Question 2

a) Describe the Access Matrix Model of security. Illustrate, using an example, how Unix access control lists (ACLs) can be represented in this model. Why is it reasonable to use this model to represent an access control policy mechanism? (15 marks)

- An Access Matrix is a formal security model of protection state in computer systems, that characterizes the rights of each subject with respect to every object in the system. Here, permissions are defined for any kind of operation, not just read, write, execute. For example, permissions push, pop, etc., for a stack object.

- An access control list (ACL), with respect to a computer file system, is a list of permissions attached to an object. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation. For instance, if a file has an ACL that contains (Alice, delete), this would give Alice permission to delete the file.

- An Access Matrix shows all possible flows of the state of a system at any given time.

b) A simple multilevel secure database management system is to be designed. Each row in a database table is assigned a separate security-level, and subjects at any security-level may access the table (but not necessarily every record in the table). For example, consider the following Customer-information table (custid is primary key).

custid	*level*	Name	Company
0031	topsecret	Monty	Springfield Nuclear
1002	secret	Wolfcastle	Planet Springfield
1022	unclassified	Moe	Moe's Tavern

Given the usual ordering between the specified security levels, a secret process may read the Moe and Wolfcastle entries but not the Monty entry, and so forth. You should assume that when a new tuple is inserted into the table it is assigned the security-level of the subject inserting it.

i. Propose suitable multilevel security rules that govern how database operations UPDATE and SELECT should behave. (8 marks)

- A subject s attempts to SELECT a row i from table. Access is OK if security level of i <= security level of s.

- A subject s attempts to UPDATE a row i from table. Access is OK if security level of s <= security level of i.

ii. Given that primary key values are unique in a table, explain how a Trojan-Horse running at

top-secret could establish a covert-channel and signal two bits of information to a subject operating at secret. (Hint: recall the multilevel file-system discussed in lectures). Suggest how the covert channel might be closed. (7 marks)

- Answered in Mac Exercises

c) Describe how the MLS DBMS mechanism in Part b) of this question could form part of a Chinese-Wall policy mechanism that ensured that a user cannot access customers records from competing companies. In answering the question be sure to explain why the Chinese Wall is enforced. (15 marks)

The companies from the MLS DBMS in part B could be could be placed into a table, defining what company information each company member has access to. As can be seen below, Monty working for Springfield Nuclear may have access to Springfield Nuclear information, but may not have access to Planet Springfield information.

_ ~ _	**Springfield Nuclear**	**Planet Springfield**	**Moe's Tavern**
Springfield Nuclear		**X**	**X**
Planet Springfield	**X**		**X**
Moe's Tavern	**X**	**X**	

Here the Chinese Wall policy is enforced to ensure that it is not possible for Springfield Nuclear worker Monty to pass on any Springfield Nuclear information to Planet Springfield worker Wolfcastle, leading to a conflict of interest.

Question 4
An Internet facing server gamer.com hosts an online multiuser game service on Port 666.

a) A string containing client data received at this port on gamer.com is passed by the service, without checking, to the C function:

```
void play ( char *str )  {
        char buff [ 60 ];
        strcpy ( buff, str );
        // .....
}
```

Describe how a buffer-overflow in this function could enable a remote attacker gain control of the gamer.com host. (15 marks)

Notes

A buffer overflow occurs during program execution when a fixed-size buffer has had too much data copied into it. This causes the data to overwrite into adjacent memory locations, and depending on what is stored there, the behavior of the program itself might be affected.

The return address in the frame points to the next instruction to execute after the current function returns. This storage and retrieval of the address of the next instruction on the stack introduces a vulnerability that allows an attacker to cause a program to execute arbitrary code.

Successfully modifying the return address allows the attacker to execute instructions with the same privileges as that of the attacked program. If the compromised program is running as root, the attacker might use the injected code to spawn a superuser shell and take control of the machine. In the case of worms, a copy of the worm program is installed, and the system begins looking for more machines to infect.

Answer 1

strcpy causes a buffer overflow into FP and RET. Here the buffer overflow causes the RET address to be overwritten, where the attacker provides their own RET address, eg. to execute a shell. By experimenting with the right parameter value (SH-HACK2), an attacker can cause a buffer overflow and get the program to execute their own code. If the program is a suid-root shell then the attacker gets access to a root shell, and essentially has access to the entire system.

Answer 2

Should identify why this program has a buffer overflow and give an outline of how the buffer overflow is carried out: can cause new instructions (eg a execv) to be pushed on the stack and executed (assuming that the argv carefully overwrites the saved stack frame pointer, etc.)

Should explain what the consequences of the buffer overflow is, that is, that the attacker can provide the attacking argv as form value on the web-site and thereby get the server to execute an instruction specified in the argv by the attacker.

b) There is a concern about SYN-flood based denial of service attacks on gamer.com. The administrator uses a (flawed) implementation of SYN-cache whereby the state associated with a half-open connection is stored in the hashtable bucket indexed as h(i.p), where h() is a one-way hash function, i is the IP address of gamer.com and p is the requested port on gamer.com. Describe how an attacker can still carry out a SYN flood on this host and outline how the SYN-cache scheme should have been implemented. (15 marks)

In this case, the half-open connections are not being stored in a secure manner. They are being stored as the hash(ip address of gamer.com, requested port). This is not secure because an attacker can easily figure out the hash values stored and thus override legitimate half-open connections from legitimate visitors of gamer.com, in order to carry out a Denial of Service attack.

The hash value on the incoming packet should have been computed using the source and destination addresses, the source and destination port, and a randomly chosen secret. Here the attacker will not know which bucket packets are being placed in.

c) Suppose that the probability of a SYN-flood based denial of service attack on gamer.com during a calendar year is 0.01, with a resulting loss of earning of $100,000. Deploying a SYN-cache reduces this probability to 0.001, however, it is recommended that the server is upgraded (once off cost $900), in order to support the SYN-cache processing requirements. Suppose that SYN-cookies entirely eliminate the risk of SYN-flood based denial of service, however, their continual use impact connection performance with a yearly loss of earnings of $5,000.00.

Use this information to carry out a Risk Assessment and advise the owners of gamer.com on how best to mitigate the risk of denial of service. (15 marks)

- Option 1
    - Risk-flood = 0.01 * 100,000 = 1,000
- Option 2
    - Risk-cache = 0.001 * 100,000 = 100
    - Upgrade to SYN-cache cost = 900
- Option 3
    - Upgrade to SYN-cookies = 5,000 yearly

From the above Risk calculations, it is clear that the current system risk amounts to $1,000 per year. Upgrading the system to allow SYN-cache would cost $100 per year, with a once off $900 cost i.e. 1,000 - (100 + 900) = 0. There would be 0 savings made in the first year due to the server upgrade, however there would be savings of 900 per year thereafter. On the other hand SYN-cookies would completely remove this risk, however this would come at a cost of 5,000 per year.

From the above assessment, it can be clearly seen that option 2 is the best option. While a breakeven is achieved in the first year, clear savings are made in the following years.