DSV SU/KTH *IV2018(part 2)/Säk1b/IV2024*
Information and Computer Security
Autumn Term 2008

Alan Davidson

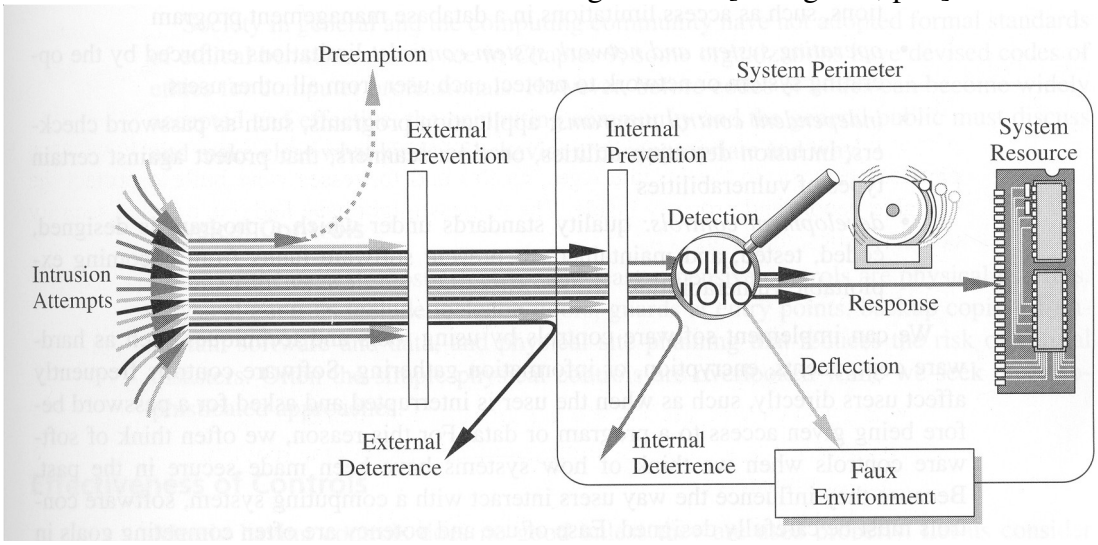
Suggested Solutions to the Exam 2008-10-22 and Comments on the Marking

Note that since these answers can cover several aspects of possible answers and also discuss these answers, they can contain more than would normally be required for full credit. Indeed, it is normally a strength if the student can give concise exam answers, whereas the following text seeks to be a thorough, comprehensive discussion.

Problem 1

> One possible measure to protect an IT system is an Intrusion Detection System. Suggest and motivate three other general, effective measures that a systems administrator can use to protect an IT system before the stage at which an IDS becomes necessary. Include at least one non-technical measure. Good answers will show a breadth of possible measures.

At the first lecture we saw an illustration from Pfleeger's book [Pfleeger07, p25]:



We see that before the detection phase (in the question represented by IDS) we have several points at which earlier controls could be introduced. We could generalise these to be *pre-emption*, *prevention* and *deterrence*. Since the question asked for breadth, a good answer might suitably present protection measures from each of these classes.

A pre-emption method (and a non technical one) might by to publish information to prospective attackers that can dissuade them. It might be an explanation that nothing of value is kept within the system, or convincing information that anyone found responsible for an intrusion can expect do be dealt with harshly (an equivalent from the physical world is posting a “Shop-lifters will be prosecuted” sign in a shop). Some students pointed out how such messages might instead act as a challenge and an enticement to some attackers. This may be true, but since no method is perfect, a case can nevertheless be made that this could be an effective method.

Whether a measure should be classified as prevention or deterrence may not always clear. I suggest that measures that clearly meet the threat of an attack are prevention methods, whereas measures

that lessen the effectiveness of an attack are deterrence methods. If prevention is concerned with understanding attacks and countering them, deterrence is about understanding the value of resources and making it difficult to compromise.

An effective prevention method can be a firewall. Intrusion attempts can come from networks and many times will be follow a pattern that differs from normal network traffic. Blocking traffic that is not part of normal network communication with an organisations servers will prevent such attacks.

An effective deterrence measure is to implement a proxy server at a point on the network that is to some degree on the outer limits of a local network. In this way network traffic that is external to an organisation, and therefore less likely to be trusted, is directed to a server that both contains less valuable resources and is in a part of the network that is less sensitive.

Some answers brought up faux environments such as honey pots as possible measures. It is however very difficult to see how an intrusion attempt can be deflected to such an environment before it has been detected as an intrusion.

A number of students chose to answer with more than the three measure that the question asks for. I should perhaps therefore emphasise that answering with more than the question asks for is always a worse answer. Apart from anything else, it is indicative of a lack of understanding of the subject and of your own knowledge. It also creates unnecessary extra work for an examiner (assuming that the answer is not rejected immediately as an improper answer to the question, which it maybe should be). The only examination strategy that I can imagine is fair and just in such a situation is to pick the three worst answers and mark according to them. Rather than trying to convince students that there are very many arguments for them not to start making assumptions outside of the framework of a question, I would like to just remind students that it is always a good idea to (simply) answer the question as it is written.

Several students seemed to take this question as an opportunity to write pages and pages on things that they knew details about, such as details of choosing passwords, or Saltzer and Schreoder's principles for the design of security software. Such off-subject dumping of knowledge never improves an answer. It does annoy the examiner that there is so much irrelevant text to wade through when marking exams! Since it definitely reduces readability, there is a case to be made for not marking such answers at all.

Problem 2

> The concept of key-space for an cryptographic algorithm is similar to that of the space of all possible passwords that an authentication method allows.
>
> a) Identify and motivate a factor (or factors) that effects the strength of keys and of passwords which are similar for both of these concepts.
>
> b) Identify and explain a factor (or factors) of the space of all possible passwords that can make the authentication method weak, and that you would not normally expect to see occurring in the key space for a cryptographic algorithm.

a) Length is surely the easiest factor to cite. The basic space of all possible passwords is a function of the number of characters that may be used in a password and the number of positions of such characters, i.e. the length. Modern symmetric cryptographic methods in support of IT do not share same factor of the number of possible characters in that they are normally binary numbers, i.e. there are only two “characters” to choose between, *1* and *0*. The size of the key space, and thereby the strength, is however similarly effected by the length of the key.

There are some subtle points of similarity and difference between password length and key length that could suitably be ignored for the sake of the exam question, but which some exam answers went into. Passwords will normally have variable length, meaning that the basic space of all

possible passwords will be a product of all the possible (*character set size* raised to the power *number of positions*) from e.g. *number of positions = 0* to *number of positions = 256.* Modern *c*ryptographic keys, on the other hand, tend to have a fixed length. DES for example has an effective key length of 56. Some encryption algorithms do allow variable length keys.

b) One possible answer: Cryptographic keys are generated by mathematical functions that are designed to give close to random selection of keys. Within the space of possible keys, these mathematical functions ensure that any one key is as likely to occur as another. Passwords on the other hand are most commonly chosen by people, not by mathematical functions. A normal requirement for passwords is that the creator be able to remember them. For this reason people will tend to limit their choices to patterns of characters that are easier for themselves, or for people in general to remember. For this reason, some passwords within the space of all possible passwords will be more likely than others. The space therefore becomes a topology rather than an even space. This is why dictionary attacks can be assumed to be more affective than brute force attacks for passwords. There is no equivalent to dictionary attacks for guessing at cryptographic keys.

Problem 3

> Describe each of the following IT security related terms. Also, clearly relate each of your descriptions to a closely connected IT security concept of your own choosing, and give an example of an application of these tools/threats/concepts:
>
> - Steganography
> - Risk Analysis
> - Formal Verification
> - Common Criteria

This question is of a kind that checks knowledge of some of the more detailed terms and concepts that are part of the course material. When marking the answers I assumed that being able to provide good descriptions, comparisons and examples of at least three of these terms during the exam could be regarded as fulfilment of the course goals and therefore worthy of a pass grade.

Steganography

Description - Steganography is the name for a class of methods that are used to hide the existence of information.

Comparison – The word c*ryptography*(in direct translation *hidden writing*) has come to mean the art and science of hiding the content of a collection of information (often a message), i.e. the fact the information exists is not hidden, but the information itself is difficult to extract. Cryptographic messages can therefore be subject to traffic analysis techniques, i.e. deriving information from the existence of traffic even if the content is not possible to interpret. This is in contrast to *steganography* (directly translated *covered writing*), here the fact that a message exists is hidden.

Example – It is possible to hide information within jpeg encoded pictures. This method is known to have been used by terrorist to spread text messages within seemingly innocuous pictures.

A number of answers mixed up examples of steganography with the description. Note that it is **not** the art and science of hiding messages in pictures. That is just one of the many possible steganographic methods.

Risk Analysis

Description – To quote [Bishop05, p15]:

> *To determine whether an asset should be protected, and to what level, requires analysis of*

the potential threats against that asset and the likelihood that they will materialize. The level of protection is a function of the probability of an attack occuring and the effects of the attack should it succeed.

Comparison – Vulnerability analysis is the process of testing for security flaws in a system, often with the aid of automated tools. The major difference between risk analysis and vulnerability analysis is the aspect of factoring in the likelihood. Otherwise we can say that risk analysis will tend to be applied at an organisational level, whereas vulnerability analysis is an activity directed at computer systems and networks. The goal of risk analysis is to ascertain what should be protected, whereas vulnerability analysis gives an indication as to what flaws should be fixed.

Example – As part of a risk analysis, an organisation might determine that a major asset is their customer database. Losing this to competition would be a threat to the whole organisation's survival. However, since the customer database is kept in the charge of trusted personnel and on an old computer that is not connected to the Internet, and with backup on a save media that is kept well secured at another physical location, the likelihood that the customer database will be lost or discovered by competitors will be very small, and any more protection measures would be superfluous.

Formal Verification

Description - Method by which the semantics of the design of a system is expressed with mathematical stringency, and thereby the design and implementation can be verified to be correct.

Comparison – The process of testing is a hunt for flaws, but as Dijkstra has written: "Program testing can be used to show the presence of bugs, but never to show their absence" [Dijkstra72]. Formal verification can on the other hand (at least in theory) prove that there are no flaws in the implementation of a design. With test one can put as much effort into the process as one deems suitable for the level of assurance required for the situation. Formal verification is regarded as an expensive measure that is most suitably applied in situations where the very highest levels of assurance are required.

Example – The reference monitor of a trusted operating system is a small but vital component. It must mediate all accesses to memory, and if it can be subverted then the security of the system overall will be in question. By specifying the design of the reference monitor in terms of predicate calculus one can symbolically manipulate that design in order to prove its completeness, i.e. that the process can not be subverted. A static analysis of the semantics of the implemented code can be compared to that design to show the presence or absence of bugs even before the code has ever been run.

Common Criteria

Description - The Common Criteria is an internationally standardised method by which the security functionality of software can be formalised, assessed, and certified.

Comparison – The Trusted Computer System Evaluation Criteria - TCSEC (also know as “The Orange Book”) was a predecessor to the Common Criteria that was developed in the United States in the late '70s and early 80's. One of the main differences between the methods is that TCSEC was designed for the evaluation of secure operating systems, whereas the Common Criteria is intended as a method for evaluating software in general.

Example – An organisation that wishes to fulfil their needs for secure filtering of network traffic might formalise their requirements as a Protection Profile (PP) for a firewall, according to the Common Criteria Guidelines. They may also have this PP certified by an CC authorised third party in order to assure that the PP itself upholds the CC requirements for such a document. Such a firewall PP will normally be general enough to match the requirements of any number of

prospective customers.

A software vendor who wishes to show that their product (e.g. firewall) is trustworthy may describe its security functionality in a Security Target document. They can then evaluate their product (in CC terms the Target of Evaluation – TOE) in relationship to the Security Target, or rather have it evaluated by an authorised third party, and even have their product certified as CC compliant. Especially if the ST was derived from an existing PP, the product can therefore be shown to meet customer requirements. The CC also specifies 7 different Evaluation Assurance Levels (EALs) which define at what level of confidence an evaluation can say that the security functionality requirements have been met. Note that an EAL does not specify a level of security so much as an level of assurance that the ST stated security functionality requirements for a piece of software have been fulfilled.

Note that the Common Criteria is not intended as an evaluation method for complete software or computer systems, but only for individual software.

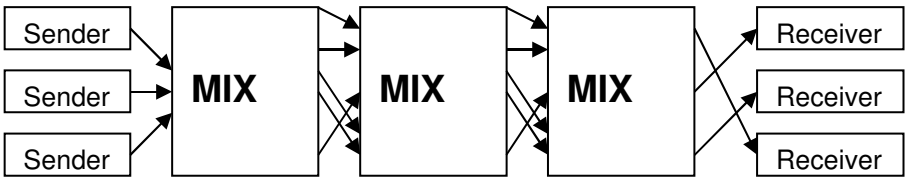
General comments...

Many students entirely missed the requirement for a related concept and example, or did not make it clear from their presentations what concepts they were attempting to relate to or where the example began and ended. Perhaps this is because students are studying from previous suggested solutions to exams from the course before the Bologna system was introduced? In previous years the older version of this problem was less exacting, and likewise the suggested answers. It is therefore important to note that of late the format of this question has changed. To emphasise the newer requirements, I have attempted to be extra clear in the above suggested answer. Once again the importance of being sure that students answer the question as written is emphasised.

"Relate each of your descriptions to a closely connected IT security concept of your own choosing" does not mean that you can simply state that another concept is related. You must of course explain **how** they are related too.

Problem 4

The diagram below is illustrates the basic architecture of a privacy enhanced email system. Explain why it is an advantage to have a sequence of several mix nodes between sender and receiver. Explain furthermore how cryptography can improve the anonymity service.



In very general terms, we can understand that having several mix nodes implements greater defence in depth. If the security of one mix node should fail then we have assurance in that the remaining nodes will be sufficient to uphold the anonymity service. Some of the possible ways in which a single node might fail are:

- A system might be subverted, and record the IP addresses of the sender together with the recipient. If there were only one node then this information would be enough to undermine the anonymity service. If however either the sender or recipient were another mix node in a chain of nodes, then all of those nodes would have to be subverted in order for the anonymity service to be undermined.
- By eavesdropping both incoming and outgoing traffic to a node it may be possible to match factors such as the delivery times and message sizes, and thereby match the sender with the

recipient. In order to accomplish this match with a sequence of mix nodes it would be necessary to either eavesdrop incoming and outgoing traffic at all of the involved nodes, or to have to guess at who the sender and recipient will be and eavesdrop immediately in their vicinity. With a mix of nodes, the route that a message will take will not be known a priori, and eavesdropping will be all the more difficult.

- If a node were a pseudonymous remailer (not usual in *mixnets*, but conceivable), that means it would keep a table of the senders' addresses mapped to connected identities. Messages would be stripped of any information pertaining to the true sender and re-sent with an anonymous identity. One advantage is that it is simple in such a system to allow for replies to anonymous messages by forwarding them through the same server and swapping back anonymous identities for true addresses before forwarding. However, a single server that contains such tables might be forced to reveal their associations by national authorities, as was the case with a classic anonymity server at penet.fi (see e.g. http://en.wikipedia.org/wiki/Penet_remailer) and the reason for it discontinuing the service. If a chain of servers were spread geographically, then the international cooperation that it would require to have all the necessary associations revealed will make the process all the more difficult.

Given the threats from eavesdropping both within a node and in the traffic to and from a node, a good anonymity service must not allow enough information to be gleaned from a message to derive the link between sender and receiver at any point in its delivery. If the message were sent in clear then all of that information would be available in the message as it comes from the sender. It is therefore necessary to hide not only the recipient of a message, but also any of its content that could be matched in two versions of that message eavesdropped en route.

With the help of asymmetric cryptosystems one can encrypt a message with aid of the public key of a mix node, knowing that only that node will be able to decrypt the content. The recipient of the message in the next step from that node can also be encrypted with the aid of said public key. In this way, only the original sender and the receiving node will ever know who is the next recipient in the chain, as it is encrypted for all other parties. Now by successively asymmetrically encrypting the recipient and the message content for each node in a chain, each node will unveil the message for the next step by decrypting the message it receives. Traceable links between the message as sent and as delivered are effectively masked.

An added bonus of the asymmetrical cryptography scheme is that only the legitimate recipient node will be able to decrypt their layer of the message. So long as the public keys that are used are legitimate, anyone who attempts to spoof a mix node to thereby attack anonymity will not be able to.

Problem 5

> Pick three of Saltzer and Schroeder's eight principles for the design and implementation of security mechanisms, and explain how failure to observe each of these principles can result in three classes of common program vulnerabilities.

This question does not ask for any old vulnerability, but *common program vulnerabilities.* A good starting point therefore would be to pick program vulnerabilities that can be motivated as common. Here we take the three most common vulnerabilities from NISTs statistics over reported vulnerabilities thus far during 2008 (see http://nvd.nist.gov/statistics.cfm). Students are not necessarily expected to have these statistics at hand for the exam, but luckily enough these common vulnerabilities were among those discussed at lectures.

The principle of complete mediation requires that all accesses to objects must be checked to ensure

that they are allowed [Bishop05, p203]. SQL Injection is a common program vulnerability that can occur when input taken from a client is assumed to be well formed and catinated into a SQL query. Carefully crafted input can adapt the query to result in database accesses that were not intended by the programmer. We can put this down to the fact that the access to the database was not properly checked to ensure that it was allowed. Carefully checking the client side input to make sure that it is properly formed and therfore allowed will alleviate the vulnerability.

Short notes - The same principle can be linked to several other of the most common vulnerabilities: Cross-Site Scripting, buffer errors, input validation, path traversal, code injection. The fact that so many of the common vulnerabilties can be linked to this principle suggests that it should be part of a good answer. Another principle that can be linked to SQL Injection is that of fail-safe defaults. Following this principle one would not allow the input to be part of the executed statement, but intead only use pre-constructed SQL statements.

The principle of least common mechanism states that mechanisms used to access objects should not be shared [Bishop05, p206]. One common cause of buffer overflow vulnerabilities is that languages such as C and assembler have mechanisms for accessing memory that are all too easily shared. Vulnerabilities can occur when parts of memory can be overwritten by user input (assisted in part by not keeping to the principle of complete mediation - see above) and thereby effect the course of execution in ways that the programmer had not predicted. Many other languages such as Java and Lisp have automatic memory management which means that mechanisms that share memory are not shared and not susceptable to the same kinds of buffer overlow vulnerabilities.

The principle of least privilege states that a subject should be given only those privileges that it needs in order to complete its task [Bishop05, p201]. NIST has the class of vulnerabilities *Permissions, Privileges, and Access Control*, which covers flaws from improperly handling among other things, privileges. There would seem to be a direct mapping from the principle to the vulnerability here. Giving programs more priviliges than they need opens up the possibility for such programs to be manipulated in order to subvert access control mechanisms.

There is some unintentional leeway for interpretation in the question. Clearly it is about three of Slatzer and Schroeder's principles, and the intention was that each principle should be matched to one class of vulnerability. However, one can interpret the mapping to be to up to 9 vulnerability classes in total (depending on how many of the classes might be duplicated for differing principles). All such answers were therefore accepted. Credit was given for the quality of the argument rather than for the number of associations.

Note that Open Design does not say that the many eyes principle is a good way to save your code from vulnerabilities, but that the security of a mechanism should not be dependent on its secrecy. In discussion during the course we have drawn parallels to this principle and the idea of many eyes, but one should be careful about what the principal actually states.

References

Bishop05 Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

Dijkstra72 E. W. Dijkstra, "Notes on Structured Programming," *Structured Programming*, O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Editors, Academic Press, 1972, pp. 1-82.

Pfleeger07 Pfleeger, C. P. and Pfleeger S. L., *Security in Computing, Fourth Ed.*, Prentice Hall, 2007

DSV SU/KTH *IV2018(part 2)/Säk1b/IV2024*
Information and Computer Security
Autumn Term 2008

Alan Davidson

Very Brief Suggested Solutions to the Exam 2008-12-06 and Comments on the Marking

The answers suggested here may be briefer than would normally be expected from students. I have tried to summarise the most important aspects of problems so that students may compare the content their own answers to these. Lack of time prevents me from doing more.

Problem 1

> Though the access control matrix is a powerful model, it is seldom used in practice for implementing access control in computer systems. Explain what access control matrices represent and suggest reasons why they are not usually implemented. Describe a commonly implemented access control mechanism, as well as how it is related to the access control matrix.

An ACM represents the rights that subjects have over objects in a secure system. Since there can be very many separate subjects and objects in a system, and very many different types of rights, the access control matrix can potentially become a very computationally expensive data structure to use, and a very costly structure to manage.

One common mechanism is the Access Control List, which is a data structure associated with each object. It is commonly associated with file objects, in which case it normally represents what users may do with each file. If we view the ACM as a table of rights with subjects listing the rows and objects listing the columns then an individual ACL for an object represents a single column of the ACM. Rather than listing the rights for each individual user, such an ACL will very often introduce a generalisation of subjects, such as the groups of unix type system. This lessens space needed as well as lessening the difficulty of managing rights, with some loss of expressive power. The ACL can therefore be viewed as an abbreviated version of the ACM column.

Several answers gave the impression that the ACM only represents mappings between files and users. Some effort has been spent on the course (as well as in the course literature) in viewing the ACM as a general model that applies to all types of subjects and objects. Such answers were therefore marked as weak.

Problem 2

> The course book describes four general classes of threats:
>
> > ...*disclosure,* or unauthorised access to information;
> > *deception*, or acceptance of false data;
> > *disruption*, or interruption or prevention of correct operation;
> > and *usurpation*, or unauthorised control of some part of a system. [Bishop05,pp4-5]
>
> Exemplify each of of these classes of threat with a description of a tool or method that was used during the first laboratory assignment of this course. Where a class of threat was not exemplified in the practical assignment you should instead illustrate it with a description of a realistic scenario. Good answers will discuss a variety of tools and methods.

A brief list of some of the tools that could have been discussed are...

Disclosure: Key Logger, Password revealer, sniffer, nmap

Deception: Arp spoofing, password dialogue spoof

Interruption: Netbus, shut down system

Usurpation: netbus

Problem 3

> Describe each of the following IT security related terms. Also, clearly relate each of your descriptions to a closely connected IT security concept of your own choosing, and give an example of an application of these tools/threats/concepts:
>
> - One-time pad.
> - Biba
> - Biometrics
> - SQL Injection

Since this is a shortened exam solution, rather than provide descriptions of the terms here I refer to the descriptions of these terms in Wikipedia.

Some concepts that could suitably be related to these with lengthy discussions are: the Vernam Cipher, Bell LaPadula, "What the entity is", The Principle of Complete Mediation. Examples that could be elaborated upon could include: How a spy uses a physical one-time pad, how a document that needs its integrity protected is treated in a four level Biba model, fingerprints, and inserting ' OR 1=1--' into an SQL authentication script.

Several answers suggested that one-time pads are an authentication mechanism. That is a very limited view of them. They should primarily be regarded as an encryption method.

Despite rewording this exam question (relative to previous exams) to be as clear as possible, very many students did not answer this question properly. It is clearly stated that each concept should be related to another, and an example given, yet many only provide a description. I suspect this may be because students have studied earlier exam papers and solutions where only descriptions were required, and are following them rather than answering the question as it is in current exams. If good and full answers are given for several concepts, then I can allow a pass even if one of the concepts is missed or wrongly answered. Only giving descriptions implies a very high risk of failing this exam question.

Problem 4

> Imagine that you are a member of a software development project that is designing client side software that is required to communicate securely with your company's server. One of your programmers says that she is confident that by putting together a very complex mix of different substitutional and transpositional ciphers that she can create a sufficiently strong symmetric cryptographic algorithm that can secure the communications. It is proposed that each copy of the client software that is sold will be prepared with its own unique cryptographic key, hidden within the software. On the basis of this algorithm and key, the client will be automatically authorised when sending properly encrypted messages to the server, and thereby allowed to make sensitive updates to a server side database.
>
> Given this description, (quite apart from any practical problems that you may be able to guess at) identify and argue for ways in which the project may be heading for security problems where they could instead be observing sound security design principles.

The complexity suggests that economy of mechanisms is lacking. The fact that the key is hidden in software indicates that the secrecy of this design will be of great importance, i.e., the principle of open design cannot be applied. That only this mechanism is sufficient for authentication indicates that the principle of separation of privilege is not adhered to.

Other ideas that were viewed favourably:

Though it is not one of Saltzer and Schroeder's principles we are aware from the cryptography part of the course that is it not a good idea to design your own cryptographic algorithms unless you have background as a cryptographer and mathematician. There is nothing in the text to suggest that the programmer does not have this background, but I guess the odds are against it.

It seems from the text as though the programmer is on her own, which would suggest that it is difficult to apply separation of duty (one again not a Saltzer and Schroeder principle, but nothing

says that the principles discussed have to be S&S). It is probably a good idea to have another person check that the design and coding works. Note however that the question is about the design of the software, not the management of the development process, so it does not really fit the question.

Some said that the fact that the client was allowed to update important information server side indicates that the principle of least privilege is not being followed. But if the client needs to update important information, then it will need the privileges to do so. If the text had said that the client was given the privileges to update important information even though it only needed to fetch a time-stamp, that would have been indication of ignoring least privilege.

Problem 5

> If you have to select a single practical method* for assuring the quality of security of software explain which one would you select and why. Provide also a motivation in comparison with the other methods.
>
> * Note that in this exam question general assurance schemes such as The Common Criteria and SSE-CMM are assumed not to be single practical methods, and may therefore not be included as part of your answer.

This question was by Albin Zuccato as a test of the subject that he lectured on, i.e. a discussion on inspection, testing and verification. The addition was by Alan in order to clarify that this was not to be mixed up with the aspect of assurance that was covered on other parts of the course. Despite the clarification many examinees apparently were unable to associate the subject matter with Albin's assurance lecture, and nevertheless gave answers that were associated with assurance schemes rather than practical development methods. I was finally very lenient in the marking of answers that had any thread of sensible answers, but for good marks only practical methods were acceptable.

Some answers unfortunately interpreted the question from the perspective of a customer, i.e., how do I know that the software I am buying has good security. This is an interpretation of the question that was not intended, but is admittedly possible, so this kind of answer was accepted. In this interpretation answers will unfortunately be more difficult and far more vague than the intended interpretation, which meant that it was difficult to gain good marks.

There is of course no given answer to whether inspection, testing or verification should be selected, but answers should show an understanding of each of them and a reasonable motivation for choosing one of them over the others.

Reference

Bishop05 Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

DSV SU/KTH *Säk1B/IV2024*
Information and Computer Security
Autumn Term 2009

Alan Davidson

Suggested Solutions to the Exam 2009-10-20 and Comments on the Marking

Note that since these answers can cover several aspects of possible answers and also discuss these answers, they can contain more than would normally be required for full credit. Indeed, it is normally a strength if the student can give concise exam answers, whereas the following text seeks to be a thorough, comprehensive discussion.

Problem 1

> Bruce Schneier has coined the term *Security Theatre* which he describes thus:
>
> "Security theater consists of security countermeasures[1] intended to provide the feeling of improved security while doing little or nothing to actually improve security." [Schneier03,p38]
>
> a) Discuss what influences in society may encourage security theater.
>
> b) Suggest in outline a more methodical approach to implementing security measures that would hopefully avoid the negative effects of security theater.

We might hope that any resources that are spent on security would help to improve security, but if Bruce Schneier is correct that is not always the case. In general we could surmise that where security theater[2] occurs the parties that are responsible for the implementing security gain more from giving the impression that they are doing something useful rather than actually doing something. There may of course be several reasons for this, but the question is specifically about influences in society, so in the following discussion we can exclude personal motives, such as the criminal (e.g. I will sell you an antivirus software that has trap-doors in it so that I can gain access to your computer without you knowing it) or lack of competence (e.g. I don't know how to secure my bosses computer, but he has told me to do it so I should do something that looks good to keep him happy).

a) The feeling of security clearly has a value to people and may make them act in desirable ways, such as to buy things on the Internet or to fly in aeroplanes, where feelings of insecurity might otherwise hinder them from doing so. However, security is no doubt too complicated a subject for the normal person in the street to understand enough to make judgements on whether security countermeasures actually work. At the same time there is much to suggest that that person in the street in not willing to pay much for security. At a very general level we can therefore identify *ignorance* and *greed* on the part of the consumer, and *commercial gain* for suppliers as contributing factors.

For example, after a number of terrorist attacks on aeroplanes[3], including those of September the 11th 2001, the general public may have an idea that they should seek other means of communicating and travelling. There are not only commercial interest in keeping the public flying, but national interests to do with the economy in general and investment in the infrastructures that allow for

---

1 The phrase "security countermeasures" can be a little confusing to foreign speakers of English, and I spent some time addressing this in the exam rooms. You could claim that Schneier has been a little sloppy in his English usage here, as "security measures" or "threat countermeasures" make better literal sense. Nevertheless it is quite natural to read this as meaning "countermeasures for the sake of security" which is surely Schneier's intention.

2 I use the American spelling since that is what Bruce Schneier naturally used when coining the concept. As an English speaker I am very tempted to alter the concept slightly and call it Security Theatre)

3 Examples from the field of IT security are normally preferred in exam answers, but this example happens to be one that Schneier himself refers to as an example of Security Theater.

communication and travel. Making security measures obvious to the traveller, such as prohibiting sharp objects on a plane (even though a ballpoint pen can be lethal in the hands of an experienced assassin), or limiting the amount of liquids allowed for each person (even though a colluding group of passengers could carry a dangerous amount of explosives when all their liquids are added together) could be part of a Security Theater that would make us feel that so much is done that we can continue to travel by plane, even suffering the indignity and the inconvenience that such measures might entail.

Note however that these Security Theater measures may have a role in *raising the level of security conciousness* in people, so that one would be extra vigilant. Understanding that security is important and that there are measures all around one, a person might be more inclined to report someone who was trying to set fire to their shoelaces on a plane than on the street. This is one example of when Security Theater might have an indirect real effect.

Another possible real effect is when fake security measures are employed to utilise the fact that attackers might be ignorant of the difference. For example, one can buy a flashing LED light to install in your car window to emulate the kind of light that shows that an advanced car alarm system has been activated. Just the flashing light might discourage some attackers from breaking into the car. But with this example we are getting away from the *influences in society* part of the question again.

Security Theater may be used when there is a real threat, but it is possibly more likely to be used *to counter a perceived threat*. There may be elements in society that have a vested interest in making the general public afraid of a threat, even when the threat is in reality not so great. It is easier for journalists to sell us stories that tell us that we should be scared of something, so in general we might expect the media to provide us with a picture of our society that is more scary and dangerous than the truth. Politicians, especially those in opposition to those that are in power, can gain from promoting a picture of insecurity which might make us believe their policies are sound. We might vote for a party that will promise to invest in law and order if we perceive that there is a growing crime problem. We might even give an elected government extended powers over us if we perceive that there exists a threat that the state can solve when they have such powers. For example, we might normally expect that our privacy rights would prohibit anyone from monitoring our network traffic. If, however, we believe that there is an imminent danger of terrorist attacks in our society we might be more complaisant when laws are introduced that allow government wire-tapping. Security Theater could therefore be used to implement a hidden agenda.

This is a discussion that can be held on several levels, and this example answer is only one of them. Whatever the level of discussion good answers are expected to show an understanding of the holistic nature of the subject of IT Security.

A few student discussions suggested that Security Theatre would not cost anything and is therefore better than doing nothing. Note that security theatre costs. Perhaps it costs less that real security, but it is a waste in real security terms, i.e. it could well be worse than doing nothing.

b) The problem here is that the security measures are presumably not having an effect on the security of a situation that is proportional to the cost of those measures. The quick and simple answer to this question is therefore to apply *cost benefit analyses* (though exam answers should of course also give an indication that the examinee understands the idea behind and what such an analysis involves). For a short explanation of what this involves you can refer to the course book [Bishop05, pp14-15]. Whether a cost benefit analysis is tractable in the situations where security theater can be found is a debatable matter, but the “hopefully” in the exam question allows us to stick our necks out a bit and assume that it could.

Some answers suggested that *risk analyses* would be required. Risk analysis can be seen as a counterpart to cost benefit analysis, and as such these answers were judged to be on the right track and an indication of an insight into the holistic nature of the subject.

On a more general level we could say that both cost benefit and risk analyses are reliant on good *security metrics*. This is an active field of research, and a tough subject. It is not a subject that figured greatly during the course though, so understandably answers that discussed the area of security metrics were few and far between.

A number of answers proposed the Common Criteria as a relevant security measure. As an approach to avoid Security Theater the CC is very limited, but it is certainly methodical. One could even claim that a method that certifies certain operating systems as secure, but the small print explains that this is only when computers are not connected to any peripherals, is helping to contribute to Security Theater! Answers that tied in the CC were given some credit, but were not judged as showing the same degree of the holistic insight of answers such as cost benefit analysis.

One answer suggested that the area of ethics could help against Security Theater. The argument was interesting and relevant, but it is hard to claim that it is methodical, as in the problem text.

Whatever actual method was chosen for the answer, points were awarded for convincing arguments, so long as they kept to the problem text.

Problem 2

> Having knowledge of which language is used in a clear text can assist a cryptanalyst in deciphering the corresponding ciphertext. This is true whether the cipher is based on substitutional or on transpositional methods (or a mixture of both). Explain in outline (i.e. you need not go deeply into cryptanalytical techniques) how.

Frequency analysis on natural language texts can for example ascertain how often a particular letter is likely to occur. In English, the most frequent letter is 'e', occurring 13.1% of the time [Davies&Price89]. In a simple substitutional method where each letter of the clear text has a direct mapping to a symbol[4] of the ciphertext, if we find a symbol occurring 13.1% of the time and if we know the language to be English, then we can surmise that that symbol was most likely mapped from an 'e'.

Patterns in language are not limited to frequencies of single letters. We find that certain combinations of letters are also language dependent. In English the letters 't' and 'h' occur together in a *digram* far more often than they do in, for example, Swedish. This also applies to *trigrams* such as 'the' and 'nce'. Imagine then a simple transpositional encryption method that has changed the order of the letters of a text. If we know that the text is English then if we can find a key and algorithm that would cause a large proportion of the letters 't' and 'h' to occur together then we have efficiently limited the space of possible algorithms and keys.

Note that the above explanation assumes that the encrypted text is long and non specific enough to exhibit close to the normal frequencies of a natural language. If a message is very short knowing the language help us in the specific way described here.

The above is just two examples of the kinds of patterns that could be utilised. Several student answers had other good examples, such as the fact that letters in English might contain the word “Dear” followed by the name of the recipient.

A number of students confused the cracking of cryptographic ciphers with the cracking of passwords, and therefore gave answers involving *dictionary attacks*. There is a vague connection in that dictionary attacks could possibly be used to crack the key used in an encryption under the assumption that the key itself can be set by a user, and that the user then chooses a work unwisely. However, this is a very big assumption, and it only helps if you already know which encryption method has been used. The problem text is specific that it is about you knowing the language that

4 Several answers assumed that substitutional methods only substitute letters with other letters. This is not necessarily true. You could use any symbol set in substitutions. A number of answers even confused substitutional methods with alphabet shift methods such as the Caesar Cipher. That is just one very simple example of substitution.

the encrypted message was in, not the language of the key. What is more, dictionary attacks are not based specific language dictionaries so much as lists of likely words (including the kind that you would not find in any dictionary, such as 'London' or 'alan99') and as such are largely language independent. 'Dictionary attacks' is therefore a very wrong answer that signifies that the examinee we very confused at the exam, and mixed the subject matter from separate parts of the course and subject matter. I only mention it here because there were several such answers, and students should understand that their answers are based on quite a serious misunderstanding of the course subject matter.

Problem 3

> Suggest and describe two alternative methods that could be used to authenticate that a server that I access on the network is not an imposter. Discuss your methods' respective strengths and weaknesses.

There was naturally a wide spectrum of possible answers to this question, and as usual good marks were given for well structured and well motivated discussions. The problem text includes the term 'alternative' which in such a problem is assumed to be a relative term, i.e. some pairs of answers are more alternative than others, so the more the two answers are different from each other, the better.

One way to structure this answer is by reference to the four basic classes of authentication classes that we have studied during the course. Of course, naming these classes is not a useful part of an answer unless it is used to reason about the methods described.

The following is a selection of the kinds of answers given and comments on these answers.

Something the entity knows

A number of answers suggested that passwords would be a possible way to the server to authenticate itself. This is possible, but not a very practically feasible method. It would require that a server 'knew' a distinct password to each user that wanted to connect to it. It could not share a single password as that would mean that anyone who knew that password could masquerade as the server. So the amount of work involved to create, share and transfer specific passwords for each possible user makes this a very cumbersome method. Only answers that described the inherent awkwardness of this method would gain good marks. Since most who suggested passwords did not enter into any such discussions I was more inclined to suspect that this answer came from a misunderstanding either of the question, or else of how passwords work. It is of course far more common for users to authenticate themselves to a server with a password than vice versa.

There were a few answers that instead required the user to first id themselves so that the server could return individual info that could be recognised by the user. This suffers from some of the same problems as passwords though. There would have to be some phase when the user provides the server with the personal knowledge required, so how is the server trusted in the first instance when that information is provided. There is also a question of how the information should be sent so as to avoid it being eavesdropped and later replayed in a masquerade attack.

Something the entity has

Certificates and private keys - If the server has a private key and can keep that key well protected, then it could prove that it has possession of that key through cryptographic means. The user could encrypt a short, unique message with the public key that corresponds to the server's private key, and send that encrypted message to the server. The server is the only holder of the private key that can decrypt that message, so if the server can return the message unencrypted then that is proof of possession of the key. The SSL protocol uses similar mechanisms for authentication, and also to secure the session traffic.

The public key used to encrypt the message must be trusted as being the proper counterpart to the server's private key. This is normally achieved through a Public Key Infrastructure. The public key is part of a certificate that is signed by another trusted party. Trust in that signature may in turn be ascertained by verification through another party's public key. The intricacies of possible PKI

schemes is outside of the scope of this question.

One of the main weaknesses of this scheme is in the trust of the certificates that contain public keys. It may be possible to fool a user into accepting a false certificate, in which case masquerading and MITM attacks will be possible.

Answers on the private key theme were varying in depth and detail. Writing that the server could send a certificate to authenticate itself was not sufficient to show an understanding of the mechanism of authentication. Some answers confused the idea of having a certificate and having a private key. A private key can be kept within a certificate, though when it comes to private keys the distinction between the two is minor. A certificate just contains some useful information about that key, such as the algorithm used to create it and its period of validity. The distinction is far more important where it comes to public keys since a certificate also embodies the means to trust the public key that it contains. The certificate is a cryptographically signed data structure.

There are other methods that can be used to prove possession of the private key apart from the one described above. Some of the schemes as described in exam answers were weaker and vulnerable to replay attacks.

Something the entity is

This class of authentication methods is the most difficult to apply to a server. It seems more relevant to apply this class to people than to machines. Some exam answers did go into the subject of biometrics, but I could only assume that these came from misunderstanding the question.

Perhaps the closest we can come to an identifying attribute is the machine's ethernet MAC address, or rather the MAC address for the machine's network interface hardware. This address should be unique, and therefore could be seen as an identifier. However, the address is not made unique for the purpose of identification, but to avoid addressing conflicts in networks. This can explain why it is simple to falsify mac address numbers. Since such addresses can be falsified, they are not very trustworthy at all.

Where the entity is

IP address could be viewed as an indication of where a machine is, or at least which network it belongs to. Different number series are assigned to different countries and different organisations. Falsification of such number series is possible, but not entirely trivial. For two way communication to work with a false number series an attacker would have to assume some amount of control of network routers and gateways. We can therefore have some trust in IP addresses, though by no means complete trust.

A few exam answers suggested that one could test where a machine is on the network by executing a *traceroute* to the desired server. Traceroute reports back to the requester how traffic to a given machine is routed through the network, i.e. through which machines it passes. This is not a method discussed during the course, but I assume that anyone who has studied computer networks would be familiar with this service. This might not be a kind of authentication that one could expect the average user to employ. Note that an attacker who has control of a router could falsify the traceroute.

Other answers

A *domain name* could be an identifying attribute, but as we have seen during the first assignment, one that is easy to falsify. Some students proposed that with DNSSEC we can be assured that domain names are correct and not falsifiable. DNSSEC is a scheme that utilises digital signatures so that servers verify themselves to DNS servers, and so that information from such servers can be trusted. This is another subject that has not been covered during the course (beyond possibly a mention) but that is unarguably a good suggestion. The current downside of this scheme is that it must be deployed in all relevant nodes and DNS servers before it can be used. Though Sweden is relatively well advanced in the deployment of DNSSEC, the scheme is by no means in common use as yet.

Some answers mentioned the possibility for authentication through a trusted third party. This could presumably be by means of PKI as described above, or schemes like Kerberos. Most of these answers were very vague on exactly what third party trust implied, and were not judged good evidence of understanding.

A number of answers cited challenge-response as a good method to authenticate the server. Discussions on how this worked varied in quality. Several seemed to miss the point that challenge-response does not necessarily authenticate both parties. I have a key-calculator from my bank. They send a key to me, that I transform with my calculator as proof that I possess that specific calculator. I am only authenticating myself to the bank. If an imposter posing as my bank sends me a false code I will enter it into my calculator and return the result to the imposter. The imposter just has to accept that code. At no point in this am I guaranteed that I would discover that I am communicating with an imposter. If I were to enter a different number from the one given and return the calculated code, and if the receiver accepts that code then I could understand that it is not the real bank. No student answers went so far as to suggest this action though.

Some students gave more than two alternative methods. Some students believe that more answers must be better. This is not the case; in fact it is the opposite. If the question asks for two answers and you give three than that must be taken as an indication that you are not able to distinguish which of the two are the best answers. You are therefore leaving it up to the examiner to decide for you, and in that situation the only objective alternative is to choose the worst two answers of the three. Giving more answers than asked for will therefore very often mean lower, rather than higher, marks.

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. Structure each of your answers with headings *description*, *related concept*, and *example.*
>
> - The *append* privilege
> - DMZ
> - The Principle of Separation of Duty
> - The Common Criteria

The append privilege

Description

The append privilege is one that allows a subject to write to an object, but only to the end of its contents. It does not allow the subject to alter or delete any of the existing contents. It is one of the kinds of privileges that define the access rights subjects can have over objects. We are used to seeing such privileges listed in access control matrices.

Related concept

The *write privilege* will normally allow the subject to put anything at any position in the object, and even remove existing content even to the point of deleting the object. The write privilege can therefore be viewed as more powerful and even more potentially destructive than append.

Example

Accounting systems can usefully use the append privilege. When logging on to a system the login process will early on in the process use an append privilege to write an entry to a log file over all logins. Even if the login can be subverted later on it its process, assuming that no privilege escalation is possible, the fact that a person logged on cannot be altered and may be useful

information to discover an attack.

Several answers stated that append is a privilege in Unix systems. This is not the case. Several thought that the game high-score list that we used as an example in a lecture was a good example of the use of append. However, even during the same lecture we discussed that append would not stop you from cheating by writing a false high-score to the file. It just means that the previous values could not be changed.

DMZ

Description

DMZ stands for Demilitarized Zone, and in an IT context refers to an area of an organisations network that separates the local network from the external network, and includes protective mechanisms. The DMZ (which Alan has claimed is a bad name for the concept, but the misnomer has unfortunately stuck) normally comprises an area between and including an internal and an external firewall. Between them can be a number of servers, such as web servers and mail servers that necessarily should be given connection to the network outside of the organisation's. They are thus protected from the outside network by the outer firewall, while the organisations local network is protected from possible security breaches of the security of the DMZ servers by the inner firewall. Both firewalls are often hosted by a single computer in order to simplify the administration and running of the DMZ.

Related concept

Proxy servers are intermediary servers that relay requests from clients to other servers. Proxy servers can also act as a kind of firewall that mediates traffic to and from those servers. A DMZ typically protects a proxy server on the outer perimeter of an organisation's network.

Example

If an organisation has a website that is important for its operation and also needs email connection with the outside network, the outer firewall of a DMZ could ensure that only http and smtp requests are allowed to pass into the DMZ. The web server within the DMZ might need to access database information from within the organisation, while users within the organisation may access their incoming emails that are on the email server in the DMZ. In this case the inner firewall would only permit database accesses from the web server and IMAP and POP session instigated from within the organisation's network to the email server. It could disallow other kinds of traffic.

The Principle of Separation of Duty

Description

The principle of *separation of duty* says that if more than one operation is required to complete a task then those operations should be executed by more than one person. This assures the integrity of the task in that it would take two parties in collusion to improperly complete the task.

Related concept

Saltzer and Schroeder's design principles for security systems include the *principle of separation of privilege* which states that a privilege should not be granted on a single condition. This is based on a similar idea as the separation of duty, in that it provides a similar kind of defence in depth. Whereas separation of duty requires that two people execute separated operations within a task, separation of privilege can involve a single person but several factors. I must present both my bank card and my PIN code to and ATM machine to gain access to my account. That is separation of privilege, but not separation of duty.

Example

In the economy department of a company many invoices are paid. The task of paying an invoice can be divided into two operations: check the veracity of the invoice, and pay the invoice. If one person checks that the invoice is correct and another pays the invoice, then the system is protected from any one of these people presenting the company with a false invoice that gives their own bank account as the receiving account. If the invoice checker tries to fool the company then the invoice payer will notice that the receiving account is not that of a legitimate company. It the invoice payer tries to cheat the company then the invoice checker will notice that the invoice itself is false.

The Common Criteria

Description

The Common Criteria (CC) is an internationally standardised scheme to provide assurance in security software. Its two main sections define standardised means of describing functional requirements and assurance requirements. A specific product's security requirements are described in a standardised Security Target document. Generalised security requirements for a type of product can be described in a Protection Profile. A product can be certified as fulfilling its Security Target. There are different levels of assurance requirements which means that when a product is certified it is certified as meeting one of the CCs seven Evaluation Assurance Levels (EALs), where EAL1 provides the least assurance, EAL7 the greatest.

Related concept

TCSEC, also known as *The Orange Book*, was an earlier standard designed by the USA Department of Defence. This standard was designed primarily to measure the assurance of the security mechanisms of operating systems. In contrast, the CC is more generally applicable to all kinds of software systems. TCSEC is no longer in current usage and has been superseded by the CC.

Example

A government organisation may have a need for firewall software. They therefore create a Protection Profile (PP) where they describe their requirements according to the CC documentation standard. A company intends to compete for orders for firewalls from this government organisation and therefore designs a firewall that fulfils the requirements in the PP. They describe the firewall's security related aspects in a Security Target, and a CC certified third party organisation verifies that the Security Target and the product that it describes meet a certain EAL.

Problem 5

In practice an individual's need for privacy can be

- Culturally dependent
- Situation/Scope dependent
- Time dependent

Describe example situations that clearly illustrate how the need for privacy can vary according to these factors.

The 'clearly' of the problem text is an important aspect. One should consider in what ways this can be made clear since clarity in answers will be a deciding factor for the grade.

Privacy needs may be individual, but society nevertheless makes general assumptions about them, in laws, in how far it applies surveillance methods, etc. If we can find situations that highlight how the assumptions made in society can have drastically different effects dependent on the given factors, that should make it clear.

For extra clarity, answers can also relate to both the kinds of privacy that we have defined in the

course, i.e. *spacial privacy* and *informational self determination.*

Cases closer to reality are more convincing than supposition based on fictitious scenarios.

Time is unfortunately too short for me to provide a proper discussion in this version of this document. As a guideline to what kinds of answers where judged as good I have summarised below examples that illustrate differences in both types of privacy for each of the three factors. Note that such brief notations as sometimes occur below are not sufficient for your exam answers, but here they will hopefully give you clues to what examples your discussion could be based upon to achieve good marks.

Culturally dependent,

- spacial privacy
  - Nakedness and public displays of affection will offend people in some cultures where you would expect laws and law enforcement to protect you from having to experience them. In other cultures e.g. nudity may be the norm.
- informational self determination
  - in some cultures a picture is regarded as containing the essence of one's soul and therefore not something that anyone should take and have control over. For others an id photo (as opposed to photographs of me in any number of other situations) is a common and undisturbing means of proving my identity.
    Several answers quoted this same situation as an example of violation of spacial privacy, but it seems to me that it must be the picture that is the problem more than the taking of the picture...?
  - You wages in Sweden is part of public record whereas in other countries it is considered secret information
  - Asking someone's age in USA or England is commonly considered an invasion of privacy, but not so in Sweden or China

Situation/Scope dependent

- spacial privacy
  - Imagine that you are in a field, listening to birdsong. You may well consider it a public nuisance if someone came walking by playing loud music from a portable stereo. The very same field might be used for an outdoor rock concert where I would expect to hear loud music.
  - In an avant guard theatre production I might expect actors to accost members of the audience, and even purposefully embarrass them. If I were similarly accosted by the same person while minding my own business in the street I would see it as an affront to may personal privacy.
  - Calling a person for a chat at any time vs when that person is bereaved.
  - Calling your systems administrator at home when there is a crisis situation at work, counter calling them because you cannot get a game to install properly on your home computer. (This could also be situation dependency of self determination if we are talking about the use of the telephone number)
  - To illustrate the aspect of scope: If I am disturbed by someone talking loudly in their mobile phone, the problem is not so great if I feel that I can move a few steps away to avoid the nuisance. If there are people talking loudly in mobile phones wherever I go then the problem is serious.

- informational self determination.
    - Helping to solve a crime, you may offer up all sorts of information on where you were, what you saw etc. Were a person to ask the same kind of information without a given purpose, it could even be harassment.
    - I may be want to let people know my political affiliation if I am campaigning for office, but in the voting booth your political affiliation is nobody's business but your own.
    - Data that would clearly not be sensitive for most of us, such as our name and address, could be very sensitive for someone who has moved to escape from domestic violence.
    - In a confessional...
    - Police looking up data on you for a) an investigation, b) out of curiosity.

Time dependent

For extra clarity in this factor we can try to specify different kinds of time dependency. Privacy could depend on what time it is we are living in, the lifetime of information, the time of day, or the duration of exposure.

- spacial privacy
    - 40 years ago having television cameras follow you around would have been unthinkable. Today people seek positions in reality programs such as Big Brother.
    - 4am in the morning phone calls from a sales person might be illegal, whereas at 4pm they might at worst be a nuisance
    - I can stand a smoker for three minutes, but not three months.
- informational self determination,
    - The last 3 numbers on the reverse side of your credit card used to be regarded as less important, whereas today we know that it is safer to scratch them off for my privacy. (not a good example though – more about security than privacy).
    - before an auction ends my bids are private, afterwards a part of the verification of the correctness of the auctioning process
    - inventions up to the time the patent is granted.
    - Your boss asks what are you doing during work time vs. during your free time.
    - For the duration of a job application an agent may spread your information, but not afterwards.

In marking I try first try to understand the description, and first ascertain as to whether it pertains to privacy or confuses it with other concepts. Then the answer is assessed according to whether it actually addresses the varying need for privacy according to the given factors. Then the clarity of the answer is judged. Good arguments will therefore show clear distinctions between the influence of the factors, and make universally applicable arguments. If the examiner can find holes and interpretations that would make the answer less than clear then the answer will be awarded less marks.

Several answers mentioned the concepts of *spacial privacy* and *informational self determination*, but without applying it to their answers. I found that strange and a little confusing.

References

Bishop05 Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

Davies&Price89
Davies, D.W. & Price, W.L., "*Security for Computer Networks*", Wiley, 1989

Schneier03 Schneier, Bruce. *Beyond Fear: Thinking Sensibly about Security in an Uncertain World.* Copernicus Books. 2003

DSV SU/KTH *IV2018(part 2)/Säk1b/IV2024*
Information and Computer Security
Autumn Term 2009

Alan Davidson

Very Brief Suggested Solutions to the Exam 2009-12-07 and Comments on the Marking

The answers suggested here may be briefer than would normally be expected from students. I have tried to summarise the most important aspects of problems so that students may compare the content their own answers to these. Lack of time prevents me from doing more.

Problem 1

> Explain the purpose of a security policy from several different perspectives, for example, with a formal definition and a pragmatic viewpoint.

Bishops definition is:

"A *security policy* is a statement that partitions the states of the system into a set of *authorized*, or *secure*, states and a set of *unauthorized*, or *nonsecure*, states." [Bishop05, p45].

Though students may not be expected to memorize such definitions by heart for the exam, close approximations would be good as a answer to the formal part.

Some security policies will be directed towards the members of an organisation, specifying what the are allowed and not allowed to do if the organisation is to remain secure. It will therefore be written in a high level language that should be easy for people to understand and apply. There will also be measures to support this policy, such as regular education about it, and even punishment systems for not adhering to it. The purpose is to ensure that members of an organisation cooperate in upholding a secure organisation.

Some security policies will be specified in a lower level, formal language. The purpose of these is to carefully specify details of how to uphold security in order that these details could be effectively applied to a mechanism that supports the policy. One such example might be a policy that specifies what Internet traffic should be allowed in a language that is easily translated to the configuration parameters of a firewall.

Problem 2

> A student in higher education might expect to be given space on a file server where files for e.g. working with hand-in assignments can be kept. Suggest and motivate reasons why such a file-server should apply on the one hand discretionary access control, or on the other hand mandatory access control.

With the example of files for a hand-in assignment, some of these might be group assignments. In order to facilitate working on the assignment members of the same group might need to share files for that assignment. With discretionary access control the file owners (in this example the students) would themselves be able to set file privileges that would allow the other students within the group to access and even to update the files. Giving the responsibility to the student is practical, especially where students form project groups themselves. As they form groups themselves they can also alter the privileges, and there is no need for excessive administrative routines such as involve any systems administrator in order to set the privileges needed.

The above example make the assumption that one can expect students to be both honest and competent. Otherwise file privileges might be either purposefully or accidentally set to allow other students than are in that assignment group to read the group's working files. If all student groups are expected to work on the same problem in their respective groups, there may be students who are

willing to break the rules and copy others' work rather than do their own. If this is a major problem then a mandatory access control system might suit better, where examiners or system administrators define rules that set the privileges of files so that only authorised group members can access them, and no others.

Many student answers to this question discussed the relative security of the two schemes without discussing the relative administrative overhead. In such a discussion, surely mandatory access control would win out every time(?).

A number of students apparently have difficulties in distinguishing between the mechanisms for changing rights (e.g. MAC & DAC) and the actual rights on objects. Marks could not be given for answers that show such confusion.

I was a little surprised to find that many students seemed to have some difficulty in understanding the concept of a file server, instead assuming all sorts of other strange functionality in their answers. This occasionally had the effect of making the arguments all the more vague.

Problem 3

> Among the many fields contained in a certificate there is an ID. Explain why this ID field is necessary, and what kinds of requirements should be put on the ID value. Give examples of suitable ID values, and of how the associated certificate would be used.

In computer security the term *certificate* is primarily understood as referring to *public key certificates*. Private keys can also be kept in a certificate structure for the sake of convenience, but in terms of public key infrastructures the most important use of certificates is for public keys.

In order to use a public key we need to have trust that the corresponding private key is owned and exclusively available to the entity with which we wish to communicate. There is no way to ascertain who the originator of a public key is from simply examining it. We need some way to ascertain who the originator is. The first step is therefore to find a way to uniquely reference the originator, so that we can never become confused about who has the corresponding private key. The name Alan is by no means unique, nor is Alan Davidson, so these would be unsuitable values. We can be fairly sure that alan@dsv.su.se will only ever refer to a single individual, so that may be a more useful identifying name. X500 Distinguished Names are an alternative scheme in order to be able to uniquely reference entities on a global scale.

Binding unique originator IDs to public keys is the method by which we can know which key to use when communicating with that originator. It is clearly important that the name be bound to the public key in a trustworthy manner. This will typically be achieved by a trusted third party digitally signing the data structure that contains the public key, the ID attribute, and other useful data, to make a certificate.

Once the trusted digital signature has been verified, we could use the certificate's public key to encrypt a symmetric key that has in turn been used to encrypt a message. We can then be safe in the knowledge that only the holder of the private key, i.e. the party that the ID attribute identifies, can use that private key to decrypt the session key, and thereby decrypt the message.

We can also use the public key from that certificate to verify digital signatures that have been made with the corresponding private key. The digital signature contains the ID of the signer that is then used as an index to find the corresponding certificate. By decrypting the signature with the certificate's public key the signature is verified.

I had the impression that several students had the impression that certificates are things that sometimes are passed to you when surfing on specific sites on the Web, and their answers to this problem were influenced by this view. A minimum requirement to show that you understand the issues of this question is to relate it to asymmetric cryptosystems.

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.*

- Military security policy
- Dictionary Attack
- Challenge response
- Computer worm

Military security policy

description

A military security policy is a policy that is primarily concerned with upholding the confidentiality of the system that it is defined for. It is not a pure confidentiality policy though, since some integrity aspects will normally be included.

relationship to commercial security policy

A commercial security policy in contrast to a military is primarily concerned with upholding the integrity aspects of the system it is designed for. Whereas the military policy is mostly confidentiality and some integrity, the commercial policy will be mostly integrity with some confidentiality.

example

In terms of information, a military organisation that is intending to send a guided missile to its enemies will presumably need to spend large amount of effort in ensuring that the location of the existence of the missile base and their plans are kept secret from the enemy. The assets of the organisation are most vulnerable when the enemy can discover such information and thereafter plan defence, counter or pre-emptive measures. Confidentiality would therefore seem to be of great importance. However, certain integrity issues are also of relevance, such as the integrity of the data that guides the missile. If the enemy could manipulate that data, plans and assets of the organisation could be endangered.

Dictionary Attack

description

A dictionary attack is a method primarily used for discovering passwords whereby lists of the most likely combinations of characters in that password are first tested by trial and error. More likely combinations come about in password creation methods that are less than random, most notably due to human tendencies to choose easily remembered passwords such as those found in a dictionary, or simple transformations thereof, such as by adding numbers to the end of names. Note that the “dictionary” in a dictionary attack (i.e. the database of likely words and transformations that are tested) should not be assumed to be an actual digital dictionary of real words. The dictionary in the attack is instead compiled specifically for the purpose of testing the most likely passwords first. We would therefore expect passwords that are not included in normal dictionaries to occur in this attack dictionary (e.g. “querty” or variations on the username). Dictionaries should not be assumed to be static either, but could even be specifically constructed for specific attacks.

Dictionary attack tools are useful not only for crackers, but also for systems administrators who might need to systematically check the strength of the system's user's passwords.

relationship to brute force attack

A brute force attack also tries to discover passwords through testing systematic combinations of possible characters. As opposed to a dictionary attack, a brute force attack will not make assumptions about one guess being more likely than another. It may well instead first test the empty password, followed by all single characters, followed by all combinations of two characters, and so on. Tools for dictionary attacks will often revert to brute force attack strategies once all the likely possibilities that it knows about have been exhausted.

example

If a cracker were to manage to gain access to a system she might well attempt to retrieve a copy of that system's password hash list (the one used in the authentication process). Knowing what algorithm has been used in creating those hashes, the cracker could use a dictionary attack to guess likely passwords, hash each guess, and check whether each generated hash matches any in the password hash file. Whenever a match is found, the cracker will have discovered the password behind the original hash, and will have access to yet another account, or in a worst case scenario, gain access to administrator accounts. On systems with many users, the chances that any of the users will have chosen a relatively easily guessed password will increase.

Examples of passwords that a dictionary attack could be expected to easily discover: guest, password, secret, Stockholm, qwerty, alan69, alanalan, nala, m0th3r...

Challenge-response

description

Challenge-response is a type of protocol that can be used in an authentication process. The secure system asks a question, or requests some action of a subject seeking authentication, where it is assumed that only the subject would know the correct answer or be able to complete that action. In a strong challenge-response authentication the challenge would be one that an imposter would not be able to guess beforehand. If a challenge-response method can provide a unique challenge for each authentication attempt then the method will be immune to replay attacks.

relationship to password based authentication

Password authentication could be seen as challenge-response in its simplest form (I note that Wikipedia includes it as an example [Wikipedia09]) since the authenticating server first prompts for a password, whereupon a password is the response. Note however that the password challenge is entirely predictable, and an attacker who manages to discover the password will know that it can be used at any time for the same authentication process. It is therefore questionable whether password based authentication is a meaningful example of challenge-response.

example

When conducting transactions with my Internet bank the bank will present me with a series of digits which I am then expected to enter into a so called *security token*, i.e. a small calculator-like device that they have provided for my use. When the digits are entered into the security token it applies a mathematical function to those digits and shows a resulting different sequence of digits. The function applied is unique for the security token, so when I respond to the bank by providing the resulting digits, that is taken as proof that I am the holder of that particular security token.

Computer worm

description

According to Bishop's definition:

“A computer worm is a program that copies itself from one computer to another” [Bishop05, p373]

We may assume that this definition (in common with other definitions from the same chapter on malware) assumes an element of ill intent. A worm is commonly programmed with the ability to exploit security vulnerabilities in order to spread itself from one computer to another.

relationship to computer virus

Some definitions of the term computer virus will cover many kinds of malware, However, the more precise definitions will usually establish that a computer virus attaches itself to program hosts. In that case we can say that while viruses have programs as hosts, worms will not attach themselves to a program but to the system itself. Since worms have the ability to spread themselves from system to system, without relying on a a host program to first be activated, they will commonly spread much faster than the classic computer virus.

example

The so called Morris Worm infected the Internet while it was still young in 1988. It exploited known vulnerabilities of Unix systems, including a very rudimentary dictionary attack. Though it was not written to be harmful, accidental side effects caused it to become an effective denial of service attack for a significant portion of the Internet of the day, and an estimated 10% of Internet connected machines were infected.

Problem 5

> Suggest and describe schemes by which a software manufacturer might establish potential customers' trust in the security of their products.

The relevant scheme that we have primarily studied during the course is the Common Criteria. Please see other sources such as wikipedia or the course book for descriptions.

There are other schemes that could be discussed, such as SSE-CMM, but other schemes were not included in the course or in the reading notes, so they were not expected as answers.

There might be techniques, methods, tricks, or vague pieces of advice that can be applied to instil trust but unless they have in some way been standardised they can hardly be regarded as *schemes*. Since answers were in this sense dependent on the student's understanding of the term *scheme*, I did stretch the concept somewhat, given that not all students will have sufficient command of the English language. Nevertheless, answers would be expected to show an understanding of the Common Criteria to gain a pass.

References

Bishop05	Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.
Wikipedia09	Wikipedia article *Challenge-Response*, http://en.wikipedia.org/wiki/Challenge_response, visited 2010-01-07.

DSV SU

SEC:I/IV2024
Information and Computer Security
Autumn Term 2010

Alan Davidson

Suggested Solutions to the Exam 2010-10-22 and Comments on the Marking (version 1.0)

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

> One practical security measure might be to encrypt a computer's hard drive so that only after correct authentication would the content be decrypted. Based and structured on the CIA triad, discuss and explain how this measure can be said to effect system security, both positively and negatively.

Confidentiality.

Even if someone should manage to bypass the authentication mechanisms that protect the computer's hard drive (such as by removing the hard-drive and inspecting it under another system) the data will not readable. The effect on confidentiality can therefore be said to be positive.

Integrity

The integrity of the data on the hard drive could be said to be lessened since all the data is transformed and will only be retrievable in its correct form if the correct decryption process is applied. Care should also be exercised in choice of the encryption algorithm used. It would be unfortunate if methods such as Cipher Block Chaining were used, which might mean that if an error occurred in the data the error would propagate in the decryption process, thus making more data than just the original error unretrievable.

On the other hand the encryption/decryption process can be seen as an integrity check. If anything were to be changed on the encrypted hard drive, the decryption mechanism would most likely make the error all the more detectable. It is possible that the decryption mechanisms used could signal the error. That is not as good as being able to reconstruct the lost data, but it is better than nothing.

As with confidentiality, there is an extra mechanisms to ensure that authentication cannot be bypassed. Only authorised users can change the unencrypted data, so assuming that those users are trusted, only trusted users can change the data. This is therefore also a measure to ensure that untrusted users are not able to manipulate the unencrypted data.

Availability

The availability of the data on an encrypted disk will be dependent on some kind of cryptographic key. For confidentiality the key must be kept separate from the disk until such time as the decryption is called for. The availability is now dependent on two factors, the disk and the key, instead of just one in the case of an unencrypted disk. If either one of the disk and key pair are lost, then the availability will be lost.

There will be a process of constant encryption and decryption of data to the and from the disk. This will take computing resources that could otherwise be used for other processing. This means that it will take some extra resource, such as time, to use an encrypted hard drive as compared to an unencrypted drive. Availability is therefore adversely affected.

One novel answer suggested that the encryption gave the possibility to safely carry the hard drive

around in the knowledge that if it were lost or stolen the data would still be safe. This portability factor could be seen as improving availability.

A few strange answers seemed to suggest that the encrypted disk had lesser availability than an unencrypted disk because unauthenticated users could (presumably illicitly) have access to data in the unencrypted disk. Availability is of course only about ensuring access to those who are authorised, not everybody.

Problem 2

> Suggest diverse possible ways in which a computer might be authenticated within a network, and discuss your methods' relative strengths and weaknesses. Good answers will cover a breadth of possible methods.

Categories of authentication methods are presented during the course, and these can by all means be used to present an answer to this question that has the breadth that the question asks for. However, it becomes clear that these categories are more easily applied to humans than to computers. Whether one categorises, for example, an IP address as something a computer knows, has, is, or where it is is of less importance. I will use those headings to give the following examples some structure, though they are by no means assumed to be necessary for good answers.

Something the entity knows (Though of course whether a computer really know anything is a matter for interpretation)

An entity can hold onto a secret data token – one form of which we often call a password – that is shared with the system that the entity needs to be authenticated with. The authenticating entity presents the secret token to the system, or else (for added confidentiality) shows that the entity 'knows' the secret without actually presenting it, with the aid of particular protocols designed for that purpose. Under the assumption that no other entities know that same secret then that entity can be uniquely authenticated.

Secret data token mechanisms can be relatively cheap to implement. Since computers do not have the same kinds of problems in remembering tokens as humans do in remembering passwords, many of the problems that humans have in the choice of passwords need not be a problem here. The method is nevertheless dependent on the secrecy of the token, which might in practice be difficult to achieve and maintain. To use this method to authenticate a single computer in a network is not as simple as its common use in authenticating users to single computers. If the computer is to be able to authenticate itself to every node in that network then it must presumably share individual secrets with each of those nodes; a situation that would surely require considerable effort to install and manage.

A number of answers suggested that passwords would be one method for authentication of a computer in a network. A majority of such answers were confused, and seemed to lose track of whether it was a computer or a user that was the subject of authentication. Without clarity on this point answers were not given high marks.

Something the entity has

MAC address. Each network interface that a computer has is given a supposedly unique MAC address number. Presenting that number on request is a possible method of authentication. MAC addresses are readily available on all networked computers. However, the uniqueness of such numbers can be questioned since the routines that are used to ensue that duplicates do not occur are not entirely trustworthy. What is more, MAC addresses can easily be simulated in software, so they can be impersonated by malicious parties that have administrator privileges.

Cryptographic keys. If the computer holds its own private key from an asymmetric key pair, and the remaining nodes in the network have access to the corresponding public key, then there are simple protocols by which the computer can show that it is the holder of that private key. For example, a network node could send a once off, original string as a authentication challenge to the computer. The computer then creates a digital signature on that sting and sends it back. If the

digital signature can be verified with the public key then the computer is authenticated as the holder of that private key. This is a strong and simple authentication method, but it is dependent on

1. The private key must be kept secure. This can be difficult in practice.
2. Some method of instilling trust in the authenticity of the public key is necessary, such as Public Key Infrastrucutres. This can be complex to implement and run in practice.

Something the entity is (or does)

We normally associate this category with the field of biometrics, and it is therefore questionable if this category should be applied to inorganic computers at all. In some situations we might be able to authenticate with the essence of a computer:

Chip manufacturers can create tamper-proof versions of individualised chips. In principle they implement the same idea as the private key solution above, but the private key is built into hardware in a manner that does not allow it to be directly accessed.

Inasmuch as computers have their own individuality, it may be possible to authenticate them from collections of their individual characteristics. What operating system is being run, what network services, normal response times, etc might all be factors that together could give an indication that we recognise a certain computer. This seems to be a possible impromptu method, i.e. that I might be able to run such checks on a machine that I was suspicious of, but as a primary method of authentication it does not seem to trustworthy since machine profiles will be subject to change and impersonation as soon as it is known that the method is being used.

Where the entity is

To some degree we can view the IP address of a computer to be a representation of where it is in a network. For example, local networks will often have their own IP address space, which means that if all local routers, switches and bridges are functioning reliably, we can view a machine with local IP address as being authenticated as coming from that local network. This is a simple and ubiquitous method, but since network equipment may not be trustworthy, and since we know that IP addresses can be spoofed, one can understand that it has not proved to be a strong method.

Another way to test where a computer is in a network is to check the route that data packets take through the network on their way to the given computer, and thereby authenticate them a being in 'the right place'. The program *traceroute* is one example of a method to do this. This is another convenient impromptu method that requires some technical understanding to run the check, and it is not foolproof as the results returned by the routing network to the requesting program could be spoofed.

Some answers to this problem made statements of the kind that certificates or challenge response protocols can be used for authentication. This is indeed true, but many such answers were so vague as to give no idea as to how the authentication is actually achieved. Only answers where it is made clear that the student understands the methods involved could achieve high marks.

A few answers made reference to clients and servers as if there was an obvious relationship between the two in this problem scenario. Since these roles are not significant for the problem I found such answers confusing and difficult to understand. The problem is of one computer that should be authenticated in a network of other machines.

The suggested solutions I have given above are relatively low level. Other solutions, such as discussing Kerberos and its pros and cons, were also quite acceptable.

Problem 3

> One of the important rights that you might expect to find represented in an access control matrix is *own*. What this right actually entails can depend on what type of access control is applied. Explain and exemplify how the *own* right will have different practical implications depending on whether DAC, MAC, or ORCON is applied.

In DAC it is at the owner's discretion to set the rights of the object to whatever they deem appropriate. E.g. if I was the owner of my own diary object, DAC would allow me to choose what kind of access other should have to it. I could presumably keep access to it from my mother but choose to allow read access to my best friend.

In MAC there are system defined rules that stipulate and enforce what rights an owner of an object may set for other subjects in the systems (and themselves for that matter). An accountant within a company may own files that have details of what upper management earn, but that does not mean to say that the accountant can choose to allow for other workers in the company to read or to set such figures.

In ORCON the rights that an owner may set and for an object can be decided by the originator of that object. E.g. A recording company is the originator of a music CD. I pay money for a copy of this CD and become its owner. I may choose to let a friend listen to the CD, but the originator does not give me the right to allow that friend to make a copy from my copy. Indeed, there may be technical support methods that practically hinder me from doing so.

A number of answers only discussed the right to transfer ownership and how that is affected by the different policies. The other kinds of rights are of course relevant too.

There was some confusion over whether owner means (or usually means) creator. The very idea of ORCON surely implies that they are not necessarily the same (if we assume the originator to be the creator). Otherwise, I do not think the distinction is especially relevant for a discussion on MAC or DAC. I suspect that very many of the files that I am the owner of on my computers have been copied from other sources so that I would not regard myself as the creator of, say, a recorded television program or of photographs that my family have passed on to me.

There was occasionally some confusion over the problem's connection to the rights of the special role of administrator. As Bishop points out [Bishop05, p241] the *root* or *administrator* role in popular operating systems is only relates to ACLs in a limited manner, i.e. these are pragmatic constructions to allow for the proper setup and running of the systems themselves. As such we normally view them as separated from such principles and mechanisms as those discussed in this problem

Given the above description of MAC one might be tempted to call the system the owner of the object, in which case there does not seem to be much difference between DAC and MAC, just whether the owner of the object is the system or the subjects within the system. I suggest that this is an erroneous view that leads to confusion. One point that should dispel this misconception was given in one exceptionally enlightened exam answer: Who the owner of an object is can effect what MAC rules should be applied. For example, if a military general owns an object there may be system-wide specific MAC rules that allow that general to lower the object's security classification to allow all to read it. The same rule does not apply to object owners who are military personnel of a lower rank.

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:*

- Certification Authority
- Kerberos
- SQL Injection
- Evaluation Assurance Levels (EALs)

Certification Authority

Description

Certification Authorities (CAs) are normally associated with Public Key Infrastructure schemes. Such authorities are trusted parties that issue digital certificates. Certificates bind a public key to an identity and thereby provide a means for trust in a public key. Trust in a public key implies trust that the corresponding private key is in the ownership of the individual that is identified in the certificate, and these are necessary requirements for asymmetric cryptosystems to be practical.

Relationship to PGP certificate signatories

The early versions of PGP used a *web of trust* model to provide similar trust in keys as is achieved by Certification Authorities in Public Key Infrastructures. Certification Authorities have hierarchical relationships to each other and to the entities whose keys they sign. At the top of such hierarchies there will be a Certification Authority whose certificate is not signed by another but is a so called *self-signed certificate*, i.e. signed by the CA itself. In the original PGP scheme, certificates were signed by any party that could verify that the certificate was trustworthy. There would therefore not necessarily be any such top node, by a web of signed certificates. A PGP signatory is therefore similar to a CA in the function of signing a certificate, but different in terms of the trust relationship between signatories.

Example

When Internet browsers are distributed and installed they normally come with top node CA certificates that implies that the CAs that are represented in those certificates are implicitly trusted. Well known examples of CAs whose certificates are included with browsers are Thawte, Verisign and RSA Security.

Kerberos

Description

Kerberos is a system that implements Single Sign-On authentication, i.e. it allows a user to authenticate themselves once and thereby be given access to services throughout a network. Kerberos has similarities to capability based access control systems in the way that once the use is authenticated it distributes limited lifetime tickets to the user that grant access to services.

Relationship to passwords saved in "key rings"

Another method that allows users to authenticate once is the principle of key rings. Here a user's passwords are kept securely encrypted on a local computer. When a key or primary password is provided the system can use that key to access the key ring passwords. The key ring will be integrated with the local system so that that authentication with the individual passwords is automatic from the perspective of the user. Key rings are like Kerberos in that they implement a kind of single sign-on, but different in many ways, such as key ring mechanisms are implemented on a local trusted machine, whereas Kerberos is implemented within a network, and user authentication can be executed from a number of client machines.

Example

DSV and KTH use a Kerberos system that allows students to authenticate themselves once and then have access to file servers, the wiki server, to the video server, etc.

SQL Injection

Description

SQL Injection is the name of a common software vulnerability. It is normally associated with web applications. Such applications have a database back-end and web content is created on the fly using data from that database. Furthermore, what data is used is dependent on the some of the user input at the web client side. If the client side input is used by the web server application to construct an SQL query (without careful input validation), it may be possible for the user to construct the input so that it interferes with the server's construction of SQL queries in such a way as to create new queries that the system designer had not anticipated. Such queries may implement a malicious attack on the web application.

Relationship to script injection,

Whenever user input is used to construct code that is subsequently interpreted, if it is not carefully checked to ensure that the input is of a form that is expected and therefore safe, there is a possibility that a malicious user could utilise such script injection vulnerabilities to create and run code that the system designers did not intend. We can therefore view SQL as only one kind of the more general threat script injection. Script injection also covers similar situations with languages such as php, ruby, sh, etc,

Example

If a server side SQL statement were constructed from the template:

```
'SELECT * FROM foo WHERE in = ' . <user-input> . ';'
```

Then entering the user input

```
1; DROP TABLE users
```

might cause two SQL queries to be executed where only one was originally intended, and the second statement may create considerable damage.

(N.B. This is a good example, but it is not assumed that all students will be able to give such concise examples, especially those who have not studies SQL.)

EALs

Description

EALs are the Evaluation Assurance Levels of the Common Criteria (CC). They are the seven basic levels (1-7) that define ascending levels of trust. When a software system is evaluated it is done with a specific target level in mind. The higher the level the more trust can be invested in the system, but on the other hand the more time and investment must be spent on the evaluation.

Relationship to the evaluation classes of TCSEC

The (now obsolete) TCSEC assurance evaluation scheme also defined a number of levels in its evaluation classes. TCSEC, also known as “The Orange Book”, has CC EALs only 6 levels, though they roughly correspond to the top six CC EALs. The CC EALs are applicable to all kinds of software, whereas TCSEC and its classes are intended for evaluation of operating systems.

Example

Microsoft Vista has been CC evaluated at CC EAL 4+, which means that it has been certified at level 4 but with some additional evaluation factors included.

An extraordinary proportion of answers did not follow the problem instructions with regard to the headings. The headers are intended to on the one hand encourage the answers to be more directed (i.e. the *relationship to [chosen concept]* heading incites examinees to write about a relationship, and not just name a concept) but also to make the discussions more directed and structured so as to make the marking of the answers possible. It there are no headings then the examiner has to make extra effort in interpreting the text to find the required parts. If the heading does not specify the concept that the examinee has chosen, the examiner has to interpret from the text which is the

concept. Though the writer of the exam answer might think that it is easy to understand these things from the context, it very seldom is for the reader. The result is that the answers cannot reasonably be marked within the the limited time allotted for the task. The examiner's only recourse is cut short the reading of such answers, saying that if the parts of the answer are not clear then the examinees marks must suffer, even when a deeper reading might show that the student has indeed understood and covered the subject well. All this goes to emphasise the importance of reading and following exam instructions exactly. It is unnecessary to fail the exam on technicalities.

Choice of the concept and the example is an important part of the answer to these problems, as well as the way they are then described. They should both provide further insight into the given concept. For this exam the choices made were generally substandard.

Problem 5

> There exists legislation (both Swedish laws and European directives) that uphold the privacy rights of individuals. Describe an example of a realistic situation where computer users might either regularly transgress against such legislation, or suffer the consequences of others who transgress against them. Indicate the essence of the legislation that is being transgressed (i.e., even though you may not know the name or designation of a law you should be able to characterise it).

Note that this is not a course where you are expected to study details of legislation, but some relevant laws were discussed during the lecture on privacy, so to gain marks for this problem students should show that they can put such laws into context and show their relationship to current privacy issues. I did allow some degree of deviation from the problem text where good examples were related to other international legislation, but since I cannot check all such answers and since we looked a Swedish and European laws during the course such answers were not awarded as high marks.

Laws and directives that were touched upon during the lecture on privacy were:

EU Data Protection Directive 95/46/EC, including informed consent, purpose specification and purpose binding, Data minimization, no processing of “special categories of data”, transparency, requirement of security mechanisms, and supervision.

EU Directive 2002/58/EC on Privacy and Electronic Communications.

EU Retention Directive 2006/24/EC

The Swedish law of free public access to information (“Offentlighetsprincipen”)

The Swedish law of privacy of personal information (“Personuppgiftslagen (PUL)”)

The Swedish law on electronic communication (“LEK”)

Swedish PUL as well as European directives on data protection make it illegal to publish or export peoples' personal information without their prior consent. Modern social media trends mean that it is easy to forget such laws when publishing information or photographs that include friends. If I were to upload a photograph of a clearly identifiable friend depicted in a situation that clearly is not publicly available, and even tag them as being in that photograph, then even in the process of uploading to, say, Facebook, I am exporting sensitive personal information. Given that web sites and communication channels often cross international borders without our knowledge, it is very easy to accidentally export personal information.

According to Swedish and European law one may not send unsolicited advertising via email. Nevertheless we surely regularly suffer from such spam that is a clear breach of not only laws but also the principle of spacial privacy. The legal issues may not be clear though, since many times it will not be clear what country the spam comes from and therefore what laws apply. Furthermore, the issue of what is solicited seems to have become clouded. I myself regularly receive email advertising from companies I do not recognise where at the end it is stated that I have requested such emails, even though I have no recollection of ever having done so. The Swedish anti-spam law

does in fact come from one that is not included in the lists that Albin gave; it is part of the Marketing Control Act (2004:103, if I understand correctly). I do recall Albin describing American spam laws as being “opt-out” whereas Europeans went with “opt-in”.

I do not recall if Albin spoke of this at the privacy lecture, but a couple of exam answers brought up the issue of cookies. Cookies are collections of data that web sites can request be saved to client's local memory and read on request. Not only that but they are one common method used to track users' surfing habits. Swedish law requires that whenever a Swedish web server makes use of cookies the client must be informed that cookies are being used, what the purpose of the cookies are, and how they can be avoided. These days many Swedish sites do not follow this law. A possible explanation for this is that modern web management tools make it easy for non programmers to create web sites without them understanding that cookies are included in the underlying mechanisms.

A current case in the media is how Google has got themselves into hot water on several occasions by transgressing against local data protection laws. For example, they have operatives that travel around collecting information on among other things the locations of wireless networks. It has lately been revealed that in the process they have saved data traffic that was detected on those wireless networks. This has occurred in many countries, including in Europe. One might question whether such cases meet the requirement set in the problem text of being regular transgression, but it certainly was high profile.

A number of student answers brought up the issue of Copyright. Note however that copyright is not the same thing as privacy rights. Privacy is about protection of individuals, whereas copyright is about protection of property. I marked good discussions on copyright fairly generously even though it could be claimed that they showed a lack of understanding of privacy issues.

Another fairly common mistake was to write of regular transgressions by malware and intrusion attacks. These may be relatively regularly occurring problems, but here to the discussion shows a confusion with privacy issues and other problems. Intrusions are more to do with property rights. If someone broke into your home and stole your belongings I suggest that you would not try to prosecute them for invasion of privacy, and the same applied to computer intrusion.

Some answers were wrote of the privacy principles of *spacial privacy* and i*nformational self determination* as if they were laws, but they are of course not. They are merely one way to classify the subject to assist us in communicating about it.

Reference

Bishop05 Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

DSV SU

SEC:I/IV2024
Information and Computer Security
Autumn Term 2010

Alan Davidson

Suggested Solutions to the Exam 2010-11-27 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).

Please note that sources are quoted and referenced in these suggested answers. This is to provide accurate documentation only. In no way are students ever expected to remember and provide accurate quotes in exam papers for this course.

Problem 1

> Give motivated arguments that updates are both good and bad for system security.

It may be effective to avoid many software vulnerabilities during the design phase, given that many such vulnerabilities are relatively easily predicted. Nevertheless, trying to predict *all* possible environmental situations and threats at the design phase is increasingly difficult as the software or its environment grows in complexity. In such situations the most effective security measure for dealing with newly discovered vulnerabilities can be to develop and distribute updates that provide protection from those vulnerabilities.

Speed will be of the essence in developing, distributing and installing updates. It will not normally be possible for those who are responsible for the security of a system to analyse the content and effects of an update before installing it. This means that systems administrators have little choice but to trust the update and install it. The integrity of every system is therefore at risk with every update. There have been cases where important system updates have caused greater problems than the problems they were intended to solve, before they were retracted.

It is safe to assume that fixes to problems will involve more code that the version with a problem. In that case, the complexity of the software is increased with an update. According to Saltzer and Schroeder's Principle of Economy of Mechanism complexity is best avoided in the design of security mechanisms. In this respect updates that cause added complexity can be said to have an adverse effect on security.

When security updates are published they are often documented in terms of the vulnerabilities that they correct. This documentation could give pointers towards understanding how that vulnerability could be utilised[1].

One answer made the argument that updates fill disks and use CPU resources, which can be seen as a threat against availability. This is an example of how a novel answer can gain marks for a reasonably structured argument (even though this does not seem to be a very important factor in itself).

Problem 2

> Imagine that a person intends to implement their own symmetric encryption algorithm in order to keep large amounts of data safe on a hard drive. To ensure that the decryption and encryption processes are quick and effective she has decided to keep to a fairly simple substitutional

1 Though many times documentation of that vulnerability will in fact have been published before the update has been released.

method. The security and practicality of this method will to a significant degree be dependent on certain qualities of the keys that are used. Suggest and motivate what such qualities can be.

The two principle factors of importance for such relatively simple keys is length and randomness.

The longer the key the larger is the keyspace, and a perfectly random key means that no one key within that keyspace is more likely than another. The larger the keyspace the more difficult it is to execute a brute force attack with reasonable resources and within reasonable time.

Some answers suggested that longer keys will mean less repetition of patterns in the cryptotext. We have seen during the course that such repeating patterns are a problem for such methods as the Vigenere cipher. However, we have also seen such methods as Cipher Block Chaining that can be employed to reduce the occurrence of repeated patters. We see by this that the frequency of repeating patterns is not only dependent on the keysize, but can be dependent on the substitutional algorithm employed.

Simply stating the longer the key the better is ignoring practical issues. If the key was as long as the data on the hard drive then the encryption would be very strong, but totally impractical since it would take another hard drive to keep the key, and keeping the key safe would be just as difficult as protecting the original data. The key should be so long as to prohibit a brute force attack while at the same time being of a practically manageable size.

A number of answers clouded things by discussing issues like avoiding the reuse of keys, or being careful to transmit them carefully. These are not issues that are especially relevant to the problem of encrypting a hard-drive. Such answers suggest that the writer has difficulties differentiating encryption of messages from other encryption situations.

Some answers brought in discussions of properties of passwords. Unless the answer also explained the assumption that the key should be chosen by a user (a very unnecessary and unlikely assumption) such answers were assumed to be confused as to the difference between passwords and cryptographic keys.

Problem 3

Why is it important to change passwords after a period of time? Give well motivated arguments based on explanations of the kind of threats that passwords that are not regularly changed can suffer from.

Each time a password is used it is to some degree exposed and is therefore at risk from eavesdropping. For example, a password is typed in at a keyboard allowing others the opportunity to see what keys are pressed. The password may unavoidably be transmitted though insecure media on its way between the authenticated and authenticating nodes, allowing it to be sniffed on the way. If a password is written down on some medium as a necessary aid to remember large numbers of them, then whenever that medium is referred to the location of the password may be revealed. Sometimes one might accidentally type a password to one system as input to another, possibly revealing that password to a less secure system. Sometimes, one might even accidentally type a password in the wrong text field, causing it to be revealed on a screen. All these situations go to show that the more a password is used, the greater is the risk that it can be revealed to others. Note how this is a problem that is largely unrelated to the actual strength of the password, i.e. how well formed it is[2]. Regularly changing passwords reduces the risk that a password is with time discovered, or that a discovered password can be utilised for an extended period.

Even if a password is not discovered directly through eavesdropping, there may be possibilities to discover a password through other means. A system that allows multiple login attempts could be fooled by directly attempting large numbers of likely passwords until one proves to work. It may be possible to gain access to hash codes of passwords (such as are used in the authentication process on unix systems, albeit modified with a salt value), in which case it can be possible to test hashing likely passwords until one that produces the same hash is discovered. There are commonly

2 Though admittedly a password like *qwerty* would be more prone in an over-the-shoulder eavesdrop.

available tools that implement such attacks on passwords, either based on a dictionary attack or a pure brute force attack. Processes to discover passwords in this way can be computationally expensive, but given enough computing resources and enough time even moderately strong passwords could be discovered. Regularly changing passwords assures that the password is changed before a long running process has a reasonable chance to discover it.

Losing your password is not like losing your keys, i.e. if you lose your keys you will soon discover the problem, whereas it may be possible to make illicit use of a compromised password without your knowledge. It is a safer strategy to assume that all passwords will with time be compromised, and therefore subject to regular change.

Password authentication methods are known to be a relative weak spot in system security insofar as they rely on the security awareness of the individuals who create and keep their own passwords. There any many methods by which a password can be discovered, including those mentioned above. Changing a password regularly is at least a method of damage containment should a password be revealed, i.e., illicit use of that password is limited until such time as it is next changed.

Many student arguments were only vaguely related to the motivation to change passwords. For example, it might be argued that users might accidentally give away there passwords to social engineers, and for that reason it is a good idea to change the password. I regard this to be a very weak line of argumentation since in such cases it is clear that the most important measure is to not give away passwords. Arguments that show the unavoidable weakness of old passwords were regarded as the best.

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.
>
> - Covert channel
> - Non-repudiation
> - Worm
> - Pseudo-anonymous remailer

Covert channel

Description

The Bishop definition is “A *covert channel* is a path of communication that was not designed to be used for communication” [Bishop05, p288]. One form of this is as a secret path of communication that piggy-backs on another, well understood and known communication path, such as hiding messages in the output of a web site.

Relationship to steganography

Steganography is the collective name for methods by which the fact that a message is being sent is obscured. The similarity between the two concepts is so great as to be confusing. They can be seen as two ways to describe the same phenomenon; whereas steganography is a collective name for one form of secret communication methods, covert channel is more concerned with the communication channel perspective. Steganography is used in the implementation of covert channels. A systems administrator who wants to check that there is no possible information leakage from one system to another might audit the system from the perspective of what possible communication channels

exist, and therefore look for possible covert channels, rather than considering the dangers of steganography.

Example

Two virtual machines running on the same hardware may be assumed to be well segregated and unable to transfer any information between each other. However, since they ultimately share the same cpu it may be possible to establish a covert channel. Run a process on one of those machines that loads the cpu in a predictable manner. Another process on the second virtual machine may be run that notes changes in cpu usage and can interpret them as messages from the first process.

Non-repudiation

Description

Non-repudiation is a general service that can be supplied by security systems. It ensures that subjects in a secure system can be shown to have taken certain actions, i.e. it cannot be denied that they have taken such actions.

Relationship to authentication

Authentication is another general security service where an entity in the real world is bound to a subject of a secure system. Authentication can be said to be a necessary pre-requisite to non-repudiation. Without a secure binding of entities of the real world, no actions of subjects within the secure system would be attributable to real agents.

Example

A teacher denies having received a student's hand-in and accuses the student of being mistaken about having sent it in on time. Luckily the message system used provides a copy of the sent mail for the student to keep, but also digitally signs the email together with a secure time-stamp of the time that the system received the message from message for delivery. When that signature is verified as having come from the system the teacher can no longer deny that the student fulfilled her obligations. The system, or the teacher may have misplaced the email, but it cannot be denied that the student sent it properly.

Worm

Description

According to Bishop's definition:

“A computer worm is a program that copies itself from one computer to another” [Bishop05, p373] We may assume that this definition (in common with other definitions from the same chapter on malware) assumes an element of ill intent. A worm is commonly programmed with the ability to exploit security vulnerabilities in order to spread itself from one computer to another.

Relationship to computer virus

Some definitions of the term computer virus will cover many kinds of malware, However, the more precise definitions will usually establish that a computer virus attaches itself to program hosts. In that case we can say that while viruses have programs as hosts, worms will not attach themselves to a program but to the system itself. Since worms have the ability to spread themselves from system to system, without relying on a a host program to first be activated, they will commonly spread much faster than the classic computer virus.

Example

The so called *Morris Worm* infected the Internet while it was still young in 1988. It exploited

known vulnerabilities of Unix systems, including a very rudimentary dictionary attack. Though it was not written to be harmful, accidental side effects caused it to become an effective denial of service attack for a significant portion of the Internet of the day, and an estimated 10% of Internet connected machines were infected.

Pseudo-anonymous remailer

Description

"A *pseudo-annoymous* (or *pseudonymous*) *remailer* is a remailer that replaces the originating electronic mail addresses (and associated data) of messages it receives before it forwards them, but keeps mappings of the anonymous identities and the associated origins." [Bishp05, p27]

This is therefore a means to anonymously send emails to a party, who can then reply to such an email. The reply is routed to the pseudo-anonymous remailer, which then maps the original sender's address into the message, and forwards it there.

Relationship to type 1 remailers

A *type 1* or *Cypherpunk remailer* also facilitates the sending of anonymous emails. In contrast to the pseudo-anonymous remailer it does not replace the sender address with a pseudonym but simply strips all sender information from the email header fields. The type 1 remailer therefore keeps no mappings between sender addresses and pseudonyms. It is not possible to reply to such anonymous messages.

Example

Possibly the most famous of pseudo-anonymous remailers was anon.penet.fi. During the 1990's it operated a well known service that was used for anonymous email services by senders from all around the globe. After several incidents where the site was subpoenaed to reveal identities of email senders by providing information of the mappings that it kept, those who ran the service decided that since anonymity could not in practice be guaranteed the service was discontinued.

Problem 5

> Saltzer and Schreoder's principles for the design and implementation of security mechanisms are summarised as:
>
> - The Principle of Least Privilege
> - The Principle of Fail-Safe Defaults
> - The Principle of Economy of Mechanism
> - The Principle of Complete Mediation
> - The Principle of Open Design
> - The Principle of Separation of Privilege
> - The Principle of Least Common Mechanism
> - The Principle of Psychological Acceptability
>
> Pick the three principles that you consider would be most important for the design of a secure and efficient firewall, and fully motivate the choice that you make including reasons why other principles are assumed to be less important.

Pass marks were given where answers showed an understanding of the principles, even if there was some minor confusion on which principle was which. Good marks were given where it was clear that there was a correct understanding of the principles and how they apply to firewalls. High marks were given where convincing arguments for the top three were given.

A complete analysis of how the principles apply to firewall design is far beyond the scope of an exam question. Therefore there are many possible ways to briefly reason on the subject, and if the

arguments were consistent, clear and reasonable, they were good enough for this problem. The following suggested answer is therefore to be regarded as only one possible way to reason.

Viewing a firewall from a functional point of view (rather than from e.g. its administration point of view) its main purpose is access control of network traffic, both incoming and outgoing. Let us begin by relating the principles that are more directly associated with access control.

Firewalls will not normally be endowed with enough information to make complex decisions on the nature of the processes that are communicating through them. They have relatively simple criteria to decide if a port should be open for a communication stream, or not. In this respect there does not seem to be enough room to apply the principle of least privilege (whereby a process should be given only those privileges need to complete its task). The principle would appear to be of lesser importance here.

The relative simplicity of the access control in a firewall also suggests that the principle of fail-safe defaults will less relevant. The principle says, in summary, that white lists are better for security than black lists. But it will be very difficult to characterise the access control only in terms of what kind of traffic is to be allowed in a white list. It will often make more sense to also describe what should be refused access in a black list.

The principle of least common mechanism requires that mechanisms used to access resources not be shared. This might be interpreted to mean that the firewall should not share hardware with other software, and this seems to be good general advice, except that for personal firewalls their design and purpose is to run on the same hardware that they protect. It seems to be difficult to hold this as a general principle for firewalls in practice.

The principle of separation of privilege states that one should not grant permissions on the basis of a single condition. It may be a good idea to grant access to a port based on several specific conditions, such as that a certain port is only open for a process that can show that it is from a trusted address, is not previously known to have executed a denial of service attack, and requests an SMTP port. However, in many situations the conditions for access will be simple, and complicating them for the sake of this principle does not seem motivated. This is once again a principle that is difficult to apply consequently for a firewall.

The one among the access control related principles that does seem generally applicable, and indeed the very essence of what a firewall is, is that of complete mediation. It is very important that a firewall filter all traffic without exception. There should be no loopholes in network communication that a firewall does not check. Admittedly there will be depths of complexity in dangerous traffic that it will be beyond the ability of the firewall to check, but that does not mean to say that it should not check all traffic within the limits of its abilities.

It is difficult to argue convincingly here for Open Design. The principle states that the security of a mechanism should not depend on the secrecy of its design or implementation [Bishop05, p204]. But surely there is little in a firewall design that could be enhanced by attempting to keep the design secret, so this becomes a non-issue[3]

From the point of view of the firewall user, psychological acceptability is surely paramount. If a firewall causes network to noticeably slow down, or else block traffic that is legitimate, then users will not stand for it, and ultimately (if they are able) disable the firewall. If we were to shift perspective to the administration of a firewall for a moment, even here the psychological acceptability factor is extremely important. A major factor in the failure of firewalls is from configuration errors by administrators. This indicates that if a firewall is desgined to ease correct configuration it will be a more secure tool.

This discussion leaves only the principle of economy of mechanism. A firewall should be an

3 Some answers held the line that open design is a good way to get free help in bug-hunting, but that is surely a very tenuous principle on which to base the design of a very vital security tool. The “many-eyes” principle has so far not been shown to be reliable enough (e.g. bugs are not infrequent in linuxes despite the code being entirely open). The principle of open design does not suggest that we should be open because “many-eyes” improves security. It says that trying to implement security by hiding things is generally not a good idea.

effective, efficient and bug-free security tool. This indicates that complexity of design and implementation should be kept to a minimum. Firewalls are not the kind of tool where one would like to see feature creep at the expense of well written, economic code[4].

From the above it is motivated that economy of mechanism, complete mediation and psychological acceptability are the three most important factors for the design of a secure and efficient firewall.

Reference

Bishop05 Matt Bishop, Introduction to Computer Security, Addison Wesley, 2005.

4 It is clear from a number of answers that some students are still equating economy of mechanism with simplicity of use. The two should not be confused. I claim that in practice the two ideas are in conflict – in order to make a program easy to use one will most likely have to introduce very complex coding.

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.

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.

DSV SU/KTH *SEC:I*
Principles of Computer Security
Autumn Term 2012

Alan Davidson

Suggested Solutions to the Exam 2012-10-31 and Comments on the Marking

The answers suggested here may be briefer than would normally be expected from students. I have tried to summarise the most important aspects of problems so that students may compare the content their own answers to these. Lack of time prevents me from doing more.

Problem 1

> Define the terms *Security Policy* and *Security Mechanism*, and motivate why it is important to differentiate between the two concepts.

According to the definitions from the course book [Bishop05]:

A security policy is a statement of what is, and what is not, allowed.[p7]

and also a later refinement of this definition

A security policy is a statement that partitions the states of the system into a set of authorised, or secure, states and a set of unauthorised of nonsecure, states.[p45]

and

A security mechanism is a method, tool, or procedure for enforcing a security policy. [p7]

I do not normally expect students to remember such definitions word for word, so any answer that could convey similar concepts would be given credit.

We understand from the definitions that the mechanisms are only our best attempt at supporting what the policy describes. There will very seldom be a perfect match between the policy and the mechanisms that enforce it. This implies that there will be some unauthorised states that the mechanisms may not able to protect against, and conversely some authorised states that mechanisms may disallow.

A common misconception is that it is the mechanisms that define the security of a system. By differentiating between the two concepts one can understand that just because and action is not disallowed by the mechanisms that does not mean to say that it allowed.

A small number of answers had the same worrying misconception, i.e. that languages and models such as Bell LaPadula should in some way be examples of mechanisms. I took that to be evidence of quite a basic misunderstanding,

Problem 2

> During the course we have made a clear division between methods for authentication and for access control. However, for some security mechanisms it may not be as clear whether they can be classified and described as authentication mechanisms or access control mechanisms.
>
> Consider the key-cards that are distributed to staff and students at the DSV department to allow access to the building and its rooms.
>
> Characterise the key-card system in terms commonly used to describe authentication methods, and then in terms commonly used to describe an access control mechanism. For each of these viewpoints motivate why the system is best characterised in the terms that you have used.

As an authentication mechanism one might characterise the key-card system as both authentication through something the entity (card holder) knows in terms of the PIN code, and something the

entity has in terms of the plastic card. Note how for some rooms and at certain times it is enough to swipe a valid card in order to gain access. However, for some places and at some times a higher level of security is deemed necessary, and then both the card and the PIN code are required to gain access. In this case we see how the authentication mechanisms utilises the principle of Defence in Depth, with a close association to the Saltzer and Schroeder's Principle of Separation of Privilege.

Other ways to classify authentication methods are s*omething that the entity is*, or *where the entity is*. There is a photograph of the card holder on the card, so one might claim that there is an element of who the entity is, i.e. what their face looks like that links them to the card. But then we should question how this aspect is checked by the authentication system. It is only humans that can view this picture and make the connection, and in practice there is no part of the system that requires human interaction. Perhaps if we were all required to wear our cards visible while in DSV rooms, and if the policy that we all sign required us to monitor and report if we see someone with a card that is not theirs, then we could call it part of the system.

One might make the case that part of the authentication is *where the entity is* given that the rooms that are being accessed have a physical location, and one must be outside the door to authenticate. Though some might develop this as an argument I suggest that the physical location is an intrinsic property of the system, rather than a useful attribute for authentication. The card and PIN code are not more correct authentication methods because they are in the Forum building (Well made arguments to the contrary were of course given credit too).

It seems to be a fairly simple protocol, and though the subject is sometimes prompted to enter a PIN code with the way that the keypad symbols light up, if we were to view it as a challenge-response protocol it is such a simple and predictable protocol that it does not seem valuable to analyse the system in such terms.

In terms of access control we can view the system as an Access Control Matrix (ACM) where the subjects are users and the objects locks on the corridors and rooms. The obvious rights are *open lock*, but sometimes you may notice that teachers can *set room to unlocked*, and *set room to locked,* for example to allow and disallow access to lecture halls.

The course has taught us that access control is seldom implemented as a matrix. We might therefore consider whether the lock system is best characterised as an Access Control List (ACL) or Capabilities. As an ACL the rights are primarily associated with the objects, which in this case is the locks. Is it likely that each lock has a list of users, or groups of users, associates with the rights for such subjects? It may not be easy for a student to tell from experience. Being a member of staff my experience is that changes to the system are made at a central system but that it can take a very long time for those changes to propagate to the actual lock. This suggests to me that it is indeed very likely that the information is downloaded to the locks.

For the system to be capability based the rights would in some way be associated with the users, or rather in the cards that users hold. The card would then list rooms (or classes of rooms) and the rights for each of those objects. Since the only means for the card to contain such information is on the magnetic strip it does not seem very likely as this is such a simple mechanism. Moreover, one would expect that any changes to rights (such as being allowed special access to the sandbox laboratory in 408 during the SEC:I assignments) would have to involve some change to the configuration of the card. Since students did not have to put their card into a card writer to gain new access, one could surmise that it is very unlikely that the system is capability based.

One might make the case to the access control system being role-based. IT seems that there are likely to be a number of basic roles: student, teacher, technical staff, cleaning staff, etc. Our arguments motivating this might be that some students become teaching assistants, and it would make sense for a role-based mechanism to aid in quickly switching those rights. Without closer inspection of the workings of the system it is difficult so say more.

We have also used terms such as DAC, MAC, and ORCON to describe strategies to manage the changing of rights. In broad terms, with DAC the object owner is free to define the rights, in MAC

it is the system owner, and with ORCON the originator of the object. The originator of the lock system is the designer and manufacturer of the lock. It does not make sense to assume that they would control the rights. To differentiate between DAC and MAC we would have to know who owns the individual locks, and how that differs from the owners of the whole system. I would suggest that the difference is of very little significance here, i.e. it is most likely that DSV can be viewed as both the owner of the locks and of the system as a whole. In this example we cannot therefore distinguish between DAC and MAC. I would not normally expect SEC:I students to manage this depth of insight during the exam. Sensible, motivated discussions that suggested either DAC or MAC were given credit.

Some examinees confused which entity was to be authenticated in their discussions. Since we are talking about access control to rooms, it makes sense to say that the entity is the person. In that context it is not the card that is authenticated, but the card holder.

Problem 3

> One method that can protect a system from the effects of malware infection is to ensure that any changes to software on secondary memory are detected and reported so that the system administrator can effectively respond. Such a method would work even if the infections is caused by previously unknown malware.
>
> a) Suggest in outline a reliable mechanism that can detect changes to software.
>
> b) Discuss possible weaknesses that this method has as useful malware protection.

a) This is clearly a problem of the integrity of programs. During the course we have touched on some means to provide integrity services. This problem does not seek to directly uphold integrity (e.g. by ensuring that only trusted processes have access to the resources), but to detect changes. Therefore I would claim that the given answer to this problem is:

The mechanism might keep cryptographic checksums or hash values of programs and regularly and automatically check them. When a discrepancy between the hash value of software on disk and the hash value that is recorded in a database is detected then the system administrator can be alerted. Such a hash value might be inserted into the database during the first installation of the software, or even before fetching the software. Many online sources of software display hash values of that software in order for customers to ensure that the software that is downloaded is the same as the software at the source.

Some answers that were along these lines suggested that the software need not be checked regularly but instead only before it is executed. The strategies may be a question of personal taste. My own motivation for regular checks is that it is less annoying to be warned during a security check than it is just as one wants to run a program, only to be told that an infection is suspected and the program must not be run.

Answers that suggested keeping track of file sizes were considered very weak. We know from the course that recording and checking the size of a file is hardly a reliable mechanism since among other things the malware might be able to change the file but without changing the file size. Such an answer shows that the examinee has entirely missed the connection to integrity services, and the role that cryptography can play, which was after all a major theme of the course.

Unfortunately a number of examinees attempted a trivial answer “Use antivirus software”! I cannot accept this as a valid answer since it is not a sensible answer to the problem “Suggest in outline a reliable mechanism that can detect changes to software”.

b) We can imagine several reasons why this strategy might prove to be less effective:

i) If this strategy is well understood by malware creators then it may be difficult to protect the database of hash values or the process that compares them. We need to be able to trust the hash values, so we would also need to have trustworthy integrity checks on that database and process so that malicious software cannot fool the strategy by altering the hash values or subverting the check mechanism.

ii) The current 'culture' of software distribution allows for frequent updates of software, sometimes to include new features, but very often in order to fix problems that have been discovered after the software has left the developers. Many times the problems will be security problems, which means that not updating means that security will suffer. The hashcode mechanisms will presumably have to be designed so as to allow for software updates, and thereby hash value updates, without raising bogus alarms. Such a fix would help the psychological acceptability of such a mechanism, but also open up another possible means for malware to bypass the checks.

iii) The distinction between programs and data is of importance, since we might normally assume that data changes are frequent, whereas program code tends to be immutable. However, the clear dividing line between program and data has become all the more muddied over the years. For example, a modern formatted text document might include instructions to the program that manipulates it, which ostensibly makes it both data and at the same time an interpreted program.

iv) Regular automatic scans and checks require computing resources. Were the checks to make a noticeable difference to the user's of a system the user may be less inclined to allow such checks to run.

v) If the mechanism only checks the integrity of legitimate programs it can only be used to detect the existence of computer viruses. Other kinds of malware such as worms would not be detected since they (by definition) do not attach themselves to programs.

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:
>
> - Hard Certificate
> - Saltzer and Schroeder's Principle of Psychological Acceptability
> - Phishing
> - Buffer Overflow

Hard Certificate

Definition

A hard certificate is a certificate that is kept on a hardware device that is specially designed so as to protect the certificate from being copied or changed, but still allows that certificate to be utilised for cryptographic operations. A certificate is a data structure that primarily contains one of the two keys in an asymmetric key pair. In the context of *hard certificate* the contained key is presumed to be a private key, which therefore puts high demands on the confidentiality and integrity of that certificate.

Relationship to Soft Certificate

A soft certificate is likewise one that contains the sensitive private key, but it is kept on the secondary memory of the system that uses that key. Soft certificates are generally kept encrypted on secondary memory until such time as they are needed, affording them some security. However, if the secondary memory can be accessed and the cryptographic key discovered the soft certificate is

compromised.

Example

During a lecture I showed a device that is used to authenticate bank transactions. This device is a smart card reader that is connected to a computer with a usb connection. A particular kind of credit card (one with BankID on it) is inserted into the reader, and a PIN number is typed into the device whenever an authentication is required. The chip on the smart card contains a hard certificate. The certificate itself is never available to the computer or the card reader. The required cryptographic functions are executed by the chip on the card itself, which is the only component that has access to the certificate.

Saltzer and Schroeder's Principle of Psychological Acceptability

Definition

Though a word-for-word quotation of Bishop's definition is not expected, answers should give the gist of:

The principle of psychological acceptability states that security mechanisms should not make the resource more difficult to access that if the security mechanisms were not present [Bishop05 p206].

Relationship to Saltzer and Schroeder's Principle of Economy of Mechanism

To summarise the principle of psychological acceptability, the mechanisms should make security easy for the user. The principle of economy of mechanisms says (in essence) that the mechanisms should be simply constructed. Simply constructed and simple to use are not necessarily the same kind of simplicity. Indeed they may be in conflict given that mechanisms that make security easy for the user will more than likely be very complicated in their construction. A command line controlled operating system is relatively simple in its construction, but most would find its security mechanisms difficult to interact with, so we have extremely complicated window, pointer, icon etc systems to allow one to more simply interact with them.

Example

A firewall that is difficult to configure because the interface is unintuitive is likely to be insecure. Likewise, if it requires so many resources that it makes the computer slow down considerably, then a user might decide that the security win is not worth the loss of fast communications.

Phishing

Definition

Phishing is a form of social engineering that is perpetrated through electronic media. Its name alludes to the principle of fishing in that one casts out attractive bait ostensibly in a non-directed manner in the hopes that a victim will act upon it. Phishing normally involves some kind of spoofing of a legitimate party in the hopes of gaining private information from uses.

Relationship to MITM attack

The Man/Monkey in the Middle attack involves redirecting traffic that should be passing between two parties through a third malicious party. The traffic can thereby be eavesdropped and manipulated. This is in contrast to phishing where (in one common form of the attack) a user is fooled into visiting a website that is in the control of the attacker which implements a version of a website that the user can mistake as a legitimate website. In contrast to MITM, phishing does not involve the legitimate website in the communication.

Example

A common example of phishing is that an email is sent out (most probably as spam) to a large

number of recipients where the contents are a spoof of a warning from an email account administrator, saying that for some reason the user's email account will be closed down if action is not taken. The email may contain a URL that takes the user to a site that is also a spoof of a legitimate site, but that asks the user to enter information that the perpetrator can use to their own advantage. That might be as simple as the users password to their email account. Given that attackers can continue to perpetrate any number of attacks that could cost the user dearly.

Comment: Most students did not distinguish between phishing and spoofing or social engineering in general, so to be strict most did not really know what phishing was, even though I am sure that we have all experienced it. Such exact definitions were however not required in order to gain credit.

Buffer Overflow

Definition

Buffer overflow is a vulnerability that can affect programs that are written in languages that do not have automatic memory management but where memory management is at the control (and responsibility) of the programmer. The problem arises where a program, in writing to a memory buffer writes beyond the limits of that buffer, thereby overwriting adjacent memory. One of the most dangerous effects of such a vulnerability is that an attacker may supply specially crafted data that the program copies into memory that thereby cause the program to execute instructions that are included in that crafted data. The effect can therefore be that an attacker can execute any code that they like with the same privileges as the attacked process. Program checks of input data to ensure that it will fix into buffers before copying to them is an effective means of avoiding many buffer overflow problems.

Relationship to Code Injection

Another general class of vulnerability is code injection. Here too an attacker can craft data that may effect the execution of a program. The difference is that in code injection the problem comes from allowing input to the program to be interpreted as program instructions. Code injection therefore only affects interpreted languages such as SQL or python.

Example

A program requests a year to be entered by the user, and assumes that the user will only enter four digits and thereto allocates four bites of memory. The program does not however check that only four characters are entered but continues to copy user input to memory one character at a time, moving forward in memory one byte at a time until the end of user input is met. The memory adjacent to the four byte buffer will be overwritten if more that four characters are input. What effect this has can very from bringing about a memory exception to manipulating program execution.

Comment: Several answers gave the impression that buffer overflows most often or always cause an execution error that aborts the process. This is far from the truth. I usually characterise this as being the best possible i.e. least dangerous outcome. Far more dangerous is that buffer overflows might allow an attacker to execute any code that they want to with the same privileges as the attacked process. Somehow a large number of you got the idea (perhaps from a misunderstanding of something said at lecture?) that buffer overflows main problem is that they can crash or halt processes. **This is a very serious misconception**. Directed buffer overflow attacks are far more serious than that!

Problem 5

> Though IT technology in general has introduced many threats to personal privacy, tools can and have been developed that enhance our ability to control what information about us is spread to others.
>
> a) Describe in general terms what *remailers* do to enhance privacy, and explain why such services

can be assumed to be both good and bad for society.

b) Cryptography is an important element in remailers where a high level of privacy protection is necessary. Explain why and how.

a)

Remailers strip all information that can identify the sender from the headers of emails in order to make the messages anonymous. Note however that if the users themselves were to include information that can lead back to them in the body of their email then no remailer will help in stripping that info.

The simplest types of remailers keep a mapping between the sender's email address and a pseudonym, thereby allowing replies sent to that pseudonym to be returned to the original sender. Such pseudonymous remailers are no longer popular since the demise of the penet.fi after the site was forced by Finnish courts to reveal details of their address mapping tables.

More advanced remailers all use networks of servers that forward the email through each other . This is done to ensure that no one point in the network will be the goal for intrusion attempts that would thereby make the system insecurity. In order to break the anonymity one would have to crack a large number of the servers.

By allowing anonymous emails the system affords users some degree of information self-determination, at least in terms of what information about them is automatically included in email headers.

Bishop has a section in the course book *Anonymity for Better of Worse* [Bishop05 p230] which I refer the reader to for a discussion with several ideas that examinees could use to relate to the good and bad of remailers.

Several answers confused privacy and anonymity with confidentiality. Such answers were considered to show a lack of understanding of the privacy discussion that is part of the course.

b)

As stated above, the idea with using a network of remailers is that no one remailer should have enough information so that a compromise of any one system (or for that matter all remailers except one) could reveal enough information to identify a sender. If someone were to sniff traffic between remailers there should not be any possibility to analyse data and thereby identify where data originates from. Encryption of the messages between all parties in an anonymising network ensures that such analysis will be intractable.

A common scheme is that the user who is to be kept anonymous will encrypt the recipient's address and the body of a message with the public key of the last remailer in a chain of remailers. This ciphertext is then in its turn encrypted (together with the address to the last remailer) with the public key of the next to last remailer, and likewise this is encrypted in sequences leading all the way back to the first remailer used. This allows each remailer to 'unpeal' the outer layer of encryption and forward the contents to the next remailer, or ultimately the recipient. In this way any one remailer can only know where the message they received came from and where they send it.

Note how anonymity is preserved even if the message from the last remailer is sent to the recipient in clear. That tells us that confidentiality of the message itself is not the goal (as some students mistakenly presumed).

For part a the best answers are concrete explanation of what actions remailers take, thereby showing that you understand what a remailer is.

For part b, one must show an understanding of what a remailer is to have a sensible discussion, and the dangers of network sniffing.

Reference

Bishop05 Matt Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

DSV SU/KTH

IntroSec
Introduction to Information Security
Autumn Term 2015

Alan Davidson

Suggested Solutions to the Exam 2016-01-15 and Comments on the Marking

Some answers suggested here may be more comprehensive than would normally be expected from students in order to cover several relevant aspects of the problem.

Problem 1

a) Explain what aspects of a system are balanced in a risk analysis in order to practically apply reasonable security safeguards.

b) Discuss reasons why this balancing of relevant aspects can in practice be difficult to achieve.

a) As expressed in the course book: “A basic model of risk management involves a user's calculating the value of all assets, determining the amount of harm from all possible threats, computing the costs of protection, selecting safeguards (that is, controls or countermeasures) based on the degree of risk and on limited resources, and applying the safeguards to optimize harm averted” (Pfleeger, Pfleeger and Margulies, 2015, p23). If the costs of protection are too great in the balance of how effective that protection is against the loss of asset value, then one has already 'lost'. There is no such thing as perfect security, unless it is finding the perfect balance in a risk analysis.

b) One way to take this discussion could be:

Values of assets are not necessarily easy to put figures on. Aspects that could be affected in a security situation include such intangibles as public confidence in your organisation. A sense of feeling secure might in some situations be valuable, as separate from the true ability of a security measure to protect an asset, e.g. visible security measures at an airport might make customers more inclined to feel safe and therefore fly, even if they might not be totally effective. How much some aspects of assets are worth could be dependent on the philosophical, religious, ethical, profiteering etc outlook of the parties involved, where there is no clear consensus on what is best. One might for example question why so much security expenditure has followed the 9/11 attacks in USA when americans themselves seem to be responsible for killing some ten times as many people each year on the roads (WHO, 2015) without a comparable expenditure on hindering such deaths compared to deaths attributed to terrorism.

I am puzzled and despondent that so many answers to this question simply brought up the CIA triad. It should be very clear from the course book and lecture discussions that the important element to a risk analysis is that of the value of assets. I have a worrying suspicion that the only reason that students have answered this way is that the keyword “balanced” in the problem text could be associated to a separate discussion from lectures, i.e. where we showed how we should not presume that security is a point of perfect balance in the centre of the CIA triad. Having said this, there are other possible points of association between the CIA triad and risk analysis. In the course book, the authors use the triad as part of a mapping in an illustrative table that they suggest for one single phase in a risk analysis (p674). It would surprise me if this was the source of students' erroneous answers since that part of the book is not part of the course, i.e. it is not within the course reading notes.

Since there was such a distinctive pattern to so many students' wrong answers for this problem I felt obliged to question if anything that had happened on the course could be the source of students getting the wrong idea. If that were the case one might think that this could be seen as an 'unfairly'

difficult problem to set. In this perspective I ended up trying to read the most I possibly could into answers to see if they could be said to be worthy the lowest possible passing grade, even when the answer was wrong. I therefore decided that answers that showed capable reasoning around the wrong answer could conceivably be given a pass grade. I nevertheless worry that the real reason for the low occurrence of good answers is that some students might assume that exam problems will not be set based on passages of course book, even when they are covered in the reading notes.

Problem 2

> Even within relatively small organisations it is quite reasonable for a single user's Internet traffic to pass through three separate firewalls before reaching the Internet.
>
> a) Explain in terms of general security design principles why this strategy could be considered good security practice, but identify and explain also any principles that suggest that this could be bad for security.
>
> b) Explain the likely roles of the specific firewall types involved.

a) In the most general terms we can state things like: Redundancy in security mechanisms is a wise strategy, as is Defence in Depth (here in the IT security sense of the term rather than the military one). But since the problem specifically relates to security design principles we associate with such principles as those defined by Saltzer and Schroeder (Pfleeger, Pfleeger and Margulies, 2015, p212), more depth is required. There are all sorts of ways to reasons sensibly around the principles in this context, the following being one attempt.

The Saltzer and Schroeder principle that is most closely associated to the ideas of Defence in Depth is that of Separation of Privilege. If access to all objects should depend on more than one condition, then access of data from the Internet to the local computer (as well as vice versa) will pass through several firewalls on the way, each of which will presumably have their own differing set of access conditions.

We might make the case for the firewalls being an implementation of the principle of complete mediation. It is tricky to claim this without in part answering part b of this problem, since without knowing which resource the firewall mediates the traffic to, it is hard to claim it is complete. If the firewall is on the Internet side of a proxy server then one would expect all traffic from the Internet to be routed through that firewall, and that firewall to in some way check all of the incoming traffic. The firewall can then be said to be implementing complete mediation of Internet traffic for the proxy server. A similar point can be made in case the firewall is of the personal sort. Such a firewall should mediate all network traffic going to that individual computer, i.e. Internet as well as from the local network.

The principle of least common mechanism can also be cited, inasmuch as different network filtering responsibilities can be separated between different firewall devices rather than sharing the same resources.

When it comest to the principle of Psychological Acceptability, three separate firewalls could mean a security problem. If, for example, it is found that legitimate traffic is not being passed, the reasons for this could have one out of three sources. But searching for errors will be difficult. It can also be more difficult work for an administrator to correctly configure three firewalls compared to one.

Some students argued that having several firewalls is likely to slow down network traffic and thereby negatively affect psychological acceptability. In fact, three firewalls would work in parallel, so they presumably would slow things down less than one more complex one.

Some students also cited the problem of upholding the principle of economy of mechanism in the firewalls. I think this is a tricky argument. The problem text discusses having three firewalls, so I can question why having three firewalls would be more of a problem for economy of mechanism than having one. Each individual firewall could be said to be simpler than a single firewall that tried to do all the filtering, and that seems to uphold economy of mechanism. Some argued that the security problem is complex, so that the firewall mechanism will be complex, which is against

economy of mechanism. But once again the problem is about the strategy of using several firewalls, so this argument does not directly address that strategy, but firewalls in general.

b) One reasonable interpretation of the roles could be these are the inner and outer firewalls of a DMZ, and the personal firewall of the user's own workstation. For the sake of brevity in this suggested answer sheet I ask students to refer to the course book for descriptions of these firewalls' roles.

A number of students described the firewall types that we have covered on the course, i.e. packet filters, stateful inspection, etc. I would say that these are more *types* than *roles*. It is unfortunate if students have the idea that these are roles. That breakdown of firewall types is more about showing a historical progression of firewall technology that explains how they functionally are doing things on several layers of abstraction. These days it is unlikely that you would be happy with a firewall that only does packet filtering.

Problem 3

> Describe and motivate useful ways to relatively easily configure and use a personal web browser and mail agent in order to ensure reasonable levels of privacy and security while at the same time upholding reasonable levels of usability.

There are many ways to suggest solutions to this problem, just so long as they are well motivated, and can be said to keep to within the requirements that are set out in the problem text. We have covered several during the course, such as:

- The danger of client-side interpreted script languages such as Java, ActiveX, JavaScript.
- The problem of web page redirects that do not allow the opportunity user to inspect malicious URL content.
- The problem of how links to images in an html formatted email can signal to a server that as specific recipient has read an email at a specific time.

Other things we have studied also become relevant for this problem, such as we understand the advantages of using the https protocol over http, and the dangers involved in accepting (and thereby trusting) self-signed certificates.

In terms of upholding spacial privacy arguments can be made for configuring for spam filtering in an email client and ad blocking in a web browser. In terms of informational self determination one of the simpler and most important configuration options one can expect to be discussed is that of cookies. It is understandable that students propose different strategies based on which browser they are used to, given that what browsers allow one to configure differs.

Arguments that proposed the use of Tor and remailers were generally poor. One can surely hardly cite these as possible mechanisms without addressing the parts of the problem text that says “relatively easily configure and use” and “upholding reasonable levels of usability”. Such tools are for special situations were privacy is needed. You would suffer great problems in your everyday tasks if you were to always use these.

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 *definition*, *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:

- Ransomware
- Security by Obscurity
- Buffer Overflow
- Cryptanalysis

Suggested solutions for problem 4 are not complete in this version of the document. Please see the course book and other relevant sources in the reading notes for definitions and examples.

Problem 5

Consider the following IT related ethical dilemma:

You are a security manager who is responsible, among other things, for the education of a number of staff within your organisation. You need them to know about the IT security threats that they and the organisation can be expected to be subjected to, and their responsibilities in the face of these threats. You have ordered a number of IT security textbooks to distribute to the staff to help them to understand the issues involved. However, the books are not available from the distributors, and they say that it will take several weeks for them to obtain new copies from the printers. One of the staff mentions to you that an illicit electronic copy of the book has been found and that several of the staff already have copies of this on their work computers. You find that the situation is not directly covered by any of the current organisational policy rules. You identify the following possible courses of action:

- You allow the staff to make use of the electronic copies while you are waiting for the books to arrive. Once the books have arrived and you have paid for them fully you ensure that all electronic copies are removed.
- You inform the staff that they should ensure that all such illicit material has no place on the organisation supplied computers and must be immediately removed. You do your best to educate the staff without the textbook, and then schedule in some follow-up work once the books are available.
- You cancel the order for the text books and allow the staff to use their electronic copies until you think you have completed their education. You instruct the staff that they must remove all copies from their computers after the end of the course and tell them that they must not tell anyone that they made use of this material. You imagine that this is likely to please the heads of the organisation not least because the overheads for the education of the staff will be less than you budgeted for, though you can think of other reasons.

a) Apply and document accepted basic ethical principles with careful balanced reasoning to motivate a suitable course of action.

b) Suggest how your course of action would be likely to be affected if you were an ISACA certified professional and subject to their Code of Professional Ethics. Suggest in outline what kinds of ethical rules you assume are included in such a code and that effect this decision.

A brief comment on the marking:

Answers to this problem must show an understanding of the principles for ethical reasoning that are well documented in the course book. It is quite possible to make arguments that show that each of the problem's scenarios are quite reasonable depending on whether one interprets the situation in a telelogical egoistic or utalitarian view, or a rule-based view. One cannot satisfactorily answer this problem without referring to which principles on which one bases the interpretation of the scenario.

I had to assume that discussions that held that the student's answer represented the only viable ethical point of view were evidence that the student had not properly studied the course literature on ethics.

For part b marks were given for any discussion that held evidence that the student had a good idea of the kinds of rules that are common in codes of professional ethics, and how they would effect the answer. ISACA is named in the problem since that is the specific example that given in the reading notes, but specific knowledge of any of the ISACA rules were not expected or required. If one did have specific knowledge, one could cite a rule that suggests the second scenario, i.e. “Serve in the interest of stakeholders in a lawful manner, while maintaining high standards of conduct and character, and not discrediting their profession or the Association” or else a rule that can be said to contradict the second scenario, i.e. “Support the professional education of stakeholders in enhancing

their understanding of the governance and management of enterprise information systems and technology, including: audit, control, security and risk management".

References

Please note that exam answers are not (necessarily) expected to give exact references, but these suggested answers attempt to do the reader the respect of giving references where suitable.

Pfleeger, C.P., Pfleeger, S.L. and Margulies, J., 2015. *Security in Computing.* 5th ed. Massachusets: Prentice Hall

WHO, 2015. *Global status report on road safety 2015.* http://www.who.int/violence_injury_prevention/road_safety_status/2015/TableA2.pdf?ua=1, last visited 2016-02-01

DSV SU/KTH

IntroSec
Introduction to Information Security
Autumn Term 2015

Alan Davidson

Exam Re-sit 2016-03-22
Hints and comments on the Marking

Suggested answers are not provided for this exam, only some pointers to help you validate your own exam answers. Do not assume that the comments here are complete examples of how the exam problem can be answered.

Problem 1

a) Describe what cryptographic elements are involved in the creation and verification of a *digital signature*.

b) Explain digital signatures' role in the certificates that are part of a Public Key Infrastructure (PKI).

a) The answer should show an understanding of how a cryptographic hash and encryption with a private key make a digital signature.

b) Answers should put over the idea that a certificate is a data structure that contains the owner's id and their public key, and that it is signed with a trusted party's digital signature as a means of acknowledging the signers belief that this public key does indeed belong to the signified owner.

Problem

Describe and discuss three weaknesses that biometric methods in general can be said to suffer from that might make them less than suitable as a general authentication method for IT systems. Good answers will show three separate and diverse aspects of the problem.

See Pfleeger et al (2015) p55 onwards for the arguments from the course book. Note that we have had a critical discussion on this subject matter in class, so it is legitimate to question e.g. whether the single point of failure argument is applicable to this problem. Students who could properly cite Pfleeger et al's arguments were of course given marks for their answers. A number of students cited only the examples given in the book without making the same arguments as Pfleeger et al. Such answers could not be given high marks. As an example:

As carefully stated during the course, it is very difficult to claim that a general problem in biometrics is that they do not work when some biological attribute changes. Examples often cited are “fingerprint authentication will not work if I cut my finger” or “voice recognition will not work if I get a cold”. This may (or may not) be true, but one must surely relate this problem to similar ways that other authentication methods fail on occasion, e.g. forgetting one's password. This is far more common, and yet there are usually easy ways to recover or reset the authentication data. So surely we can simply go to the system administrator and register another finger? When Pfleeger et al cite these kinds of examples it is to illustrate that biometrics “can become a single point of failure”, and note how the authors are careful to include the word *can* in this statement.

Some further general arguments include:

Once biometric data has been discovered and a method to masquerade is implemented, there is no given method by which to recover the authentication method. In contrast, if a password is discovered one can recover simply by changing the password.

Biometric data is unavoidably linked to the individual. Not all authentication situations need to uniquely identify individuals in order to give them the rights needed for the situation. E.g. a system that checks if I am someone who has legitimately paid for the right to see a movie only needs to check that, not to know exactly who I am. Biometrics can therefore be seen to introduce problems

associated with privacy.

Biometric data are subject to exposure, even outside of their use for authentication. Fingerprints are one example, where one could expect to find fingerprints on many kinds of surfaces that a person normally touches. If the biometric method is possible to falsify, as fingerprints are, then this kind of exposure will make a targeted attack difficult to avoid.

There are indications that biometrics in general do not meet widely held preconceptions that they are necessarily good.

Answers that made categorical claims for all biometric methods could not be given high marks. Good answers are more balanced and considered. For example, “biometrics are easily falsified” is not something that I would dare claim of biometric methods in general. During the course we have not claimed this, but rather said that some methods have been seen to be possible to fool with relatively small resources. Insofar as there is a general impression that biometric methods are “good”, we can therefore argue that this is not necessarily true.

A tip for such exam discussions as these – do not use words like “very”, “easily” etc to qualify your arguments. They are meaningless in an objective presentation of facts and arguments and probably more likely to be an indication of sloppy arguments.

Problem 3

> The common strategies that antivirus software (or what we have termed *anti-malware* during the course) use to detect malware can mean that they are not as effective as one might hope. Describe and motivate two important and separate aspects of malware that can reasonably be assumed to make it difficult for such antiviral systems to discover their presence.

Possible aspects where one can make convincing arguments for this problem are:

- polymorphic malware
- stealth malware, including rootkits. See section in Pfleeger et al on stealth, pp189-191
- directed malware, i.e., designed for a specific target, with little spread “in the wild”, and therefore less chance of being discovered
- fast and self propagating, so that the malware infects before the time window of detection and protection measures are implemented and disseminated.

N.B. the above would not be sufficient as an answer, but a guide to help understand the grading. The problem text calls to “describe and motivate”.

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 *definition*, *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:
>
> - Bell LaPadula
> - Denial of Service
> - Mandatory Access Control
> - DMZ

Suggested solutions for problem 4 are not complete in this version of the document. Please see the

course book and other relevant sources in the reading notes as well as the course video material for definitions and examples.

Suggestions for closely related concepts that can allow for productive discussions on the relationship:

- Bell LaPadula – Biba
- Denial of Service – Ransomware, DDos, Availability
- Mandatory Access Control – Discretionary Access Control
- DMZ – Proxy server, Firewall, Defence in Depth

This does not mean to say that naming these concepts is sufficient for an answer. They must be meaningfully related to the given concepts. I suggest that they are close enough to the given concept for the examinee to be able to show deeper understanding of the given concept through their comparison.

Problem 5

a) Give concise arguments for why one can expect that personal privacy issues are likely to become an increasing concern in the future.

b) Describe IT tools that can on the one hand afford an Internet user pseudo-anonymity, and on the other, anonymity, explaining the mechanisms that these tools use in order to achieve these levels of anonymity.:

a)

The following are summaries of arguments that have been made during the course:

In the information age, all the more detailed information on individuals is valuable for society in general (e.g. in predicting trends among a populace, such as the spread of infection) but also valuable for those who can profit from that information (e.g. manipulating information flow to make an individual more prone to choose one product before another).

All the more personal information is being registered by digital devices (mobile phones, watches, fridges, etc.) and they are becoming all the more connected.

Digital technology, with the tools to collect and process information, are becoming increasingly cost effective, allowing greater detail in the collected information, as well as greater opportunity to predict and also manipulate individuals. Such advances include the inclusion of diverse sensors , such as GPS devices, physiological sensors like heart rate, sleep phase sensors, etc., but also advances in data processing such as effective data-mining techniques.

Increasing complexity of the systems that individuals will want to, and be required to, use in common interaction in society means that it becomes increasingly difficult to have insight into the mechanisms and protocols that are used as well as the rights that might be infringed. One is all the more inclined to use things without understanding them or their possible adverse effects.

A prevalent culture of fear as propagated by anti-social elements of society, public media, and within politics may make individuals less likely to protect their privacy rights if they believe it will improve their personal safety.

Examinees often choose emotive language and subjective arguments in this discussion. It should be clear that in an academic context the more objective the argument the more weight it carries. Many discussions only attempted to discuss aspects of current practices, whereas the problem asks for indicators of increasing concern.

b)

Pseudo-anonymity implies the use of pseudonyms, thereby removing the direct reference to an individual, but not removing the possibility to trace the pseudonym back to the original identity

(example case – the Penet remailer penet.fi). This contrast to systems such as TOR which take reasonable measures to ensure that no one single party, or eavesdropper thereof, has enough information about the originating party to ever be able to reveal the connection. The idea of mixnets, including mechanisms of encryption and padding making eavesdropping to discover connections between forwarding agents is an important principle for anonymity. Pseudo-anonymity based systems presume trust in a single anonymising agent to do the anonymization and not reveal information. Anonymity systems presume trust in the principles of the systems themselves, but not any single actor within the system.

N.B. this is not sufficient as an answer, but a guide to help understand the grading.

References

Please note that exam answers are not (necessarily) expected to give exact references, but these notes attempt to do the reader the respect of giving references where suitable.

Pfleeger, C.P., Pfleeger, S.L. and Margulies, J., 2015. *Security in Computing.* 5th ed. Massachusets: Prentice Hall

DSV SU/KTH

IntroSec
Introduction to Information Security
Autumn Term 2015

Alan Davidson

Exam Re-sit 2016-08-16
Hints and Comments on the Marking

Suggested answers are not provided for this exam, only some pointers to help you validate your own exam answers. Do not assume that the comments here are complete examples of how the exam problem can be answered.

Problem 1

> Wikipedia (2016) defines Access Control Matrix as:
>
> "... an abstract, formal security model of protection state in computer systems, that characterises the rights of each subject with respect to every object in the system"
>
> Explain this definition further by breaking the sentence into suitable constituent parts and explaining what each part means, by all means including helpful examples. The result should be a text that could help anybody who does not entirely understand the original definition to gain better insight.

"abstract" – The Access Control Matrix expresses an idea. There may be practical implementations of that idea, but not necessarily.

"formal security model" - The language used to express the matrix follows stringent rules, at a formal level equivalent to mathematical or logical representations.

"protection state" - The purpose of the model is to describe a state in which resources are properly protected. The matrix describes only a single state of a system, and not the possible state transitions, e.g. what happens when an object is added to a system is not described.

"computer system" - The state described is assumed to express protection of any resource that has any place in a computer system. At such it could just as well describe the protection state of a network of computers as a single processor's registers (N.B. it is a mistake to assume that subjects are people and objects are files).

"that characterises the rights of each subject with respect to every object in the system" - each and every subject is mapped to every object with respect to a set of rights (which may be empty). Where subjects are processes and objects are files, typical examples of rights might include *read*, *write*, *execute*. The matrix's mapping of all such subjects to objects in respect of rights is commonly represented as a table where each column represent a single object, and each row a single subject, the intersection cell of which contains the rights for that subject over that object.

Problem

> Give examples that well illustrate how multifactor authentication is commonly used in practical authentication situations. Discuss also why this practice might generally be assumed to be an improvement over single-factor authentication.

The problem asks for examples in plural. Good answers therefore use several examples to illustrate how multifactor might involve diverse kinds of authentication methods, such as something the entity has with something the entity knows, but also other combinations such as "... is & ...knows", "where the entity is & ..has", or "...knows & ...knows".

Good discussions on the motivation might introduce relevant security principles, as well as careful

logical motivation.

Problem 3

a) Characterise and explain what protection simple traffic filtering firewalls (as described in the course literature, and as opposed to the hybrid products that can be obtained on the market under the name “firewall”) may be able to afford a computer network. Be explicit as possible on what kinds of security problems such a firewall best can protect from.

b) Give differing illustrative examples of attacks on a network that it is unlikely that such a firewall will be able to protect against. Include detailed motivation.

Some examinees assumed that the “simple traffic filtering firewalls” was referring specifically to “packet filtering firewalls”. The was not the case, but all the types covered in the course book were included. The question was therefore asking the student to give a general explanation for what firewalls do and how they do it. The interpretation of considering only packet filtering was not regarded as a very serious mistake, so long as the student could properly explain how this type of firewall works.

There is a wide range of discussions on attacks that firewalls are unlikely to be able to protect against. Some are considered to be better illustrations of the failings of firewalls than others. For example, social engineering attacks are indeed difficult to filter out, but this says little about firewalls. On the other hand, traffic being transmitted through an encrypted VPN (and through a firewall) can be a good example to illustrate the limits of the firewall concept.

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 *definition*, *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:

- One-Time Pad
- Replay Attack
- Complete Mediation
- Certificate Authority

Suggested solutions for problem 4 are not complete in this version of the document. Please see the course book and other relevant sources in the reading notes as well as the course video material for definitions and examples.

Suggestions for closely related concepts that can allow for productive discussions on the relationship:

- One-time pad - vigenére cipher
- Replay attack - eavesdropping
- Complete mediation – unvalidated input
- Certificate Authority - PGP

This does not mean to say that naming these concepts is sufficient for an answer. They must be meaningfully related to the given concepts. I suggest that they are close enough to the given concept for the examinee to be able to show deeper understanding of the given concept through their

comparison.

Problem 5

> Consider the following statement and discuss how this position can be justified, based on elements of the course material:
>
> "Spam emails are always an offence to personal rights, and therefore a security issue. Furthermore, spam is sometimes used as a precursor to a type of attack that makes them even more serious a security problem"

This problem invites the student to show their understanding of *spacial privacy* issues as well as such email threats as, for example, *phishing, web-bugs, executable attachments,* etc.

References

Please note that exam answers are not (necessarily) expected to give exact references, but these notes attempt to do the reader the respect of giving references where suitable.

Wikipedia 2016 Wikipedia, *Access Control Matrix,* Available at: https://en.wikipedia.org/wiki/Access_Control_Matrix [last accessed 2016-08-15]

DSV SU/KTH

IntroSec
Introduction to Information Security
Autumn Term 2018

Alan Davidson

Suggested Solutions to the Exam 2019-01-11 and Comments on the Marking

(First Incomplete Draft)

Some answers suggested here may be more comprehensive than would normally be expected from students in order to cover several relevant aspects of the problem.

Problem 1

> A more recent tendency in the security management of computer systems is to accept that the perimeter of the system (as in, for example, at the system's interface with the internet) cannot be perfectly protected. For this reason it makes good security sense to build systems that are resilient even to having malicious actors or malicious code within your system.
>
> Suggest and describe up to three different effective security methods or mechanisms that could assists in providing protection in the face of malicious agents acting within the outer bounds of an IT system.

The problem sets the context of system security. Of the models that we have covered during the course this might suitably remind you of Pfleegers figure 1-12, *Effects of Controls* (Pfleeger et al, p30). That would help you consider ideas such as deterrence, detection and recovery to lead you into methods and mechanisms such as encryption of sensitive data, implementing strict access control policies, IDS, Antivirus, backup, etc, all of which might be suitable to include in your answer.

Note that according to the problem description the threat is already inside the system. What kind of IT system is involved is up to the examinee to define, but we may reasonably consider it to be a network of computers. For that reason we can discount measures such as DMZs. Other possible measures that the examinee could bring up are more numerous to mention here. The quality of the answer will depend on how effective the method might be assumed to be, and how well they are described. A selection of possible subject to expand upon are (only very briefly) listed below.

Access control. For example a stringent MAC based system could be effective at segregating parts of the system, both between network nodes and within individual computers, so that a malicious agent that manages to gain low levels of privileges should be effectively contained.

Encryption of sensitive data.

Logging of activities, (including with intrusion detection systems, and intrusion prevention) can be spread throughout a system. Integrity checks on sensitive data sets (such as by means of well protected hashes) can aid in detecting if something that should not change has been changed.

If we see “the system” as a network of computers then Personal Firewalls could prevent the attack from spreading either to individual personal workstations, or conceivably even stop a threat from spreading out from a compromised station.

One could state that malware protection such as antivirus is a suitable measure, though during the course we have suggested that these might not be as effective as one might assume. That does not mean to say they cannot be valuable though.

Education of users could be cited as a measure, though that is extremely vague unless one can explain what kind of education can be expected to be effective.

A few suggested a sandbox as a possible measure. The descriptions around this suggestion were

generally vague. A sandbox, or jail, is normally used to segregate parts of a system that are possibly a threat, such as testing a new program that one is not sure about. If an attacker is within the outer bounds or your system then if you are able to isolate their activity so well so as to be able to put them into a sandbox then you might as well eradicate them. Sandboxes are therefore not a practical means of containing a malicious agent, so without clearer motivation, such answers were assumed to be somewhat confused about what sandboxes are used for.

A honeypot is a kind of sandboxed environment that is more specifically designed to preoccupy and contain a malicious agent while it is being analysed. The honeypot is a well known security measure, but suffers that same problem as a Sandbox, i.e. if one has good enough control over the agent to be able to segregate them, then they are not much of a threat. The difference with a honeypot is that it can be made to look like an attractive target so as to lure an intruder into it.

Some answers cited Saltzer and Schoeder's principles. Since the problem specifically asks for *methods and mechanisms,* citing principles is generally too vague to be called something so specific as a method. Citing principles without explaining how they would be applied to provide protection is too vague as an answer.

The most sensible interpretation of this problem text is that it is about system configuration. The best answers were therefore the ones that saw this, and presented suitable methods and mechanisms in this context. A number of answers drifted into aspect of software security, such as "apply complete mediation in all programs in the system to avoid dangers such as buffer overflows, SQL injection etc". Surely in modern systems I have little control over how all that software that one uses is actually written. I therefore had to assume that such answers showed a confusion about measure that were applied during separate parts of the course that are not so applicable to this problem.

Some wrote very generally about good authentication and strong firewalls, and other such issues that one surely can claim are more about protecting the perimeter of a system. If the threat is within the system one should possible assume that they have already bypassed such measures? There are certainly cases to be made for applying firewalls within the system to help protect different parts from each other (see *personal firewalls* above), or that implementing things like processes of continuous authentication might be able to hinder some malicious actors that already are in a system. However, I have to be suspicious of generally stated answers about authentication, firewalls, etc, assuming that the examinee did not fully understand the context of where such mechanisms are suitably applied.

One student wrote about locking a computer when you leave it. If you can describe what kind of threat that that measure is effective against, I can certainly award points for that.

On entering the exam hall several examinees asked for a clarification of the first exam problem, so in case others were labouring under misinterpretations of the problem text, I explained that "within the outer bounds of the system" means that the threat is already inside. The first paragraph of the problem sets the context that the threat is inside the system. Taken on its own the phrase "within the outer bounds" could possibly be taken to mean "a part of the outer bounds" rather than "on the inside of the system past the outer bounds". In normal English usage I would say that the second interpretation is so clearly the correct one that I did not even consider that one could twist it to mean the former. However, this could be an issue beyond the expected level of English language, so I have to consider that it could be an issue. However, given the context set in the first paragraph I can say that this is the only reasonable interpretation. I have to conclude that a student is only likely to go with the first interpretation if you find it much easier to answer about Firewalls, DMZs etc. and therefore lead ones own self to answer incorrectly.

Grading:

P Showing understanding of some ideas that could be relevant for the issue, but they are unclear on the context of the problem, vague in application, etc.

P+ Good clear answer on the most relevant of methods for protecting a system, but do not show great breadth in the examples shown.

P++ Enough clearly described methods to show a good understanding or the context and the possible solutions. Note that the “complete” in the grading criteria is not interpreted as giving an exhaustive answer, but one that is enough to show reasonable mastery of the subject and perspective within the time available.

Problem 2

> As authentication methods, both passwords and biometrics have their pros and cons. Discuss and motivate security problems that are inherent to password based authentication where biometric methods might be assumed to be free of these specific problems, and likewise discuss and motivate security problems that are inherent to biometric methods where password based authentication might be assumed to be free of those problems.

In no particular order, here is a list of some of the aspects that one might use to develop arguments. Please note that for brevity here I am only giving hints without discussion and motivation. **Discussion and motivation is however a vital part of the exam answer** to show that you understand the issues and are able to communicate them well.

You can change passwords if they are stolen or eavesdropped, but if biometrics can be replicated (such as through lifting and replicating fingerprints, as demonstrated in the *Mythbusters* clip linked in the course material), then there is no easy way to revoke the fingerprint data to stop the use of the false fingerprint.

You can forget passwords, but presumably not biometrics (well, maybe which finger was used for a fingerprint?).

Cost of implementation - It is certainly true that in general biometric methods will require more specialised hardware to work, though a camera can be used, and that is not specialised equipment for biometrics, and a keystroke measures can be used in biometrics, requiring exactly the same equipment as a password. What is more, card readers have keypads specifically for the purposes of entering a pin code, so I would be wary of assuming that every device that is part of an authentication process can be expected to already have a keyboard.

Privacy can be an issue for biometrics. Biometric data is linked to identity so one must reveal one’s identity when authenticating. There are authentication situations where it may not be necessary to be personally identified, such as having a door code for access to a building.

Passwords are subject to human tendency to choose guessable patterns. Guessable patterns could conceivably be a part of biometric data, but we may suppose that designers of those methods will have greater opportunities to avoid capturing guessable data form biological traits.

One can inadvertently leave traces or otherwise reveal some biometric data in situations outside of the authentication situation. Fingerprints can be left that could allow false facsimile fingerprints to be created. A highly detailed camera might be able to capture iris patterns without the subject deliberately presenting their eye. Though it is possible to inadvertently reveal passwords outside of the authentication process (as well as during that process) that could be put down to carelessness, whereas keeping biological attributes confidential is not a natural part of how we use our bodies.

A number of answers brought up the issue of biometrics being susceptible to false positives and false negatives. There is a case to be made for this, but when compared to passwords there is surely a case to be made that passwords can also suffer badly from the problem of false negatives. Accidentally tying the wrong password is surely a common occurrence, and a faulty keyboard can cause even a correctly typed password to by registered falsely.

If you site the fact that a finger can be cut or burned and spoil authentication, one must surely question if that is more of a problem than occasionally forgetting a password. Surely any authentication method must have an alternative authentication means in order to be able to reset?

Can it be claimed that fingerprints are unique? As Pfleeger (p62) says – there is actually no no definitive scientific scientific evidence for such a claim.

Please note that It has not been claimed on the course that voice recognition will fail if one has a cold. Nevertheless a number of answers claimed that this is a tangible problem with biometrics. In fact research on this subject says that voice recognition systems will use metrics that are not greatly effected by situations such as a having a cold (see e.g. https://whatsnext.nuance.com/customer-experience/five-common-misconceptions-voice-biometrics/). The same applies to facial recognition and claims that smiling, or growing a beard might render them inoperative. Sensible biometric methods would use data that is not sensitive to such changes. One should perhaps check one's preconceptions before using them in exam answers.

I found that a number of answers stated many facts about the respective authentication methods without regards to how they compared to the other. For example, if you state that users tend to reuse passwords for several systems, I do not know what you want to say about how that compares to biometrics. Surely biometrics will force you to re-use your bodily data to authenticate on different systems, so it is not at all clear what comparison students hoped to draw with such an argument.

Some claimed that biometric authentication could not be prone to social engineering attacks as compared to passwords. I do not see how one can be categorical in such a claim. If a social engineer can convince someone to use their BankID to transfer their savings, I am sure that a good social engineer could convince someone to use their biometric data in situations that are against their own best interests.

Pfleeger et al has quite an extensive discussion on the pros and cons of biometrics, so much material for an argument could be found there. It was however fairly common for students to repeat such arguments yet with some dubious reasoning. Though as examiner I understand where your arguments might have come from, if I found the description and motivation to be less than stringent you may well find that I have dotted your exam papers with comments that question whether you are uncritical or even mistaken in your interpretation of what Pfleeger wrote.

Problem 3

> An innocent and honest party can suffer security issues in the processes of both sending and receiving e-mails. Describe up to three such separate security issues, as well as measures that one can go to in order to mitigate each of those issues.

It is possible to include attachments to emails that are a threat to the recipient's systems, in particular if such attachments are executable files that are possible to run within the recipient's system, with the privileges of the user.

Mitigation may be through education of users that it can be extremely dangerous to save and execute attached files that come for a source that one does not have implicit trust for, so that a recipient knows what allowing an attachment to run code involves, and how to avoid letting it do so.

Emails that contain rendered html code can be a security threat. Web bugs are links to external servers that make use of the fact that if that if included images are linked to an external source for the image contents then the html renderer will normally fetch that image as the page is being rendered. The URL that is used can be a link to a server-side process that registers who requested to fetch the image, and when, thereby revealing data on those who view the email.

Html mails can also contain other kinds of harmful elements, such as links to external websites together with exhortations to click on such links. This could be the means of implementing a phishing attack, or a non-persistent XSS attack.

Problems in sending are primarily connected to inadvertently revealing information that can be to the sender's detriment. The SMTP protocol sends emails in clear, so any relaying party along the path of delivery, as well as any party who manages to eavesdrop SMTP traffic along the way, would have access to the content of emails. Mitigation – use additional mechanisms that allow for end to end encryption of email content.

Spam and Spacial privacy. Mitigation through spam filters and legal measures.

More to come later...

P++ Three well described problems with mitigation methods

P+ Less than three relevant problems and mitigation are well described (implying that less than three have been attempted, or that some part of the descriptions is vague or confused).

P An attempt to describe problems and mitigation that might mean something to a person who already understands.

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 *definition*, *relationship to [your chosen related concept]*, and *example.* **Your answers to each part should contribute to evidence of your deep understanding of the concept. Related concepts and examples should be chosen and explained with care to maximise the depth of your answers.**
>
> 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:
>
> - Self signed certificate
> - Attribute Based Access Control
> - DMZ
> - Worm

Suggested solutions for problem 4 are not complete in this version of the document. Please see the course book and other relevant sources in the reading notes for definitions and examples.

Self signed certificate

Definition

A public key certificate that has been signed with the aid of its matching private key, thereby allowing no more trust for this certificate than the simple data integrity check that an untrusted digital signature affords.

Relationship to

E.g. A CA signed certificate. (This part of the suggested answer sheet is not yet complete.)

Example

When creating a PGP asymmetric key pair the private key is automatically self signed by the PGP system. In this system one would tend not to trust any such certificate unless it has been verified by the owner through some alternative channel. Other PGP users may choose to add their signatures to the self-signed certificate, therefore bestowing it with greater validity within a so called *web of trust*.

Attribute Based Access Control

Definition

Attribute Based Access Control is an access control mechanism whereby one represents rights in terms of attributes that are may be present in either subjects or objects, or both, also allowing elements of context to define rights.

Relationship to

e.g. RBAC (This part of the suggested answer sheet is not yet complete.)

Comment

MAC and DAC are policies that define how rights should be updated when the system changes. They therefore have only a tenuous relationship with ABAC which is difficult to make sense of in the limits of this exam question. ABAC is closer to ideas of ACL, Capabilities and RBAC, so those comparisons make more sense.

Example

ABAC could be used to simulate a Bell LaPadula type of policy if the subjects and the objects are given attributes signifying their levels. A single line in an ABAC specification could then signify that subjects with Top Secret clearance may read all objects with a Top Secret label as well as all objects with a lower level.

DMZ

This part of the suggested answer sheet is not yet complete.

Definition

Relationship to

e.g.

Example

Comments

Worm

Definition

Relationship to

e.g. (This part of the suggested answer sheet is not yet complete.)

Example

Problem 5

> Explain why a set of cooperating hosts (sometimes referred to as a mixnet) can give a more reliable anonymity service than a single anonymising host is able to give when mediating network traffic. Good explanations will include explanations of relevant threats to anonymity that these services have to deal with.

References

Please note that exam answers are not (necessarily) expected to give exact references, but these suggested answers attempt to do the reader the respect of giving references where suitable.

Pfleeger fifth ed.

https://whatsnext.nuance.com/customer-experience/five-common-misconceptions-voice-biometrics/

DSV SU/KTH *IntroSec*
Introduction to Information Security
Autumn Term 2019

Alan Davidson

Suggested Solutions to the Exam 2020-01-11 and Comments on the Marking

(First Incomplete Draft)

This is a guide to help understand the grading of the exam papers. Answers should not be assumed to be complete, especially in terms of the full discussion about the concepts that shows that the students can communicate the depth of their understanding. On the other hand, some answers here might contain many aspects of possible answers and should therefor be considered to be more than may be required for a passing (or even a high grade-) answer.

Problem 1

> Despite statements to the contrary as can be found in the course book, there is good reason to say that the DES algorithm is unsuitable for continued general use.
>
> a) Give convincing, motivated arguments for why DES may now be considered unsuitable for general use. You may include several different lines of argument and thereby improve your possibilities to achieve higher grades.
>
> b) With terminology normally used to classify cryptographic algorithms, characterise what the DES algorithm was designed to be used as. Suggest an alternative algorithm that is more suitable for general use that fits these same characteristics, and explain factors that make this alternative suitable..

a) Even with a total key length of 64 bits the keyspace that is available to the DES algorithm is small in relation to the computing power of modern computers, or rather in relationship to the cost of sufficient computing power to be able to attempt a reasonable brute-force attack on such a keyspace. Furthermore, despite a length och 64 bits, only 56 bits are significant for the DES key, with a significant reduction in the keyspace as a result.

There were changes were made to the original design of the algorithm (e.g. changes to the so called “S-Boxes”) in collaboration with the NSA before it was brought into service by the American government. Reasons for these changes were not made public, and have been the cause of some public speculation. Some have suggested that the changes might have served to introduce some kind of easier route to breaking DES encryption, i.e. that this may have implemented some kind of mathematical back door. There is to date little evidence that the changes have made the algorithm weaker. Nevertheless, if one is inclined to distrust actors in such government institutions as the NSA, one might have reason to avoid algorithms in which they have had a hand, even when all aspects of the design of the algorithm are open for all to analyse.

As it became clear that key lengths longer than 56 bits would be desirable, attempts were made to adapt DES to meet this need. A solution was to apply the DES algorithm over three iterations (two iterations was shown to be insufficient, notwithstanding intuition), using two different keys. This is known as Two Key Triple DES, and gives an effective key length of around 80 bits. Using a third key (Three Key Triple DES) brings the effective key size to 112. Nevertheless, these key sizes are generally considered insufficiently strong for general usage given current computational costs.

b) DES is a symmetric encryption algorithm, meaning that the same key is used for decryption as for encryption. This is in contrast to asymmetric encryption algorithms such as RSA, where different - though mathematically related - keys are necessary in decryption and encryption. DES is a block cypher, in contrast to a stream cypher. This implies that it is less effective for encrypting long streams of data, where the loss of some data might be expected.

A suitable alternative to DES for modern usage is the symmetric encryption algorithm Advanced Encryption Algorithm (AES). This has been designed for use with key lengths of 128, 192 and 256 bits. The prime advantage to this algorithm is its longer key lengths and its relative speed with such keys when compared to the alternative Triple DES.

A surprising number of exams stated that the design of the DES algorithm was not open, and carried on to make comments on all the trust issues that this involved. This has concerned me that the issue may not have been made clear during the course, causing some oversimplifications and misunderstandings. The DES algorithm is entirely open. You can buy books with the whole algorithm in. As such all the mathematics of the encryption process are open for all to analyse and to check its validity. What may be brought into question is only some of the design choices that were made, such as how the reduction of significant bits from 64 to 56 was motivated, and whether changes in the S-Boxes might have had pure cryptanalytical motives or more hidden ones. Such questions are more in the realms of conspiracy theories, not evidence of elements of the design that are purposefully hidden. Therefore, when one cites such reasons for doubting the strengths of DES it is important to be clear.

Some answers indicated that the examinee had no understanding that symmetric encryptions algorithms are a vital element in modern cryptography. There were arguments that DES could be replaced entirely by asymmetric cryptographic algorithms. Such answers were regarded as revealing basic misconceptions that should not occur on such a course as this.

Many used the term “private key” to describe the symmetric key that DES and other symmetric encryption algorithms use. Though this has been used in the literature previously, the usage is confusing in the light of asymmetric cryptography’s clear division of the keys in a key pair to *private* and *public keys*. It is therefore common practice today to avoid the term private key in the context of symmetric algorithms, instead preferring the term *secret key*.

Grading:

P Showing understanding of the relevance of keysize to strength and How DES is lacking in this respect, and that time has brought down cost and how that effect the work factor required for brut force attacks.

P+ Identifying characteristics symmetric encryption algorithm and suggesting a suitable symmetric encryption algorithm that might reasonably replace DES.

P++ Apart from the above, greater insights such as the rumours around NSAs motives for some of the design choices and that DES is block based (as opposed to stream based). Discussions that explain why attempts to strengthen DES can be regarded ad not worth the trouble since other algorithms now surpass the effectivity and strength of DES extensions.

Note that the “complete” in the grading criteria is not interpreted as giving an exhaustive answer, but one that is enough to show reasonable mastery of the subject and perspective within the time available.

Problem 2

> Characterise what an *access control list* is in terms of where one might expect to find one, details of what we might expect it to contain, and give descriptions of policies that may suitably restrict how it may be changed over time.

An access control list will be associated with an access protected object, and contain mappings of which rights which subjects will have over that object. The association will be protected and the rights upheld by a secure control environment. For example, in a file system an access control list may be associated with an individual file object, indicating that some users might have read and

write privileges, while others might have only read privileges. An operating system will ensure that the association is securely upheld (i.e. cannot be tampered with), and uphold the privileges whenever a subject attempts access to that file. The collected access control lists for all objects in a system together comprise an implementation of the access control matrix for that system.

A policy of Discretionary Access Control allows changes to the access control list to be made by the object owner, or rather, the subject who has own privileges over that object. The intention with such a policy is ostensibly that the object's owner knows best what privileges are the most suitable.

A Mandatory Access Control policy implements system wide rules that stipulate how privileges may be changed. By this scheme, a subject might create a new object within the secure system, and as opposed to DAC, not have permission to decide on the rights for the object, but the system itself does. This could be said to be true of

Some answers seemed to equate MAC with multileveled systems such as Bell LaPadula. We looked at the MAC aspects of the Bell LaPadula model during the course, but it is a fallacy (something akin to the fallacy of *affirming the consequent)* to assume based on that that MAC is in any way limited to the scheme of Bell LaPadula.

Problem 3

> Describe situations where some malware types can be challenging for automated systems to detect, and explain why.

Situations and malware types that one might productively discuss (more than is done here for exam answers) include:

Difficulties in signature recognition, such as...

That the malware is of a type that has not yet been generally discovered, analysed and recognised. There are issues of time windows and directed vs non-directed malware attacks that the examinee might describe and explain why we can expect such things to avoid the normal process of malware signature discovery and dissemination.

The malware is of a type where its signature is known but difficult to detect, such as in polymorphic viruses (incl encrypted)

Diverse stealth methods, including

- the malware imbeds itself into the operating environment whereby the anti-malware can no longer rely on system functions correctly operating in order to find the malware, i.e. so called *rootkits*.
- the malware may attempt to directly impede the normal operation of anti-malware by for example silently killing the anti-malware's running processes.
- Through an understanding of how the software detects issues, the malware might attack the specific detection methods. For example, if all system files were run through a hash algorithm and the hashes saved, a recurring process may be run to check those hashes against the files in order to discover any illicit changes to those files. This method would require that the hashes themselves be kept securely, since if malware were able to alter the hashes themselves they could subvert the checks.
- If a method of malware detection is based on containing suspected malware and observing its effects then one could, for example, delay the effects until it is no longer under observation. Malware may be able to detect if it is being run in a virtual machine, and either attempt to cause damage to the host system by escaping the bounds of the virtual environment, or wait until such time as it is released form the virtual environment before taking further malicious action.

P++ At least two clearly described predominant malware types or situations with clear explanations as to the mechanisms of, and the difficulties this creates for, automated processes.

P+ Two or more relevant malware types or situations well described with some indication of why they create problems for automation, indicating understanding of the strategies of malware detection.

P The ability to relate some of the common difficulties that automated anti-malware software will have in detecting malware.

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 *definition, relationship to [your chosen related concept]*, and *example.* **Your answers to each part should contribute to evidence of your deep understanding of the concept. Related concepts and examples should be chosen and explained with care to maximise the depth of your answers.**
>
> 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:
>
> - Nonrepudiation
> - Digital Rights Management
> - Shibboleth (in the context of a name of a system)
> - Buffer Overflow

Suggested solutions for problem 4 are not complete in this version of the document. Please see the course book and other relevant sources in the reading notes for definitions and examples.

Nonrepudiation

Definition

Nonrepudiation is a service that a secure system may implement that links an entity to an action such that in the face of a denial that the entity took that action, the system can provide evidence that the entity did in fact commit to that action. This may be implemented through various different methods, such as signing with a private key that is only available to the signer and verified with at trusted public key.

Relationship to Integrity

Of the three basic security services (or *properties* as the course book classifies them) *confidentiality*, *integrity* and *availability*, nonrepudiation is most closely associated with integrity, in that it is a service that proves the voracity of records of actions within the system, and therefore represents the truth of data in relationship with “real world” events. In some summaries of basic security services texts will include nonrepudiation as a service in its own right, thereby highlighting it as a kind of integrity that can be seen as distinct from the normal interpretation. For example, ISO 7498-2 adds *nonrepudiation* and *authentication*.

Example

A bank customer uses an authentication token containing a private key that only the customer has access to in order to sign a bank transaction moving funds to a third party. The bank customer then contacts the bank, fraudulently claiming that funds are missing from their account from what must have been a bank error. The customer thereby repudiates that they were responsible for the transfer

of funds. The bank has the possibility to refute the customer's claims through the nonrepudiation service offered by verifying the customer's signature on the transaction with the corresponding public key.

Comment

A number of answers made the erroneous assumption that nonrepudiation is about not being able to refute "sending things". The concept is broader than that.

Digital Rights Management

Definition

Digital Rights Management denotes a class of mechanisms which are designed to limit privileges over a digital material for those who are not the originators of that material.

Relationship to Mandatory Access Control

Whereas Mandatory Access Control limits rights according to rules that are set by the policy of a contained secure system, DRM involves controlling what is possible to do with a material even in outside of the immediate control of the material originators.

Example

Whereas normal PDF files may in practice be copied and manipulated as the owner of the file sees fit, some e-book formats can only be read by specific software. Even after having paid for the right to read such an e-book, the originating organisation can control the specialised software to disallow copying of the text, or even limit it to only be readable within a limited time period.

Comment

Some students who presumably were not aware of the term tried to interpret it literally and made the assumption that it is something to do with access control in general. Sometimes you cannot understand a term simply by picking it apart, but need to have picked up on its usage.

Shibboleth

Definition

Shibboleth is a Single Sign On scheme for web based service.

Relationship to Kerberos

Kerberos is a well known system for Single Sign On for several services over a network. The principles of authentication and ticket granting are very similar for Shibboleth as for Kerberos. Many types of networked services may be adapted to work according to Kerberos' access control, but Shibboleth is for web based services. Shibboleth tickets are kept as cookies local to a user's web browser.

Example

At DSV one only needs to authenticate once, and then has access to Daisy, iLearn2, scipro, play, the department wiki, among other services for as long as relevant Shibboleth tickets are valid.

Comments

Buffer Overflow

Definition

Buffer overflow is a name used for both a type of vulnerability and an associated type of attack. As a vulnerability it is the result of improper checks on input data that is being written to primary memory were that data can be written beyond the bounds of memory allocated for that data. This can cause effects unintended by the programmer, most dangerous of which can be that program control is altered through the way that unprotected memory is manipulated, even giving an attacker full control over the future of the attacked process.

Relationship to Injection Attacks

e.g.

Injection attacks are also an issue resulting from improper checks on input data. In a buffer overflow attack the input is not properly restricted when being written to memory, most usually due to a programmer not exercising proper boundary checks when using programming languages that do not do such checks automatically. In injection attacks the problem is the nature of the input data even when it is properly restricted to input buffers. The injection attack problem lies in where that input data is used to build expressions that are then in some form processed by an interpreter. The programming error is to not properly check and control the content of the input data where malicious input can allow the attacker to cause unexpected outcomes from that code interpretation.

Example

If an execution stack is not suitably protected then overflowing data in a locally allocated buffer might result in the return address value being overwritten by data supplied as input to the program. It may then be possible to carefully redirect the program execution to a part of memory that has been prepared with alternative code, such as a buffer where the program has previously allowed data from outside the controlled to be input. That code might simply open a new command shell, allowing the perpetrator of the buffer overflow attack to introduce any instructions they please, and that will then be executed with a privilege level equalling that of the compromised process.

Comment

Unfortunately many answers held common misconceptions, such as that buffer overflows are about overloading resources in general, or that buffer overflows will necessarily cause a DoS (whether if be be assuming it overloads resources or by correctly understanding that it corrupts memory). It is surely important to understand that buffer overflow vulnerabilities and attacks can cause far more damage to a system besides just halting it.

Problem 5

> One of the important security functions of a multi-user operating system is to authenticate users before granting access. Suggest and motivate what 3 design principles for security will have the greatest influence on the security of the system's authentication mechanism.

The primary design principles studied during the course are Saltzer and Schroeder’s Design Principles for Security Software, and these would be the obvious central point to base arguments.

Due to time pressures, in this version of this document I will not give entire suggested answers but allude to how such answers might be motivated. Real exam answers should have more discussion and motivation.

Based on aspects of authentication that the course has touched upon I suggest that the clearest

arguments would include the principles of Separation of Privilege and Psychological Acceptability

- Principle of Separation of Privilege (even though the name may lead us to associate with access control rather than authentication) can be related to the well established principle of multi-factor authentication.
- Principle of Psychological Acceptability – given that we know that passwords are an important authentication method, and that they are weak in that people create and handle them poorly we can use this as evidence that psychological acceptability is important for an authentication method's strength.

Beyond these two more obvious matches, several of the remaining ones could be motivated, and the answer is graded according to the strength of the discussion, in how it shows good understanding of the concept of authentication and of secure design principles (whether they be from Saltzer and Schroeder or not)

- Fail Safe Defaults – As a critical security process it is important that if authentication should fail, it must do so securely. The default situation is suitably not to authenticate until identity has been proved, and not that identity is assumed and disproved.
- Economy of mechanism – As a critical security process it is important that authentication have qualities normally associated with simpler mechanisms, such as readability, small attack surface, etc.
- Complete mediation – All authentication attempts should be processed through the authentication process.

I have myself not managed to find convincing arguments for an important role for

- Least Privilege – Since authentication reasonably has to be a highly privileged process
- Least common mechanism – Problems of sharing resources has not historically been a problem in authentication failures.
- Open design – hiding secrets in the design of authentication mechanisms would be ill advised, but it has not been a tendency in authentication processes, suggesting that it is not a problem that warrants special attention.

In the grading I was forgiving of not being able to make the connection between our well established secure design principles and the concept of authentication. Even answers that showed a good understanding of the principles, or else a good understanding of the requirements of authentication mechanisms were allowed a pass.

Making any logical connection between principles and the real issues of authentication would

DSV SU/KTH *IntroSec*
Introduction to Information Security
Autumn Term 2019

Alan Davidson

Suggested Solutions to the Exam 2020-06-04 and Comments on the Marking

(First Draft)

This is a guide to help understand the grading of the exam papers. Answers should not be assumed to be complete, especially in terms of the full discussion about the concepts that shows that the students can communicate the depth of their understanding. On the other hand, some answers here might contain many aspects of possible answers and should therefore be considered to be more than may be required for a passing (or even a high grade-) answer.

Problem 1

> The exam answer that you are currently addressing may be viewed as an object in a secure system, where other relevant entities in the secure system that is this examination may be *you* (as owner of the exam answer), *the examiner*, and *any other student*.
>
> Clearly illustrate, with brief motivations, what access rights should apply to such relevant entities in order to uphold the integrity of the exam situation
>
> > a.at a point in time when you are still writing your answer.
> >
> > b.at a point in time when the problem is still current (i.e. the problem text has been released and the hand-in deadline has not yet been reached), and you have been permitted to take an unsupervised break.
>
> For each of these two scenarios, discuss how an IT security mechanism that this examination applies is being used to help enforce the policy that those rights describe.
>
> Discuss, with motivations, which types of access control policies the examination system should apply in order to determine who may set the access rights over exam answers
>
> > c.during the examination.
> >
> > d.once the exam is completed.

a)

One reasonable ACM to model the situation might be:

	The examinee´s answer	The examinee	Student x
The examinee	own, read, write	self	-
The examiner	read[1]	monitor	monitor
Student x	-	-	self

In this ACM absence of specific rights is assumed to mean no rights, i.e. the default is no rights.

One mechanism to ensure the examinee does not read another student's answer is that all that the student is able to read on their computer screen is monitored via screen sharing with the examiner and via the camera view that includes the screen. Any attempt to violate the policy by sharing your screen with another student can thus be discovered, and presumably punished.

b)

To illustrate the freedom that is allowed during a break we might introduce another entity that has previously had no rights within the system and was therefore not included – Person Y

1 Note that the policy of allowing the examiner to read the examinee's answer during the exam may be problematic from an anonymity point of view, i.e. watching a student write an exam answer links that answer to the examinee. One might prefer to say that this right only comes into play once the exam is completed and has been anonymised.

	The examinee´s answer	The examinee	Student x	Person y
The examinee	own, read	self	-	interact with
The examiner	read	-	-	-
Student x	-	-	self	-
Person y	read	interact with	-	self

For this system state, it is assumed that student is themselves not on a break.

Note how, in the policy that this ACM describes, it is assumed that anyone may read the examinees answer, except for a student not on a break. In practice this will be dependent on other real world factors, such as whether the student wishes someone else to read their answer. But that is not what this situation is trying to model. For the sake of the exam the policy, reading is an allowed privilege.

Note – This is clearly not “the correct answer” that any student would ever be expected to repeat. There are many assumptions and vagaries behind this solution that could be discussed ad infinitum, and motivate any number of variations. For example, one can well question why can the examiner not “interact with” all other parties. The answer nevertheless shows how a student can answer in a manner to show their understanding of both the situation and the ACM model. For example, just using the “interact with” privilege shows that one understands well that ACMs can be used for any set of meaningful privileges, not just those associated with file systems, such as read, write, execute, append.

In a deeper analysis a number of entities might be included besides the ones mentioned. Some students began discussing video recordings as entities, which is a legitimate part of this exam, but including too much made the discussion difficult. Another reasonable entity to include might be exam invigilator. The question did not address any difference between examiner and invigilator just because there were only teaching staff invigilating. In other exams that difference might be more prevalent, and may be more clearly an important distinction, largely when trying to uphold exam anonymity for the examiner. For exam answers confusion between the role of examiner and invigilator was expected, and was not emphasised while judging exam answers.

As an exam answer a clear way to represent the rights is as an access control matrix, where praxis is subjects in the rows and objects in the columns. Note that the obvious object is the exam answer, though insightful answers would show an understanding that other entities can be both objects and subjects.

A number of student answers included execute rights. Compared to rights like read, write, and even append, execute is unclear in this context. without making it clear what they meant that this right would entail I have to assume that students had a naïve concept of what access control is about, assuming it to be more or less exactly that they might expect to see in a file management system. As made clear during the course, students are expected not to get stuck on the idea that access control is just about files.

c)

During the exam, access the student’s answers should be carefully managed to uphold the secrecy of the answer. In terms of ownership, it makes sense to say that the students own their exam answers (as in the ACM from part a). They are after all the creators of them, and in a legal sense retain the copyright of those answers. Nevertheless, the exam situation stipulates what rights there are on objects irrespective of who the owner is, so that would indicate that Mandatory Access Control (MAC) is suitably applied during the exam.

d)

Once the exam is completed, the integrity of the system requires that the exam answers should not be altered. However, the issue of secrecy has changed. The answer must be readable by the

examiner, but beyond that, whether it remains secret or not to other entities is no longer the concern of the examination system. The answers belong to the student (i.e. the student still has own rights) and it is reasonable to allow the student to decide about whether others may read the answer or not. This would indicate that the system will have to continue to have some MAC influence to uphold integrity, whereas for the confidentiality side, Discretionary Access Control (DAC) is a reasonable policy, allowing the student to make decisions about whether others may read the answers or not. On the whole, it is therefore reasonable to allow a mixed MAC and DAC policy.

A number of students seemed keen to introduce the concept of Role Based Access Control for this discussion. During the course we have proposed that RBAC is basically a simple, practical mapping of entities to roles, whereby the ACM can include roles rather than subjects. Several students brought up RBAC as a kind of alternative to MAC and DAC. A case can be made for this, but it is one we avoid during the course since the concepts become unhappily tangled at that point. All student answers that included RBAC were poorly motivated, indicating that students may well be using a concept that they are not well versed with in an attempt to convince the examiner that they are able to bring up a concept of relevance, even if they do not really understand it, or the problem, themselves.

As with parts *a* and *b*, these answers to *c* and *d* should not be considered the only correct way to analyse the situation. Answers may legitimately vary based on, for example, how one interprets "the exam answer". The suggested answer assumes that the concept is in terms on the semantic content of an idea, rather than necessarily being e.g. the actual characters of an answer as entered into iLearn.

P The answer shows the ability to correctly use terms and models of access control within the context of this problem scenario.

P+ The answer shows the ability to give motivations that are consistent with the terms and concepts used.

P++ The answer shows the ability to give well-founded motivations for terms and concepts in all parts of the problem set and suggests the ability to apply the concepts correctly in more general contexts.

Problem 2

As stated in the course book:

"Stream Ciphers encrypt one bit or one byte at a time, block ciphers encrypt a fixed number of bits as a single chunk" (Pfleeger et al. 2015, p94).

a.Discuss relative advantages and disadvantages that generally speaking differentiate stream ciphers from block ciphers.

b.Discuss what limitations this is likely to put on the basic cipher types that can be used for stream ciphers as compared to block ciphers, and whether that can tell us anything about their likely comparative strength.

.

a)

For elements of this part of the answer, see Table 2-10, p 95, in the course book.

b)

The basic cipher types that this problems alludes to are *transpositional* and *substitutional* ciphers. We may assume that a goal of stream ciphers is that you can quickly encipher the digital bits "on the fly", i.e. with low latency. It will therefore be difficult to apply transpositional ciphers which would require the cipher to wait to gather a block of bits before they can be transposed, or "shuffled about". So if we only rely on substitutional ciphers for stream, that could indicate that for

algorithms of equivalent complexity (whatever that might mean!) the lack of being able to use the basic principle of transposition should mean that the cipher will be easier to crack. A cryptanalyst will know that the order the encrypted data has is the same as the order of the original cleartext. On the other hand, we have seen that a substitutional cipher can in principle be uncrackable in itself, without the need of transposition – the one time pad cipher shows us that. The upshot is that at the level of understanding required for this course, we can reason in this manner, but it is difficult to reach a conclusive answer on relative strengths of stream vs block ciphers.

P The answers shows the ability to correctly relate the relevant parts from the course book.

P+ The answer shows the ability to correctly relate the issues to show an understanding of transpositional and substitutional ciphers.

P++ The answer shows the ability to relate ideas of algorithm strength to principles studied during the course.

You will note that the P+ grading criteria is dependent on the student managing to interpret "basic cipher types" as being the transpositional and substitutional ones that we have studied during the course. It is surely reasonable to think that if you have understood the course material on cryptography then your mind would go to these. I nevertheless graded leniently where the connection was not immediately apparent to students. For example, some students assumed that the basic cipher types under discussion would be symmetrical and asymmetrical cryptosystems. With good understanding it should be clear that these concepts have little to do with the problems of block vs stream ciphers. We are after all not discussing key-distribution issues here, which would be a lead into discussing symmetrical vs asymmetrical. Nevertheless, bringing up symmetrical and asymmetrical cryptosystems in this context generally (and unsurprisingly) led to confused and confusing answers, so if a student took that route, it was difficult to achieve a P+, even if the had the ability to show they understood those concepts.

Some students discussed how key lengths were limited in stream ciphers. This comes form mistaken assumptions about how keys are used in stream ciphers. Keyspaces are in practice not limited in the way that many students claimed. Nevertheless, details of stream cipher algorithms are beyond the material covered on this course, so it is understandable that one might jump to such assumptions. Answers that brought in key-length issues were therefore graded leniently even though they were strictly incorrect.

Problem 3

> Two of Saltzer and Schroeder's security design principles are in the course book called *Complete Mediation* and *Ease of Use* (Pfleeger et al. 2015, p217) (in other texts known under the name of *Psychological Acceptability*).
>
> Give concrete IT security examples and discuss how careful application of each of these two principles may be expected to improve system security. Structure your answer according to the effects we can see on each of the security goals as represented by the CIA triad. In your discussion contrast with explanations of how poor adherence to the spirit of each of these two principles will likely result in worse security in the case of at least one of the CIA goals.

Complete mediation implies that all access to a resource are checked in order to ensure that they are allowed. As a concrete example, all accesses to objects in the iLearn system should follow this principle. In some complex situations this may be computationally expensive, such as when submitting an assignment. The system must check that you are allowed accesses to the course, to the assignment, that you are a member of the relevant group, etc. In order to save on computational resources a system might skimp on the principle, for example by assuming that if a previous operation was permitted, then so should the next. The problem with such a strategy is that factors in the system and environment that may be outside the control of the system can effect whether the operation should be permitted.

Consider a situation where a user uploads a group assignment, and then changes group. Assuming that the user may upload a new version based on the fact that they uploaded the previous one would be a break any integrity requirement that only group members may upload files. Assuming that uploading a version of a document should allow the student to view any versions that other group members overwrite the previous version with would mean that the student could view hand-ins from the group that they no longer belong to. This may violate confidentiality requirements that different groups' hand-ins should not be viewable across student group boundaries.

Comment: The issue of how complete mediation supports availability is less easy for students to discuss. Given the definition of availability as "the ability of a system to ensure that an asset can be used by any authorized parties" (Pfleeger et al. 2015, p7) we can understand that it is more about robust services that make sure that the asset is there when it is needed, rather than an issue of access control. This is something that I find Introsec students are often confused about, so I am somewhat resigned to the idea that only top-grade students are likely to be able to make the distinction, though I often try to highlight it in class. So having said, any answers that were not directly questionable when discussing availability were judged to be sufficient. Strong answers might explicitly state that complete mediation can be expected to have little effect on availability for the above stated reason. On the other hand, a case may be made for availability being reliant on access control mechanisms that properly support the policy so that things that should be allowed according to the policy are not erroneously blocked. Such an argument might be as follows.

If the student has joined a new group then a policy should reasonably allow that student to read any group work that has been uploaded to iLearn hand-in boxes for that group. A mechanism with "lazy" mediation might work by caching which group the user belonged to at the beginning of the session, and link all future operations to that group. This could fail if the user were to change group in mid session, and thereby erroneously hinder the availability to the new group's uploaded material. With complete mediation the system would check all factors relevant the access control policy at the time of the access request.

Ease of use can cover several aspects of a security mechanism. Clearly if a mechanism has a poorly designed user interface it may lack ease of use. Underlying architectural design and implementation might also make mechanisms less than easy to use such as if it should cause appreciable delay in the normal operation of the system. My example will be based on this latter idea.

A firewall might conduct extensive checks on network traffic. This may for example require complete mediation so that each and every IP package is checked to ensure that it follow the policy for access to a network. Moreover, since network threats may be complex, the checks on the network traffic may require computationally complex operations to be able to ascertain if it is policy compliant or not. All this computation may slow the network traffic, and should the firewall also be sharing the resources of a device, such as in a personal firewall, it may also noticeably hinder other processes from making full use of the computational resources. The result may be that the firewall itself becomes a tangible threat to availability. A user who is need of fast network response times and dedicated resources to their primary tasks (i.e. the thing they seek to achieve in the first hand, where security is not the foremost demand) may seek to bypass the firewall mechanism for some operations, or indeed disable the firewall entirely.

If cryptographic operations are easy to use then a user may be encouraged to use them in order to ensure that messages are encrypted for the user, thereby supporting confidentiality. If it is easy t sign emails then a user may allow the operation to be completed automatically on all emails, therefore ensuring the integrity of both the content and that it comes from the originator. Without such checks, email protocols largely allow for insecure communications. At present, relatively few email users use cryptography for emails, indicating that the adoption and operation of cryptography tools is largely deemed not to be worth the effort.

P The answer show understanding of the principles, though unclear or confused in some minor aspects.

P+ Good understanding of the principles exhibited, including relevant examples.

P++ Clear perspective on the relationship to CIA principles, and illustrative examples.

Problem 4

> Define 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 *definition*, *relationship to [your chosen related concept]*, and *example.* **Your answers to each part should contribute to evidence of your deep understanding of the concept. Related concepts and examples should be chosen and explained with care to maximise the depth of your answers.**
>
> Please note that in general a 50% complete answer will be required to obtain a pass mark for this problem
>
> - Continuous authentication
> - Rootkit
> - Asymmetric cryptosystems
> - Informational Self Determination

Suggested solutions for problem 4 are not complete in this version of the document. Please see the course book and other relevant sources in the reading notes for definitions and examples.

Continuous authentication

See Pfleeger et al. 2015, p245

Possibly fruitful related concept to discuss: *Session based authentication* – though as a concept this may be beyond the average Introsec student. *Password based authentication* could also be fruitful when contrasted with *continuous authentication*, giving the chance to bring in concepts like *session highjacking.*

As example one might describe a web based session where each http request from the client to the server might be automatically signed by the client software.

Rootkit

See Pfleeger et al. 2015, p170, p334, etc.

Possibly fruitful related concept to discuss: *Antivirus protection,* and thereby discuss the special difficulties involved in detection of malware the run with system privileges.

The “Sony rootkit” might be an example, assuming that the student is able to describe a little more of the circumstances than just giving the name.

Asymmetric cryptosystems

See Pfleeger et al. 2015, p89

Possibly fruitful related concept to discuss – well too obvious really: *Symmetric cryptosystems*

As example one could briefly illustrate how the keys of asymmetric cryptosystem’s key-pairs are used in encryption and signing.

Informational Self Determination

See course video Introduction to Privacy.

Possibly fruitful related concept to discuss – also pretty obvious: *Spacial privacy*

As example one might take a privacy enhancing tool such as those that limit the sharing of information through cookies, and show how it supports Informational self determination.

Comment: Not many students manage to get full marks for these questions, which I may take as an indication that many take their understanding of relevant terms for granted, until they are tested on a little more depth to that understanding. I do not think it should be difficult to get a passing grade. One thing that you have to do is follow the instructions, and fill in each of the parts with a short a text as you can manage that shows that you have a sensible perspective over the term. One recurring problem is that students do not describe the given concept's **relationship** to a chosen term, as the concept required. If you simply describe another term that does not help. The chosen term must be used to show your deeper understanding of the given term.

Problem 5

> It is an examiner's obligation to ensure that students reading a university course meet the intended learning outcomes as set out for that course. In the unusual situation that the current Covid-19 pandemic puts the university in, an examiner's obligations have not changed, but we may assume that the amount of work involved in meeting those obligations has multiplied to beyond whatever resources the university makes available to the examiner.
>
> Discuss how applying at least two different, well established theories of ethical principles to the above described situation might result in differing use of IT security systems in order to assist the examiner.
>
> Higher grades may be achieved either by greater analytical depth of the situation and its consequences, or by discussing the outcomes for more than two theories.

See Pfleeger et al. 2015, pp746-756, for the established theories of ethical principles that were studied during the course, and exercises on their application. Other theories and principles were of course acceptable, just so long as the examiner could ascertain that they could reasonably be classed as "well established".

In the following examples of how one might structure arguments I become aware that it is difficult to separate the experience and perspective of the examiner. Note however that it is not the conclusions that are brought that the problem is graded upon, but the ability to reason outside of personally established patters and according to generally accepted principles, so these are examples of bringing different ethical perspectives into a line of reasoning.

The telelogical utilitarian viewpoint would cause one to consider the balance of benefit for all people. The benefit of exams for a society is that anyone should be able to rely on the fact that a student who has passed a course has attained the intended learning outcomes for that course. Trust in bodies of higher education, and of the educational role of universities, and of individual student's level of achievement are all bound to the accuracy of the test of learning that the exam involves. Even if the examiner may assume that it is unlikely that anyone would cheat on such a test, society's trust would require that it is *seen* that the university takes every reasonable precaution to avoid the possibility of defrauding the test, thereby assuring society and students that their grade is fairly and equally judged. Contrast these factors against the negatives involved in an examination situation, for example, the possible inaccuracies of the test instrument when examinees may be placed in an unusually stressful situation, that the possibilities of cheating mean that the system should force a situation where individuals suffer privacy intrusions by the monitoring of their actions. Furthermore the methods to achieve such a trustworthy test may not be established or readily at hand for the examiner, so great effort may be required, even beyond what recompense the university gives for that effort. Therefore, with the utilitarian viewpoint one might surmise that the benefit to society of an extensive exam procedure should outweigh the transient discomfort of both students and examiner. Wherever the examiner can see a situation where the exceptional limitations brought on by a pandemic might allow any lesser trustworthiness in the process than in well established normal

examination situations that society has come to accept as fair, the examiner should make every effort to establish new methods that compensate for the extra difficulties. IT systems should be used as far as possible to ensure that the possibility of cheating on the exam is reduced. IT systems should be employed to provide evidence that no student may access more information (including interaction with any individuals beyond trusted exam invigilation staff) that would be available to them in the exam hall, by limiting and surveilling their actions. Any situation where students necessarily are in a position to access more information (such as during mandated breaks), the examiner should ensure that situations are controlled in a manner where it is not possible to affect the validity of the test, such as through access control over parts of the test over time. Checks must be conducted to ensure that the student is properly authenticated as the same student that will receive the grade for their exam submission, and that the exam submissions are authenticated as belonging to that student. Login authentication would be insufficient given that login credentials can be passed to another party, or following a login and examination session could be passed over to another party. Some form of continuous authentication for the examination session should therefore be employed. Authentication data should be difficult to falsify, even with the willing participation of the student. A live stream of the student's face while verifiably active in participating in the exam may be employed, so long as this can be achieved in accordance with personal privacy legislation such as society has mandated. Mandatory access control mechanisms should be applied to student submissions to reasonably ensure that they are authentic and confidential.

With an egoistic viewpoint, the examiner may balance the immediate discomfort of adapting to a new situation with the projected repercussions from different parties. The examiner would not be directly concerned with society's expectations of higher education in general nor the university itself, but assume that such matters are the responsibly of leadership. Negative recriminations from employers are unlikely so long as the examiner acts according to the norms that he/she observes being employed by their peers. Copying the method that requires the least effort that has not resulted in any recriminations from leadership would be commensurate with egoistic principles. The examiner may thereby spend all the more time and effort on other activities that result in greater personal gratification or gain. Inasmuch as leadership actions are contingent upon the influence of students bodies then the examiner may consider the student's reaction to the examination process and how it impacts the examiner's personal interests. Even personal relationships with students may be a factor that effects the examiner. One may gain from being liked, or from one's courses being popular. Such factors may suggest that the examiner should give as minimal checks on learning outcomes as can go ostensibly un-criticised. IT security systems that demand any effort on the part of the examiner would not be employed. Existing norms and systems might be employed such as submission of examination through iLearn, relying on the standard authentication and access control that that system supplies. Extra confidentiality services might be employed by the examiner himself/ herself so long as the examiner judges that it may be to his/her own detriment if his/her motives were revealed. This might be by obscuring what other activities the examiner might be occupied with at such times as the leadership and students might otherwise assume that they should be busy with conducting a reliable examination.

With an act-deontological[2] perspective the examiner might be tasked with balancing the assumed right that the examination must uphold all principles that they envisage good examination should, balanced against the right that the effort involved to adapt to the new situation should not be asked of the individual employee without concessions either to the demands set, or of the recompensation offered. This might mean that the examiner would either implement the best and fairest exam that they are able to devise with the aid of all IT security mechanisms at their disposal (similar to the utilitarian case). Alternatively, if the contrasting right is deemed the greater, i.e. that the employer is in the wrong in setting unreasonable requirements upon their employees, the examiner might refuse to put their name to any exam that he/she would regard as not meeting professional ethical standards, accepting possible repercussions such as being let go or required to hand in his/her

2 i.e. the individual rule-based theory as Pfleeger et al. describe – the term is used e.g. on page 758, but seemingly not explicitly defined in the theoretical part of that text.

notice. Refusing to give the exam would of course result in no IT security mechanisms being employed for that purpose.

With a rule-deontological perspective, the examiner may look to factors that belong to presumed universal truths such as fidelity, justice, doing no harm, self-improvement, making reparations etc. Since factors of justice and fidelity would indicate that the exam must be fair across all students and must be just, this would once again indicate that every reasonable effort should be made to ensure that the grades given are fair and that no-one student who contravenes those principles can be awarded a too high grade as a result of fraudulent actions. On the other hand, the students' case may be considered in that the intrusion that IT security systems involve might bring harm to them. Privacy rights may be encroached upon, such as viewing things in a students home that have no bearing on the legitimacy of the exam, but that the student would prefer not to have revealed. The student may suffer by means of the effort involved in meeting extra exam requirement whereas that extra is not accounted for in the goals, allotted time, synopsis etc. for that course. Meanwhile, a goal of self-improvement might suggest that a situation that brings new challenges should be faced with efforts to find new and innovative solutions. Testing ideas and learning new lessons might be important enough to take some risks, so long as the examiner is in a position to make reparations for any unintended ills that may occur. This perspective might therefore result in the examiner making use of all their understanding of it security mechanisms to implement a new equivalent examination format, that together make as good a match with the policy as they can conceive. This might involve considering surveillance methods that are as closely matched to the goals of examination as they can manage (within a limited budget). Familiar IT systems may be used so that the student is relatively well prepared for the exam situation and should be assisted as far as possible in avoiding accidental release of private information that is irrelevant for the exam.

Reference

Pfleeger et al. 2015 Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies. Security in Computing, 5 edn., Upper Saddle River, NJ: Prentice Hall. 2015