Apiary Party

The game challenges you to create intelligent and robust agents in a network security context. The game is played on a graph containing different kinds of nodes representing a computer network. There are also two agent roles within the game: a defender who tries to protect the network, and an attacker who tries to compromise and exploit the network. Put on your white hats and show off your 1337 skills.

Each node contains two important properties:

a) Security Value (SV): A measure of the node's strength ranging from 0 to 20. If an attacker hacks the node (represented by a D20 dice roll) with an outcome greater or equal to the SV the attacker gains access to the node.
b) Point Value (PV): This value represents the importance (points) of the node. Attackers that successfully hack a node gain the PV of the node.

Nodes are representative of the kinds of computers or devices found on a typical network.

1. Public Nodes: These are the starting nodes from where the attacker will be granted automatic access such as an external gateway or similar device meant to connect out. They have SV of 0 and PV of 0.
2. Networked Convince: Contains little or no secure information but is connected to the network i.e. thermostat, local office printer, appliance (fridge, coffee maker, TV), etc. SV and PV range from 1 to 8
3. Personal Device: Various workstations with different kinds of data such as the conference room Skype machine, a tablet, a simple workstation, etc. SV and PV range from 5 to 15.
4. Secure Device: CEO's laptop, Security Cameras, sysadmin workstation, etc. SV and PV from 12 to 19
5. Database: Controls an important database with private and confidential data, etc. SV from 12 to 19 and PV range from 20 to 29.

Defender:
The defender will be given a network and a budget based on the size of the network. Defenders can apply several security features on this network at a cost per each action. These security features are:

a) Add a Honeypot: Honeypots are fake nodes created used to entrap Attackers. They also act as Intrusion Detection Systems (IDS) and Attackers that gain access to a honeypot will trigger an alert which will end the game.
b) Add a Firewall: This action will remove a specified edge from the network, but a node in the network must always have at least one edge in the network (it is on the network for a reason).
c) Strengthen a Node: This will increase the node's SV. By applying this action, such as applying security updates or changing policies, the defender can lessen the probability of an attacker gaining access to this node.

The cost for each of the features are different **and will change** but they will always meet the constraint that $a \geq b \geq c$:

Attacker:
The defenders modified secured network will be then given to the attacker to attack (white hat pen testing of course). Attackers start with an initial budget based on the network size and can perform various actions at a cost. The attacker will start from any of the public nodes and the game is over once there is not enough budget to perform more actions or an alert is triggered. They can see the graph topography but they will only know the point values of the nodes and which are Database Nodes. Attackers can perform the following actions while they have enough in their budget:

a) Attack a node: Using a D20 dice the attacker will try to roll greater than or equal the node's SV. Successful rolls will award the attacker with access to the node and the node's PV as well as access to the node's connections (edges).
b) Probe Values: Gain knowledge of the node's PV.
c) Probe Honeypot: Identify if the node is honeypot or not.
d) Super attack: The attacker can roll a D20+ dice to increase his chances to hack a node successfully. The value of the die will be parameterized and **will** change.

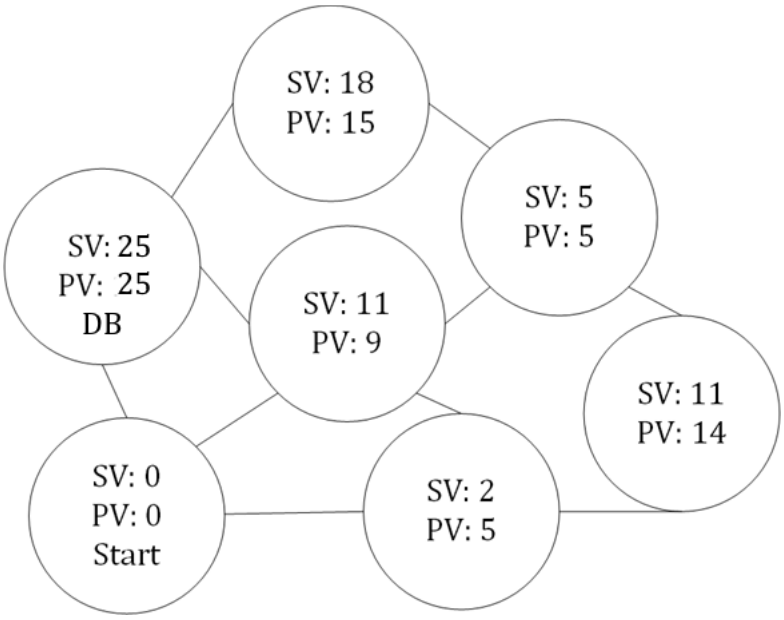
The costs of the above features will be parameterized and **will change**. However, they will always meet the following constraints: $d > a$, $b \geq c$, and $d > b$.

All invalid actions will be charged an invalid action cost and no action will be taken.

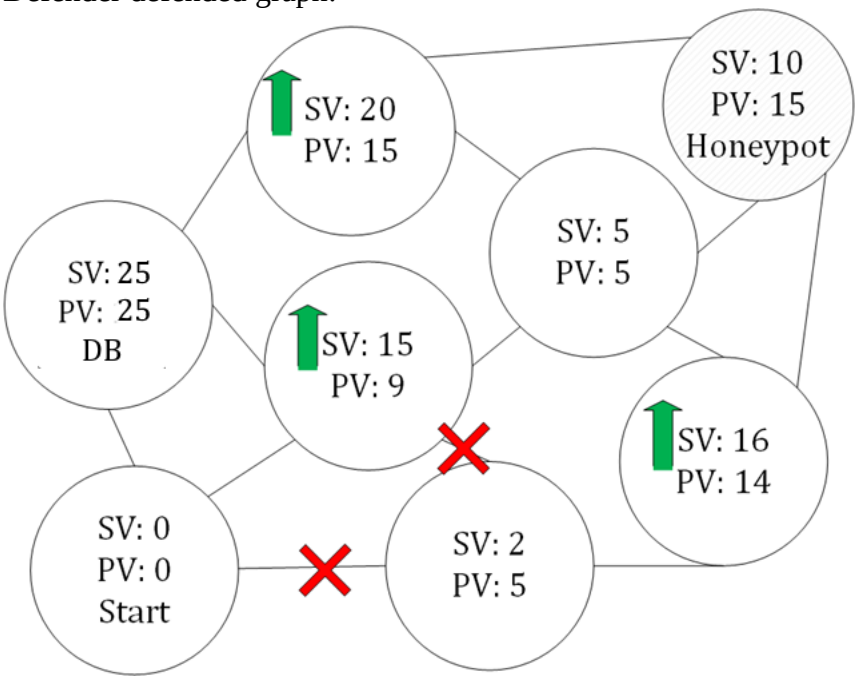
Analysis:
The Game Master will keep track of all game actions. It will first execute all defender code on each generated graph created defended versions of those graphs. Then it will execute each attacker on each defended graph. Afterwards it will perform an analysis of the results and measure the performance of all agents.

Helpful hints:
The costs for actions will change! Your agents should respond to differing costs. If honeypots are expensive then you as a defender might not want to buy a lot of honeypots. If honeypots are expensive then you as an attacker might reason that there are probably not a lot of honeypots. If strengthening is expensive then you might not want to spend a lot on strengthening as a defender. If strengthening is expensive then you as an attacker might assume that the security values are probably close to the point values. If there are more nodes than the number of nodes parameter for generating graphs, then you might assume that some of them are honeypots. LOOK AT THE PARAMETERS!!!!

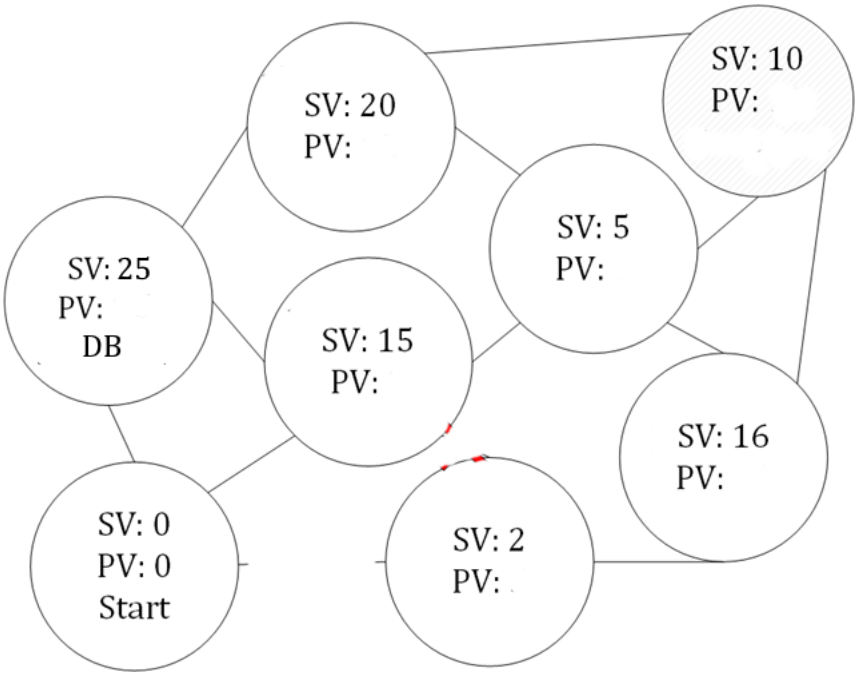
Example:
Defender initial graph



Defender defended graph:



What the Attacker sees:


SV: 20
PV:
SV: 10
PV:
SV: 5
PV:
SV: 25
PV:
DB
SV: 15
PV:
SV: 16
PV:
SV: 0
PV: 0
Start
SV: 2
PV: