Assignment 2

General Instructions

There are three deadlines for this assignment (following group registration):

1. Task choice – 09:00, 2023-11-30
2. Task completion – 09:00, 2023-12-11
3. Peer review – 17:00, 2023-01-12

and compulsory elements:

1. Meetings with both a reviewed and a reviewing group.

Task choice phase

First register your group in the Assignment Two Groups activity in iLearn. Many of you will no doubt choose to stay in the same group as assignment 1, but you must nevertheless register anew in this activity, and may like to try to grab the same group number as you had for assignment 1. If you find that you have different ambitions for assignment 2, or have found that your cooperation in assignment 1 has not worked as well as you might have hoped, this is an opportunity to now find another partner, though be aware that time to do so is very short this time. If you do split from your assignment 1 partner you are expected you to keep that partner fully informed of your actions, so as to avoid confusion about who they will work with on assignment 2.

For this project you are to complete one out of five tasks as described below. First view the task descriptions and by all means do preliminary investigations into what kind of documentation you can find to help you complete the task. If you have previous experience with any of the tasks mentioned in this document, you are encouraged to choose something outside of your comfort zone. Make a choice for which task your group prefers to complete. If a large number of groups choose the same task the staff may be obliged to limit the numbers of registrations for that task, and even refuse those groups who were among the last to register for that task. If you have read the course *Säk2* you may not choose to complete the GPG task since it closely resembles topics already covered in that course.

After the deadline for task hand-in a secondary phase with a peer review will be initiated. You will then be reviewing another group's work, and you may not review the same kind of task as you have completed. You may suggest to the course staff which of these tasks you would prefer to review, which the staff may take into account when matching groups for the review stage, but we reserve the right to match according to other factors. Säk2 students may not choose to review the GPG task.

Only one group member from each group shall make your choice on behalf of both group members via the Assignment 2 task choice activity in the course page in iLearn.

Task completion phase

Task completion is expected to take one and a half days of study for each group member.

Work is to be completed in pairs. This entails that group members shall collaborate closely on all tasks. Only minor practical elements of the projects may be delegated between group members, and then only where both members are assumed to be equally able to complete and to report the delegated tasks. Besides minimal advice, groups are not permitted to assist other groups in the completion of these tasks.

Each group is to document their task in Portable Document Format (pdf). All group members' names must be included on a separate header page at the front of the hand-in document. If you were originally registered for the course **before the Autumn term 2023** then make it clear on this header page which term you were originally registered.

Your documentation shall concern the experiment that you have designed (in the case of tasks 3, 4, and 5) and completed. You can assume that the reader understands the subject matter, so do not include any unnecessary filling in your documentation. Just motivate and describe your actions so that the reader can understand what is required to repeat the experiment, and why it should be done that way. Include a *Time Summary* section a general estimation of what parts of your task took how long to complete in terms of person hours. The report must also contain a C*onclusions* section where the group discusses what security lessons may be learned by completing this experiment. The report is to be written as concisely as possible and with correct and appropriate language.

The report must include proper academic referencing, meaning that statements and quotations not directly attributable to the authors themselves must be supported by an indexed reference list that clearly shows the source. No text generated by AI tools may be a part of your hand-in. If there are any signs that the authors have included ideas, text or illustrations that are derived from other sources than their own mental processes without proper attribution, their work will be subject to investigation according to the university disciplinary proceedings. To avoid any doubt in what such referencing entails students are encouraged to review the Wikipedia page https://en.wikipedia.org/wiki/Citation.

In summary, your document will contain a header page, then a body with sub-headings:

1. Design (for tasks Anonymisation, Metasploitable and Social Engineering)
2. Experiment results – where your steps are discussed, not just stated.
3. Time Summary
4. Conclusions – The group's reflections on what has been learned from the experiment as a whole.
5. References – A list a sources that have been cited in the main body of the text.

Submit the group's assignment solution to *Assignment 2 Hand-in* in the course page in iLearn. Any additional files that are needed by the examiner in order to verify that your work meets the assignment requirements must also be a part of your hand-in. Submit such files separately, not as e.g. compressed packages.

All group members must confirm the group's submission in the relevant iLearn submission activity. Only confirm once you are sure that the submitted files are those that the group members agree are a true representation of the group's work.

Peer review phase

For the review stage groups must be prepared to communicate and cooperate with their reviewer group in order to ensure that they understand both the task and your group's solution. Each group will therefore have at least two discussions: One with the group whose work they are reviewing and one with the group who is reviewing their work. Students may come to an agreement between themselves that this meeting may be online rather than the default expectation of it being conducted in the Kista campus during normal study hours. Both members of each group are required to be present for the estimated 30 minutes during which the task you are reviewing will be summarised. Only in the very rarest and unavoidable occasions can dispensation to be absent from this session be granted, which requires that the student several days beforehand gains the examiner's written consent to be absent.

The review process in its entirety (i.e. studying the task description, studying the reviewed group's documentation, attending course staff run-through of the requirements, meeting groups, and documenting) is expected to take one and a half days of study for each group member.

Further information, including more detailed support for the review process will be published after the completion phase deadline.

The Experimentation Environment

Those who complete the AppArmor exercise will need to have administrator privileges on a Linux system. It is therefore assumed that you will use the virtual Ubuntu system in the environment you have used for assignment 1.

The GPG exercise does not require the virtual machines that you have used for assignment 1. You may choose to install GPG on your own systems, but it is already installed on the Linux systems that all students have accounts for and can be accessed from the computer rooms on the second floor.

The Metasploitable task must be conducted in the same environment that you have been using for assignment one since that is where you will find a copy of the prepared vulnerable machine.

For the anonymity tools experiment you must use the virtual environment from assignment 1. This is because the university regulations do not allow students to anonymise their traffic. The network traffic from the provided virtual environment is kept separate from the normal university networks, instead being diverted though our security laboratory network. Nevertheless take heed of the instructions to report the tools you will be using and when, since the central university network monitoring may otherwise take your traffic to be contrary to the university network usage policy.

The social engineering experiment makes no special requirements of any computer environment used. Note however that ethical considerations are the primary limitation to what you may do.

Grading criteria

The general grading criteria as specified in the “Course Goals and Criteria” document apply to this assignment. Students are advised to re-read the section “Assignments” before commencing work and before submitting.

The grades for assignment 2 are dependent on both completion and review phase hand-ins.

Assignment 2 will be graded as either *pass*, *insufficient* or *fail*.

Pass

A pass mark is awarded if both the group’s own task and the review phase work is judged

- completed according to the assignment instructions and documented independently of other groups.
- the documentation is clearly presented and easy to read.

Insufficient

A good attempt at completion is evident, but where the solution does not meet the requirements of completion the course staff is able to attribute this to justifiable misapprehensions of the goals of the assignment or due to insurmountable difficulties during the project that have been brought to the attention of the course staff in a timely manner.

Where any hand-in is graded *insufficient* the group will be required to hand in a second time after the end of the course where they address the original hand-in's insufficiencies. In some situations (depending on the nature of the insufficiency) you may also be required to meet as a group with an examiner in order to clearly establish the nature of the issues that meant that you were unable to meet the requirements. An assignment that cannot be given a pass after the second hand-in opportunity will graded as fail with no further opportunities to complement.

Fail

A fail grade will be given if:

- The group has clearly failed to reach the requirements for at least an insufficient grade.
- An attempt to mislead the staff is evident, such as documenting so as to make it appear that more work has been done that in reality, or such as submitting as a group although students have not properly shared the tasks.
- Plagiarism is evident.

A fail grade entails that the students in that group will not be given further opportunities to complete an equivalent assignment before the next time the course is held.

Past Deadline Hand-ins

Due to the time dependencies between phases on this assignment, hand-ins that are not properly submitted by deadline will in general be regarded as automatic fails.

In case verifiably unpredictable problems cause a group or a member of a group to unavoidably miss the any of the deadlines, one extra opportunity may be arranged after the end of the course to allow those groups to complete the assignment requirements. This alternative opportunity may involve alternative requirements, such as being required to complete several tasks if there are no suitable other group's work to review. Students who are unable to meet the requirements for this alternative assignment completion will be given no further opportunities to complete the course before the next time the course is held.

Individual Task Instructions

Task 1: AppArmor

Profile
This task will require reading of some dry documentation and coding of a configuration file in order to test what a mandatory access control system can look like. We assume it will primarily attract those students who have some coding experience, and good linux experience. You will have to find your own way around the sources of AppArmor documentation on the net that best suit your level of ability.

https://wiki.ubuntu.com/AppArmor
For this task you are to write an AppArmor profile for either vi, vim, nano, or (advanced) gedit.

There is an account *student* on your Ubuntu Linux virtual machines. Create a folder in this user's home directory called "`groupXX`" (where XX is your group number), for example "`/home/student/group10/`", and create a set of rules such that the editor you have chosen is only able to create and write to files in that folder and not in any other folder on the system. This means that it should not be possible to write files to subfolders in that folder, such as `/home/cs2lab/group10/subfolder/`. Using comments on each line in your profile code explain what each line in your ruleset does and why you included it.

In your documentation include a description of how you arrived at your solution, including sources referred to and tools used. Include an excerpt of a system log that shows how an attempt to use the editor to write outside of its permitted directory has been refused.

Based on understanding gained from this exercise include a short report that describes how the behaviour that you have seen in AppArmor's mandatory access control can be used to improve system security in a real-life situation. Include also a discussion on what strengths and weaknesses the group perceives in its use.

Include your AppArmor profile code within your report but also include it as a separate file in your submission in case the staff wish to use it for testing purposes.

Task 2: GPG

Profile
This task will give students the opportunity to show that they can apply the principles of public key cryptography that we have studied during the course. GPG2 is installed on the Linux computers at DSV (though not in the virtual Linux images that you have been using during assignment 1), but some may like to install the software on their own personal computers. Best understanding of the processes involved are expected if you can use the command line interface to the GPG2 program. It is quite possible to solve the problem by using a graphical interface, but do not expect to be able to understand what is happening when you use an interface that hides all the details. Since you need to document your understanding for this assignment, graphical interfaces will put you at a serious disadvantage.

Your task is to learn and to apply PGP through the tool GPG2 to ensure that you are able to use it for communications that are safe, trustworthy, and practical. You must show good adherence on all levels to proper practice in applied cryptography.

You are to create a GPG key pair for each member in the group and ensure that they are kept in accordance to their purpose. Each group member shall verify your partner's public key and then sign them. For this task you are also to cooperate with another group to do a key signing challenge by trading public keys and verifying their origin. You are to send a file to the other group, signed by both members of the group, and encrypted in a way that allows both of the other group members to decrypt it. The other group shall likewise send such a file to your group. Ensure that:

- you take measures to properly protect your own private keys.
- you can read the other group's file and verify that it is authentic.
- all steps make proper expedient use of the cryptographic tools so that your tasks can be executed in no more steps than are necessary (so make good use of the documentation). Recipients of encrypted and signed files should be able to unpack and verify such files with a minimum of effort (including mental effort) and with a maximum of practical usage.

Include in your report explanations of how you created and suitably protected your keys, verified each others' keys, how you verified the other group's keys, steps you took to ensure that your keys are verifiable, how you encrypted and signed your message to the other group, and how you verified and decrypted the other groups' message. Also briefly explain what you found challenging along the way. Encrypt your final submission (which should be a single pdf) making use of the course staff key that can be fetched from https://www.dsv.su.se/~alan/AlanIntroSec.asc, and both group members sign it. Submit the encrypted file, your signatures, and whatever else might be necessary for the course staff to read and verify the integrity and authenticity of your file. Besides these files include a clear exact copy of the pdf file containing your report. Though this in principle undermines the secrecy of the document that you went to the trouble of encrypting, it means that the course staff are guaranteed to be able to read about what you have done even if something goes wrong in the encryption or decryption. Upload the files directly into the hand-in activity, i.e. zip files or similar are not to be used.

Hint: Following the videos and seminars on cryptography it is well understood that one cannot encrypt a file as large as your report with pure public key cryptography, but it will take a combination of asymmetric and symmetric cryptography in practice. Be aware that you do not have to implement this mix of algorithms manually, but GPG will by default do the right thing for you. The tricky part can be to understand exactly what the program has done. Try to find the right arguments to the program that will help you to extract information about what is going on when you ask for, e.g. an encryption.

Task 3: Breaking into Metasploitable: A Penetration Test Experiment

Profile
This task allows students to apply some of their knowledge of common system vulnerabilities in the role of "white hat hacker", i.e. to inform the system owners of what vulnerabilities should be rectified. Since Metasploitable is a Linux based system, and the most powerful tools to help you are on your Kali system, this will no doubt suit groups who are somewhat more proficient in Linux.

Your task is to devise and document an informed and structured strategy for breaking your way into a vulnerable system. Only once this strategy has been devised and documented - under a document sub-heading *Design* - shall the group then follow through by using it to break into the supplied Metasploitable system.

Your first task is to look into how penetration tests can be suitably structured and documented. We freely admit that the system you are provided with is very easy to break into so that simply trying an ad hoc attack method is very likely to succeed. Searching the web for known deliberately included vulnerabilities in metasploitable systems is also easy. However, though you might be itching to get started you are first required to discover and document what a more structured and methodical approach can be. You are welcome to surf the internet with suitable search strings such as "penetration test design". Section 24.2 in the course book is also a suitable source. You need not spend long with these studies, but your penetration test plan should show some form of motivated structure before you begin any practical tests. **N.B. chaotic or ad-hoc attacks on this target are not part of this assignment. The *planning* of what you will do is a required element.**

N.B. Due to a late discovered technical issue the remaining instructions are likely to be subject to update in the days following the release of this assignment (2023-11-27). You should count on being able to complete a version of this task, though not necessarily as described there below. We expect to have addressed the issues by the time groups reach this stage of the task.

Ultimately you are to obtain a username and password from our pre-prepared Metasploitable system. You are also to look for important information left on the system in the form of four flags. These flags are located somewhere within the /etc folder or within files in the folder and contain secret information about previous Lab staff and the NOD building. You will find your own copy of the vulnerable Metasploitable system among the virtual machines in the environment that you have been using for assignment 1. At your disposal you have the Kali VM. Kali Linux is a flavour of Linux oriented towards penetration testing and forensics. More information on Kali Linux can be found here. Learn, and make as much use as is feasible of the Kali tools that can be used for penetration testing.

For the documentation of this task, you are to include the following:

- A declaration of the general strategy used to attack the system (including references where that strategy has been taken from, or adapted from external sources).
- Brief descriptions of all the strategies used to gain access and to find the flags, including those that were not fruitful.
- A more detailed step by step description of the exploit used to gain access to the system.
- Username and password for at least one of the users with SuperUser privileges on the system.
- The secret contained in at least one of the hidden flags and explanation of how it was how the information in the flag is hidden (for example, was it encrypted?)
- Your own brief thoughts on the exploit used, including how likely it is that it would exist on a production server, and how dangerous it might be if it did.

Though screenshots may be helpful to show what you have achieved, you are advised to keep them to a level that in no way detracts from your document's readability. You should assume that the readers of your document will be most interested in the reasoning behind your actions, including failed actions, and deductions from what you experienced. Good solutions will document several strategies to achieve the same results, wherever appropriate.

Some concrete hints for your attack strategy:

- Try scanning the ports on your target system to see what is open and what protocols they those ports might be expected to ‘speak’
- Kali Linux contains a program called Johnny which can be used to crack password hashes.
- In unixes a dot at the beginning of a file name causes it to be hidden in normal directory listings. You should find flags and configurations that allow such ‘hidden’ file names to be shown.
- Make sure to inspect the following files:
  - The file where passwords used to be kept on Unixes and the file where passwords are kept in more modern Unixes.
  - /etc/mailname
  - /etc/nanorc
  - /etc/motd
  - /etc/fstab
  - /etc/SuperSecretInformation

Task 4: Anonymisation Tools: Keeping Your Information to Yourself

Profile
Though there are technical aspects to this assignment it allows you to find tools for the operating system environment that you prefer.

For this task it is assumed that any and all anonymous traffic is routed through the virtual environment that was set up for assignment 1. This may not be entirely necessary depending on what tools you experiment with, but all groups should take care and even seek advice before attempting any of these experiments on any networks. Before you use any anonymisation tools that effect the identification of data traffic within the Stockholm University controlled network via the provided virtual environment you are required to send an email with CC to both your group members, to cs2lab@dsv.su.se with subject field: *IntroSec – Planned Anonymous Traffic* two working days before you begin transmitting such traffic. In the message, stipulate what tools you will be using, when your traffic will begin and when you expect to have completed your experimentation.

Your task is to find 3 different anonymisation tools that you are able to motivate are especially interesting to work with and compare them in a written report. If you do not already know which tools you are most interested in you may find useful links by searching the Web for terms association with network anonymisation, or start by checking the Wikipedia web page on Anonymous proxy and that page's *See also* section. You do not have to limit yourselves to only Web anonymisation.

You must first set up, motivate and document - under a document sub-heading *Design* - a test protocol for the aspects of the tools that you regard are the most interesting to test. The test protocol is to be designed based on aspects of the tools that you chosen and to give a reasonable investigation for the time available. Factors that you should consider comparing include:

- How easy is it for a user to install and use the system securely? E.g.
  - Are there parts that would be difficult for a naïve user to complete, or complete securely?
  - Are there any kinds of delay involved that might make users too impatient to use the tool?
  - Is it possible to demonstrate where poor handling of the tool may contribute to a breach of anonymity?
- How helpful is the documentation for different user profiles?
- What does each tool protect against compared to the other tools you are looking at?
- What might the tools not protect against?
- What common misconceptions might exist about the tool?
- Is it possible to uncover the anonymised data if the user does not use the tool properly?
- A short discussion on the cost/benefit ratio of the tool's benefits versus the potential loss of availability involved in installing, configuring and using the tool.

Wherever possible and reasonable, prove identified problems with practical demonstrations.

Task 5: Social Engineering

Profile
This is an experiment and investigation that does not assume much technical prowess, but does require careful thought, strict ethical behaviour, and finesse when dealing with interview subjects.

Your task is to find and document - under a document sub-heading *Design* - two true to life situations where deliberate and malicious social engineering could be used to break, bypass or exploit an IT security policy. You are also to first design and then perform and document at least 3 interviews per social engineering scenario with *potential victims* of such a scenario in order to assess the likelihood of success, the potential damage caused and the potential victims' awareness of the exploitability of the situation.

Your overall goal with this task is to gain insight into, and to document, the public's understanding of the risks involved with IT connected social engineering attacks. You should therefore ensure that the subjects that you interview are good representatives of the potential targets of the attack. Interviewing friends and family may be suspected of giving less reliable results than interviewing members of the general public (assuming they are potential targets for your chosen scenarios). You are not trying to find experts on social engineering (as some previous students have mistakenly assumed). You are trying to find ordinary people and then finding a **safe** and objective way to interrogate them about how they are likely to act to thereby draw assumptions on the public's understanding of social engineering. This in turn should give an indication of how serious a threat your specific scenarios are likely to be.

In your report you are to describe how the situation could be exploited, if any artefacts are used for the exploit, how a victim can protect from it, your interviewed potential victims awareness of the situation, and the expected gains from a successful exploit. Also document how the dangers from your scenarios could be mitigated by a security manager's measures.

WARNING! Previous years a few groups have erroneously interpreted this task as being about directly executing social engineering attacks on unsuspecting individuals. This would be regarded as a transgression of course and university ethical principles. The very trick is to investigate social engineering **without causing even the least amount of possible danger or discomfort to others**. If you are in any doubt about the limits of ethical research you are encouraged to browse https://www.vr.se/english/mandates/ethics.html. If there is any indication that any subject may have suffered the least discomfort at your hands due to poorly devised method, your project may be summarily terminated and failed. If in doubt, check with the course staff before conducting any practical stages.