Exercise 5

Exercises Peer-to-Peer-Systems and Security (SS2011)

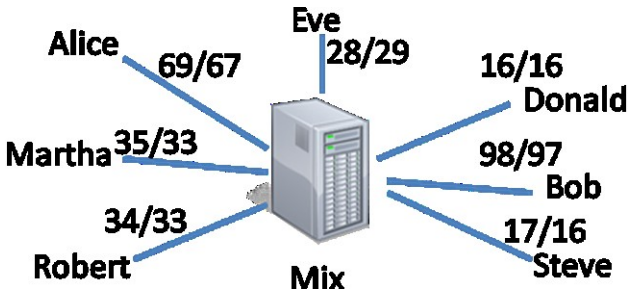
Monday 11.7 2011
Hand-in: Monday 18.7. 2011 in lecture or per mail
Exercise: Thursday 21.7. (together with lecture)

Dr. Heiko Niedermayer
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München

Rules: There will be five exercise sheets. You have to hand-in 70 % of the assignments, attend atleast 3 exercise courses and present a solution in the exercise course to get the 0.3 bonus..

Task 1 Anonymity / Encryption
Some questions with short answers.

a) Assume that Alice, Bob, and Cleta use SSL to communicate with each other. Alice sends a message via SSL to Bob and Bob forwards it via SSL to Cleta. Does the message look the same on both paths (Alice→Bob, Bob→ Cleta) for a global observer? Is this true for all properties of the message?
b) Now look at the following graph. A passive observer was able to observe all communication around a mix (some anonymity system that makes packets unlinkable). All packets are perfectly encrypted and when passing the mix they are re-encrypted by the mix. The observer counted how many packets the entity sent to the mix (first number) and how many packets the mix sent to the entity (second number). Can you still guess who communicates with whom?



Solution:
a)
The message will look different after re-encryption ARBFDBR → TRTRPKG.
However, the size of the message and the time of the message transfer may indicate that this is the same message.
b)
Yes, although there is a chance to be wrong and there may be other reasons for the rates. Here, most likely Robert and Martha, Alice and Bob, Bob and Eve, Donald and Steve are communicating.

Task 2 Eclipse Attack on Chord
Assume that you want to attack a node in Chord and eclipse it from the rest of the network. You have as many resources as you like, but significantly less than 50 % of all nodes.

a) What do you have to do to be able to intercept all of his outgoing messages to other nodes? (Eclipse the outgoing links)
b) What do you have to do to prevent packets towards the node reach the node? (Eclipse ingoing links)

Solution:

a) Position Sybil nodes in 2^i distance
b) Position as predecessor

Task 3 Eclipse Attack on Kademlia

Assume that you want to attack a node in Kademlia and eclipse it from the rest of the network. This is a bit harder than in Chord and will most likely be less perfect. You have as many resources as you like, but significantly less than 50 % of all nodes.

a) What do you have to do to be able to intercept his outgoing messages to other nodes? (Eclipse the outgoing links)
b) What do you have to do to prevent packets towards the node reach the node? (Eclipse ingoing links)

Solution:

a)

In Kademlia, you make it into the bucket of a node, if you talk with it. Thus, as attacker you have to do lookups and other requests to be added. Here, you contact the victim with many Sybil nodes from various ID ranges to take over the buckets of the victim. Your nodes have to be long-lived as only unresponsive nodes are replaced by newer nodes.

b)

The same strategy as in a), yet you do this with other nodes. Here, you present a lot of Sybil nodes from the ID range of the victim, so that the other nodes will store your nodes into their buckets.

Task 4 Bootstrap Tree and Social Network Graph

In the lecture we discussed defences against Sybil and Eclipse attack based on the bootstrap graph and based on social network graphs. Brief answers are sufficient.

a) Why is it not possible to *only* use the bootstrap graph to route to a certain ID (node with certain ID)?
b) Why is it not possible to *only* use the social network graph to route to a certain ID (node with certain ID)?
c) How could you use either the bootstrap graph or the social network graph in your normal DHT routing to defend against routing attacks?

Solution:

a)

because the bootstrap graph does not tell you where an ID is, because it is not ordered or built according to these Ids. So, you would have to search in all subtrees to find a node with an ID. To find the closted node to an ID, you would have to look for all nodes and then select the best node.

b)

The same argument as in a) also holds for the social network graph.

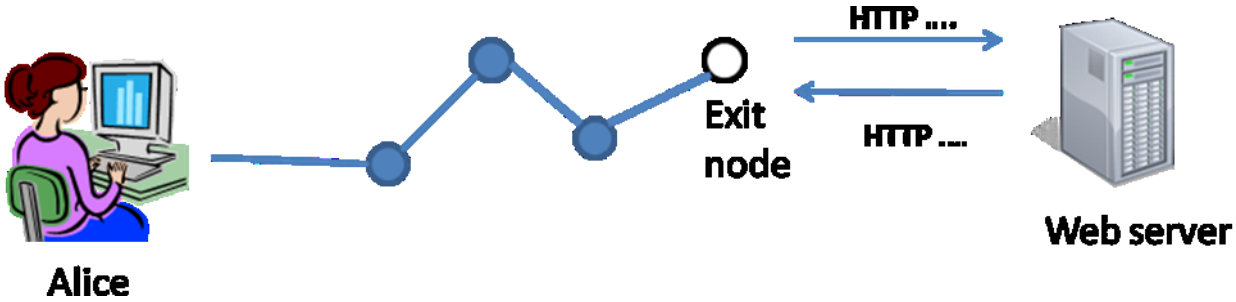
c)

Bootstrap graph: Do iterative routing and e.g. every second node that is asked for better nodes to the target is not a closer node according to the routing, but a node from the bootstrap graph. The nodes selected in the bootstrap graph are randomly selected from different subtrees, parent nodes are also considered to be more trusted and selected from time to time.

Social network graph: Only add non-suspicious nodes in your routing table as preventive measure. If you do iterative routing you can apply a similar strategy as in the bootstrap graph case, ask the next node according to the routing, then ask a neighbour node in the social graph, then the best according to the routing, then again a node from the social graph... for better next hops to the target.

Task 5 Exit Nodes and Anonymity

The following graph sketches the situation:



Alice is using an anonymity system (like Tor) to access a web server. Lets assume the communication within the anonymity system along the dark thick lines is all-encrypted and highly secure and anonymous. To exit the anonymity network towards the normal Internet, exit nodes are used in such systems. The exit node (white node) operates as proxy that executes the HTTP requests to the web server for Alice.

Question: How can this so-called exit node attack Alice's private data or break her anonymity if she is not careful? (Hint: consider what the exit node can read)

Solution:

The exit node will see all communication with the web server in cleartext if HTTP is used. Personal data posted to the server or even unencrypted passwords (not uncommon in forums) would be leaked.
If HTTPS is used, the exit node would not see the traffic and, thus, this problem would be resolved, yet using HTTPS depends on the website, not primarily on the user. The exit node could also try to stage a man-in-the-middle attack on the protocols, which would succeed if either the users accepts the wrong certificate or the exit node also controls a CA that is considered trusted in the root store of the browser (list of trusted CAs).