

Cover page

| Question | Question title | Marks | Question type |
|----------|----------------|-------|--------------------------|
| i | Cover page | | Information or resources |

Programvaresikkerhet (50%)

| Question | Question title | Marks | Question type |
|----------|-------------------------|-------|------------------|
| 1 | Binary exploitation | 14 | Text Entry |
| 2 | Web applications | 8 | Text area |
| 3 | Pentest methodology | 6 | Inline Gap Match |
| 4 | Fuzzing | 9 | Text Entry |
| 5 | Passord i minnet | 8 | Text Entry |
| 6 | To-faktor autentisering | 5 | Text Entry |

Kryptografi (50%)

| Question | Question title | Marks | Question type |
|----------|------------------------|-------|---------------|
| 7 | Krypto2022_1 | 14 | Composite |
| 8 | Krypto2022_2 | 12 | Composite |
| 9 | Krypto2022_3 | 16 | Composite |
| 10 | krypto hash-funksjoner | 8 | Essay |

i Cover page

Department of Computer Science

Examination paper for IDATT2503 Software security and cryptography

Examination date: 03.12.2022

Examination time (from-to): 15:00 – 19:00

Permitted examination support material: D

It is allowed to use the calculator Casio FX- 82ES

Academic contact during examination:

Donn Morrison 45548895

Ole Christian Eidheim 905 51 635

Dag Olav Kjellemo 47681639

Academic contact present at the exam location: No

OTHER INFORMATION

Get an overview of the question set before you start answering the questions.

Read the questions carefully and make your own assumptions. If a question is unclear/vague, make your own assumptions and specify them in your answer. Only contact academic contact in case of errors or insufficiencies in the question set. Address an invigilator if you wish to contact the academic contact. Write down the question in advance.

Withdrawing from the exam: If you become ill or wish to submit a blank test/withdraw from the exam for another reason, go to the menu in the top right-hand corner and click "Submit blank". This cannot be undone, even if the test is still open.

Access to your answers: After the exam, you can find your answers in the archive in Inspera. Be aware that it may take a working day until any hand-written material is available in the archive.

1 Binary exploitation

Consider the following source code for a compiled binary executable:

```
01 #include <stdio.h>
02 void flag(long int printflag)
03 {
04     if(printflag == 1)
05     {
06         printf("flag is s3cr3t\n");
07     }
08 }
09 int main()
10 {
11     char buff[32];
12     gets(buff);
13 }
```

The architecture is Intel x64. Position independent executable (PIE) is disabled. ASLR is enabled. There is no stack canary.

Construct an exploit to be read by gets() on standard input in the following text fields. Each text field must contain exactly 8 bytes. For non-printable characters (e.g., addresses or null bytes), use the hex encoding 0x1122334455667788. The layout must set up the stack so that the printf() is reached from within the function flag(). The address of flag() is 0x6666666666666666.

The following gadgets exist (hint, you need at least one of them).

```
0x7777777777777777 # mov rbx, 0x1; ret
0x8888888888888888 # pop rdi; ret
0x9999999999999999 # xor rdi, rdi; ret
```

The buffer "buff" should be filled with "A"s. The first 8 bytes are completed for you. The register rbp must be filled with "B"s.

AAAAAAAA

| | | |
|--|----------------------|--|
| <input type="text"/> | (AAAAAAAA) | 32 bytes of A to fill the buffer |
| <input type="text"/> | (AAAAAAAA) | |
| <input type="text"/> | (AAAAAAAA) | |
| <input type="text"/> | (BBBBBBBB) | <--- This will be RBX (base pointer) |
| <input type="text"/> | (0x8888888888888888) | <-- x64 calling convention: function arguments in registers, with the first argument being RDI, so we use the gadget to pop the next value on the stack which will let us pass the conditional check printflag == 1 into R |
| printflag set to this val --> <input type="text"/> | (0x0000000000000001) | |
| <input type="text"/> | (0x6666666666666666) | <-- address of flag(), which is the last gadget in the ROP chain, will be returned to when the ret instruction of the gadget at 0x8888888888888888 is executed. |

2 Web applications

1. Explain the danger of a cross-site scripting vulnerability in a large web application like Blackboard Learn, where users having different levels of access (administrators, teachers, students) interact. 2pts

Fill in your answer here

1. Any execution of Javascript within the context of an authenticated user gives the attacker control over all actions the victim can perform. For an application like Blackboard, a lower privilege user like a student could gain privileges of an instructor, teaching assistant, in the worst case, administrator. The impact could range from altering of grades to remote code execution on the server.

2. Describe the dangers of running server software as an administrative user like "root". 2pts

Fill in your answer here

2. Server software receiving network requests or input from untrusted sources should never run with administrative because if a vulnerability exists in the processing of input this could lead to full compromise of the server.

3a. What is a command injection vulnerability? 2pts

Fill in your answer here

3a. Command injection can occur when untrusted user input is not sanitised before being passed to a function call such as `system()` or `execve()` in C, or any of the corresponding calls to execute system commands in other languages.

3b. Explain what protection a developer of a web application can use against command injection. 2pts

Fill in your answer here

3b. A web developer must sanitise user input based on the context of where that input is used. Generally, special characters should be filtered or escaped. It is advisable to use sanitisation techniques that are built-in to the language used.

Maximum marks: 8

3 Pentest methodology

Provide the steps of a typical pentest methodology in the correct order.

 [Help](#)

Persistent access phase

Exploitation phase

Evidence collection and reporting phase

Agreement phase

Scanning phase

Planning and recon phase

1.

Agreement phase ✓
2.

Planning and recon phase ✓
3.

Scanning phase ✓
4.

Exploitation phase ✓
5.

Persistent access phase ✓
6.

Evidence collection and reporting pha ✓

Maximum marks: 6

4 Fuzzing

Given the libFuzzer test:

```

01 #include <stdbool.h>
02 #include <stdint.h>
03 #include <stdlib.h>
04
05 bool starts_with_exam(const char *str, size_t len) {
06     if (len >= 3 &&
07         str[0] == 'e' &&
08         str[1] == 'x' &&
09         str[2] == 'a' &&
10         str[3] == 'm')
11         return true;
12     return false;
13 }
14
15 int LLVMFuzzerTestOneInput(const uint8_t *data, size_t size) {
16     starts_with_exam((const char *)data, size);
17     return 0;
18 }

```

When the test runs through a debugger, the debugger will stop at line (10), and then `str` will

point to the string value (exa) and `len` will have the value (3).

Maximum marks: 9

5 Passord i minnet

Given the source code:

```
01 #include "crypto.hpp"
02 #include <iostream>
03
04 bool read_and_check_password() {
05     auto password = std::string();
06
07     std::cin >> password;
08
09     auto hash = secure_hasher(password);
10
11     auto ptr = &password[0];
12     for (std::size_t i = 0; i < password.size(); ++i)
13         *(ptr + i) = '\0';
14
15     return is_correct_hash(hash);
16 }
```

To make sure that the password is fully cleared from memory, the keyword

(volatile) must be added before `auto` at line

(11).

Maximum marks: 8

6 To-faktor autentisering

The three authentication factors are:

- Something the user (vet, har, er, knows, has, is)
- Something the user (vet, har, er, knows, has, is)
- Something the user (vet, har, er, knows, has, is)

Maximum marks: 5

7 Krypto2022_1

a) (4 poeng)

Briefly explain what a stream cipher is, and how it compares (advantages and disadvantages) to a block cipher.

Enter text here

We are given the following recursive sequence/LFSR:

$$z_{i+4} = z_i + z_{i+1} \bmod 2$$

b) (3 points)

Given the key $K = z_0 z_1 z_2 z_3 z_4 = 0101$.

What is the period of the sequence generated by the LFSR ? (15)

Show your calculations.

Enter text here

c) (3 points)

Encrypt the message **101001** with the keystream generated in the previous question, using XOR as operation.

Enter calculation here

d) (3 points)

You have eavesdropped, and got the encrypted message **110010** using a key unknown to you. You know that the first three bits of the plain text are 001. You want to change these, so that the plaintext will have **011** as the first three bits after decryption. The remainder of the message is to be unchanged.

Show how you can modify the ciphertext to achieve this.

Enter text here

e) (3 points)

Explain briefly how you can make a stream cipher from a block cipher.

Enter text here

Maximum marks: 14

8 Krypto2022_2

Use the affine cipher with $\mathcal{P} = \mathcal{C} = \mathbf{Z}_{32}$ and key (5,11), so that the encryption function is given by

$$\mathbf{Enc}(x) = 5x + 11 \pmod{32}$$

a) (2 points)

Why is (14,4) not a valid key in this encryption scheme?

Enter text here

b) (2 points)

Encrypt the message "1 10" (consisting of two blocks) i ECB-mode

Enter text here

We will now consider attacks on the affine cipher.

c) (2 points)

What is a weakness in using a cipher in ECB-mode? What types of attack can be performed given a large number of ciphertexts? Assume the ciphertexts are in a natural language such as English or Norwegian.

Enter text here

d) (4 points)

You are given the following pairs of known plaintexts $x_1 = 2$ and $x_2 = 5$, and their respective ciphertexts $y_1 = \mathbf{Enc}_k(x_1) = 10$ and $y_2 = \mathbf{Enc}_k(x_2) = 3$, encrypted with secret key k .

Find the key $k = (a, b)$, and decrypt the following ciphertext: $y = 23$

(Hint: Two equations in two unknowns)

Enter text here

e) (2 points)

Explain a way you can make the affine cipher more secure to attacks.

Enter text here

Maximum marks: 12

9 Krypto2022_3

a) (2 points)

What are two major differences between RSA and AES cryptographic systems?

Enter text here

b) (2 points)

What mathematical "problem" is the security of RSA based on?

Enter text here

Alice has the following public RSA key: $n = 323$, $e = 5$

c) (3 points)

What is Alice's private key? (194)

Show your calculations for how you found this key:

Enter text here

d) (3 points)

Encrypt the message $y = 100$ with the key given above.

Enter text here

e) (4 points)

Describe two attacks on the "school-book version" of RSA as described in the course and the formula sheet. For each, give one way that the vulnerability to the given attack can be reduced.

Enter text here

f) (2 points)

Can we use a single prime number, i.e. $n = p$ in RSA, instead of n being a product of two primes?

Select one alternative:

- ☒ Yes, it will work as normal, given similar security if p is big enough. ✔
- ☐ No, encryption and decryption will not work. n has to be a product of primes.
- ☐ Yes, it will work, but the security will be poor.

Explain your answer:

Enter text here

Maximum marks: 16

