Exercise 1

Consider a scheme similar to Homework 3, Part a where M systems are subject to a series of N attacks. A system is discarded as “unsecure” if it reaches a penetration score of P before reaching, instead, a security score of S. Simulate and represent the probabilities of a system being discarded, for various values of P, example: P = k\*10 (k=2,…,10), conditional on the 3 cases for S: S = 20, S = 60, S = 100 (or any other value of S of your choice that you find useful to explore (it could be a user parameter).

To simulate the probabilities of a system being discarded for various values of the Penetration Score P conditional on the three cases of the Security Score S, I decided to set a fixed number of systems and attacks to 1000, to use 20, 60, 100 as values for the Security Scores and P = k\*10 (k = 2, …, 10) as Penetration Scores.

So, for every Security Score and for every Penetration Score, I simulated an attack using a random attackScore, so the probability that a system is attacked, from 0 to S + 1. The decision to set as upper bound for the Attack Score S + 1 is linked to the line chart, since if, for example, the Security Score ia a very low value, such as 20, the Attack Score will likely to be higher than Security Score much often than if the Security Score is 100, leading to a line in the chart near to 0 that doesn’t allow to see clearly the relationship between the probability of discarding a system and the increasing of the Penetration Score.

The logic behind the algorithm is:

If the Attack Score is higher than the Security Score, the system is secure, so we don’t have to discard it

Otherwise, the system has been penetrated, so we increase the Penetration Score

If the system reaches the Penetration Score, it will be automatically discarded and we have to increase the discardedCount which represent the number of systems discarded for that Security Score.

The Discarded Probability, so the probability that the systems are discarded given the values of the Security Score and the Penetration Score is calculated as the Discarded Count by the total number of the systems M.

Moreover, in the Line Chart we can see the behaviour of the Discarded Probability related to the three values of the Security Score S1 = 20, S2 = 60, S3 = 100 as the Penetration Score P becomes higher. In fact, we obtain the expected result, since as P increases, the probability of discarding a system decreases, since it will be very difficult that a system reaches an high Penetration Score, before reaching the Security Score.

Instead, in the Bar Chart, we can see the number of the systems discarded for each Security Score. Also in this case, we obtained the expected result, since the number of discarded systems increases as the Security Score increases, linked to the fact that higher the Security Score is, higher is the number of systems that can’t reach that value of S, so that are “unsecure” for an high Security Score.

Research

Pubblicità

Impostazioni sulla privacy

Find out on the web about the “Gambler’s Ruin Problem”. See if you can see any analogy with this exercise and make your personal consideration about what your simulation is suggesting to you.

The Gambler’s Ruin problem is a probability problem that deals with a betting scenario that involves two players, the “gambler” and the “house,” engaged in a series of fair bets. The problem seeks to understand the likelihood of the gambler losing all their money or reaching a certain goal before the house does.

The setup of the problem is:

Two players: The gambler and the house.

Initial conditions: The gambler starts with a certain amount of money, and the house starts with a certain amount.

Betting rules: In each round, the gambler and the house make a bet. The outcome is a win for the gambler with probability p and a loss with probability 1−p. The bet size is typically fixed.

Goal: The gambler’s goal may be to reach a certain amount of money or to bankrupt the house, and vice versa.

Stopping conditions: The game continues until one of the players reaches their goal or goes bankrupt.

One of the key results of the Gambler’s Ruin problem is that, under certain conditions (such as a fair game where the probability of winning is the same in each round), the probability of the gambler reaching their goal or going bankrupt is solely determined by the initial amount of money they have compared to the total amount of money in the system.

There are some similarities between the Scenario simulated in the previous exercise and the Gambler’s Ruin Problem, which can be seen in:

Probability of Ruin:

Gambler’s Ruin: Refers to the probability of a gambler losing all their money or reaching bankruptcy.

System Security Scenario: Relates to the probability of a system being compromised (ruined) before reaching a certain Security Score.

Series of Events:

Gambler’s Ruin: Involves a series of betting events where the gambler either wins or loses each round.

System Security Scenario: Involves a series of attacks where a system may or may not be compromised in each attack.

Transition Probabilities:

Gambler’s Ruin: The probabilities of winning or losing in each round, which determine the transition from one state (amount of money) to another.

System Security Scenario: The probabilities of a system being compromised or remaining secure in each attack, determining the transition between different Security Scores.

Initial and Target States:

Gambler’s Ruin: Involves starting with a certain amount of money and the goal of either reaching a specified amount or going bankrupt.

System Security Scenario: Involves starting with a certain security score and the goal of either reaching a specified Security Score or being compromised.

Boundary Conditions:

Gambler’s Ruin: Typically has boundary conditions, such as winning when reaching a certain goal and losing when reaching bankruptcy.

System Security Scenario: Has boundary conditions, such as a system being compromised at a certain Penetration Score and being secure when reaching a specified Security Score.