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.

1) wxGTK: User-assisted execution of arbitrary code.
http://www.securityfocus.com/archive/1/513491/30/0/threaded

Availability	wxGTK is not working/functioning as expected. Some change has been inserted causing the Create() function to raise an integer heap based overflow ultimately leading to an crash / failure of the system. This is a kind of DOS(denial of service) attack where it violates the availability policy. **For Example**: The attacker might take benefit of this bug and try to open a crafted JPEG file using a program that uses wxGTK, possibly resulting in the remote execution of arbitrary code with the privileges of the user running the application. Using this approach, the attacker might make the system fail by manipulating with it.

2) Cisco Security Advisory: Cisco IOS XR Software Border Gateway Protocol Vulnerability.
http://www.securityfocus.com/archive/1/513411/30/30/threaded

Availability	The vulnerabilities mentioned in the bug report may result in the continuous resetting of BGP peer sessions. This may lead to in consistency within the routing and ultimately the network will be affected. The network will not function as expected resulting in a DOS (denial of service) attack where it violates the availability policy.

3) Intuit Lacerte 2017 for Windows security issue.
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-11338

Confidentiality	The entire customer list contains sensitive information such as SSN, address, job, title, phone number, email address, phone/email address, etc. The information is nothing but confidential data which should be confidential for the user. But as there are chances of attacks such as sniffing or man in the middle attack, if this happens – it would result into violating the confidentiality policy.

2.Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file *alicefs*, and Bob and Cyndy can read it. Cyndy can read and write the file *bobfs*, which Bob owns, but Alice can read and execute it. Only Cyndy can read and write the file *cyndyfs*, 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.
a. Create the corresponding access control matrix.

Access Control Matrix:

	Alicefs	bobfs	Cyndyfs
Alice	Ox	rx	
Bob	R	ox	
Cyndy	R	rw	Orwx

b. Cyndy gives Alice permission to read *cyndyfs*, and Alice removes Bob's ability to read *alicefs*. Show

the new access control matrix.

Access Control Matrix:

	Alicefs	bobfs	Cyndyfs
Alice	Ox	rx	R
Bob		ox	
Cyndy	R	rw	Orwx

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

a. 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 r from A[q,d] ;
        delete w from A[q,d] ;
        delete x from A[q,d] ;
        delete a from A[q,d] ;
        delete m from A[q,d] ;
        delete o from A[q,d];
        delete t from A[q,d] ;
end;
```

b. 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 m in A[p,d] then
        delete r from A[q,d] ;
        delete w from A[q,d] ;
        delete x from A[q,d] ;
        delete a from A[q,d] ;
        delete m from A[q,d] ;
        delete o from A[q,d];
        delete t from A[q,d] ;
end;
```