SOLUTIONS TO CHAPTER 9 PROBLEMS

1. Popular news services require integrity and availability but not confidentiality. Backup storage systems require confidentiality and integrity but not necessarily 24/7 availability. Finally, banking services require confidentiality, integrity, and availability.

2. (a) and (c) have to be a part of the TCB and (b), (d) and (e) can be implemented outside the TCB.

3. A covert channel is an unauthorized communications channel that can be created in a system by observing and manipulating measurable performance characteristics of the system. The key requirement of a covert channel to exist is to have some shared system resources such as CPU, disk, or network that can be can be used for sending secret signals.

4. It is just entered into the matrix twice. In the example given in the text, *printer1* is in two domains simultaneously. There is no problem here.

5. Full protection matrix: 5000 × 100 = 500,000 units of space.
ACL:

50 × 100	(1% objects accessible in all domains; 100 entries/ACL)
+ 500 × 2	(10% objects accessible in two domains; two entries/ACL)
+ 4450 × 1	(89% objects accessible in one domain; one entry/ACL)
= 10,450 units of space	

The space needed for storing a capability list will be same as that for ACL.

6. (a) Capability list
(b) Capability list
(c) ACL
(d) ACL

7. To make a file readable by everyone *except* one person, access-control lists are the only possibility. For sharing private files, access-control lists or capabilities can be used. To make files public, access-control lists are easiest, but it may also be possible to put a capability for the file or files in a well-known place in a capability system.

8. Here is the protection matrix:

	Object			
Domain	**PPP-Notes**	**prog1**	**project.t**	**splash.gif**
asw	Read	Read Exec	Read Write	Read Write
gmw	Read Write		Read Write	
users	Read		Read Write	
devel		Read Exec		Read

9. The ACLs are as follows:

File	ACL
PPP-Notes	gmw:RW; *:R
prog1	asw:RWX; devel:RX; *:R
project.t	asw:RW; users:RW
splash.gif	asw:RW; devel:R

Assume that ∗ means all.

10. If *asw* wants to allow *gmw* but no other members of users to look at splash.gif, he could modify the ACL to asw:RW; devel:R; gmw:R.

11. Bell-LaPadula model only: (a), (b), (h)
Biba model only: (e), (i)
Both: (c), (d), (f), (g)

12. The server will verify that the capability is valid and then generate a weaker capability. This is legal. After all, the friend can just give away the capability it already has. Giving it the power to give away something even weaker is not a security threat. If you have the ability to give away, say, read/write power, giving away read-only power is not a problem.

13. No. That would be writing down, which violates the ∗ property.

14. A process writing to another process is similar to a process writing to a file. Consequently, the ∗ property would have to hold. A process could write up but not write down. Process *B* could send to *C*, *D*, and *E*, but not to *A*.

15. In the original photo, the R, G, and B axes each allow discrete integral values from 0 to 255, inclusive. This means that there are 2^{24} valid points in color space that a pixel can occupy. When 1 bit is taken away for the covert channel, only the even values are allowed (assuming the secret bit is replaced by a 0 everywhere). Thus as much of the space is covered, but the color resolution is only half as good. In total, only 1/8 of the colors can be represented. The disallowed colors are mapped onto the adjacent color all of whose values are even numbers, for example, the colors (201, 43, 97), (201, 42, 97), (200, 43, 96), and (200, 42, 97) now all map onto the point (200, 42, 96) and can no longer be distinguished.

16. the time has come the walrus said to talk of many things
of ships and shoes and sealing wax of cabbages and kings
of why the sea is boiling hot and whether pigs have wings
but wait a bit the oysters cried before we have our chat
for some of us are out of breath and all of us are fat
no hurry said the carpenter they thanked him much for that

From *Through the Looking Glass* (Tweedledum and Tweedledee).

17. The constraint is that no two cells contain the same two letters, otherwise decryption would be ambiguous. Thus each of the 676 matrix elements contains a different one of the 676 digrams. The number of different combinations is thus 676! This is a very big number.

18. The number of permutations is $n!$, so this is the size of the key space. One advantage is that the statistical attack based on properties of natural languages does not work because an E really does represent an E, and so on.

19. The sender picks a random key and sends it to the trusted third party encrypted with the secret key that they share. The trusted third party then decrypts the random key and recrypts it with the secret key it shares with the receiver. This message is then sent to the receiver.

20. A function like $y = x^k$ is easy to compute but taking the kth root of y is far more difficult.

21. *A* and *B* pick random keys *Ka* and *Kb* and send them to *C* encrypted with *C*'s public key. *C* picks a random key *K* and sends it to *A* encrypted using *Ka* and to *B* encrypted using *Kb*.

22. One way to sign a document would be for the smart card to read in the document, make a hash of it, and then encrypt the hash with the user's private key, stored in the card. The encrypted hash would be output to the Internet cafe computer, but the secret key would never leave the smart card, so the scheme is secure.

23. The image contains 1,920,000 pixels. Each pixel has 3 bits that can be used, given a raw capacity of 720,000 bytes. If this is effectively doubled due to compressing the text before storing it, the image can hold ASCII text occupying about 1,440,000 bytes before compression. There is no expansion due to the steganography. The image with the hidden data is the same size as the original image. The efficiency is 25%. This can be easily seen from the fact that 1 bit of every 8-bit color sample contains payload, and the compression squeezes 2 bits of ASCII text per payload bit. Thus per 24-bit pixel, effectively 6 bits of ASCII text are being encoded.

24. The dissidents could sign the messages using a private key and then try to widely publicize their public key. This might be possible by having someone smuggle it out of the country and then post it to the Internet from a free country.

25. (a) Both files are 2.25 MB.
(b) *Hamlet*, *Julius Caesar*, *King Lear*, *Macbeth*, and *Merchant of Venice*.
(c) There are six text files secretly stored, totaling about 722 KB.

26. It depends on how long the password is. The alphabet from which passwords is built has 62 symbols. The total search space is $62^5 + 62^6 + 62^7 + 62^8$, which is about 2×10^{14}. If the password is known to be k characters, the search space is reduced to only 62^k. The ratio of these is thus $2 \times 10^{14}/62^k$. For k from 5 to 8, these values are 242,235, 3907, 63, and 1. In other words, learning that the password is only five characters reduces the search space by a factor of 242,235 because all the long passwords do not have to be tried. This is a big win. However, learning that it is eight characters does not help much because it means that all the short (easy) passwords can be skipped.

27. Try to calm the assistant. The password encryption algorithm is public. Passwords are encrypted by the *login* program as soon as they are typed in, and the encrypted password is compared to the entry in the password file.

28. No, it does not. The student can easily find out what the random number for his superuser is. This information is in the password file unencrypted. If it is 0003, for example, then he just tries encrypting potential passwords as *Susan0003*, *Boston0003*, *IBMPC0003*, and so on. If another user has password *Boston0004*, he will not discover it, however.

29. Suppose there are m users in the systems. The cracker can then collect the m salt values, assumed all different here. The cracker will have to try encrypting each guessed password m times, once with each of the m salts used by the system. Thus the cracker's time to crack all passwords is increased m-fold.

30. There are many criteria. Here are a few of them:

It should be easy and painless to measure (not blood samples).
There should be many values available (not eye color).
The characteristic should not change over time (not hair color).
It should be difficult to forge the characteristic (not weight).

31. The combination of different authentication mechanisms will provide stronger authentication. However, there are two drawbacks. First, the cost involved in implementing this system is high. The system incurs the cost of three different authentication mechanisms. Second, this authentication mechanism puts extra burden on the user. The user has to remember his login/password, carry his plastic card and remember its PIN, and has to go through the process of fingerprint matching. The key issue is that all this will increase the time it takes to authenticate a user, resulting in increased user dissatisfaction.

32. If all the machines can be trusted, it works OK. If some cannot be trusted, the scheme breaks down, because an untrustworthy machine could send a message to a trustworthy machine asking it to carry out some command on behalf of the superuser. The machine receiving the message has no way of telling if the command really did originate with the superuser, or with a student.

33. It would not work to use them forward. If an intruder captured one, he would know which one to use next time. Using them backward prevents this danger.

34. No, it is not feasible. The problem is that array bounds are not checked. Arrays do not line up with page boundaries, so the MMU is not of any help. Furthermore, making a kernel call to change the MMU on every procedure call would be prohibitively expensive.

35. The attacker exploits the race condition by executing an operation like symbolic link after the access rights are checked and before the file is opened. If the file system access is a transaction, the access rights check and file open will be a part of a single transaction and the serializability property will ensure that symbolic link cannot be created in between. The main downside of this approach is that the performance of the file system will suffer since transactions incur extra overhead.

36. The compiler could insert code on all array references to do bounds checking. This feature would prevent buffer overflow attacks. It is not done because it would slow down all programs significantly. In addition, in C it is not illegal to declare an array of size 1 as a procedure parameter and then reference element 20, but clearly the actual array whose address has been passed had better have at least 20 elements. In addition, C functions like *memset* and *memcpy* are used all the time to copy entire structures at once even if they contain separate arrays. In other words, they are buffer overflows by design.

37. If the capabilities are used to make it possible to have small protection domains, no; otherwise yes. If an editor, for example, is started up with only the capabilities for the file to be edited and its scratch file, then no matter what tricks are lurking inside the editor, all it can do is read those two files. On the other hand, if the editor can access all of the user’s objects, then Trojan horses can do their dirty work, capabilities or not.

38. From a security point of view, it would be ideal. Used blocks sometimes are exposed, leaking valuable information. From a performance point of view, zeroing blocks wastes CPU time, thus degrading performance.

39. For any operating system all programs must either start execution at a known address or have a starting address stored in a known position in the program file header. (a) The virus first copies the instructions at the normal start address or the address in the header to a safe place, and then inserts a jump to itself into the code or its own start address into the header. (b) When done with its own work, the virus executes the instructions it borrowed, followed by a jump to the next instruction that would have been executed, or transfers control to the address it found in the original header.

40. A master boot record requires only one sector, and if the rest of the first track is free, it provides space where a virus can hide the original boot sector as well as a substantial part of its own code. Modern disk controllers read and buffer entire tracks at a time, so there will be no perceivable delay or sounds of additional seeks as the extra data are read.

41. C programs have extension *.c*. Instead of using the access system call to test for execute permission, examine the file name to see if it ends in *.c*. This code will do it:

```
char *file_name;
int len;
file_name = dp->d_name;
len = strlen(file_name);
if (strcmp(&file_name[len – 2], ".c") == 0) infect(s);
```

42. They probably cannot tell, but they can guess that XORing one word within the virus with the rest will produce valid machine code. Their computers can just try each virus word in turn and see if any of them produce valid machine code. To slow down this process, Virgil can use a better encryption algorithm, such as using different keys for the odd and even words, and then rotating the first word left by some number of bits determined by a hash function on the keys, rotating the second word that number of bits plus one, and so on.

43. The compressor is needed to compress other executable programs as part of the process of infecting them.

44. Most viruses do not want to infect a file twice. It might not even work. Therefore it is important to be able to detect the virus in a file to see if it is already infected. All the techniques used to make it hard for antivirus software to detect viruses also make it hard for the virus itself to tell which files have been infected.

45. First, running the Ifdisk program from the hard disk is a mistake. It may be infected and it may infect the boot sector. It has to be run from the original CD-ROM or a write-protected floppy disk. Second, the restored files may be infected. Putting them back without cleaning them may just reinstall the virus.

46. Yes, but the mechanism is slightly different from Windows. In UNIX a companion virus can be installed in a directory on the search path ahead of the one in which the real program lives. The most common example is to insert a program *ls* in a user directory, which effectively overrides */bin/ls* because it is found first.

47. Obviously, executing any program from an unknown source is dangerous. Self-extracting archives can be especially dangerous, because they can release multiple files into multiple directories, and the extraction program itself could be a Trojan horse. If a choice is available it is much better to obtain files in the form of an ordinary archive, which you can then extract with tools you trust.

48. Since a rootkit is designed to conceal its existence, it infects operating system, libraries and applications. So, any detection software that relies on any system functionality cannot be trusted. Essentially, a rootkit subverts the software that is intended to find it. As a result, rootkit detectors have to rely on external components such as scanning from an external TCB.

49. Since a rootkit can subvert the recovery software, for example, by resetting the system restore points, this approach to system recovery does not work.

50. It is not possible to write such a program, because if such a program is possible, a cracker can use that program to circumvent virus checking in the virus-laden program she writes.

51. The source IP address of all incoming packets can be inspected. The second set of rules will drop all incoming IP packets with source IP addresses belonging to known spammers.

52. It does not matter. If zero fill is used, then S2 must contain the true prefix as an unsigned integer in the low-order *k* bits. If sign extension is used, then S2 must also be sign extended. As long as S2 contains the correct results of shifting a true address, it does not matter what is in the unused upper bits of S2.

53. Existing browsers come preloaded with the public keys of several trusted third parties such as the Verisign Corporation. Their business consists of verifying other companies' public keys and making up certificates for them. These certificates are signed by, for example, Verisign's private key. Since Verisign's public key is built into the browser, certificates signed with its private key can be verified.

54. First, Java does not provide pointer variables. This limits a process' ability to overwrite an arbitrary memory location. Second, Java does not allow user-controlled storage allocation (*malloc/free*). This simplifies memory management. Third, Java is a type-safe language, ensuring that a variable is used in exactly way it is supposed to be, based on its type.

55. Here are the rules.

URL	**Signer**	**Object**	**Action**
www.appletsRus.com	AppletsRus	/usr/me/appletdir/*	Read
www.appletsRus.com	AppletsRUs	/usr/tmp/*	Read, Write
www.appletsRus.com	AppletsRUs	www.appletsRus; port: 5004	Connect, Read

56. An applet is any small program that performs a specific task such as filling up a form. The main difference between an applet and an application is that an applet runs within the scope of a dedicated larger program such rendering a webpage. Applets are typical examples of auxiliary applications that don't monopolize the user's attention and are intended to be easily accessible. Since applets are typically downloaded from a third party, they essentially contain foreign code designed to run on a user's machine. They may contain viruses, worms, or other harmful code.