DSV SU ***SEC:I/IV2024***
Principles of Computer Security
Autumn Term 2010

Alan Davidson

Suggested Solutions to the Exam 2011-08-25

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

> Classify each of the following as a violation of confidentiality, of integrity, of availability, or of some combination of these three. Explain and motivate your classifications.
>
> a) A student copies another student's assignment hand-in and submits it as their own.
>
> b) Close to the hand-in deadline a student (who is not ready with their hand-in) finds a buffer overflow bug in the system that receives hand-ins and manages to utilise that bug to crash the server at will, thereby earning more time to work on the assignment.
>
> c) A student eavesdrops the examiner's password and uses it to change his/her grade from an F to an A.
>
> d) A teacher pretends to be a student in an on-line forum that is meant only for students in order to keep track of what students are saying about him/her behind his/her back.
>
> e) A teacher, assuming that students use their computers to chat and play games during lectures rather than as study tools, disconnects the wireless lan in the lecture hall, even though the students are fully within their rights to use computers and the network.

Discussions around this problem will be relative to what requirements (i.e. what policy) are set on the system in question. For these suggested answers I base them on an idea of how things usually work at universities that I think we can all agree on. Some universities will no doubt operate differently from these assumptions, so I shall try to comment possible variations along the way.

a)

There is clearly an integrity issue here, since we can surely assume that a hand-in will be required to be written by the student that puts their name to it. The copied work has a mismatch between the author of the work and the name supplied as the author of the work, so the data is inconsistent.

Is there a confidentiality issue here? Possibly not. At DSV we seldom require students to take responsibility for the confidentiality of their work. On the contrary, on the SEC:I course we encourage students to communicate and share the knowledge they gain from assignment work. Many students like to hold onto their own work and not show it to others, so informally they might say it was a confidentiality problem if they tried to keep their work to themselves and the other student managed to copy it anyway. It would depend on how the student gained access to the assignment work. If the student did not just happen to see the work as it was printed out on a public computer or even ask the other student if he/she could have a copy, but actively contravened the policy by breaking in to system to copy it, then I think most people would assume that there were confidentiality issues here. But from the official policy on assignment work I suspect that the problem is assumed to be with the hand in, rather than with the fact that the student managed to read other student's work.

Having said that, some examiners from other universities that I have spoken to make the specific requirement that students must make every effort to keep their work confidential. This means that if two students hand in the same piece of work the examiner does not have to enter into arguments about who wrote the hand-in. Both are guilty, one for not protecting their work, the other for copying it, and even though we may not know which is which, both can be punished. In this scenario there is a clear confidentiality issue.

b)

Crashing the server to hinder others from handing in is an availability issue. An attack on availability is usually called a Denial of Service (DoS) attack. It is commonly assumed that DoS attacks are through overloading systems with data traffic, but any method that stops a service that should be available from being available is denying service.

We could also argue that there is an integrity issue, since a buffer overflow corrupts a system's memory in a way that the system designer had not intended. With more checks that the input data meets required parameters, the buffer overflow fault could have been avoided.

c)

The fact that we are dealing with a password and that the action required to come by it is eavesdropping, we can assume that the data should have been secret, so we primarily have a confidentiality issue. Integrity issues follow on from the confidentiality issue. Knowledge of the password seems to be enough to authenticate as the examiner, so the system is acting as if the subject is the examiner when it is not. The data introduced, an A grade instead of an F, is surely incorrect data and therefore an integrity issue.

I have heard some people suggest that if a student were to manage to crack into a system and change their grade on a security course then they should be worth the grade they assign themselves, so in some way we could claim that the grade is correct. I suggest that such people have a naive, technology based view of what IT security is. On the SEC:I course, if someone did this then they would not only be contravening the University and departmental policies, but they would have shown that they had not applied important aspects of the course, not least the parts on policies and ethics.

d)

The teacher is pretending or masquerading as a student so the very language used here tells us that she/he is not what she/he purports to be, i.e. we have an integrity issue. The problem text then goes on to say that this was done in order to gain access to information that would otherwise not be available to her/him. Access to that information would not normally be available to the teacher (maybe because if one knew that the teacher was reading input one would not say things about him/her) so as a result of the integrity issue we also have a confidentiality issue.

e)

This is a pure availability issue, and another DoS attack. The texts says that the teacher has contravened the students' rights to have access, so what the computers and networks are being used for (so long as it is withing the bounds of the policy) is not an issue.

Problem 2

> A standardised X.509v3 certificate as used in public key cryptographic infrastructures has many fields, two of which are the *Issuer's Distinguished Name* and *Subject's Distinguished Name*. Explain in as simple and concise terms as you can manage what these particular fields contain and what role they play when checking the integrity of a digitally signed document. You do not need to show specific understanding of the X.509v3 certificate in your answer, but an understanding of public key cryptography in terms of these data.

The Subject's Distinguished Name field links the public key contained in the certificate to its owner. Reference to the owner must be to a unique entity and not possible to confuse with others. This field is the only means of identifying who is the holder of the corresponding private key. Without this information it would not be safe to e.g. encrypt a message with the contained key since you would not know which entity was able to decrypt that message. Likewise when checking the integrity of the signed message of the problem text the public key of the signature is used to check that the message is exactly the same as when it was signed, but it is the name field that specifies exactly whose private key was used to sign.

The fields of the certificate are bound together by the whole data structure being signed by another party, i.e. the issuer, and this is the entity that is referenced with the Issuer's Distinguished Name field. The binding through a signature is what allows us to trust the link between the owner field and the public key. But in order to verify the signature we need to know exactly which entity has signed the certificate in order to retrieve their certificate to obtain the public key that matches the key used for the signature. As with the subject the issuer must be uniquely referenced with this name field.

In order that all name fields have the requisite unique reference to an entity, they must not only be in a standard form that allows for uniqueness, there must be some global scheme to ensure that, for example, the same name cannot be given to two different entities. In simpler versions of certificates than are specified in x509, email addresses are used as names, and email providers are trusted to ensure the uniqueness of any one address. In x509, the format is a so called Distinguished Name with formally specified sub-fields. The Distinguished Name is verified as referencing an entity by a Certification Authority (CA).

Problem 3

> Explain the differences between computer viruses (according to the usage of the term on the course and in the course literature) and computer worms. Suggest and explain reasons why worms have become more common than viruses in the latest years.

You can check the definitions of computer worm and computer virus in the course book. For the sake of this suggested answer I shall characterise the difference thus:

A virus uses programs as hosts for its propagation whereas a worm uses a system as host. In practice this means that a virus will have to wait for an agent to activate the infected program for it to become active and thereby spread to other programs. A worm on the other hand infects active systems and can activley work to spread itself to other activated systems. Since worms do not have to wait for activation they can potentially spread very quickly from system to system. Since it is common to install antivirus software on systems, malware that take a long time to spread stand a greater chance of being identified, analysed, and included in the lists of malware that properly updated antivirus software can deal with. A new fast moving worm may be able to spread quickly enough so as to beat the update cycles of anti-virus software.

A number of answers were very preoccupied with the effects of virii and worms, claiming that they usually do one thing or the other. All of these were quite naïve answers based on nothing from the course material, but from own misapprehension. There is nothing in the definitions of these malware (as used on the course) that means that they would have different kinds of effects. They are both programs and depending on the level of privilege they manage to attain can do anything on a computer system that any other kind of software can do.

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.
>
> - Rootkit
> - Sandbox
> - Bell LaPadula
> - Caesar Cipher

Rootkit

A rootkit is malicious software that is installed, or installs itself, on a system where it then alters the behaviour of a system on such a low level of its operating system as to make its discovery very difficult.

A virtual machine emulates a complete system, running within a system. A rootkit can similarly emulate certain functions of its underlying system, but only those functions that allow it to subvert the system and with malicious intent.

In a famous case Sony BMG distributed system altering software on music CDs. Though Sony BMGs intent may have been to protect their content other parties found that the mechanisms introduced could be used to subvert the systems they were installed upon. Though Sony BMGs intent may not have been malicious (though certainly ill-advised) the software could be used for malicious purposes, and is today known as the Sony BMG rootkit.

Sandbox

A Sandbox (in IT security circles) is an environment created with either software or hardware, or both, which is purposefully separated from any environment where unauthorised software or actions could otherwise breach the security policy. Untrusted system elements can be run, and often monitored with the aid of this separated environment.

A virtual machine is a complete emulated software system which can act as one kind of sandbox environment. Virtual machines are normally not specifically constructed to act as sandboxes so should be configured and used with care.

Java applets are small programs that can be downloaded and run by web browsers. If such programs should not be trusted the java environment will run them within a sandbox where they have a limited instruction set and limited access to resources.

Bell LaPadula

Bell LaPadula is a model for specifying confidentiality policies. It divides both subjects and objects of the system into separate privilege levels and limits the rights of how subjects may read and write to objects, thereby upholding confidentiality.

Biba is a model that is very similar to Bell LaPadula in the way that it separates subjects and objects into levels. In fact, Bibia was based on the Bell LPadula model. Bibi is however a model for specifying integrity policies, as opposed to the confidentiality policies of Bell LaPadula. The difference in the two is embodied in the rules for read and write right. Bell LaPadula rules can be summarised as no read up and no write down, whereas Biba specifies no read down and no write up.

Four levels of Bell LaPadula privileges could be top-secret, secret, confidential, and unclassified. A user classified at the confidentiality level would be allowed to read objects at the confidential and unclassified level, and write to objects at the confidential level and above.

Caeser Cipher

The Caeser Cipher is a very simple method where a text is encrypted by replacing its letters with letters from the alphabet 3 positions on, the last three letters shifting to the first three. It is rumoured to have been invented by Julius Caeser himself.

The Caeser Cipher is a very simple (an oft used example of) a substitutional cipher, Another simple cipher is known as the scytale where a strip is wrapped around a staff, the text written on the wrapped strip, and then the strip is unwound, leaving the message unreadable since the letters are jumbled. This, in contrast to the Caeser Cipher is an example of a transpositional cipher.

In the Ceaser Cipher in the English alphabet, the cleartext SECRET would be transcribed to the ciphertext VHFUHW.

Problem 5

> If you have built a secure software system, two possible ways to convince customers that your security is good may be to distribute the system as open source, and to Common Criteria certify

your system. Discuss advantages and weaknesses of these two schemes for assuring customers of your system's security.

The Common Criteria is an international standard with a widespread organisation of certified expertise available to assist in the specification and certification of security requirements as well as software that fulfils those requirements. Software that is certified with the Common Criteria will have its security assured according to worldwide understood principles. Though some details of the system's design are made public in the description of the Security Target, the implementation is only made available to the certifying body, and is therefore kept secret from those who might take advantage of knowledge of the implementation, such as competitors or crackers.

The process required for Common Criteria certification is however costly and time-consuming, which implies that competing products could be cheaper and quicker to market. If customers are not overly concerned with assurance certification then they could be expected to choose other products. What is more, Common Criteria documentation takes some expertise to interpret and understand the implications of so it can be claimed to be unsuitable for software that is adopted by anything other than large organisations who can afford that expertise.

Current versions of the common criteria are not easily adapted to current software life-cycle processes. This means, among other things, that if you distribute a patch or an update to your system it is no longer CC certified, or must be re-certified.

The Common Criteria is not a guarantee of security but a method of providing levels of trust in the software.

By producing open source software you are allowing prospective customers or other critical parties to view and analyse your code and help in finding faults and improving it. The "many-eyes-principle" suggests that software will be more secure if more people have the opportunity to find security defects.

The many-eyes-principle is not a proven method to improve security in that you are not guaranteed to have your system viewed by those knowledgeable and helpful enough to improve the system. There is of course also the risk that those who would seek to subvert your system's security will gain a deeper insight into its workings if they have access to the source code, and even be able to subvert its security more easily.

If you intend to make profit from your system then making it open source will mean that you will not be able to make money by selling copies of the software. It may not be so easy to make a profit through other means, such as providing support and consulting services on the running of your software.