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

Peer-to-Peer Systems and Security

Chapter 3
3.2 Attacks against Anonymity

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

Vorlesung SS 2009

Overview

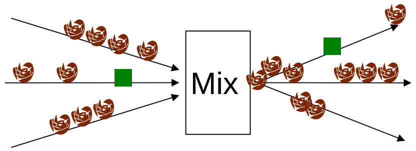
- Motivation
- A selection of attacks
- Conclusion

Motivation

Motivation

- Secure systems are as strong as the easiest attack against them…. one possible notion of security…
- Anonymous systems in their pure form do not resist all attacks.
- A detailed system design needs to defend against the attacks important for the system.

Flooding or (n-1) attack/trickle attack



Flooding attack

- … corresponds to n-1 attack introduced in the mix section of chapter 3.1.
- Attacker floods the system to reduce anonymity set and preferably own all but one messages in the system (or subpart of the system).

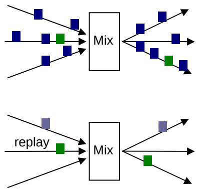
Trickle attack

- Trickle (deutsch: Rinnsal)
- An attacker tries to make the message of interest the only message in system, e.g. a timed mix.

Replay attacks / Subpoena attacks

Replay attacks

- An attacker records messages and resends them.
  - At a suitable point in time, maybe with little or no traffic.
  - Correlation between observations.
- Mitigation
  - Hash messages and accept hash only once per interval.
  - Frequent key changes → accept only message with current keys.



Subpoena attacks

- Subpoena (deutsch: Zwangsmaßnahme/Vorladung)
- An attacker records messages. Use legal methods or other human layer methods to get keys (or content).
- Mitigation
  - Link encryption with short-lived (ephemerial) keys, periodic key rotation.

Partition attack

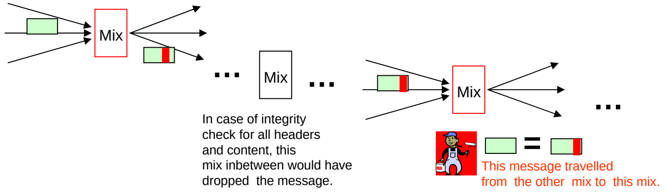
Partition attack

- A partition attack uses the partitioning of a property in the system.
- Partition of client knowledge
  - Set of re-routers known to client → clients will use different re-routers, the combination may leak information.
  - Attacker may determine knowledge of client and use this to identify its messages.
- Mitigation
  - Identical algorithms for updating and obtaining knowledge.
  - As much knowledge as possible.
  - Directory servers to collect and distribute knowledge.

Tagging Attack

Tagging Attacks

- Flip bits in other headers (for the next hops) or content.
- Recognize the message on a node later on the path due to the error it detects in the header.
- Mitigation
  - Integrity check all headers and content at each hop (for each layer of encryption)



In case of integrity check for all headers and content, this mix inbetween would have dropped the message.

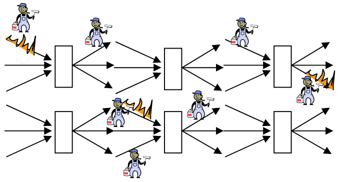
This message travelled from the other mix to this mix.

Pattern and Timing Attacks

Pattern Attacks

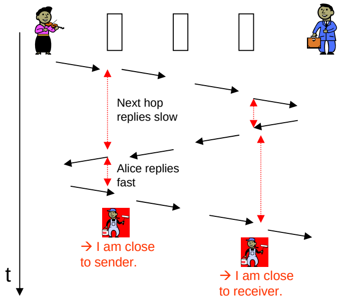
- Insert (or observe) pattern at link, node, or flow.
  - The pattern is related to items of interest.
- Observe the pattern later somewhere else.

  → flow or some of the messages also pass the observed point



Timing Attacks

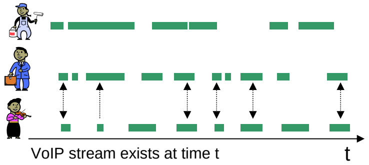
- Timing attacks use different timing behaviour to deduce a situation.
- A relay close to the receiver will see the reply faster.
- A relay close to the sender will see the reply faster.



→ I am close to sender.

→ I am close to receiver.

Intersection attack / Confirmation attack



(Longterm) intersection attack

- ❑ Observe users of interest.
- ❑ Compare their behaviour
  - ▪ Track uptime of users, packet send and receive events
- ❑ Use correlation to link users
  - ▪ Users that are statistically closest to each other might be communicating.
  - ▪ … Correlation is not a proof….
- ❑ Mitigation (short term)
  - ▪ Parallel communication with many others, dummy traffic.

Traffic confirmation attack

- ❑ Assume, you know A and B. You have a suspicion.
- ❑ Lets confirm it. Use intersection and/or pattern attack to check if they are linked.

Attacking hidden services

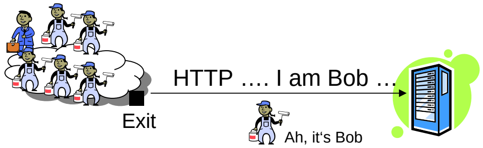
Hidden services

- ❑ A hidden service is an anonymous service (receiver anonymity).

Attacking hidden services

- ❑ Question: „Which server is serving the hidden service?“
- ❑ Perform variants of intersection attack on service and a list of suspects.
- ❑ Use a property of the service that can be observed for the service as well as for the suspects.
  - ▪ Uptime of service / server.
    - • Ping anonymous service and servers in candidate list.
  - ▪ Find patterns in response times, …
  - ▪ Recently published: use deviation of clock drift by querying timestamps (optional part of TCP standard, can be avoided by not supporting this part in your stack).

Information Leakage at higher layers



Information leakage at higher layers

- ❑ Problem
  - ▪ Information in the application can contain linkable / trackable data to sender or receiver.
    - • e.g. user name, browser ID, cookies, etc.
  - ▪ If public services are requested, traffic from exit node to server is likely to send this information unencrypted.
- ❑ Mitigation
  - ▪ Filter such information, e.g. with a privacy enhancing proxy.
  - ▪ Ensure by using HTTPS or similar protocols to tell your ID or name only directly to the server (if that fits to the desired anonymity).
    - • e.g. I may want to hide that I am reading my webmail, but the webmail server should know my identity.

Conclusion

Conclusion

- ❑ Attacks
- ❑ Mitigation
- ❑ … list not complete.