Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Peer-to-Peer Systems and Security

Chapter 2
2.2 Attacks and Attack Mitigation

Dipl.-Inform. Heiko Niedermayer
Dipl.-Inform. Ralph Holz
Prof. Dr. Georg Carle

Vorlesung SS 2009

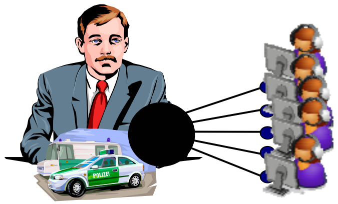
Overview

- ❑ Motivation
- ❑ Attacks
  - Overview
  - Routing Attacks
  - Poisoning Attacks
  - Sybil Attack
  - Eclipse Attack
- ❑ Conclusion

Motivation

Standard client/server assumption

- ❑ A „smart server“ enforces security.



Peer-to-Peer / Decentralized networks

- ❑ In pure form, no „smart server“
- ❑ Then who can enforce security?
  - All members?
  - How to reach decisions?
- ❑ Also means lack of TTP

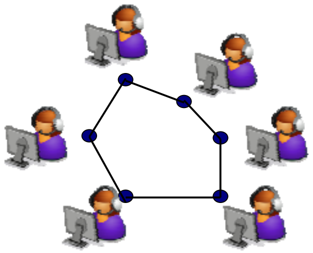
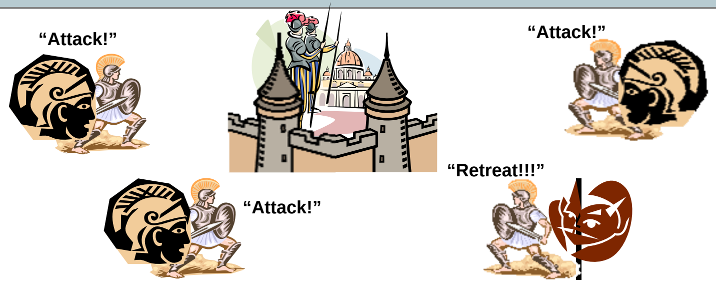


Illustration: Byzantine Generals Problem



- ❑ How can you decide what to do when you cannot trust the information you receive? – Byzantine Generals Problem
- ❑ Byzantine armies besiege a city and must decide to attack or not.
  - If only a small number of armies attack, they are lost.
  - Some generals may be traitors; they try to trick the others into a false decision. Wanted: secure protocol that allows to reach the correct agreement.
- ❑ Proven: if more than 1/3 of generals are traitors, there cannot be such a protocol.

Attacks – Overview

Attacks can take many different forms.

What can be attacked?

- Routing between nodes
- Storage
- Service Quality
- Behaviour / Participation
- Existence of network itself
- …

Attacks on Routing

Routing attacks

- Misroute messages
  - Change target while forwarding
  - Either randomized or according to some plan
- Drop messages
- Propagate wrong information for routing
  - See Routing Table Poisoning
- etc.

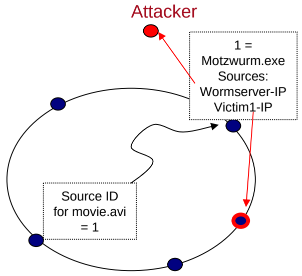
Defenses

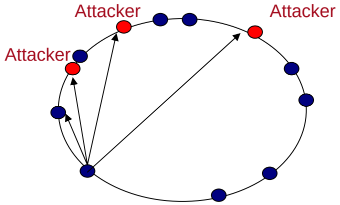
- Use iterative lookups instead of recursive lookup
  - Kademlia uses iterative lookups; Chord uses recursive lookup
- Routing via multiple paths
- Check if certain constraints are being met (→ get closer with each hop)
- …

Poisoning attacks

Poisoning attack

- Attackers use false information to break the integrity of the system.
- Index Poisoning
  - Store bogus information in the DHT
    - E.g. links to nodes that do not have a file, redirect nodes requesting an item to attacker nodes, link meta-information to wrong item.
- File Poisoning
  - Spamming the network with fake and corrupted files.
- Routing Table Poisoning
  - Add attacker nodes to the routing table of a node, e.g. using the knowledge of structural constraints and mechanisms in DHTs.
  - Interesting for surveilling a node or denial of service.





Attacks and Identities

Often, attackers use the convenient position of their own NodeID to stage the attack.

→ Attackers shouldn‘t be able to choose their own position too easily.

→ Secure and verifiable NodeIDs

Authentication in decentralized networks is a problem… as we know.

Limiting identities even more…

- Maybe limited by identity = hash(IP address)?
  - IP spoofing raises some barriers for the attacker.
- A server as Authority and Identity Provider? → still no real limit, limit IDs on what information?
  - Payment server?

Certain attacks are based on the problem of verifying and limiting identities.

- Sybil attack
- Eclipse attack

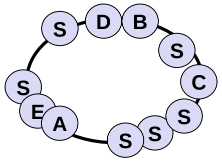
Sybil Attack

Background

- 1973: Flora Rheta Schreiber publishes her book „Sybil“. Sybil is a woman with 16 separate personalities.

The Sybil Attack

- Insert a node multiple times into a network, each time with a different identity
- Potential Goals
  - Helps to perform other attacks and to position a node for particular attacks like Routing Table Poisoning
  - Attack connectivity of the network
  - Attack replica set
  - In case of majority votes, be the majority.
- The Sybil attack is an efficient attack against Peer-to-Peer and other decentralized networks.



Here, node S is in the network with 6 different identities.
→ Sybil attack

Sybil Attack

Can authentication help?

- Only if identities cannot be created (cheaply).
- Otherwise, simply create many identities and authenticate yourself with any of your identities.

Limit the number of identities?

- Use real physical identities
  - Who enters the data?
    - Anyone can register with nonsense
  - Limit to IP address or IP:Port?
    - But many nodes behind a NAT possible
    - IP:Port allows 1000s of identities per IP (A real limit?)
- Use external identities?
  - Limit to email adresses?
    - A real limit?
- Make it costly to create identity?
  - Solve computational puzzles
- Make people pay money for registration
  - Step backwards towards central server.

Sybil Attack – Work by Douceur

„One can have, some claim, as many electronic personas as one has time and energy to create.“

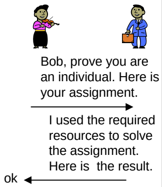
Judith S. Donath 1998 in a work on virtual communities

John Douceur introduced the Sybil attack together with a formal analysis of the problem.

Basic assumptions

- Communication, storage, and computational resources are limited
- Now assume that any entity has to prove its identity by providing a certain amount of resources (→ Proof-of-Work). The proof is fulfilled by presenting a bitstring proving the work.
- Any such constraint on a system has to be small enough so that the *minimal capable entity* can also prove its identity.

Observation: → a strong entity can provide enough resources for multiple identities


Bob, prove you are an individual. Here is your assignment.

I used the required resources to solve the assignment. Here is the result.

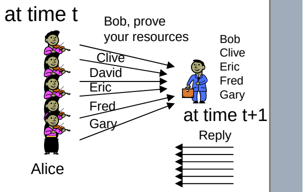
way more resources (CPU, RAM, bandwidth)

Sybil Attack – Work by Douceur (Results)

Case 1: Direct identity validation

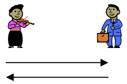
- Identity has to be proven to each peer directly.
- Case 1a: Check all identities simultaneously
  - To accept, Alice challenges all identities at time t.

  → Attacker is limited to as many faulty identities as it outperforms the minimal capable entity, because it has to prove all of them.



- Case 1b: Sequential checking of identites
  - To accept an identity, Alice challenges it at some time.

  → At any time, only the resources for the minimal capable entity are checked. Thus, an arbitrary number of faulty identites can be created.

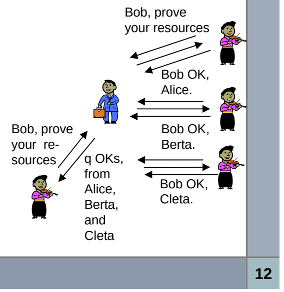


Case 2: Indirect identity validation

- Identity is accepted if q other identities accepted it or it is proven as in case 1.
- Case 2a: All entities check all identities simultaneously

  → Arbitrary number of identities can be obtained if either the number of faulty identities f is larger than q, or the attacker strong enough.
- Case 2b: Entities do not coordinate, so each entity checks all identities at other points in time.

  → Even a minimally capable attacker can support multiple identities (~ prove each identity only to q others, to each other entities only one identity is proven (partitions of size q), use them to prove to rest, and hope for non-overlaps).


Bob, prove your resources
Bob OK, Alice.
Bob OK, Berta.
Bob OK, Cleta.
Bob, prove your re-sources
q OKs, from Alice, Berta, and Cleta

Sybil Attack – Work by Douceur (Conclusion)

Concluding the results by Douceur

- Using resource limitations to defeat the sybil attack requires conditions that are extreme and unrealistic.
  - All entitities operate under nearly identical constraints.
  - Simultaneous check of all identities, across the entire system.
  - In case of indirect validation, q > the number of system-wide failures / attackers.
- Another issue is that proof-of-work approaches waste a lot of resources.

→ Without a central authority that certifies identities (binding real-world person to nodeID), no realistic approach exists to completely stop the Sybil attack.

Usage of external identifiers

Barier
for people to join
and enter the network
Yes = large, maybe
too large for success
of the network
No = small

	Barrier for participation	Verification	Can Limit Sybil attack
Central	Yes	Yes	Yes
Decentralized, IDs determined by external factors	**No**	**Yes**	**?**
Decentralized, freely chosen IDs	No	Yes	No

External identifiers

- IPv4 address
  - Multiple nodes behind NAT → same ID, necessary to allow a set of nodes
- IPv6 address
  - Due to huge address space and privacy options no real limit → only use the first bits and then allow one or a small number of nodes with same prefix.
- …

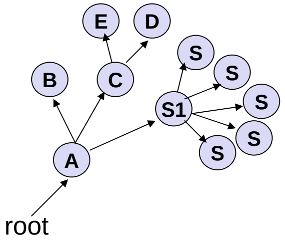
Some assumptions about Sybil Attacks

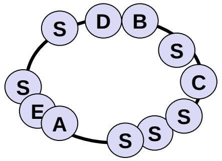
Assumption about the bootstrapping

- The first Sybil node enters via an arbitrary bootstrap node.
- The rest of the nodes will prefer to join via another sybil node.

Bootstrap tree

- Tree where nodes are a child of the node they used to bootstrap.
- In the tree below, A would have been the first node. B,C, and the first sybil S1 joined via A. The rest of the Sybil nodes join via S1.

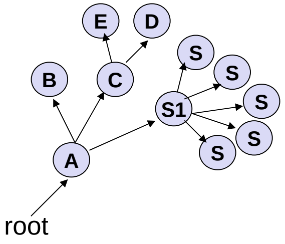




Some assumptions about Sybil Attacks

Sybil nodes and the bootstrap graph

- Ron Anderson et. al argue that the properties of the bootstrap graph can be used to route around sybil nodes.
  - Basic idea: Iterative queries using nodes from different subtrees in the bootstrap graph along with nodes closer to the target.
- However, if the bootstrap node is not enforcing any access control policies or is based on social relationships, there is no need for sybil nodes to join via each other.



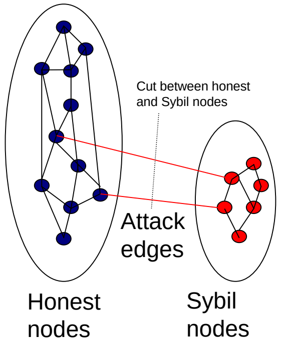
Fighting the Sybil Attack with social networks

Social networks

- Nodes are (real-world) identities and edges are social relationships between these identities (e.g. knows, trusts, is friend with).

SybilGuard

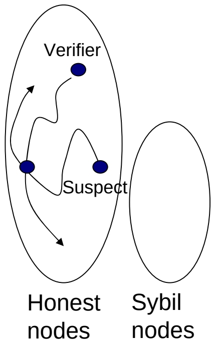
- Assumption
  - Sybil nodes primarily know each other.
  - Since they correspond to only few real-world personas, their cluster will have fewer edges to other clusters than the clusters of honest nodes.
    → Small cut between the subgraph of honest nodes and the subgraph of sybil nodes.



Fighting the Sybil Attack with social networks

SybilGuard

- Basic idea
  - Use the overlap of random routes to determine if a node is in the subgraph of honest nodes or in the subgraph of sybil nodes.
- Overview
  - The social network is based on real-world friendship (strong trust relationship).
  - For the random routes, each node has a fixed random permutation of input-output-mappings in the social network.
  - Thus, each node has a fixed random route.
  - To verify other nodes than the direct neighbors in the social graph, the other node „suspect“ and the „verifier“ check their random routes for an intersection. If one exists, the suspect is accepted as an intersection is more likely to happen if both are honest or both a sybils.

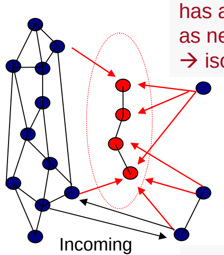


→ The security of SybilGuard is only probabilistic. If the assumptions hold, it can help to fight the sybil attack.

Eclipse Attack

Eclipse Attack

- In an Eclipse attack, an attacker tries to separate a node or group of nodes from the rest of the network.
- Potential victims
  - A specific group of nodes / certain area of the Peer-to-Peer network
  - Arbitrary nodes (easier)
  - Data item (easier)
- If successful, the attacker controls
  - most or all neighbors of its victims.
  - most or all traffic to/from its victims.
- Thus, the attacker „eclipses“ correct nodes from each other‘s view (zu deutsch: „verdunkelt“).



Node only has attackers as neighbor → isolated

Incoming links may still exist, as well as some outgoing links to correct nodes

Sub-network almost only has attackers as neighbor

How to stage an Eclipse Attack

Options for the attacker

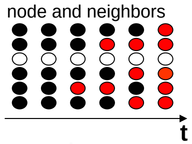
- Use neighbor discovery and routing table maintenance to position malicious nodes into the routing tables.
  - Exact method depends on routing, maintenance, and security protocol.
  - Choose appropriate node IDs to position malicious nodes.
  - Introduce fake nodes to poison routing tables, etc.
  - Stay long in the network / appear as super peer / …

  → A small group of nodes can do this without staging a Sybil attack!
- Use Sybil Attack to increase number of malicious nodes.

Eclipsing and Iterative Routing

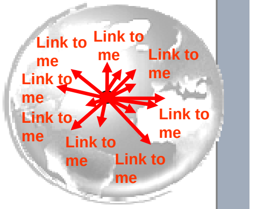
- Problem: Sender controls lookup in iterative routing and she expects better next hop nodes as reply.
- Solution
  - Introduce fake nodes to mislead requests.
  - Mislead requests along a chain of properly positioned attackers.

Defending against the Eclipse Attack



Observations

- Unless it is performed by bootstrap nodes, an Eclipse attack takes some time as the attacker has to infiltrate routing tables of other nodes.
- A node of an attacker in an Eclipse attack tries to make more nodes link to it and as a consequence can have significantly more input links than normal nodes.
- The attacker may be not be completely distributed all over the world and, thus, attacker nodes may be from similar IP subnets, geographic locations, etc.



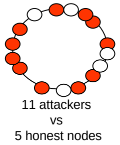
Some proposed counter measures (I)

- Use churn – leave the network and rejoin via reliable bootstrap nodes.
- Perform (anonymous or hidden) audits on neighbors to check if their number of input links is suspicious.
  - Hard to check and expensive.

Defending against the Eclipse Attack

Some proposed counter measures (II)

- Fight the Sybil attack → important, but not sufficient.
- k-buckets with update strategy like in Kademlia → good old nodes stay in the neighborhood.
- Constrained routing tables like in Chord → an attacker cannot have significantly more input links.
  - However, in combination with Sybil attack, this can be useful to force a victim to route to the attacker.
- Proximity constraints → Do not fill your routing table with nodes in similar distance or similar IP range.
- …
- Note: These counter measures may help, but do not solve the problem of the Eclipse attack completely. Similar to the Byzanthine Generals Problem, given enough colluding attackers, defense becomes impossible for many P2P attacks.



Data Eclipse Attack in Kad

Data Eclipse

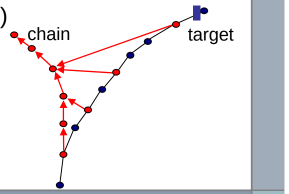
- Eclipse data item instead of node.
- For most DHTs this is a simple storage attack.
  - Attack storage of item by positioning oneself so that one is responsible for item.
- Problem with Kad
  - No responsible node, cached and stored on many nodes in a range close to item ID.
- Sufficient: Be on most paths to the data item.

Data Eclipse in Kad network

- M. Steiner (2008): conducted Eclipse attack in Kad
  - Fill buckets of other nodes with attacker nodes, e.g. by doing a lot of queries.
  - Simple strategies and small number of nodes were sufficient to eclipse data items.

→ Kad restricted nodes in a bucket to two nodes from the same IP subnetwork.

- However, this defense was broken by Kohen et al. (2009)
  - Use chain of conveniently positioned attackers.
  - When a message arrives at a malicious node, route along the chain → until time-out.
  - Never more than one malicious node per bucket.
  - Shown to work in the real Kad network.



Conclusions

- There are many attacks on Peer-to-Peer networks.
- Most attacks use the fact that „good“ and „bad“ information is hard to distuingish as no trusted authority exists.
- With the Sybil attack an attacker can dominate a network.
- With the Eclipse attack an attacker can isolate nodes, subnetworks, and data items.
  - Can be very effective
  - Defenses reduce efficiency of P2P network