PISP EXERCISE: 01

1. **Read the following three bug reports. For each of them, decide whether an attack exploiting it violates confidentiality, integrity, availability, or some combination thereof. Give reasons for your decision.**

    a. Windows SMB Remote Code Execution Vulnerability that has been Exploited by WannaCry

⟶ A flaw in MicrosoE's SMBv1 convention, which permits computers to communicate with one another, allows pernicious malicious actors to remotely execute their own programs on the server. They can regularly fulfill this by sending the server a particular form of message. We hazard losing control of the server in the event that they can seize total control and run with it as they like. Sometimes, noxious on-screen characters may subtly screen framework activity and attempt to compromise other machines on the same network.The security upgrade improves the way SMBv1 bargains with certain extraordinary demands to settle the frail spot. It makes beyond any doubt private data remains private since it stops unauthorized sharing of information **(confidentiality)**. It moreover guarantees that the information you ought to be able to induce is as it were changed by individuals who are permitted to do so **(Integrity)**.

    b. The Heartbleed Vulnerability

⟶ The internet's communication strategies are affected by the Heartbleed vulnerability. Potential attackers' memory can be examined by websites influenced by Heartbleed. Hence, it's conceivable that programmers will discover the encryption keys. By uncovering the encryption keys, risk-performing artists could pick up and get access to usernames and passwords for frameworks. Depending on the degree of authorization of the stolen credentials, threat actors may initiate unused attacks, tune in on discussions, and accept the personality of clients inside the framework. In this occurrence, delicate data is at hazard, and on the off chance that the assailants get the usernames and passwords, they can modify them, leading to a **loss of integrity.**

    c. Cisco Security Advisory: Cisco IOS XR Software Border Gateway Protocol Vulnerability

⟶ There may be an issue with the Border Gateway Protocol (BGP), which helps in coordinating internet traffic. The BGP associations may have to be be more than once restarted in case this issue is abused. This may result in unusual issues with information transport and benefit disturbances for a few systems. This kind of attack would basically damage the availability guideline since, as already famous, it over and over resets the

BGP traditions. This issue may result in mistakes when sending information over the web and may cause certain networks to glitch. In the event that this helplessness is more than once misused, it'll result in repeated connection restarts, which is equivalent to benefit intrusion. This sort of assault damages the necessity to preserve accessibility, or the state of objects continuously being operational **(Loss of Availability)**

2. Consider a computer system with three users: Naomi, Jacob, and Ruth. Naomi owns the file naomifs, and Jacob and Ruth can read it. Ruth can read and write the file jacobfs, which Jacob owns, but Naomi can read and execute it. Only Ruth can read and write the file ruthfs, which she owns. Assume that the owner of each of these files can execute it. Note that there are four kinds of access rights in this question: read, write, own, and execute.

- Create the corresponding access control matrix.

user/files	**naomifs**	**jacobfs**	**ruthfs**
Naomi	**own,execute(--xo)**	**read,execute (r-x-)**	
Jacob	**read (r---)**	**own,execute(--xo)**	
Ruth	**read (r---)**	**read,write (rw--)**	**own,execute,read, write(rwxo)**

- Ruth gives Naomi permission to read ruthfs, and Naomi removes Jacob's ability to read naomifs. Show the new access control matrix.

	naomifs	**jacobfs**	**ruthfs**
Naomi	**own,execute(--xo)**	**read,execute(r-x-)**	**read (r---)**
Jacob		**own,execute(--xo)**	
Ruth	**read (r---)**	**read,write(rw--)**	**own,execute,read, write(rwxo)**

3.Consider the set of rights {read, write, execute, append, modify, own, truncate}.

- Using the syntax in Section 2.3 of the text book (Introduction to Computer Security), write a command delete_all_rights (p, q, d). This command causes p to delete all rights the subject q has over an object d.

→

```
command delete_all_rights(p, q, d)
        delete read from A[q, d];
        delete write from A[q, d];
        delete execute from A[q, d];
        delete append from A[q, d];
        delete modify from A[q, d];
        delete own from A[q, d];
        delete truncate from A[q, d];
end
```

- Modify your command so that the deletion can occur only if p has modify right over d.

→

```
command delete_all_rights(p, q, d)
 if modify in A[p, d]
 then
        delete read from A[q, d];
        delete write from A[q, d];
        delete execute from A[q, d];
        delete append from A[q, d];
        delete modify from A[q, d];
        delete own from A[q, d];
        delete truncate from A[q, d];
 end if
end
```