Outline

➢ Motivation.
➢ Overview of cryptocurrencies.
➢ Cryptocurrency and distributed services.
➢ ABC: a cryptocurrency-focused threat modeling framework.
➢ Conclusions.
Once Upon A Time
Centralized Currency
Decentralized Currency
But, is it secure?
Bitcoin History

  - By Satoshi Nakamoto.
  - Described a distributed cryptocurrency system not regulated by any government.
- The system went live in January 2009.
- Now “Satoshi Nakamoto” is only associated with certain public keys on Bitcoin blockchain.
  - She/He/They was/were active on forums/emails till 2010.
- Currently there are 2000+ cryptocurrencies (https://coinmarketcap.com/).
Bitcoin in a Nutshell I

- A distributed currency exchange medium open to anyone to join.
- Utilize basic cryptographic primitives to control the money flow in the system.
- Main components:
  - **Participants**: miners and clients.
  - **Transactions**: virtual currency tokens that move funds around.
  - **Blockchain**: an append only log.
  - **Mining**: extending the blockchain.
  - **Consensus**: agreeing on the current state of the Blockchain.
Bitcoin in a Nutshell II

- No real identities are required, just a key pair.
  - Usually the hash of the public key is the participant’s address.
- Losing the private key of a specific address means losing the coins associated to this address forever.
  - Wallets take care of tracking coins, issuing transactions, etc.
- Clients, or simple payment verification (SPV) nodes, are concerned with their transactions only.
  - Do not mine or hold full copies of the blockchain.
- Miners, or fully validating nodes, track everything.
  - Hold full copies of the blockchain.
  - Mine and run the consensus protocol.
Bitcoin Pictorially
Virtual Coins

- Digital tokens, or transactions, that can be spent by providing signatures.
- No notion of accounts, track chains of transactions.
  - Other cryptocurrencies do it differently, e.g., Ethereum have accounts for users.

The Blockchain

- It is an append only log containing a full record of all transactions.
  - Full history is needed to handle double spending.
Mining

- Miners extend the blockchain by mining new blocks.
  - Proof-of-work in Bitcoin.
- Miners solve a hash puzzle,

\[ \text{SHA-256(SHA-256 (new block header))} < \text{Difficulty Target} \]

- Difficulty is adjusted periodically.
- This is needed to prevent Sybil attacks.
- Miners collect rewards: mining rewards + transaction fees.
- Total Bitcoin to mine is capped by 21 million BTC.
  - Currently there are around 17.4 million coins in circulation.
**Consensus**

- Miners hold, hopefully, consistent copies of the blockchain.
  - Only differ in the recent unconfirmed blocks.
- A miner votes for a block implicitly by building on top of it.
- Forking the blockchain means that miners work on different branches
  - Caused by network propagation delays, adversarial actions, etc.
  - Resolved by adopting the longest branch.
- Security is subject to the assumption that at least 50% of the mining power is honest.

Bitcoin Sisters

- Bitcoin has several limitations related to:
  - Supported functionality, anonymity, transaction fees, mining and consensus, scalability, etc.
- This motivated the community to develop new cryptocurrencies, to name few:
  - Ethereum, Litecoin, Zcash, IOTA, Storj, Golem, etc.
- Will focus more on systems that provide distributed services on top of the currency exchange medium.
Cryptocurrency and Distributed Services
System Types

- Two types of cryptocurrency integration with distributed services:
  - **Resource-backed cryptocurrencies.**
    - Provides a distributed services(s) on top of the currency exchange medium, e.g., file storage, video transcoding, etc.
    - Mining itself could be tied to the service put in the system, e.g., Filecoin.
  - **Monetary-incentivised distributed systems.**
    - Reward users for participation in the system.
    - Use a cryptocurrency as a decentralized payment service instead of centralized ones.
    - Not concerned with the details of the cryptocurrency system.
Goals

- Achieve decentralization.
- An advancing step toward useful proof-of-work mining algorithms.
  - Provide a public good.
  - Reduce energy consumption.
  - Achieve decentralized mining (think about mining pools).
Threat Modeling and Cryptocurrencies

- Threat modeling is an essential step in secure systems design.
  - Explore the threat space to a system and identify the potential attack scenarios.
  - Helps in both guiding the system design, and evaluating the security of developed systems.
- Cryptocurrency/blockchain-based space experienced a huge number of attacks.
  - Financial incentives lead to more motivated attackers.
  - Security is more challenging in resource-backed cryptocurrency.
ABC: Asset-Based Cryptocurrency-focused Threat Modeling Framework
What is ABC?

- A systematic threat modeling framework geared toward cryptocurrency-based systems.
  - Its tools are useful for any distributed system.
- Helps designers to focus on:
  - Financial motivation of attackers.
  - New asset types in cryptocurrencies.
  - Deriving system-specific threat categories.
  - Spotting collusion and managing the complexity of the threat space.
    - using a new tool called a collusion matrix.
- Integrates with other steps of a system design; risk management and threat mitigation.
ABC Steps

Running Example: **CompuCoin**

- A cryptocurrency that provides a distributed computation outsourcing service.
- Parties with excessive CPU power may join as servers to perform computations for others in exchange for CompuCoin tokens.
- The mining process is tied to the amount of service these servers provide.
Step 1: System Model Characterization

- Identify the following:
  - Activities in the system.
  - Participant roles.
  - Assets.
  - Any external dependencies on other services.
  - System assumptions.
- Draw a network diagram(s) of the system modules.

**Functionality description.** Outlined in CompuCoin description introduced earlier.

**Participants.** Clients and servers.

**Dependencies.** May rely on a verifiable computation outsourcing protocol.

**Assets.** Computation service, service rewards (or payments), blockchain, currency, transactions, and the communication network.
Step 2: Threat Category Identification

- Define broad threat classes that must be investigated.
- ABC defines these classes around the assets.
- For each asset, do the following:
  - Define what constitute a secure behaviour for the asset.
  - Use that knowledge to derive the asset security requirements.
  - Define threat classes as violations of these requirements.
## Step 2: Running Example Application

<table>
<thead>
<tr>
<th>Asset</th>
<th>Security Threat Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>Service corruption (provide corrupted service for clients).</td>
</tr>
<tr>
<td></td>
<td>Denial of service (make the service unavailable to legitimate users).</td>
</tr>
<tr>
<td></td>
<td>Information disclosure (service content/related data are public).</td>
</tr>
<tr>
<td></td>
<td>Repudiation (the server can deny a service it delivered).</td>
</tr>
<tr>
<td>Service payments</td>
<td>Service slacking (a server collects payments without performing all the promised work).</td>
</tr>
<tr>
<td></td>
<td>Service theft (a client obtains correct service for a lower payment than the agreed upon amount).</td>
</tr>
<tr>
<td>Blockchain</td>
<td>Inconsistency (honest miners hold copies of the blockchain that may differ beyond the unconfirmed blocks).</td>
</tr>
<tr>
<td></td>
<td>Invalid blocks adoption (the blockchain contains invalid blocks that does not follow the system specifications).</td>
</tr>
<tr>
<td></td>
<td>Biased mining (a miner pretends to expend the needed resources for mining to be elected to extend the blockchain).</td>
</tr>
<tr>
<td>Transactions</td>
<td>Repudiation (an attacker denies issuing transactions).</td>
</tr>
<tr>
<td></td>
<td>Tampering (an attacker manipulates the transactions in the system).</td>
</tr>
<tr>
<td></td>
<td>Deanonymization (an attacker exploits transaction linkability and violates users’ anonymity).</td>
</tr>
<tr>
<td>Currency</td>
<td>Currency theft (an attacker steals currency from others in the system).</td>
</tr>
<tr>
<td>Communication network</td>
<td>Denial of service (interrupt the operation of the underlying network).</td>
</tr>
</tbody>
</table>
Step 3: Threat Scenario Enumeration and Reduction

- For each threat, define scenarios that attackers may follow to pursue their goals.
  - Be comprehensive, consider collusion and financial motivation.
- ABC devises collusion matrices to help with this step.
- Analyzing a collusion matrix involves:
  - Enumerating all possible attack scenarios.
  - Crossing out irrelevant cases and merge together those that have the same effect.
  - Documenting all distilled threat scenarios.
Collusion Matrix

(a) External System

Client (Alice) → Service request → Service → Payment → Server (Bob)

(b) Service Theft Threat

<table>
<thead>
<tr>
<th>Targets</th>
<th>Client</th>
<th>Server</th>
<th>Client and Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client and Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client and External</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server and External</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client, Server, and External</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Service Theft Matrix

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
<th>Client and Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
## Step 2: Running Example Application

### Service Theft Threat Collusion Matrix

<table>
<thead>
<tr>
<th>Attacker</th>
<th>Target</th>
<th>Client</th>
<th>Server</th>
<th>Client and Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Server</td>
<td>Clients cannot be targets because they do not serve others.</td>
<td>Servers and external cannot attack because they do not ask/pay for service.</td>
<td>Reduced to the case of attacking servers only, clients do not serve others (cannot be targets).</td>
</tr>
<tr>
<td>Server</td>
<td>Client and External</td>
<td></td>
<td>(1) Refuse to pay after receiving the service. (2) Issue invalid payments.</td>
<td></td>
</tr>
<tr>
<td>Server and External</td>
<td>Client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client</td>
<td>Client and External</td>
<td></td>
<td>Reduced to the case of an attacker client. A client does not become stronger when colluding with other servers or external entities.</td>
<td></td>
</tr>
<tr>
<td>Client and External</td>
<td>Server and Client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client, Server, and External</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 4: Risk Management and Threat Mitigation

- An independent task of threat modeling.
- However, financial incentives affect prioritizing threats and their mitigation techniques.
  - Use game theory-based analysis to quantify the pay-off an attacker may obtain.
  - Use detect-and-punish techniques to address certain threat types.
- For example, in CompuCoin:
  - Locking payments in an escrow neutralizes threat 1.
  - Having a penalty deposit that is fortified upon cheating addresses threat 2.
  - Both require careful design and economic analysis.
Last Stop
Conclusions

- Cryptocurrencies provide a disruptive work model.
  - But also exhibit complicated relations between, financially motivated, untrusted parties.
- Great potential and huge arena of applications.
  - However, deeper thinking is needed to assess when/where to apply.
  - Threat modeling is a critical step to enhance their security.
- Are they just a hype that will fade away?!
  - Still provide an elegant proof of concept.
Questions?

aNd ThANk yOU :)