the bitcoin network or selectively filtering a victim's view of the blockchain, eclipse attacks are a useful building block for Engineering block races, splitting mining power, Selfish mining, 0-confirmation double spend, N-confirmation double spend.
RESSOURCES	Ethan Heilman, Alison Kendler, Aviv Zohar†, Sharon Goldberg "Eclipse Attacks on Bitcoin's Peer-to-Peer Network"

Experimental weakness: Bitcoin Hijacking

TECHNIC	Routing attack (Interior Border Gateway Protocol (iBGP) and the routing rules)		
HOLDER	Researcher for ETH Zurich	YEAR	2017

VICTIM	Low powered miner	COUNTRY	USA
IMPACT	Create partition inside the network to create two distinct blockchains		

DESCRIPTION	Routing attacks tend to target the routing protocol like the Interior Border Gateway Protocol (BGP). Due to the near-impossible character of the delaying routing attack, we will here be interested in a more specific type of routing attack: Partitioning attack. The goal of those types of attacks is to create a partition inside a network by isolating them thanks to BGP Hijacking (create a node which, by his forged IP address, takes the priority into the data forwarding). By isolating them, they become invisible into the network and every information that they receive is filtered and possibly modified by the hijacked nodes. Inside a blockchain, the goal by partitioning the network could be to create multiple “sub-network” without the same data inside their respective blockchain, resulting in a voluntary fork. A research paper conduct by researcher from the ETH Zurich and the Hebrew university tend to demonstrate with test conducted on their own Bitcoin nodes than hijacking 39 prefix is enough to isolate a set of nodes possessing roughly 50% of the network total mining power. By doing that, they warn us that after analysis, those types of hijacking are already influencing the BTC network. Attack like that could create a sort of 51% attack where the powerful isolated partition comes online with a longer blockchain and overwrite the existent blockchain, annulling in the process the not listed transaction leading to double spending attack.
RESSOURCES	Maria Apostolaki | Laurent Vanbever | Aviv Zohar, “Hijacking Bitcoin: Routing Attacks on Cryptocurrencies”, ETH Zurich “Blockchain meets Internet Routing” “Hackers Scoop $20 Million in ETH From Exposed Ethereum Nodes”, June 13th, 2018

CONSENSUS LAYER ATTACKS

Ethereum Classic 51% attack

TECHNIC	Inserting 11 false transactions in the blockchain history		
HOLDER	Unknown	YEAR	2020

VICTIM	Ethereum Classic	COUNTRY	N/A
IMPACT	Attacker was able to get away with more than ~807K ETC (5.6 million $)		

DESCRIPTION	The attacker performed the following action to execute the 51% attack: 1. The attacker withdrew 807K ETC from a Crypto exchange to several wallets. 2. The attacker started mining blocks by purchasing the hash power for double the price. The total cost of mining is approx. 17.5 BTC (~$192,000) 3. The attacker created private transactions, sending money to his/her own wallets, and inserted these transactions in the blocks he/she was mining. No one saw these transactions because the attacker didn't publish the blocks. 4. The attacker sent money back to the Crypto exchange using intermediary wallets on the non-reorganized chain, which was visible to everyone. During this, the attacker had plenty of time to monetize this money—convert to USD and withdraw or change them to BTC, whatever. Long attack duration (12 hours) allowed the attacker to split operations into smaller parts to avoid any suspicion. 5. The attacker published his/her blocks with the version of the transaction created in step #3 and executed the chain re-organization. It means that transactions on step #4 were replaced with transactions on step #3. As this sequence of the block had more weight than the chain built by all other miners, they had to accept these blocks, effectively replacing the blockchain history with attacker's one.
RESSOURCES	bitquery.io,"Ethereum Classic 51% Chain Attack", Aug 2nd, 2020 bitquery.io, "Attacker Stole 807K ETC in Ethereum Classic 51% Attack", Aug 5th, 2020 decrypt.co, "51% Attacks a "Universal Problem" For Proof of Work, says ETC Labs CEO", Sept 7th, 2020 etccooperative.org ,"51% attack on ETC", Aug 2nd, 2020

Selfish mining - Fork After Withholding attack

TECHNIC	Fork After Withholding attack		
HOLDER	Ministry of Science and ICT	YEAR	2017

VICTIM	N/A	COUNTRY	South Korea
IMPACT	Earn unmerited reward for fake mining		

DESCRIPTION	On August 2017, under the MSIT (Ministry of Science and ICT) of South Korea, the ITRC (Information Technology Research) support program and supervised by the IITP (Institute for Information & communications Technology Promotion Center) support program, a research paper was published about selfish mining in the blockchain and its more advanced variant: the Fork After Withholding attack (FAW). Selfish mining is when people, to earn more reward inside a Mining Pool, will withhold a Full Proof of Work (FPoW) and submit it when another person finds a block, to hopefully create a fork that will be validated and earn the reward. While theoretically feasible, selfish mining is highly impractical. Indeed, to be efficient, one needs to have a higher computational power than the target to have better chance to be taken. It's where the researcher, with a modified algorithm of the selfish mining, made it practical and possibly more profitable than the Block Withholding (BW) attack where one submits only partial proof of work to earn unmerited reward. The FAW is based on 3 behaviors and the computing power splitting of the attacker. The attacker will first split his computational power: one part is for the innocent mining, and one is to join a mining pool and generate a FPoW that he will keep inside it. Three figure cases can occur: • The first is when someone exterior to the infiltrated Mining Pool finds a block, the attacker will publish the FPoW, creating a fork. If his fork is chosen, then he earns the reward for finding the block. • The second one is when someone of the targeted mining pool finds the FPoW, the attacker discards his own one and earns the reward for participating in the finding of the FPoW. • The last case is when the attacker finds the FPoW by innocent mining, he publishes it and discards his forged one from the infiltrated Mining Pool, earning the reward for finding the block. With that, the attack is at least as profitable as a BW attack in the second case and third cases but becomes more profitable in the first case, making it globally more profitable. This attack can be performed on a single mining pool like previously described but also on multiple pools and even between pools.Nowadays, without changing the reward system or the crypto currencies architecture, those types of attack do not have reliable counter solution else than the manager administrate his mining pool and cutting the attacker from the pool, which can be a temporary solution.
RESSOURCES	Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, Yongdae Kim, "Be Selfish and Avoid Dilemmas: Fork Afer Withholding (FAW) Attacks on Bitcoin", Aug 31, 2017 Anna Katrenko, Mihail Sotnichek, "Blockchain Attack Vectors: Vulnerabilities of the Most Secure Technology"

GOUVERNANCE ATTACKS

Ethereum fork in 2016 and creation of Ethereum Classic

TECHNIC	N/A		
HOLDER	N/A	YEAR	2016

VICTIM	Ethereum	COUNTRY	N/A
IMPACT	Creation of Ethereum Classic		

DESCRIPTION	Ethereum Classic (ETC) grew out of an ideological and ethical rift in the Ethereum community that provokes controversy to this day. In 2016, a significant hack was carried out on a third-party application running on the Ethereum (ETH) blockchain, which resulted in the theft of millions of dollars'worth of ether, or ETH. In response, the Ethereum blockchain undergoes a hard fork to reverse the hack transaction and remove it from the official ledger and return the stolen ETH to their original owners. In contrast, the other branch of this fork kept the official ledger, that included the hack, unchanged — aiming to preserve a 100% immutable ledger. In other words, the two resulting blockchains differed in only one way: one still contained the record of the hack and the stolen ETH, while the other essentially wound back the clock as if the hack had never happened. The edited blockchain preserved the Ethereum moniker, while the original/unchanged blockchain became known as Ethereum Classic. The controversial split of Ethereum and Ethereum Classic boils down to a philosophical debate which weighs two divergent visions: • A distributed ledger's revised blockchain which was manually altered in a way that erases a successful cybertheft. • A truly immutable blockchain with a permanent record of the network's entire history, including a successful cybertheft. Proponents of Ethereum Classic argue that the ETC hard fork hypocritically enabled the very thing that blockchain technology is meant to prevent — subjective human manipulation. As a result, many idealists stand by Ethereum Classic and its associated cryptocurrency, ETC.
RESSOURCES	"Ethereum classic and the ethereum hard fork", Jun 11th, 2020 "https://blockworks.co/ethereums-hard-fork-is-bound-to-be-implemented-despite-opposition/", Aug 4th, 2021 "https://www.futura-sciences.com/tech/questions-reponses/cryptomonnaies-existe-t-il-deux-ethereum-eth-differences-eth-etc-16037/", Sept 17th, 2021

Beanstalk Farms: Flash loan to obtain majority of decision chair

TECHNIC	Flash loan to obtain a controlling stake in the project		
HOLDER		YEAR	2022

VICTIM	Beanstalk Farms	COUNTRY	N/A
IMPACT	Flash loan to obtain a controlling stake in the project		

DESCRIPTION	An attacker managed to drain around $182 million of cryptocurrency from Beanstalk Farms. Like many other DeFi projects, the creators included a governance mechanism where participants could vote collectively on changes to the code. They would then obtain voting rights in proportion to the value of tokens they held. The attack was made possible by another DeFi product called a “flash loan,” which allows users to borrow large amounts of cryptocurrency for very short periods of time (minutes or even seconds). Flash loans are meant to provide liquidity or take advantage of price arbitrage opportunities but can also be used for more nefarious purposes. According to analysis from blockchain security firm CertiK, the Beanstalk attacker used a flash loan obtained through the decentralized protocol Aave to borrow close to $1 billion in cryptocurrency assets and exchanged these for enough beans to gain a 67 percent voting stake in the project. With this supermajority stake, they were able to approve the execution of code that transferred the assets to their own wallet. The attacker then instantly repaid the flash loan, netting an $80 million profit. Based on the duration of an Aave flash loan, the entire process took place in less than 13 seconds.
RESSOURCES	theverge.com, “Beanstalk cryptocurrency project robbed after hacker votes to send themself $182 million”, Apr 18th, 2022 theregister.com, “https://www.theregister.com/2022/04/18/beanstalk_loses_182m_flash_loan”, Apr 18th, 2022

Terra Blockchain Halted To 'Prevent Attacks'

TECHNIC	N/A		
HOLDER	N/A	YEAR	2022

VICTIM	Terra	COUNTRY	N/A
IMPACT	Service interruption		

DESCRIPTION	The TERRA blockchain has an on-chain Proof Of Stake type of governance. Its related token is the LUNA whose value dropped by 98% on the 9th of May 2022. The managers of the blockchain decided to temporarily stop the block production in order to avoid any rogue takeover of the blockchain. Indeed, Proof Of Stake type of governance means that decisions are likely to be taken by validators with delegation from the biggest token owners. As the LUNA price was very low, malicious actors had the opportunity to operate a massive purchase of a token, delegate their power of decision to a partner in crime and take control of the blockchain. The blockchain was eventually restarted after the new delegations functionality had been disabled.
RESSOURCES	www.forbes.com, "Terra Blockchain Halted To 'Prevent Attacks' After Luna Token Crashes Nearly 100% Overnight", May 12th, 2022 coindesk.com, "Terra Blockchain Resumes Following 9-Hour Halt", May 13th, 2022

Steem hostile takeover and creation of Hive

TECHNIC	hostile takeover using the "ninja mined" tokens		
HOLDER	N/A	ANNÉE	2020

VICTIM	Steem	PAYS	N/A
IMPACT	Creation of Hive		

DESCRIPTION	When he bought Steemit company, Justin Sun acquired a large amount of STEEM, the main token of the Steem blockchain. This amount was "ninja mined" at the creation of the Steem blockchain to allow a control of the blockchain in the event of an attack on this delegated proof of stake blockchain. It had never been used by the Steemit company but was a threat on the decentralization of Steem. Answering to this threat, the historical delegates asked Justin Sun about his intentions. They were not satisfied by the answers and, indeed, Justin Sun initiated a hostile takeover using the "ninja mined" tokens. Thanks to the defense mechanisms of the Steem code base,

	such a takeover could not happen overnight and the historical delegates decided to create a hardfork where the "ninja mined" tokens were transformed into a development fund algorithmically controlled by the community via a voting process. Thus Hive was born. Both blockchains share the same history and the users could decide which one to use, or even both. Most of the historical community moved to Hive, which is now controlled only by the community and much more decentralized than Steem (before or after the fork), nonetheless Steem continues to be used today.
RESSOURCES	https://peakd.com/communityfork/@hiveio/announcing-the-launch-of-hive-blockchain, March 17, 2020 Luke Stokes, https://peakd.com/steem/@lukestokes/to-cz-binance-answers-to-your-twitter-questions-about-steem, March 7, 2020

DECENTRALISED APPLICATIONS

Lazarus Group and the Axie Infinity hack

TECHNIQUE	Compromise of specific validator systems used by the Ronin network

HOLDER	Lazarus Group	YEAR	2022

VICTIM	Axie Infinity	COUNTRY	South Korea
IMPACT	Control of the validation process		

DESCRIPTION	Blockchains based on Ethereum have their own validators. In the case of the Ronin network, there were nine. To exert control over a Blockchain you can conduct what is called a 51% attack. If you control 51% of the validators available on a network, you control the consensus and you control which transactions are validated. This is likely what occurred at Axie with the attackers issuing forged transactions to the Ronin bridge and validating them using the five validator nodes they controlled. The attackers at this point withdrew the 173,600 ETH and 25.5m in USD Coin (USDC) that were 'frozen' inside the Ronin bridge smart contract out into the Ethereum network. Not all the attacks on the validator nodes were identical. The attackers compromised the private keys of four nodes and attacked a specific feature of the fifth decentralized node. Several underlying issues allowed the attack to succeed. A small set of validators makes a 51% attack easier to conduct. The network's small-scale leads to a centralization of validator nodes within the decentralized system. This point played against network security. It's a pure numbers game, fewer validators in total, less to get to the 51% required. It is reported that several of the validator nodes were operated by the same entity, in the same region of the world. This would have made it much easier for the attackers, who only needed to compromise that entity and its systems.
RESSOURCES	thisweekincryptofraud.substack.com,"Lazarus Group and the Axie Infinity hack", May 4th, 2022 idstrong.com , "Lazarus Hackers Responsible for Million Axie Infinity Attack", Apr 18th, 2022

Wyvern Protocol

TECHNIQUE	Use a deprecated method		
HOLDER	Unknown	YEAR	2022

VICTIM	Opensea NFT owners	COUNTRY	N/A
IMPACT	32 users had been affected and stolen		

DESCRIPTION	The attack appears to have exploited a flexibility in the Wyvern Protocol, the open-source standard underlying most NFT smart contracts, including those made on OpenSea. One explanation (linked by CEO Devin Finzer on Twitter) described the attack in two parts: 1 - targets signed a partial contract, with a general authorization and large portions left blank. 2 - with the signature in place, attackers completed the contract with a call to their own contract, which transferred ownership of the NFTs without payment. In essence, targets of the attack had signed a blank check — and once it was signed, attackers filled in the rest of the check to take their holdings.
RESSOURCES	theverge.com, "$1.7 million in NFTs stolen in apparent phishing attack on OpenSea users", Feb 20th, 2022 cnet.com, "OpenSea Says at Least $1.7M in NFTs Stolen in Phishing Attack", Feb 21st, 2022 Nadav Hollnder, "https://twitter.com/NadavAHollander/status/1495509511179755530" Feb 20th, 2022

Hot wallet attack: BitMart

TECHNIQUE	Protocol lack of control		
HOLDER	Unknown	YEAR	2022
VICTIM	BitMart	COUNTRY	South Africa
IMPACT			
DESCRIPTION	An attack was launched in January 2022 on the hot wallet of BitMart, an exchange platform. This attack, discovered by Peckshield, a blockchain security and auditing company, targeted the Ethereum (ETH) and Binance Smart Chain (BSC). The amount stolen by the cyber-criminal was first estimated to 150 million dollars, but Peckshield's instigation raise the loss to 200 million dollars. Speckshield investigation determined that the attacker exchanged every ETH and BSC stolen by ETH on the exchange site *1inch* for then sending the ETH on Tornado.cash, a protocol enabling the deposit of ETH and the withdrawals with another address even without ETH balance, making it near-impossible to link the sender and the receiver.		
RESSOURCES	Sergio Gochenko, "Bitmart Loses $200 Million in Hack Performed by Unknown Attackers", Dec 6h, 2021 Jamie Redman, "Privacy-Centric Crypto Mixing Protocol Tornado.cash Plans to Deploy on L2 Platform Arbitrum", Nov 29th, 2021		

CENTRALISED APPLICATIONS

Cream Finance

TECHNIQUE	Cream Finance attack consisted of a flash loan transaction leveraging a price oracle vulnerability in the Cream Finance protocol		
HOLDER		YEAR	N/A

VICTIM	Cream Finance	COUNTRY	N/A
IMPACT	Manipulate the price of an asset		

DESCRIPTION	Cream Finance is a decentralized protocol that provides lending and borrowing capabilities in a permissionless manner. Cream has a lending pool where you can provide liquidity with yUSD tokens, as well as use these yUSD tokens as collateral to borrow other assets. The hacker used a flash loan attack that took advantage of a badly implemented oracle price proxy. The oracle proxy calculated the pricePerShare using on-chain calls in 4Pool and yUSD contracts. The attacker sent a token to the contract address directly instead of passing through the defined contract calls that keep track of the accounting properly. This allowed the attacker to manipulate the price, therefore using yUSD to borrow from many markets.
RESSOURCES	Medium.com, "Understanding the Cream Finance Hack", Oct 29th, 2021

OpenSea attack : buying at older and cheaper prices

TECHNIQUE	Backend vulnerability to buy products at previous prices		
HOLDER	Unknown	YEAR	2022

VICTIM	Opensea	COUNTRY	N/A
IMPACT	Ability to buy products at previous (lower) prices and resell them, defrauding legitimate asset owners		

DESCRIPTION	A threat actor has exploited a vulnerability in the backend of OpenSea, the internet's largest NFT marketplace, to buy products at previous (lower) prices and then resell them at higher values, defrauding legitimate asset owners. The exploit appears to originate from the ability to re-list an NFT at a new price without cancelling the previous listing. Those previous listings are now being used to purchase NFTs at prices specified at some point in the past -- which is often well below current market prices. DeFi developer Rotem Yakir released a detailed thread on Twitter explaining the OpenSea bug, writing that it "stems from the fact that previously you could re-list an NFT without canceling it (which you can't now) and all the previous listing are not canceled on-chain." "Previously, you could have re-listed an NFT without canceling the previous list. Sometimes but not always, if you cancel your new listing, the old one will not appear on the UI but is still valid,"
RESSOURCES	Catalin Cimpanu, "Hacker abuses OpenSea to buy NFTs at older, cheaper prices", Jan 24th, 2022 Coindesk.com, "OpenSea Bug Allows Attackers to Get Massive Discount on Popular NFTs", Jan 24th, 2022 Rotem Yakir, "https://twitter.com/yakirrotem/status/1485559864948629512", Jan 22th, 2022

WALLET (HARDWARE AND/OR SOFTWARE)

Harmony's Horizon Bridge Hack

TECHNIC	Private key theft for approving transaction		
HOLDER	Possibly Lazarus Group	YEAR	2022

VICTIM	Harmony	COUNTRY	
IMPACT	$100 million was stolen from Harmony Bridge among more than 10 crypto coins.		

DESCRIPTION	This attack is placed in the top 10 most expensive DeFi hack. The bridge used to need only 2 of 5 validators to approve any transaction. After having initiated multiple transactions of diverse crypto currencies, the hacker stole 2 validators' private keys and managed to decrypt it. With those two validation accounts, he managed to initiate and approve a 100 million dollar transaction. He then swapped those stolen coins for ETH using decentralized exchanges through Tornado Cash A research linked the Lazarus Group to this attack, because of the similarities between that attack and other ones perpetrated by the North-Korean group.
RESSOURCES	TechCrunch, "Hack exploits Harmony Blockchain Bridge", June 2022 Medium, "Harmony's Horizon Bridge Hack", June 2022

Trezor Hardware Wallet's Hack

TECHNIC	Using a critical vulnerability in Trezor One and Trezor Model T to extract and crack seed phrases.		
HOLDER	Kraken's security experts	YEAR	2019

VICTIM	Trezor Hardware Wallet	COUNTRY	N/A
IMPACT			

DESCRIPTION	From all the different types of wallets, hardware wallets are considered one of the most secure for two reasons: they are not always connected to the internet, reducing the component exposure to potential attacks as well as the need for the physical device to perform any transfer of funds. However, in October of 2019, Kraken Security Labs disclosed to the Trezor team the result of their successful pentesting. With physical access to the hardware wallet. The team was able to obtain the private key holding the funds in less than 15 min. The vulnerabilities found are attributed to the hardware microcontroller used by the wallet, to secure the private keys. The goal of the attack is to extract the private key from the flash memory of the microcontroller. To reach their goal the white hat team exploited known vulnerabilities of the microcontrollers. Notably using voltage glitch allowed them to turn the microcontroller into debugger mode. With such mode activated, it was possible to extract data from the flash memory, such as the encrypted key. Finally, to decrypt the private key, it is necessary to brute force it with a 4-digit pin code, which took 2 min. Even if the wallet showed vulnerabilities, no report of stolen funds was made.
RESSOURCES	Marko Vidrih, "Trezor Hardware Wallet Hacked in 15 Minutes", August 28th ,2021 KRAKENFX, "Kraken Identifies Critical Flaw in Trezor Hardware Wallets, January 31st, 2020 Joe Grand (Youtube)"How I hacked a hardware crypto wallet and recovered $2 million.",January 24th, 2022

WEB APPLICATION INTERFACE

CoinDash ICO Hack

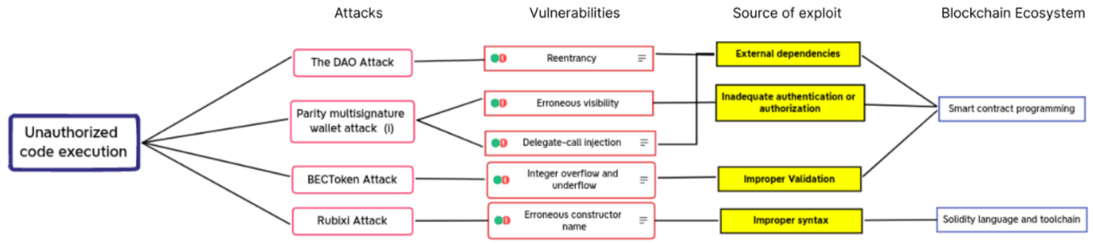
TECHNIC			
HOLDER	Unknown	YEAR	2017

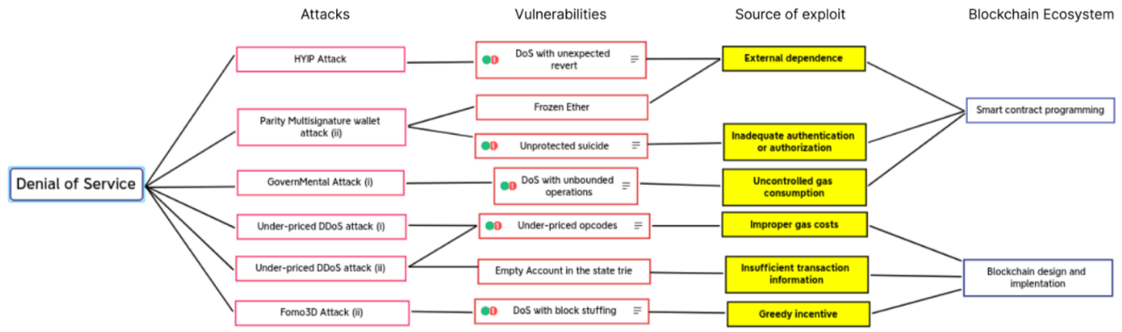
VICTIM	CoinDash	COUNTRY	Israël
IMPACT	43 500 ethers, equivalent to 7 million US Dollars at the time were stolen from investors		

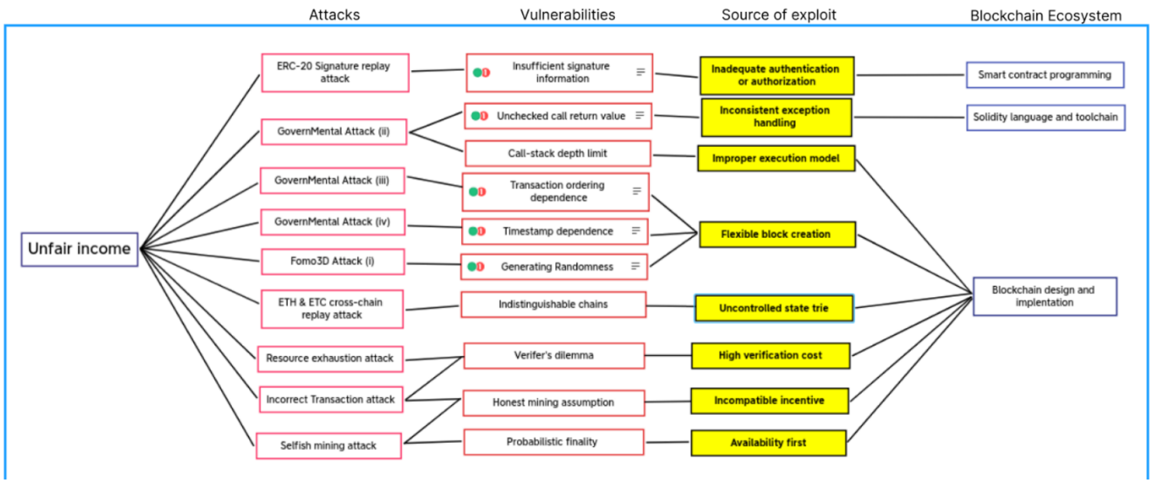
DESCRIPTION	CoinDash is a blockchain startup founded in 2016 that has guidelines to help democratize blockchain and crypto currencies by selling tools to make it more user friendly. During its Initial Coin Offering stage (ICO), CoinDash has been hacked by an unknown perpetrator. The cybercriminal tempered with the donation website, changing the receiver donation address. It resulted in $7M equivalent in Ethereum being stolen in 13 minutes before CoinDash closed the funding. The cybercriminal was able to take advantage of a zero-day vulnerability posing the question of website security. Indeed, their website was a wordpress website, easy to create but requiring further step before being properly secured. To calm the community anger, CoinDash gave investors the CDT coin that they should have received even if the fund were stolen. This attack gives us an insight on the need to secure any gateway to the blockchain because it's always the weakest element of a network that makes the overall network security.
RESSOURCES	DailyPriyab, "ICO Hack — CoinDash-ed", Jul 17th , 2017 Wolfie Zhao, "$7 Million Lost in CoinDash ICO Hack", Jul 17th, 2017 Stuart D. Levi, "Lessons From the CoinDash Initial Coin Offering Hack", Jul 19th, 2017

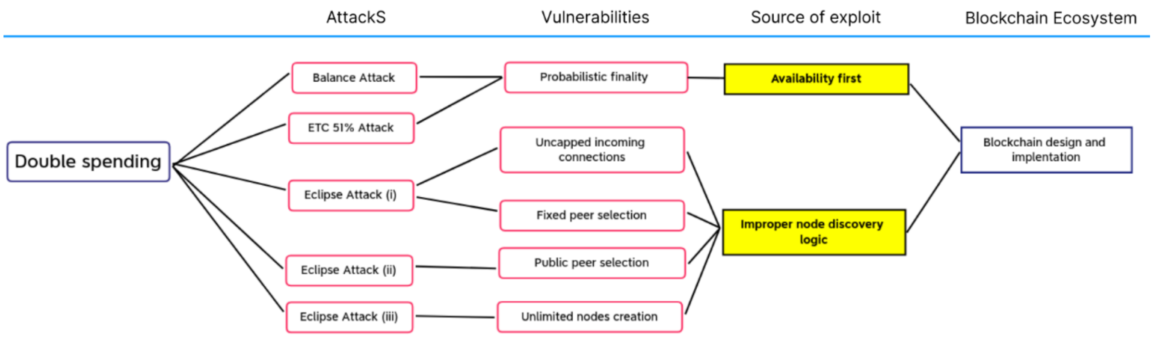
APPENDIX

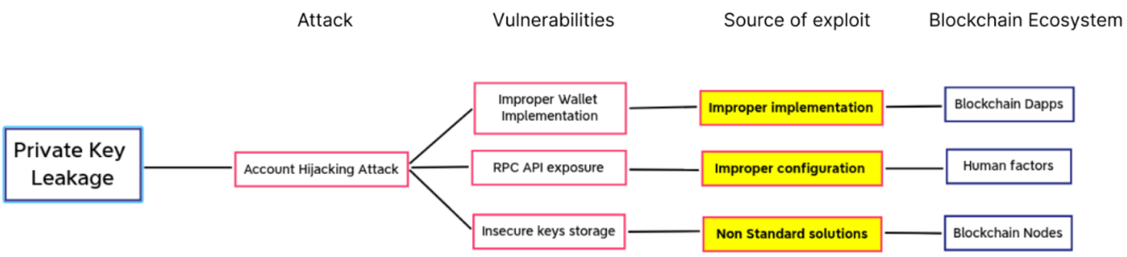
VULNERABILITIES PER CATEGORY











VULNERABILITIES PER COMPONENT – CASE OF ETHEREUM

Ethereum data layer vulnerabilities related to smart contract development:

Blockchain Component	Vulnerabilities	Blockchain Ecosystem
Ethereum Data Layer	Insufficient signature information Leaking Ether to arbitrary address Authentication though tx. origin Integer overflow and underflow Reentrancy Delegate-call injection Frozen Ether Upgradeable contracts DoS with unexpected revert Shadowing State Variables Typographical Error Requirement Violation Presence of unused variables Manipulated balance Missing Protection against Signature Replay Attacks Signature Malleability Hash Collisions With Multiple Variable Length Arguments Erroneous visibility State Variable Default Visibility Function Default Visibility Right-To-Left-Override control character (U+202E) Unprotected suicide Confidentiality failure DoS with unbounded operations	Smart contract programming

Ethereum data layer vulnerabilities related to Solidity programming language and toolchain

Blockchain Component	Vulnerabilities	Blockchain Ecosystem
Ethereum Data Layer	Unchecked call return value Incorrect Inheritance Order Uninitialized storage pointer Erroneous constructor name Type casts Floating Pragma Outdated Compiler Use of Deprecated Solidity Functions	Solidity Language and Toolchain

Ethereum data layer vulnerabilities related to Ethereum design and implementation:

Blockchain Component	Vulnerabilities	Blockchain Ecosystem
Ethereum Data Layer	Short Address Ether lost to orphan address Assert Violation Call-stack depth limit Write to arbitrary storage location Under-priced opcodes Transaction ordering dependence Empty Account in the state trie Indistinguishable chains	Ethereum design and implentation

Ethereum network layer vulnerabilities related to Ethereum design and implementation:

Blockchain Component	Vulnerabilities	Blockchain Ecosystem
Ethereum network layer	RPC API exposure Sole block synchronization Fixed peer selection Public peer selection Uncapped incoming connections Unlimited nodes creation	Ethereum design and implentation

Ethereum consensus layer vulnerabilities related to Ethereum design and implementation:

Blockchain Component	Vulnerabilities	Blockchain Ecosystem
Ethereum consensus layer	Outsourceable puzzle DoS with block stuffing Rewards for uncle blocks Verifer's dilemma Honest mining assumption Probabilistic finality	Ethereum design and implentation

CSA - BLOCKCHAIN WEAKNESS CATEGORIZATION

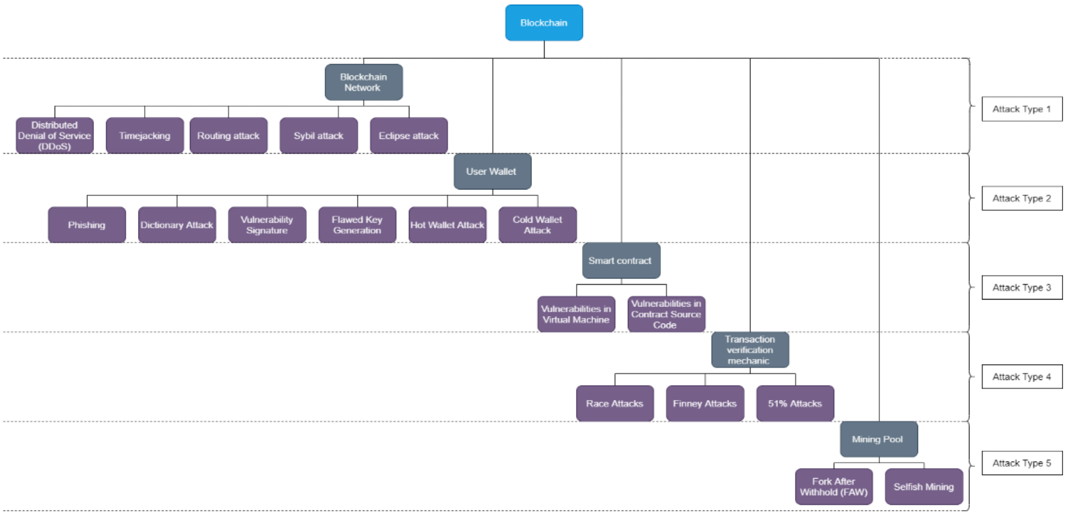
Cloud Security Alliance (CSA)[2] has documented a list of 200 weaknesses and the Common Weakness Enumeration (CWE) has referenced smart contract weaknesses under the name SWC Registry[3] (Smart contract Weakness

[2] CSA documentation: https://docs.google.com/spreadsheets/d/1HIM3BH8Cgth27ED4ruy9fXOpbOUAPAGY7merlZiE6_U/edit#gid=1028635246

[3] SWC registrary: https://swcregistry.io

Classification). The proposed catalog of attacks above uses these inputs to present a global categorization of attacks. The Catalog of attacks that we wrote has the purpose to document some of the most common and therefore used vulnerabilities to be better prepared to react and prevent them.

Figure 3 : The different types of attacks against blockchain



THANKS

This document has been created by the crypto-actif workgroup. Campus Cyber warmly thanks all the contributors

Workgroup coordinators

- Jean-Loic Mugnier – Niftag
- Stefane Mouille – Cabinet Louis Reynaud
- Christophe Pelfresne – Banque de France

Workgroup Contributors

- Alexandre Chopin – BNP Paribas
- Stephan Cohen – BNP Paribas
- Pierre Boulet – Université de Lille
- Paul Gedeon – Red Alert Labs
- Thomas Benoit – Set In Stone
- Thierry Desamblanc – Enedis

CAMPUS CYBER
5-7 rue Bellini
92 800 La Défense
Campuscyber.fr

Contact@campuscyber.fr

Retrouvez nos productions : campuscyber.fr/communs/