UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Examination in	TEK 5510 — Security in operating systems and software.
Day of examination:	Tuesday November 27th 2018.
Examination hours:	09:00 – 13:00.
This problem set consists of 11 pages.	
Appendices:	None
Permitted aids:	Dictionary.

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Answer all questions in this examination paper.

Answers can be written in English or in Norwegian.

Be concise when answering. It is often sufficient to write a single sentence, or at most a few sentences, to describe the concept that each sub-question asks for.

Question 1. Cryptography

a) Explain what public and private cryptographic keys are and give an example of an application of them. (2p)

b) What is salt, and how does the use of salt increase password security? (3p)

Answer

a) **1p** A private key is known only by the owner, used to decrypt messages.
1p The public key is publicly known, used to encrypt messages to the owner of the corresponding private key.
b) **1p** A salt is a string that is appended or prepended to the password before hashing it.
1p This is done to make precomputed rainbow tables unusable. The hash depends on the salt value, so one would need a rainbow table for the specific salt value to use the table for cracking.
1p The salt is known, so password salting does not increase the difficulty of cracking one password through bruteforcing.

Question 2. Secure development and Web security

You are hired by the university to design and develop a new web application for handing in and receiving markings and grades of home exams. Before

(Continued on page 2.)

you start you do a threat analysis of the project.

a) What are the assets and threats that needs to be considered in this project? (2p)

b) Explain the terms Confidentiality, Integrity and Availability. How do these principles relate to the assets and threats? (3p)

c) Use the OWASP principles of secure development to describe how you will design and develop the web application. (5p)

d) Explain the terms *SQL injection* and *Cross site scripting*. Will these types of attack be a threat to your web application? (2p)

Answer

a) **1p** The assets are the students' markings and grades (and possibly other personal data).
1p Possible threats are that the service becomes unavailable to the students, that the students' results fall in the wrong hands, and that the results are tampered with.
b) **1.5p** Confidentiality concerns protecting information from falling into wrong hands. Integrity concerns unauthorized tampering with data. Availability concerns that the service should be available to authorized users.
1.5p Relation to the assets and threats: the service becomes unavailable to the students (availability), that the students' results fall in the wrong hands (confidentiality), and that the results are tampered with (integrity).
c) **1p** for each security principle from the list below that is stated with a reasonable description of how it relates to the web application. Maximum 5p.

- Minimize attack surface
- Secure default settings
- Least privilege
- Defense in depth
- Fail securely
- Don't trust services
- Separation of duties
- Keep security simple
- Fix security issues correctly
- Avoid security by obscurity

d) **0,5p** SQL injection means that, because of missing input validation, additional SQL requests can be injected to do additional/unintended database queries.

(Continued on page 3.)

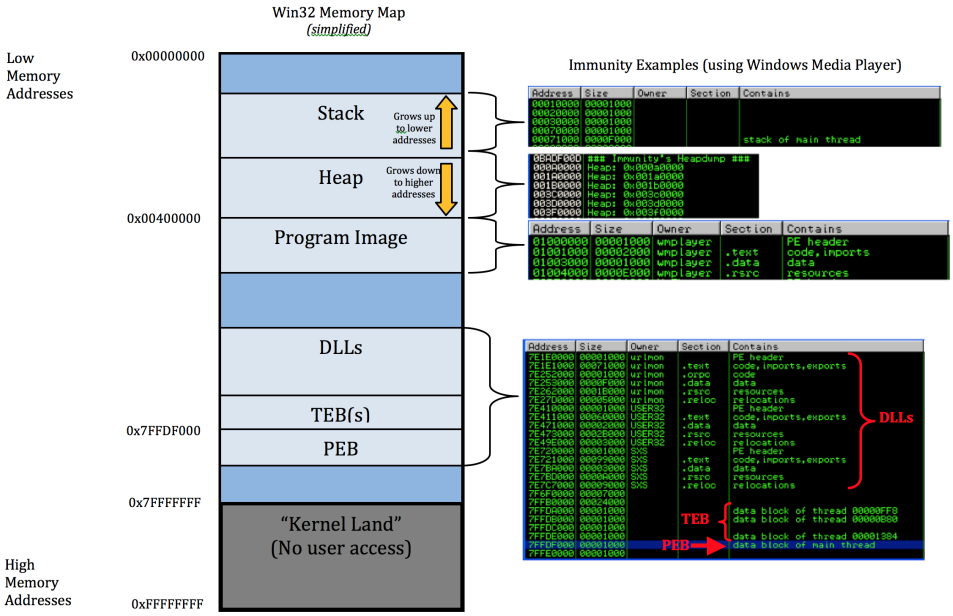

Figure 1: Figure from: www.securitysift.com/windows-exploit-development-part-1-basics/

0,5p Cross site scripting: Attacker can, because of missing input validation, provide any html element, for example to rewrite the document content or to redirect the page.
1p Good input validation can prevent both attacks. Limitation of the user input to pre-defined choices/commands for example. (Whether they apply to the web page depends on the design of the web application, for example whether the web page has a database)

Question 3. Analyzing executables
Figure 1 shows an executable file mapped into memory.

a) Explain the terms *executable* and *process*. (2p)

b) Explain the terms stack, heap, program image, DLLs, and Kernel Land that are shown in Figure 1.(4p)

c) Why can it be useful to find the strings contained in an executable? How can you find the strings? (2p)

d) In some cases the strings are *obfuscated*. What does this mean? How can you then find the actual strings? (2p)

Answer
a) **1p** An executable is a file that contains data and instructions on a format that can be executed.
1p A process is a running executable.
b) **1p** The **stack** is used for local variables and parameters for functions,

(Continued on page 4.)

and to help control program flow.
1p The **heap** is used for dynamic memory during program execution.
1p The **program image** is the executable file loaded into memory.
0.5p The **DLLs** are the libraries/dll-files that are loaded into memory (dynamically linked libraries).
0.5p Kernel Land is the most protected part of the memory, where for example core operating system components are running. As opposed to **User Land**, where normal user processes are run.
c) **1p** The strings can give information about what tasks an executable can perform. Error messages or debug strings, for example, or library names/file names.
1p The strings can be found either through the strings-view in IDA, the Strings tool, or by similar types of tools.
d) **1p** If the strings are obfuscated, they are encrypted or scrambled/encoded in order to make them difficult for an analyzer to decode. The strings are typically decrypted into clear text by a decryption function before use.
1p The actual strings can be found by feeding the encrypted strings to the decryption function. Either by writing a separate program containing a reimplementation of the decryption function, or by calling the decryption function in the executable directly. One of the alternatives is sufficient. (Or through logging during execution of the entire executable which only decrypts the strings actually used in that specific run.)

Question 4. Assembly
Consider the screenshot in Figure 2.

a) Explain the contents of the red, orange and green rectangles. (3p)

b) The next instruction to be executed is a call. To which function, and with which parameters? (2p)

c) Explain in detail how the data at address 0x00F8F968 on the stack was placed there. Which instructions, registers, and addresses were involved? (3p)

Answer
a) **1p** The red rectangle contains the addresses in memory of the corresponding bytes in the orange rectangle.
1p The byte values in the orange rectangle are the op codes for the instructions that are to be executed when the process is running.
1p The green rectangle contains the assembly instructions (human readable form) corresponding to the op codes.
b) **1p** The function to be called is strncmp (string comparison).
1p The arguments are pointers to the strings "-w" and "-r", and the number 2.
c) At 0x00F8F968 on the stack is the pointer (address) to the string "-w", 0x013B95A2. This address was pushed to the stack by the previously executed instruction *push dword ptr DS:[ESI+4]*. ESI = 0x013B9528B and ESI+4 = 0x013B952B. At address ESI+4 shown in blue in the memory dump at the lower left is the address 0x013B95A2, which points to the string "-w".

(Continued on page 5.)

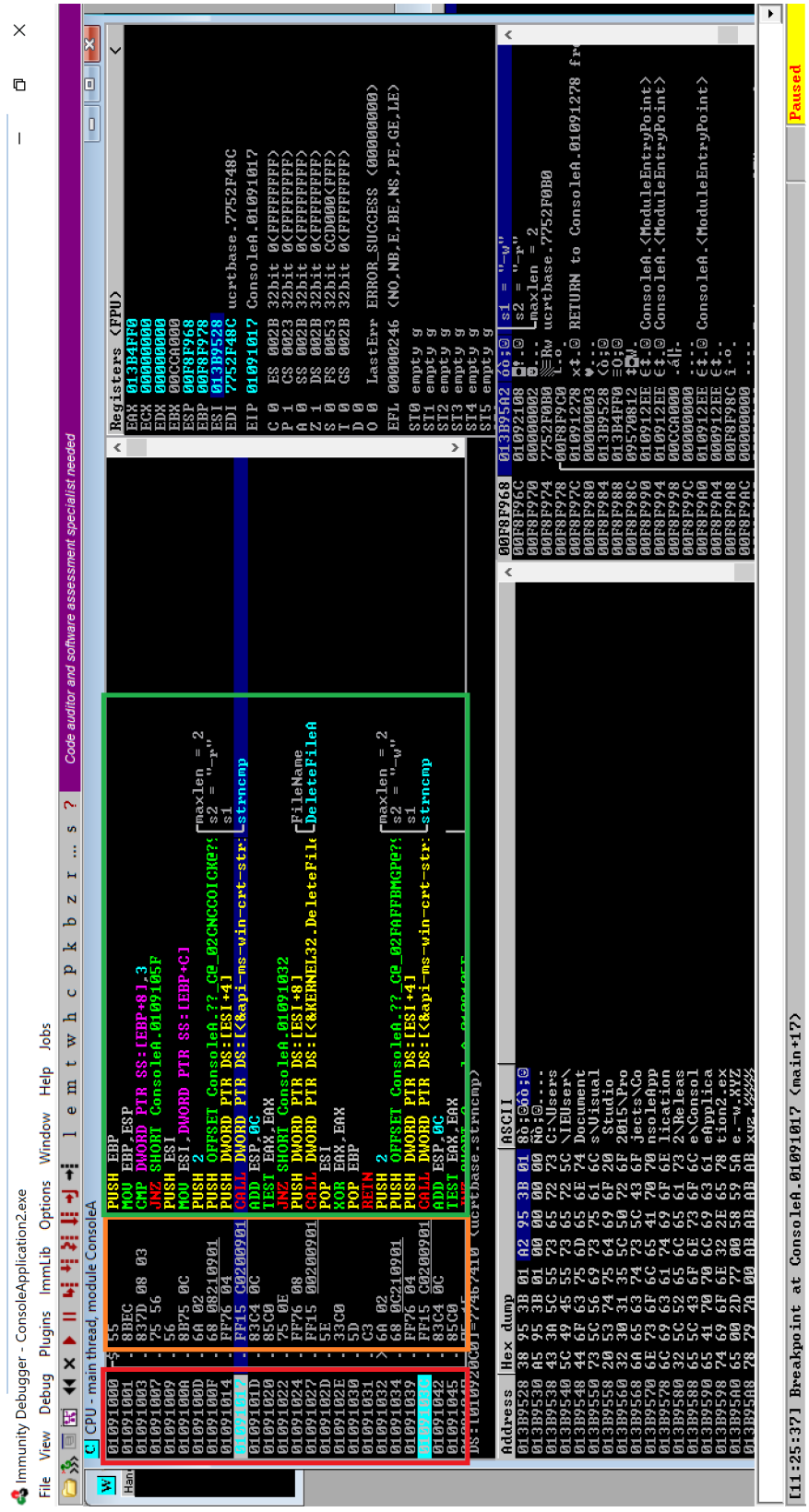

Figure 2: Immunity Debugger

(Continued on page 6.)

1p for stating that the previous instruction pushed the value onto the stack.
2p for explaining the role of ESI, using the register value and the memory dump.

Question 5. Malware and vulnerabilities

a) Give a short explanation of typical steps involved when a malware takes control over a computer. (2p)

b) Malware are created for different purposes by different types of attackers. Give examples of three types of attackers and what they typically aim to achieve with their malware. (3p)

c) What is a vulnerability? Give two examples of different types of vulnerabilities. (3p)

d) "Always update your software to the latest version as quickly as possible" is a common security guideline. Explain why this is important. (2p)

Answer
a) **1p** A *vulnerability* is exploited to gain control over the execution flow.
1p Thereafter a *payload* is executed, containing the actions that the attacker is interested in executing (e.g. encrypting the disk if it were ransomware).
In other words: The vulnerability is the way in. The payload is the reason that the attacker was interested in getting in.
b) **1p** for each example, maximum 3p. Possible examples:

- Script kiddies - because they can/for fun/...
- Hacktivists - to promote ideological or political agenda
- Organized crime - for money
- Advanced persistent threats - targeted attacks

c) **1p** A vulnerability is an exploitable bug.
2p for giving two examples of vulnerabilities. 1p for each. Possible examples: Buffer overflow, use after free, double free,...
d) **2p** The latest version contains the latest security patches. After a vulnerability of a piece of software has been published, anyone could use that vulnerability to attack anyone who has not yet updated to a fixed version. It is a bad idea to leave oneself vulnerable to publicly known attacks.

Question 6. Exploitation

a) Explain the terms *local exploit* and *remote exploit*. (1p)

b) What is meant by *privilege escalation*? (1p)

c) What is a Return Oriented Programming (ROP) chain? (2p).

d) What is Address Space Layout Randomization (ASLR)? (2p)
Answer
a) **1p** Local exploit is an attack that is done locally on a machine,

(Continued on page 7.)

through physical access. Remote exploit is an attack delivered over a network/without physical access to the computer.
b) **1p** Privilege escalation means to increase ones privileges on a computer, for example from normal user privileges to administrator privileges.
c) A ROP chain consists of a sequence of *gadgets*. Each gadget is a (often short) sequence of bytes that are interpreted as instructions, ending with a return. The addresses of the gadgets are placed in sequence in memory, such that the return at the end of one gadget makes the instruction pointer return into the next gadget. In total the gadgets perform a sequence of instructions, for example to make a piece of memory executable, and to execute a payload located there. The addresses in a ROP chains do not need to be stored in executable memory to work, but the addresses must point to executable memory.
d) ASLR is a security mechanism that aims to make the addresses of loaded modules (and thereby gadgets in the modules) less predictable. When a module is loaded into memory, the address on which it is loaded is randomized, instead of always loading it at the same address.

Question 7. Computing with integers

Show the computations with binary numbers and explain the results. With unsigned 8-bits integers:

a) 255 + 3 = ? (1p)

b) 126+4 = ? (1p)

With signed 8-bits integers:

c) 126+4 = ? (1p)

Answer

To get full score, the binary calculations must be included. a) **1p** 255+3 = (in binary:) 1111 1111 + 11 = 1 0000 0010 =(8bits unsigned:) 10 =2.
In 8bits integers, the first bit overflows, and the result is 255+3 = 2.
b) **1p** 126+4= (binary:) 0111 1110 + 100 = 1000 0010 = (8bits unsigned:) 130.
No overflow. Just as in normal mathematics.
c) **1p** 126+4= (binary:) 0111 1110 + 100 = 1000 0010 = (8bits signed:) -126
The first bit is used for sign. If the first bit is 1, the magnitude is found by the two's complement of the last 7 bits.

Question 8. Windows

a) How is a token used in the Windows access control model? (2p)

b) Explain the concept of User Account Control (UAC). Why does it exist? What is it intended to mitigate? How is it related to tokens? (3p)

You are asked to investigate a piece of malware, Lab03-03.exe. You decide to run it in a virtual machine first to get an overview of the executable. Figure 3 contains a screenshot of Process Explorer during the execution of

(Continued on page 8.)

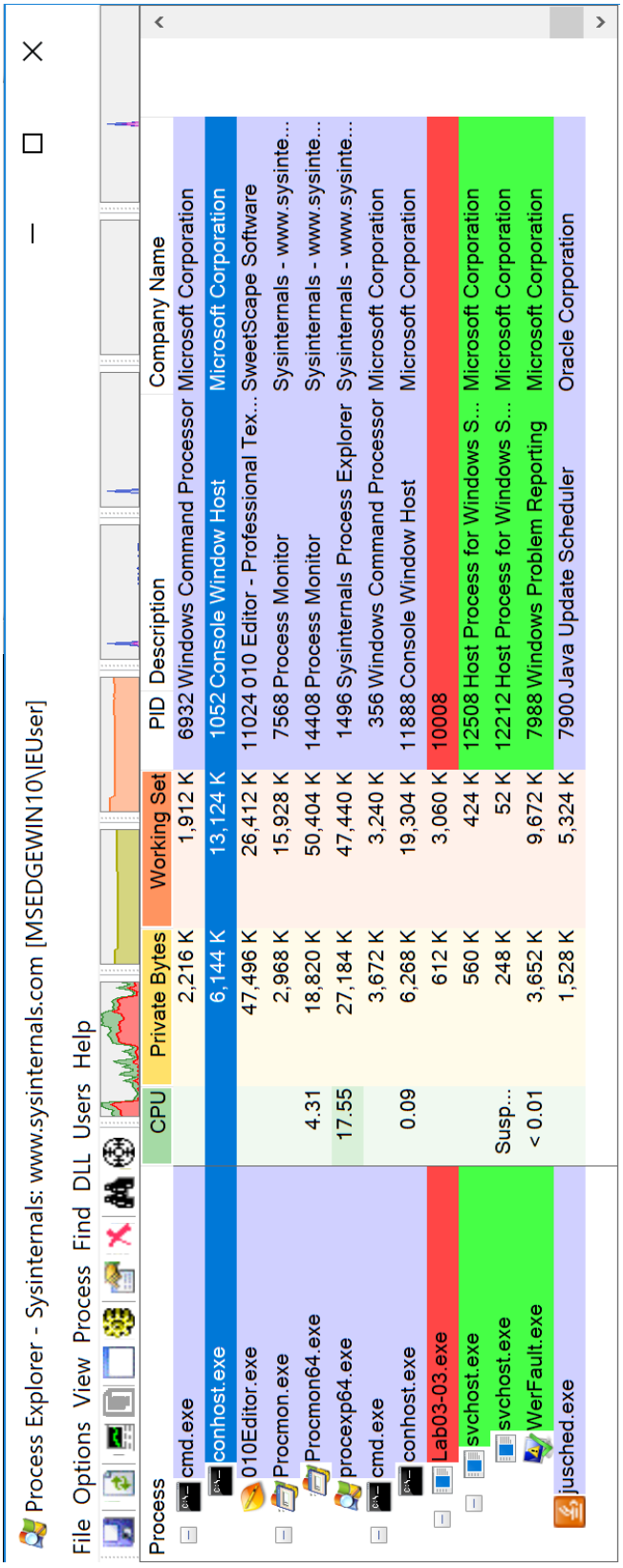

Figure 3: Process Explorer

(Continued on page 9.)

the malware.

c) What information does the screenshot in Figure 3 give about the malware? (3p).

d) How would you proceed to investigate the malware further? (3p)

Answer

a) **2p** The security credentials of a subject is stored in its token. An access token contains user SID, group SIDs, alias SIDs, and privileges. The Discretionary Access Control List (DACL) of an object consists of Access Control Entries (ACEs). Each ACE contains the SID it applies to, the type of access, and whether access is denied or granted. If the access token of the subject requesting a specific access to an object contains SIDs that according to the DACL of the object are granted the requested access, access is granted.

b) **3p** UAC is a mechanism for letting administrator users have as low privileges as possible, according to the security principle of least privilege. A standard token is used by default, and the admin token is prompted for (with the UAC pop-up) when needed to perform specific tasks. The idea is that if an attacker manages to take control over a process with the user's token, the attacker does not automatically have all administrator privileges, only the standard user privileges. This may introduce the extra step of privilege escalation in the attack, thereby increasing the difficulty. Before UAC was introduced, users tended to use administrator users (and administrator tokens) at all times, even though the administrator privileges were only needed for a few tasks, because it was too tedious to be logged in as normal user, since they needed to log out and in again as administrator if they needed to perform a task requiring administrator privileges.

c) **1p** The process Lab03-03.exe creates a new process called svchost.exe, which again creates another process called svchost.exe.

1p The name svchost is also the name of some normal, non-malicious windows processes. The fact that the malware spawns processes with this name is probably to hide and be difficult to detect. We can see from the screenshot that the Lab03-03.exe process is terminating, meaning that the two svchosts will soon be orphaned, and it will be more difficult to spot that these two processes were created by the malware and are not normal svchost processes. **1p** The WerFault.exe means that the malware has failed during execution. Probably because it was created for an older/other version of windows. This means that doing dynamic analysis like this will not show the actual purpose/functionality of this malware.

d) **3p** To investigate the malware further, we could try to run the malware on older/other versions of windows, we could try to investigate what makes it fail (for example through debugging it), or we could do static analysis. Static analysis is the only way to be sure that you can find/analyze all possible functionalities of the malware, not only the functionality that is run under the current conditions. We can use IDA to do static analysis: Find strings, export and import functions. Look for functions that interact with the system, like writing to or reading from files or the registry, or network activity. If the strings are obfuscated, try to decrypt them.

(Continued on page 10.)

Question 9. Linux

Consider the folling directory listing:

```
-rw-r--r-x 2  dole woodpeckers 8789 Feb 20 12:49  myfile
```

a) Briefly describe **who** can do **what** with *myfile* (3p)

In Linux systems users have a UID and username.

b) Explain the relationship between usernames and UIDs. (2p)

c) Which identifier is used for authentication, and which identifier is used to make access control decisions? (2p)

Answer

a) **1p** The user dole can read and write to myfile.
1p Members of the group woodpeckers can read myfile.
1p Others can read and execute myfile.
b) **1p** In Linux systems each user has a username which is mapped to a UID (User Identity).
1p Multiple users can be mapped to the same UID, but a single username can only have a single UID.
c) **1p** The username is the key to find the hashed password which is used for user authentication.
1p The UID is used when the system approves access to resources.

Question 10. Hardware security and Android

a) Briefly explain the difference between hardware and software vulnerabilities. (1p)

b) Describe the relationship between the BIOS/UEFI and the operating system. (2p)

c) Describe some security benefits and risks from using virtualization. (2p)

Android's access control model is based on Linux.

d) Explain the role of permissions in the Android security model. (2p)

e) Explain how traditional Linux users are utilized for security in Android. (2p)

f) Briefly describe the particular security challenges related to mobile devices (2p)

Answer

a) **1p** A hardware vulnerability is a vulnerability in the hardware on which software is running, while a software vulnerability is a vulnerability in the code/instructions of the software.
b) **2p** The BIOS/UEFI is responsible for initializing the system when it

(Continued on page 11.)

boots, and it loads the operating system.
c) **2p** Virtualization provides isolation between virtual machines and between virtual machine and host. This isolation provides a safe(r) environment for testing and analyzing malware. Still, the VMs and the host run on the same hardware, and they would still be vulnerable to hardware-level attacks like Spectre and Meltdown. Also the virtualization layer itself is an attack surface, and enabling features like shared clipboard weakens the isolation.
d) **2p** Applications have a set of permissions, which describe what the application is allowed to access. There are three types of permissions: Normal (not show to the user), Dangerous (user is alerted, can cost money/access personal data), and Signature (Only granted to apps that have the same key as the app that declared the permission. Primarily used to give system permissions to system apps).
e) **2p** Android typically uses one Linux user (UID) per application. Otherwise it uses a traditional Unix filesystem security model, with UIDs and GIDs and access masks for all file objects.
f) **2p** Mobile devices are always online, on different networks and multiple network types. They are often used in public and across security boundaries (private/work/meetings...). They are easy to lose/steal.

Good luck!