**Security Game Specification**

The game is based on a network graph. The network graph will contain several nodes. Two agents will be playing the game. One agent, defender, will try to protect his network by applying different security features and the other agent, attacker, will try to hack the defender’s network nodes.

Each of the node will have two types of values.

1. Security Value (i.e. SV): This value is the measurement of the vulnerability of a normal node. This value will be ranged from 0~20. To attack, the attacker needs to roll a D20 die and if the outcome of the roll die is greater or equal to the security value, attacker passes the security check.
2. Point Value (i.e. P.V.): This value determines the number of points the node will contain. If the attacker successfully passes the security check of a Normal Node, he will get the point value of that node.

There will be three types of nodes in the network.

1. Normal Nodes: These nodes are normal resources in the network. The attacker will try to hack these nodes and get points if they pass the security check. Otherwise the attacker gets no points.
2. Router Nodes: This type of nodes play a role like router i.e. making huge number of connections with other normal nodes or router nodes. Attacker will not get any points for hacking this type of nodes.
3. Entry Nodes: These nodes will be exposed to the attacker. These are the starting nodes from where the attacker will start their attack on the defender’s network. They will not have any P.V.

**Defender’s role:**

Initially, the defender will be given a network and a credit amount based on the size of the network. He can spend the credit by applying several security features on his network.

The defender can apply the following security features to enhance security of the network:

1. Apply Honeypot
2. Apply Firewall
3. Increase the Security Value (S.V.) of a node

To apply the above features the defender needs to spend some credit amount from his account. The costs limit of the above features will be parameterized based on the network size. The costs will differ from feature to feature and the cost order should be like “a > b > c”. Now describing each of the features below:

* Applying Honeypot: This feature is the highest costly feature that defender can apply. The defender can add honeypot nodes (and edges) in the network. For this he will need to spend some credits from his account. As it is said before, these costs will be parameterized. The speciality of the Honeypot is that attacker will not get any points if he successfully passes the security check of a Honeypot.
* Apply Firewall: Defender can apply firewall on the edges of a node. This action will remove those specific edges from the network. But a node must have at least one edge. This is the second highest costly feature that defender can apply on the network.
* Increase the Security Value (S.V.) of a node: Defender can increase the Security Value (S.V.) of a node. By applying this action, defender can degrade the probability of successfully passing the security checking on that specific node by the attacker.

Thus, using the above features the defender can alter the network. The defender will try to minimize the cost and maximize his utility by preventing the attacker to get more points from the Normal Nodes.

**Attacker’s role:**

After the defender is done with his work, he has no more role to play. The modified secured network will be then given to the attacker to attack. Same as the defender, the attacker will be given an initial credit amount based on the network size. He can spend the credit by doing the following things to attack the network. The attacker will initiate his attack from any of the Entry Nodes. The entire network will be unexplored to him. He can explore the network by hacking them starting from a specific Entry Node.

The attacker can do the following things to attack the network:

1. Attack a node: Using a D20 dice, the attacker will try to pass the security check of a node by rolling it. He needs to roll a higher or equal value than the security value(S.V.) of that node. If he is successful, he gets the point value of that node. Then the attacker can explore other neighbor nodes of that node.
2. Probe Security Value: The attacker can probe the Security Value (S.V.) of a node. He can only know the Security Value of that node by doing so.
3. Probe Point Value: The attacker can probe the Point Value (P.V.) of a node. He can only know the Point Value of that node by doing so.
4. Probe Honeypot: The attacker can probe a node to check whether it’s a honeypot or not.
5. Probe Edges: The attacker can probe a node to check how many neighbor it has.
6. D50 roll dice: The attacker can roll a D50 dice to increase his chances to hack a node successfully. This feature is the highest costly feature that attacker can apply.

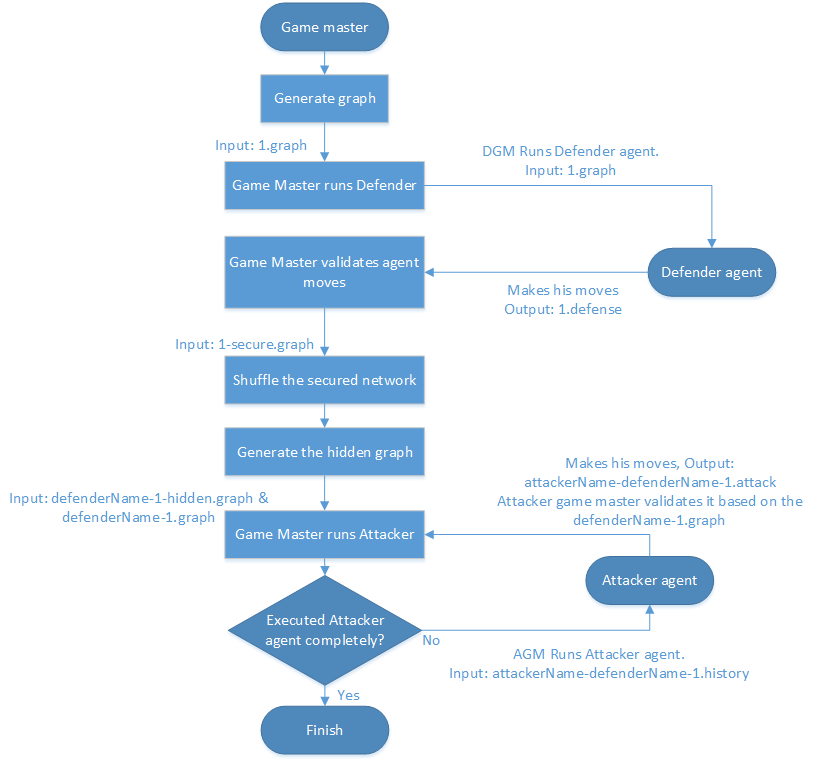
To apply the above features the attacker needs to spend some credit amount from his account. The costs of the above features will be parameterized based on the network size. The cost limit of the above features will differ from each other and the cost order should be like “f > a > e >= c >= b >= d”. The attacker will try to minimize the cost and maximize his utility.

**Analysis:**

A game master will keep track how the game goes on. The game master will run several attacker to attack a defender’s graph one by one. So a statistical analysis can be done by observing all the results of the attackers against a specific defender. Thus we can measure one defender’s performance against other defenders.

An attacker’s performance can be measured by calculating the points that he gathered from defenders Network (i.e. calculating average points, medians, and variances).

In extra, we will be able to measure the riskiest attacker agents, conservative attacker agents (e.g. most probing, least probing). For defender agent, we will be able to calculate which defender applied most number of Honeypots / firewall / S.V. value increment.

**Flowchart of the security game  
  
**