

Rettemal

TEK5510

2025

i Front Page

UNIVERSITY OF OSLO
The Faculty of Mathematics and Natural Sciences
Written examination TEK5510/9510
2025 2nd SEMESTER
Duration: December 19th at 9:00 to 13:00 (4 hours)
Permitted aids: None
It is important that you read this front page before you start.

Remember to explain your answers. Explanations are required to receive full score.

If you need to zoom in on screenshots, press 'Ctrl' and '+' at the same time. Use the '+' in numpad (the right hand part of the keyboard).

Good luck!

Question 1

1 Security Principles and Terminology

1. What are the three main security principles? (3p)
2. How can each of the main security principles be breached? Explain. (3p)
3. Explain the terms 0-day vulnerability and N-day vulnerability. (2p)

1. 1p for each of Confidentiality, Integrity and Availability.
2. 1p for each principle. One example is sufficient for each security principle. For example: Availability is breached if the disk is encrypted by ransomware. Confidentiality is breached if an attacker can extract information from a database through for example SQL injection.
3. 1p for: A 0-day vulnerability is an undisclosed vulnerability which is unknown to the vendor and the public. 1p for An N-day is a vulnerability that is known, and for which a patch might already exist.

Question 2

2 Analyzing an Executable

1. For each of the following tasks, which of the three tools Process Monitor, Process Explorer, and Regshot would be suitable (might be more than one)? Explain (Total 6p)
 - a) Check which files an executable is writing to.
 - b) Explore whether an executable starts other processes.
 - c) Determine whether an executable made changes to the registry.
2. Give three examples of signs of malicious activity that would be shown in Process Monitor, Process Explorer and/or Regshot. Explain why these are malicious signs. (3p)

1.

a. 2p for Process Monitor gives an overview of WriteFile events and shows which process is performing them.

Note: Regshot is not correct. It only shows new files that are created, not existing files that are written to, and not which executable that created the files.

b. 1p for Process Monitor shows events of started processes and which process that performed the events. 1p for Process Explorer shows the process hierarchy, including child processes that are started by parent processes, but this is only shown while the processes are running.

c. Max 2p total. 2p for Process Monitor gives an overview of registry related events and shows which process is performing them. If only Regshot is mentioned, 1p can be given for explaining that Regshot only shows registry changes, not which executable that did the changes, but the names and values of the registry key changes may indicate which process they are related to.

2.

Max 3p total. 1p for each example including the action and in which tool it can be seen. Examples: Setting the registry key for automatically running a process at startup, shown in Process Monitor, while Regshot shows that the value is set but not which process that did it. A process with the same name as a builtin Windows application, but with the exe file located in Downloads, shown in ProcessMonitor and in Process Explorer. A process starting another process before stopping itself, leaving the child process orphaned, shown in Process Monitor and Process Explorer.

Question 3

3 Vulnerabilities

Consider the function:

```
void copyBuf(char *src) {  
    char buf[4];  
    strcpy(&buf, src);  
}
```

This will have a stack frame with four entries like this:

```
0x1000 | 1 |  
0x1004 | 2 |  
0x1008 | 3 |  
0x100c | 4 |
```

1. Fill in what will be stored in field number (1p per)

- a) 1
- b) 2
- c) 3
- d) 4

2. What is a stack canary, and between what numbers would this be placed on the stack? (2p)

3. Explain the following mitigations and whether these would hinder exploitation of the copyBuf function:

- a) ASLR (2p)
- b) DEP/NX (2p)

1. 1p for each of

- a. buf
- b. saved rsp
- c. saved rip
- d. argument/src

2. 1p for correctly explaining a stack canary, for example: a value that is saved on the stack to protect important values that is checked for modification before return from the function. The canary detects overflow, but does not prevent it.

1p for saying that it will be saved between 1 and 2.

3. a. 1p for explaining that ASLR randomizes memory addresses for stack, heap, code, libraries etc.

1p for an explanation of how this would hinder exploitation. For example: saying that this would hinder us in using hardcoded addresses in our exploit or that an address leak would be necessary. A simple "this would hinder the exploit" without explanation gives 0 points.

b. 1p for explaining that DEP/NX marks memory regions/stack as non executable.

1p for an explanation of what this would hinder. One example: We could not place code in memory and would have to reuse existing code inside the binary/ROP/return to libc etc. A simple "this would hinder the exploit" without explanation gives 0 points.

Question 4

4 Integer Computation

Do the following calculations. For each exercise, show the computations in binary numbers **and** explain the result. All numbers below are in decimal. (2p for each)

1. $102 + 26$ as 8-bit unsigned integers
2. $102 + 26$ as 8-bit signed integers
3. $120 - 126$ as 8-bit unsigned integers
4. $120 - 126$ as 8-bit signed integers

2p for each of:

1. $102 + 26 = 0110\ 0110_2 + 0001\ 1010_2 = 1000\ 0000_2 = 128$
2. $102 + 26 = 0110\ 0110_2 + 0001\ 1010_2 = 1000\ 0000_2 = -128$, overflow. The signed numbers go from -128 to 127, and the first bit is the sign bit.
3. $120 - 126 = 0111\ 1000_2 - 0111\ 1110_2 = 1111\ 1010_2 = 250$, underflow. No sign bit.
4. $120 - 126 = 0111\ 1000_2 - 0111\ 1110_2 = 1111\ 1010_2 = -6$

Note: The binary calculation must be included to receive full score.

Question 5

5 Cryptography

1. What is the difference between *symmetric* and *asymmetric* encryption? (2p)
2. What differs *hashing* from the two above? (1p)
3. Are the following examples of symmetric, asymmetric or a hashing algorithm? (1p per)
 - a) Yescrypt
 - b) RSA
 - c) XOR
4. We have also learned about the Caesar cipher, also called Rot-n. Explain whether the Caesar cipher is a symmetric or asymmetric crypto scheme? (2p)
5. The phrase «crypto» has been encrypted to the phrase «pelcgb», what is the value of n? (1p)

1. 1p for explaining that symmetric encryption uses the same key for encryption and decryption.

1p for explaining that asymmetric encryption uses a different key for encryption and decryption.

2. 1p for saying that hashing should not be decrypted/is a one way scheme.

3. 1p per

- a. hashing
- b. asymmetric encryption
- c. symmetric encryption

4. The point of this exercise is to check if the student understands the concepts of symmetric and asymmetric encryption and the caesar/rot-n cipher. Any answer that shows this should be awarded 2p. An answer just stating symmetric/asymmetric gives 0 points. An example of a good answer is: The caesar cipher is symmetric in the case where n is 13, since the same key can be used for encryption/decryption. The cases where n is not 13 the scheme is asymmetric since we use a different key for encryption/decryption. Another good answer would be: It is hard to say if the Caesar cipher is symmetric/asymmetric since the key is deduced from the decryption key, therefore you could consider it symmetric in every case. Also: the caesar cipher is symmetric since you use the same key to rotate left to rotate right.

5. 1p for n=13.

Question 6

6 Passwords

1. How and where are user information and passwords stored in Linux? (3p)
2. How and where are passwords stored in Windows for a local user? (3p)
3. Explain briefly how tools like John the Ripper work. (2p)
3. What is a salt in relation to passwords, and how is it used? (2p)
4. Would salt prevent use of John the Ripper? Explain. (2p)

1. 1p for stating that user information is stored in `/etc/passwd`. 1p for stating that passwords are stored in `/etc/shadow`. 1p for stating that the passwords are stored with a salt.

2. 1p for stating that passwords are stored in the registry. 1p for stating that they are located in `HKLM\SAM` (or the Security Account Manager). 1p for stating that they are hashed without a salt.

3. 1p for stating that John the Ripper works by guessing a password, hashing it as described in the input file (which contains information about the algorithm etc.) and comparing the result with the hash that is to be cracked. 1p for stating that it may use various modes, like wordlists and rules for adding numbers etc.

4. 1p for stating that a salt is a random string/value stored with the password that is added to the password before hashing. 1p for stating that a salt is used to give different hashes even if the passwords are equal, rendering precomputed hash tables useless.

5. 1p for No. 1p for explaining that John guesses each password individually regardless of whether a salt is used, and it uses the salt to compute hashes during cracking if there is a salt.

Question 7

7 Access control on Linux

1. Explain the access rights of the file tent.make after running (2p per):
 - a) `chmod 763 tent.make`
 - b) `chmod 521 tent.make`
2. What is the function of the «sticky bit»? (2p)
3. What is the sudo command? (2p)

1. 2p for all three rights are explained correctly in each. 1p if two right. 0,5p if one correct
 - a. Owner has read, write and execute. Group has read and write. Others have write and execute.
 - b. Owner has read and execute. Group has write. Others have execute.
2. 2p for correct explanation of the sticky bit. For example: If the sticky bit is set on a directory only the owner and root/superuser can delete a file from that directory.
3. 2p for correct explanation of the sudo command. For example: The sudo command executes a command as the superuser/root.

Question 8

8 Access Control in Windows

Object

Thread A

Access token

Alice
Group B

Thread B

Access token

Bob
Group A
Group B

DACL

ACE	Access	Principal	Permissions
ACE 1	Access allowed	Alice	Read, Write
ACE 2	Access denied	Group A	Write, Execute
ACE 3	Access allowed	Group B	Execute

For each case below, what is the result of the request? Explain how each Access Control Entry influences the result.

1. Thread B requests read access to the object. (2p)
2. Thread B requests write and execute access to the object. (2p)
3. Thread A requests read and execute access to the object. (2p)
4. Another Thread C is given execute access and denied read and write access to the object. What is the token for Thread C? Explain. (2p)

The answer must be explained to get full score.

1. 2p for: ACE1 does not apply, as Alice is not in the access token of Thread B. ACE2 applies, as Group A is in the access token of Thread B, but ACE2 does not concern read. ACE3 applies, as Group B is in the access token of Thread B, but ACE3 does not concern read. The end of the DACL is reached without read access granted, therefore read access is denied.
2. 2p for: ACE1 does not apply, as Alice is not in the access token of Thread B. ACE2 applies, as Group A is in the access token of Thread B. ACE2 denies write and execute access, and access is denied. ACE3 is therefore not checked.
3. 2p for: ACE1 applies, as Alice is in the access token of Thread A. ACE1 grants read access, but still need execute access. ACE2 does not apply, as Group A is not in the access token of Thread A. ACE2 applies, as Group B is in the access token of Thread A, and execute access is granted. Since both read and execute access are granted, all requested access is granted, and the thread will be granted access.
4. 2p for: The access token must contain Group B, and not contain Alice and Group A. The token can contain any other user than Alice, for example Bob.

Question 9

9 Securing software

Securing software

1. What is the difference between a generational and mutational fuzzer? (2p)

2. Which of the threat modelling frameworks presented in this course would be suitable for modelling:

- a) Categorizing vulnerability type (1p)
- b) Damage assesment (1p)

3. Pick your two favorite OWASP security principles and give a short summary of what they are about. (4p)

1. 1p for a generational fuzzer generates testcases from scratch. 1p for a mutational fuzzer edits/mutates an existing testcase.
2. 1p for each (not necessary to state what the letters stand for)
 - a. STRIDE
 - b. DREAD

3. 2p for mentioning and explaining two of the OWASP principles correctly. Deduct 1p if the principle title is correct, but the explanation is incorrect. The principles, with a brief explanation for each, are:

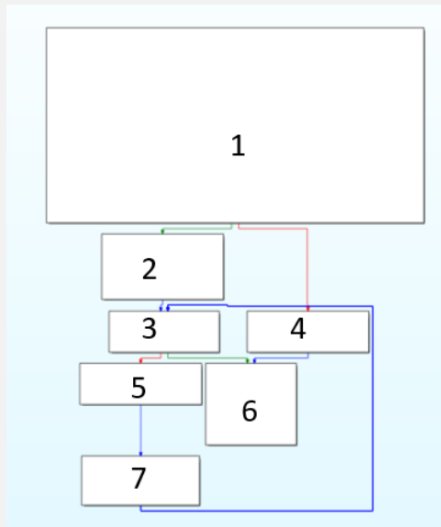
- Minimize attack surface: Less features means less attack surface
- Secure default settings: The default settings should be secure, where the user can optionally decrease security
- Least privilege: Give each principal access to only what it needs to perform its task
- Defense in depth: Have multiple mechanisms, such that if one fails, all is not lost
- Fail securely: If anything goes wrong, it should not fail into a insecure state
- Don't trust services: Do not trust input from users or external services you do not control
- Separation of duties: Use different processes/accounts for different tasks. A superuser should not be normal users
- Keep security simple: If security is complex users will normally bypass it, and it increases the risk of misconfigurations.
- Fix security issues correctly: Fix the root cause of a problem, so you will not have to fix it again
- Avoid security by obscurity: Don't rely on that keeping your code/design secret will protect your system.

Question 10

10 Reversing

The screenshot shows the graph overview of a function in IDA.

What can you tell about the function from the graph? Where does it start? Which code structures do you recognize? Where does it return? (6p)



Note:

The numbers are inserted to make it easier to reference the code blocks in your answer and have no other function.

- 1p for stating that the function starts in 1.
- 1p for stating that there is an if-test at the end of 1.
- 1p for stating that there is a loop (for/while are both accepted) in 3,5,7.
- 1p for stating that in 3 there is an if-test checking whether the end condition for the loop is met.
- 1p for stating that the function ends in 6.
- 1p for stating that 5 and 7 contain the instructions that are done for each pass through the loop.

Question 11

11 Virtualization

1. Give two arguments for when and why virtualization is useful in general (2p)
2. There are two types of hypervisors/virtual machine monitors. Explain them. (2p)
3. When should each type be used? What are their advantages and disadvantages? (4p)

- 1.
- 2p for mentioning two of the reasons for using virtualization that have been

described in this course:

- Efficient use of hardware and resources, with improved management and resource utilization.
- Improved security: Malware would infect the VM, not the host, which provides safer testing and analysis of malware. And VMs are isolated from each other.
- Distributed applications bundled with OS, which allows optimal combination of OS and application, and is ideal for cloud services.
- Powerful debugging: Taking a snapshot of the current state of the OS, step through program and OS execution, and reset system state.

2.

1p for Type 1 hypervisors run below the operating system, directly on hardware.

1p for Type 2 hypervisors run on a host operating system and uses the services of the host for e.g. memory management, scheduling, and drivers.

3.

1p for: Type 1 hypervisors should be used if only virtual machines are running on the hardware, not side by side with a host operating system (for example in data centers).

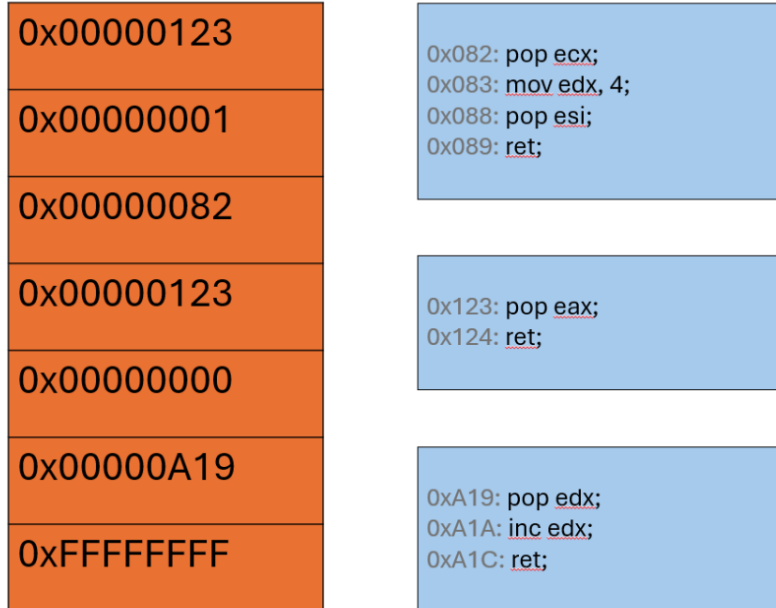
1p for: Type 1 hypervisors are usually very small, have less attack surface and high performance. Hardware support can be an issue.

1p for: Type 2 should be used for example when you want a virtualized environment side by side with a host operating system, for example when you need an isolated environment for running executables on your laptop. An example is the Oracle VirtualBox, which many students have used in this course.

1p for: Type 2 have a performance penalty, but are easier to use.

Question 12

12 Return Oriented Programming



1. What is the purpose of a ROP-chain? (2p)

2. Consider the ROP-chain in the picture. What is the value of the relevant registers after each instruction? (6p)

1. 2p for explaining that ROP chains are used to bypass NX/DEP by using executable code already found inside the binary.

2. 6p can be given. Deduct one point for each error. Give 1p if the student understands the concept of a ROP-chain, but the registers are wrong. Give 1p for writing the correct register values after running the chain without explanation or intermediate operations. Writing that the others are unknown is not necessary, but giving the registers we do not know the values of a value deducts a point.

After 0x123: `eax=0x1`, the others are unknown.

After 0x82: `eax=0x1`, `ecx=0x123`, others unknown.

After 0x83: `eax=0x1`, `ecx=0x123`, `edx=0x4`, others unknown.

After 0x88: `eax=0x1`, `ecx=0x123`, `edx=0x4`, `esi=0x0`, others unknown.

After 0xA19: `eax=0x1`, `ecx=0x123`, `edx=0xffffffff`, `esi=0x0`, others unknown.

After 0xA1A: `eax=0x1`, `ecx=0x123`, `edx=0x0`, `esi=0x0`, others unknown.

Question 13

13 Android

1. Explain the following permission types and give an example of each (2p each):

- a) Normal
- b) Dangerous
- c) Signature

2. Explain ARM TrustZone and what it is used for. (3p)

1. 1p for correct description and 1p for a correct example for each.
 - a. Normal permissions are not shown to the user. Examples: Internet, bluetooth, nfc, set alarm...
 - b. Dangerous permissions can be used access personal data or costs money. Examples: Contacts, calendar, location, phone, sms,...
 - c. Signature permissions can be declared by the apps themselves. Examples: Install packages, mount format file systems...
2. 1p for explaining that ARM Trustzone is a dedicated processor mode for security critical code. 1p for mentioning each of the following examples (max 2p): Lock screen passcode verification, fingerprint template matching, KeyStore management, DRM, Protected confirmation.