Arab Academy for Science, Technology and Maritime Transport

College of Computing and Information Technology

Course: Distributed Systems Security

Lecturer: Prof. Dr. Ayman Adel Abdel-Hamid

TA: Mohamed Dwidar

Any Plagiarism will result in a ZERO for the whole project

Implement a shooter game multiplayer game as per the following.

- The system should have at least four players where each player can do one of three things shoot, heal and move

- One player is designated as host assuming server responsibilities like multicasting events and calculating game logic to prevent cheating from others e.g. players using cheatEnging to block damage

- If a health value is tampered with it must go to its original value after the next request from leader

- Just like multiplayer gaming on old local area network games the first player acts as the host and he has to tell his friends his ip by mouth and they all join on that ip

- When players join an ip map is built as the host accepts the players and records their ips and sends the ip to all current players and they record it

- System must implement logical clocks & tie break logic to ensure events like two players making a kill shot on each other are handled correctly as follows:
    - If (L(A)==L(B)) then A<B⇔ T(A) < T(B) unless T(A)=T(B) then evaluate on id(A) & id(B)
      T(A) is the time A arrived at the host

- When a node wants to send a request, it updates its logical clock and multicasts the request (timestamped with its logical clock value) to all other nodes.

- Upon receiving a request, the recipient puts it in a queue ordered by requests' timestamp (a tie breaker will be needed for requests with the same timestamp e.g. request Id), then the recipient multicasts an ACK to other nodes (ACKs are not queued). Be mindful that an event is not executed unless it is acknowledged by all players

- All the messages transmitted must be encrypted during transmission (assume the keys don't change)
- The host uses application level multicast not network multicasts

- Game logic is as follows:
  - All players exist on a 2d (10*10) grid where the positions of everyone is known by everyone
  - Shoot and heal must affect only one player e.g. A shoot B -> -10 health for B
  - Shoot and heal are rejected if the Euclidean distance between the two is more than 3 (only the host makes that decision)
  - Heal action deals +10 health, shoot deals -10
  - Health do not go over 100 or below 0
  - A move action takes a direction up, down, left, right and moves one block in that direction only after getting acknowledged by the host

Bonus Part:

2 points for imitating TLS SSL

- Every player has a public and private key and a signed certificate
- When I receive a message from someone, I hash the contents of a certificate and decrypt the signature on the certificate and compare the hash
- Assume a constant CA public key

3 points for raft algorithm implementation to elect a new host when the designated host is down

Rubric:

- [6 Marks] RMI concepts:
  - o [3 Marks] Nodes initialization and interaction
  - o [3 Marks] Parameter Passing
- [4 Marks] TO-Multicasting
- [3 Marks] encryption & security
- [3 Marks] Game Logic
- [4 Marks] Presentation

Deliverables:

- A design document (make a diagram or any form of project design document so that I know you thought of the design first before coding)
- The project source code in a zipped archive

Test scenarios:

- Have node B send a shot to node C then have node A send a heal. both just started with a clock = 1 so it would cause a tie. see which one is dealt first
- Use cheat engine to tamper with the health value then see if it changes next in the game
- Play the round until one player wins by killing all others

Assignment starts on Saturday 10th of May with a deadline on Friday 23rd of May

You can work in groups of 2