DSV SU *SEC:I*
Information and Computer Security
Autumn Term 2011

Alan Davidson

Suggested Solutions to the Exam 2011-12-19 and
Comments on the Marking

The answers suggested here may contain more aspects or discussion than would normally be expected from students to achieve the highest grade. The intention is to give students not only explanations of why their answers were graded as they were, but also to give students plenty of food for thought (if applicable, for future exam sittings).

Problem 1

> Access control mechanisms can be instruments for upholding the confidentiality, the integrity and even the availability of IT systems. Using the access control of a computer's local file system as an example, discuss in what manner each of the CIA factors can be upheld.

The access control mechanisms in a local file system will most commonly be administered with Discretionary Access Control, i.e., the owner of each file is allowed to define what that files privileges should be. The owner can thereby limit access to that file according to what rights the system defines, and according to what classes of subjects the system allows. For example, a unix system will allow (among others) the rights to read, write, and execute a file system object. Each object can have separate set of rights for each class of users that are associated with that object. Standard unix access control allows the rights to be set for the owner of the object, for all other users of the system, as well as for a single defined group of subjects.

So how might this apply to confidentiality? A simple answer might be to say that since confidentiality implies not being able to read, one can implement confidentiality with an access control mechanism by ensuring that subjects that should not have access to information to an object do not have read privileges for that object.

A more complicated answer might take into account more than the confidentiality of individual objects and account for system-wide confidentiality, such as in the Bell LaPadula model. In this model subjects and objects are all classified into ordered layers of confidentiality. According to this model subjects with lower confidentiality may not read objects which belong to classes with higher confidentiality. Furthermore, subjects may not write to objects that are at in a lower class of objects in order that they should not thereby release confidential information. If we limit the write and read privileges of subjects over objects according to this scheme we can claim to be upholding confidentiality.

A secure system will uphold data integrity, i.e., ensure that data is correct (N.B. it is not just an issue of ensuring that data does not change, as some answers said. It can also be a matter of ensuring that data changes correctly). Integrity will depend on much more than just access control, yet access control can play a role. It can control who is allowed to write to an object, and inasmuch as only trusted subjects are allowed to write to an object, then it follows that the changes made by those subjects can be trusted. The Biba model gives us an idea of how to uphold integrity in this way. As with Bell LaPadula, subjects and objects are classified into ordered integrity (or trust) classes. Subjects at a certain level of integrity may only write to objects of an equal or lower level of integrity. If a subject does not have write permission to objects of higher integrity then that subject will not be able to taint such objects. What is more, a subject at any integrity level should not even be given read access to an object of lower integrity, lest they become corrupted by that object. It can therefore even be relevant to limit read access to some objects to ensure system integrity.

The discussion of how to uphold availability through access control is more vague and esoteric than the for the above factors. This difficulty was even hinted at in the problem text. One should not fall

into the trap of just assuming that availability is simply the complement to confidentiality. Upholding availability in a secure system involves far more than upholding confidentiality. Denial of Service is an attack on availability, and measures may be put into place to mitigate the occurrence or effects of such attacks, while such measures have no direct bearing on confidentiality. With this part of the problem one therefore had to be careful in order not to give an impression of weak understanding of availability. One good way to answer this part of the problem could be to simply state that there is no direct method to affect availability though access control.

If one were to press the point, one might make the case for an indirect role for access control in availability. For example, one might claim that there is an overlap between integrity and availability in that an untrusted subject should not be allowed 'delete' privileges over an object that must remain available. One could also say that process that uphold availability, say backup, network services, or mirroring processes should be allowed appropriate access.

Problem 2

> When discussing the principles of authentication we often assume that the subjects for authentication are users, but these are by no means the only kinds of objects that require authentication.
>
> Suggest three realistic situations where other objects than users might require authentication in IT based systems. Describe suitable authentication methods for each of these three situations. Your answer should show a breadth of understanding of varying methods of authentication. Discuss also measures that might be taken to ensure that your authentication methods are as secure and effective as is appropriate for the given scenarios.

The problem asks for a breadth of examples of authentication. One way to ensure that you cover the requisite breadth might be to ensure that the authentication methods that you choose come from the different kinds covered on the course, i.e.:

- something the entity knows
- something the entity has
- something the entity is
- where the entity is

However, under closer scrutiny these categories can be seen to be biased towards authentication of users since they can be difficult to apply for other subjects. Can anything other than a user know anything? Would one classify an IP address as something a server host has, something it is, or where the host is? The classification unfortunately seems to easily break down. The very best of answers will not use the classification blindly, but with care.

An https server might authenticate itself to a client. This is by means of the server proving that it has a certain private key. Traffic to a client is digitally signed with this private key. The client is assumed to hold the corresponding public key within a trusted certificate. The client can therefore verify that the server what it claims to be by verifying signed messages from the server. This authentication method may be susceptible to a man-in-the-middle attack, since all we really know is that the traffic originated from the server, but an impostor may be forwarding the traffic to you to. It is therefore not entirely sure that the server sending data to you is in fact the holder of the said private key. What you do know, on the other hand, is that if you encrypt data that you send in reply with the public key, any impostors who are diverting that traffic will not be able to read that data.

Other examples of 'something the entity has' scenarios that one might go into greater depth with include software serial numbers and dongles that used to authenticate copies of software, documents that are authenticated with digital signatures similarly to the https traffic. I would be suspicious of whether authenticated emails fulfill the exam problem requirements given that emails are themselves seldom authenticated, but it is more likely that we are authenticating the sender, i.e., the user.

In terms of 'something the entity is', this is another category that is difficult to translate to non-user subjects. Otherwise we often associate this category with the field of biometrics. Factors that can be

said to be authenticating and an integral part of other objects are not obvious.

In the Trusted Computing scheme peripheral devices such as high definition monitors can be required to authenticate themselves to the computer. This can be utilised in digital rights management schemes so that, for example, a computer will only play a DRM secured blu-ray medium in full definition if it is connected to an authenticated monitor, and not a recording device.

A novel suggestion for this kind of identity based authentication is that one might check a number of parameters that are know for a certain computer, such as response time, temperature profiles, etc. This is certainly in the domain of 'something the entity is', but as yet it does not have the stamp of believability.

The location of subjects is often used in authentication schemes. That location might be physical, or within logical networks, or relative to another device... Within a university building computers connected into the network on the staff floor are authenticated as being machines that can be given full access to printers on that floor, whereas computers added to other floors are required to go through a separate authentication process before they can use printers. This is a quick and efficient authentication scheme that presupposes that only responsible users of the department's resources will can access to the staff floors or use network sockets without immediate detection. It is clearly not a very exact or foolproof method, but presumably effective enough to avoid too much misuse of university resources. A more exact authentication mechanism such as the one used on other floors is clearly possible, but presumably unnecessary and may instead have a detrimental effect on resource availability.

There were a number of problems with students answers where they did not follow the problem text. Some only mentioned kinds of entities and authentication without entering into a discussion on the scenario. Some answers did not have any discussion on the effectiveness of their suggested methods even when that discussion was clearly warranted.

Problem 3

> Explain how the application of Saltzer and Schroeder's principles for the design of security systems might help to avoid the common vulnerabilities of
>
> - buffer overflows
> - poorly configured firewalls
>
> Be specific in which principles you consider to be the most relevant and why.

The following discussion is long and detailed. As with most other suggested exam answers this is not because I expect exam answers to have this level of detail, but because I want to address many of the issues that arose from the answers that students gave. Maybe I am clouding the central issues by discussing all the others. If that is the case, I can cut things short by saying the best answers were based on the link between buffer overflows and complete mediation, and the link between firewall configuration and psychological acceptability.

A number of answers were so vague and confused as to make me question whether the examinee knew what buffer overflows were. An understanding of what buffer overflows and firewalls are is of course necessary for success in this problem.

Buffer Overflow

Buffer overflows result from accepting input to a program that does not fit into the receiving memory buffers, and yet still continuing to write to that memory. Clearly the main issue here is that input is not being properly checked, while the principle of complete mediation requires that all accesses to objects (in this case memory buffers) be checked to ensure that they are allowed. It would be a mistake to miss the relevance for this principle.

Another relevant principle here is that of least privilege. The most serious kinds of buffer overflow are those that allow the user to insert code that is subsequently executed. In these situations foreign code is executed with the same privileges as those that the running program has. For this reason it is

an unnecessary risk to allow the program to run with higher privileges than are strictly necessary for the program to complete its task.

For similar reasons one can make a case for observing the principle of least common mechanism. Once a buffer overflow attack is executed the resources of the machine that is under attack are vulnerable. The less resources that are shared, the less is vulnerable.

Creating a buffer overflow vulnerability is a programming error. One could therefore propose that if the program code was simple then errors such as this would be less likely. In that case one would argue for the principle of economy of mechanism.

I can see no case for the principle of open design. The principle states that the security of a mechanism should not depend on the secrecy of its design or implementation. The buffer overflow vulnerability is not a case of failed dependency on secrecy. Some confuse the principle of open design with that of open source, and suggest that if others (“many eyes”) could read the source code then the problem would be discovered. Though we have in general discussed the role of openness in security it is a stretch to bake all such implications into the principle of open design.

I also have difficulties making convincing arguments for the relevance of fail-safe defaults, separation of privilege, and psychological acceptability in mitigating buffer overflow vulnerabilities.

Design of Firewalls

The fact that firewalls are commonly poorly configured suggests that the problem may well be in the design of the firewalls which means that they are difficult to configure. I suggest that in this situation there are two relevant Saltzer and Schroeder principles. Most obvious is that of psychological acceptability which states that a resource should not be more difficult to access than if the security mechanisms were not present. At face value the principle seems difficult to apply since which resource is being accessed may not be clear. I suggest that if we call the filtering power of the firewall the resource to be accessed then we can say that the configuration mechanisms should not get in the way of accessing the resource. In simpler terms, the firewall should be simple and transparent to configure properly.

Firewalls are security mechanisms that can play a vital role for system security. They should generally be well designed, and in that design we can make a case for the principle of economy of mechanism. Saltzer and Schroeder tell us that smaller mechanisms will have several desirable qualities, among them less likelihood to suffer from bugs. So this is a nice quality, but how does it relate to the problem of poor configuration as in the problem text? Some might claim that a system built with a simple mechanism implies that it will be simple to configure and use, but one can refute this claim. A simply built firewall such as *iptables* can be very difficult to configure since it must be done with a special rule based language where the order of the rules will have a vital effect on the firewall's actions, so speaking from personal experience I would say that this simple firewall is hard to configure. On the other hand a very complicated piece of software design can result in fancy user interfaces that give ease of configuration with good feedback. So I discount the simplicity of design aspect in easing configuration. However, in terms of the dangers that poor configuration might cause, a higher presence of bugs as one finds in more complex design could conceivably be the cause of greater problems than if the problem was poor configuration alone. From this discussion I would suggest that the connection between poor configuration and economy of mechanism is tenuous at best.

To some degree one might claim that a simple firewall could be simpler to configure. For example, the simplest of packet filtering firewalls might be relatively simple to configure because there is relatively little to consider – where a packet came from, where it is going... However, we have understood from our discussion of firewall types at that lecture that packet filtering firewalls are less powerful than stateful firewalls, application proxies and guards. So given a choice of a perfectly configured packet filter or a mostly correctly configured stateful firewall, it is not clear that the one is more secure than the other.

We can make the case for the principle of least common mechanism similarly to that of economy of mechanism. If the firewall shares many resources, it may not have a noticeable effect on the configuration of the firewall, but the failure of a configuration might be expected to have a more serious effect if the mechanisms for accessing resources are shared. If failed configuration allows an attacker to gain access to the machine that a firewall is running on, and that machine is only running the firewall, then the danger will be less than if that machine is also running proxy web-sites, customer databases, etc.

It would be a great surprise if any firewall did not implement the possibility of complete mediation, i.e. that all accesses should be checked to see if they are allowed. I suggest the the connection is so trivial and obvious that it is not meaningful to discuss it in this exam. I mean to say, is a cause of poor configuration the fact that firewalls are designed without complete mediation? - No, because they are not.

If a firewall is to be designed to work robustly in the face of possibly poor configuration then it seems to make sense that it should work according to the principle of fail-safe defaults, i.e. that it should by default refuse access, and only allow access if explicitly instructed to, rather than allowing access unless instructed not to. This principle will not give us magically well configured firewalls, but we might expect it to marginally reduce the likelihood that firewalls will be too open. On the other hand we should consider that an overly constrictive firewall is a problem in itself; that would be a threat against availability. So perhaps fail-safe defaults is ultimately not a such a useful principle as one might first hope.

Configuration of Firewalls

Another way to discuss this question is in terms of how the principles might apply to the administrator of the firewall rather than the designer of the firewall. The principles are intended for secure design but during the course we have suggested that they may be more generally applicable. In that case, maybe the principles can be used to guide in the firewall configuration.

Complete mediation in this perspective becomes more important. The administrator should ensure that all accesses to the resources protected by the firewall are properly checked.

The administrator should ensure that subjects are not given more privileges than are required to complete the task, i.e. apply least privilege. However, in the context of a firewall it is difficult to see how to apply this principle beyond the obvious. When a firewall grants access it is by allowing communication, otherwise it blocks communication. There is not exactly a range of privileges here that allow us to make much sense of the principle.

The administrator might have to take psychological acceptability into account i.e. not make the resource difficult to access than if the firewall was not present. I guess this means that you should neither block nor slow down communication that should run smoothly. The principle does not seem to add much to configuration.

Other principles have similar discussions as with the design of firewalls, or do not seem to be interesting.

Problem 4

> Describe each of the following IT security related terms. Also, for each of these terms further illustrate the concept by choosing a closely connected IT security concept and explaining the relationship between the concepts. Furthermore, give an example of an application of these tools/threats/concepts. Give concrete examples wherever possible. Structure each of your answers with headings *description*, *relationship to [your chosen related concept]*, and *example.*
>
> Some students may find it helpful to use the pre-printed problem 4 answer sheet for their answer. Those who choose not to should take care to follow the above instructions extra carefully.
>
> *Please note that in general a 50% complete answer will be required to obtain a pass mark for this problem*

- RSA

- Spacial privacy
- ACL (Access Control List)
- Integrity policy

Note how the related concepts and examples are chosen carefully so that they all together give a richer understanding of the term.

RSA

RSA is the name of the most commonly known and used asymmetrical cryptoalgorithm. It is named after its authors, Rivest, Shamir and Adleman. Very broadly speaking, the mathematics it is based upon is that large primary numbers are relatively easy to find whereas factoring of large number is difficult.

RSA as related to AES: Whereas RSA is a well known asymmetrical cryptoalgorithm, AES is a well known symmetrical cryptoalgorithm. RSA generates and uses a key pair where one is made public and the other the is kept private. AES uses single keys, the same for encryption as decryption.

Example of use: The RSA algorithm can only encrypt relatively short lengths of data. For this reason, when it is used for encryption the cleartext is encrypted with a symmetrical algorithm using a session key, and the session key is encrypted with the recipients public key. We are thereby assured that only the recipient can decrypt the session key with his/her private key. To sign a document a hash value of the document is calculated and then encrypted with the RSA private key. To verify the signature the signer's public key is used to decrypt the hash value, and the hash value can then be calculated on the document and then compared to the one within the signature.

Spacial Privacy

Spacial privacy is one of the two basic types of privacy rights. It can be summarised as “the right to be left alone”.

Spacial Privacy as related to Informational Self Determination: The other main class of privacy rights is Informational Self Determination which is not about being left alone but about having the right to decide what happens to information and/or data about oneself.

Example of spacial privacy: Spam emails are unsolicited emails that are most commonly undesired. The sending of spam can therefore be deemed to be an infringement of one's spacial privacy rights.

ACL (Access Control List)

An Access Control List is a mechanism for representing and enforcing access control rights. The specification of a subject's rights is associated with each individual object. Access Control Lists are one possible way in which to implement and economise an Access Control Matrix.

ACL as related to Capabilities: Whereas ACLs associate sets of rights with the objects, capabilities associate sets of rights with subjects. ACLs allow for generalisations over subjects, whereas capabilities allow for generalisations over objects.

Example of an ACL: If Jill is allowed to read and write file agenda.txt whereas Jack is only allowed to read it, this can be formally represented with the ACL (Jill{read,write},Jack{read}) associated with agenda.txt. Unix based systems allow an object's rights to be set for the object owner, a grouping of users, and the set of all other users.

Integrity Policy

Policies describe what is allowed and what is not allowed in system. An integrity policy is a policy that specifically concentrates on what is required and what is not permitted in order to uphold the integrity aspects of the system. The language used to express policies in general can cover a wide range from very formal to almost natural language. Integrity policies, in that they concentrate on only the integrity aspects, tend to be the more formal kind.

Integrity Policy related to Commercial Policy: The term *commercial policy* is used to characterise policies that are usual in commercial applications where the security requirements are to a larger part integrity requirements, but with some confidentiality requirements. An integrity policy is therefore insufficient on its own to represent a commercial policy.

Example of an Integrity Policy: The Biba model describes a language with which to specify integrity constraints in a policy. It classifies subjects and objects into levels of trust and in outline it upholds integrity by allowing subjects at a certain trust level to only read subjects at the same level or above while only allowing them to write to subjects of the same level or below.

Problem 5

> A private computer user has noted that there are many firewall products available and she asks you for advice on how to choose and run firewalls for the small number of computers she has in her home network. Her main worries are that some software might not be dependable.
>
> Give well motivated objective advice that can be of use in selecting a trustworthy firewalls, as well as some measures that are reasonable to take to ensure that the firewall systems run securely.
>
> You may assume that the user is an enthusiastic computer user, i.e., not entirely naïve and willing to go to reasonable effort to learn. You should not assume that firewalls will run on any particular operating system.

There are plenty of concepts, models and tools from the course that can be brought into this problem to say sensible things. Those who concentrated on home-baked advice and missed all the important associations to the course material would not earn good marks.

When it comes to instilling trust in software the most relevant part of the course matter that comes to mind is that of assurance evaluation such as with the Common Criteria. The advice could be to check through the Common Criteria website to see if there are any certified Protection Profiles for firewalls there that closely resemble the person's requirements. If there are then the next step is to search for products that are certified as fulfilling that Protection Profile.

Even if there is no matching Protection Profile the person could look through the certified Security Targets for a suitable candidate. In any event one is assured that several relevant, security related aspects of the system will have undergone scrutiny from trusted and objective parties.

A case may be made for open source products in that the 'many-eyes principle' suggests that problems in such systems are easily discovered since all are able to study the code that implements it. Note however that even open source products have been Common Criteria certified.

When it comes to advice on running the firewall...

The Common Criteria does make requirements on the documentation of the software, so the first advice must be to carefully follow the certified documentation.

After that, we can start to give some of the general advice that can be gleaned from the course material. I will just briefly mention two.

Following Saltzer and Schroeder's principles, one can give the advice that a firewall should be run on a dedicated machine that runs as little as possible besides the firewall. The reduces the complexity of the system and reduces the possible attack surface.

Do not put all your faith into a single firewall. Layering of protection is a sound security principle. If you have a single firewall at the bridge between the local network and the Internet it can filter traffic effectively at that vital point, but it will not protect computers from each other within the network. It might only take one mistaken software installation on one of the local machines to infect all of the network. For such reasons it is also wise to install personal firewalls on each of the end user computers.