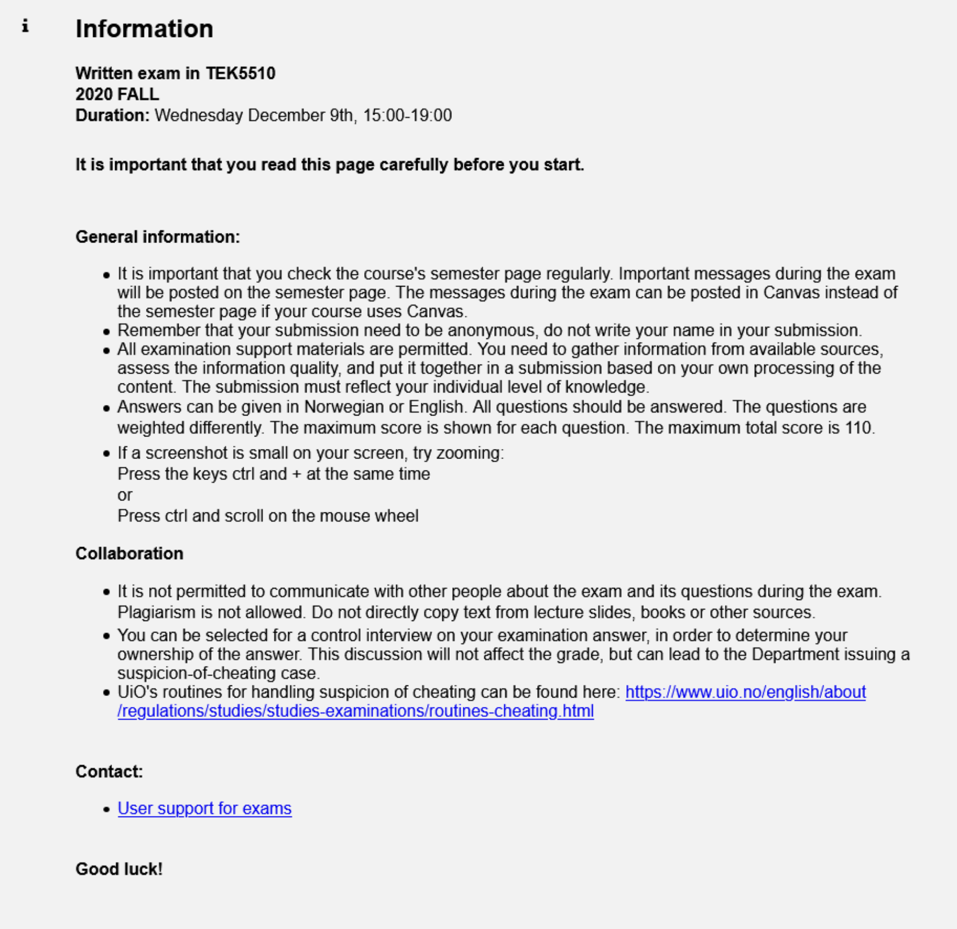


ℹ **Information**

Written exam in TEK5510
2020 FALL
Duration: Wednesday December 9th, 15:00-19:00

It is important that you read this page carefully before you start.

General information:

- It is important that you check the course's semester page regularly. Important messages during the exam will be posted on the semester page. The messages during the exam can be posted in Canvas instead of the semester page if your course uses Canvas.
- Remember that your submission need to be anonymous, do not write your name in your submission.
- All examination support materials are permitted. You need to gather information from available sources, assess the information quality, and put it together in a submission based on your own processing of the content. The submission must reflect your individual level of knowledge.
- Answers can be given in Norwegian or English. All questions should be answered. The questions are weighted differently. The maximum score is shown for each question. The maximum total score is 110.
- If a screenshot is small on your screen, try zooming:
  Press the keys ctrl and + at the same time
  or
  Press ctrl and scroll on the mouse wheel

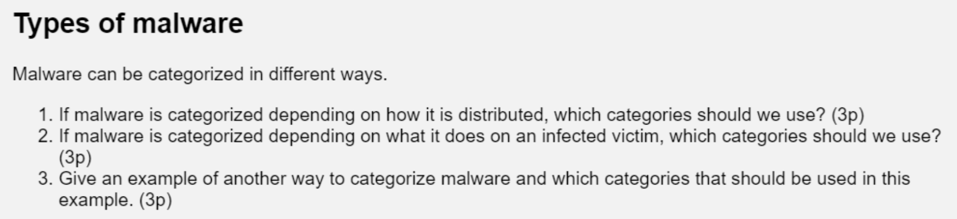
Collaboration

- It is not permitted to communicate with other people about the exam and its questions during the exam. Plagiarism is not allowed. Do not directly copy text from lecture slides, books or other sources.
- You can be selected for a control interview on your examination answer, in order to determine your ownership of the answer. This discussion will not affect the grade, but can lead to the Department issuing a suspicion-of-cheating case.
- UiO's routines for handling suspicion of cheating can be found here: https://www.uio.no/english/about/regulations/studies/studies-examinations/routines-cheating.html

Contact:

- User support for exams

Good luck!



Types of malware

Malware can be categorized in different ways.

1. If malware is categorized depending on how it is distributed, which categories should we use? (3p)
2. If malware is categorized depending on what it does on an infected victim, which categories should we use? (3p)
3. Give an example of another way to categorize malware and which categories that should be used in this example. (3p)

1. 1p for each category like through malicious web sites, e-mail attachments, unprotected/vulnerable network interfaces, Trojan horses ... Max 3p.

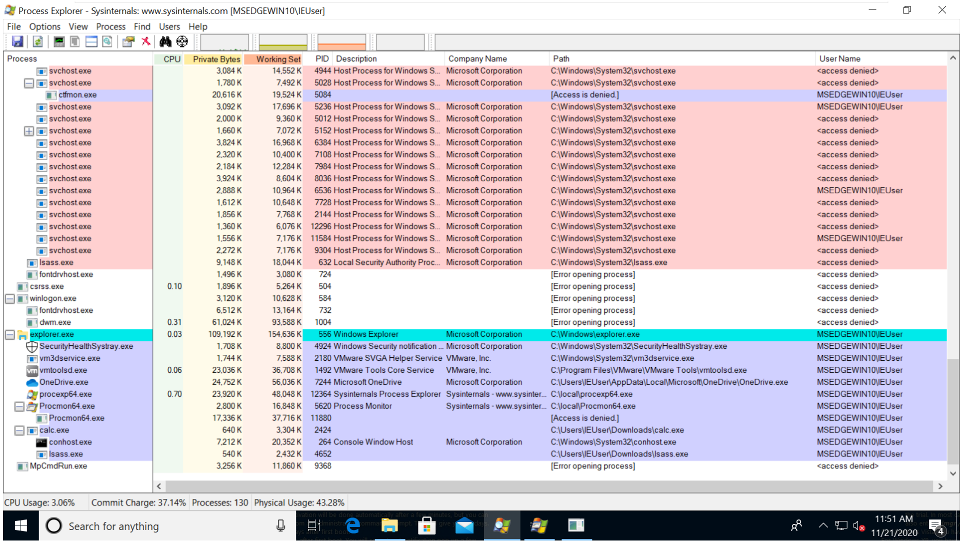
2. 1p for each category like ransomware, botnet, key logger, wiper, backdoor ... Max 3p.

3. 3p for a reasonable way of categorizing, with reasonable categories. For example:

- Categorize depending on who the target is. Targeted malware for a specific person/organization vs mass-distributed malware.
- Categorize depending on the attacker. Script kiddies, hacktivists, organized crime, advanced persistent threats.

2 Processes

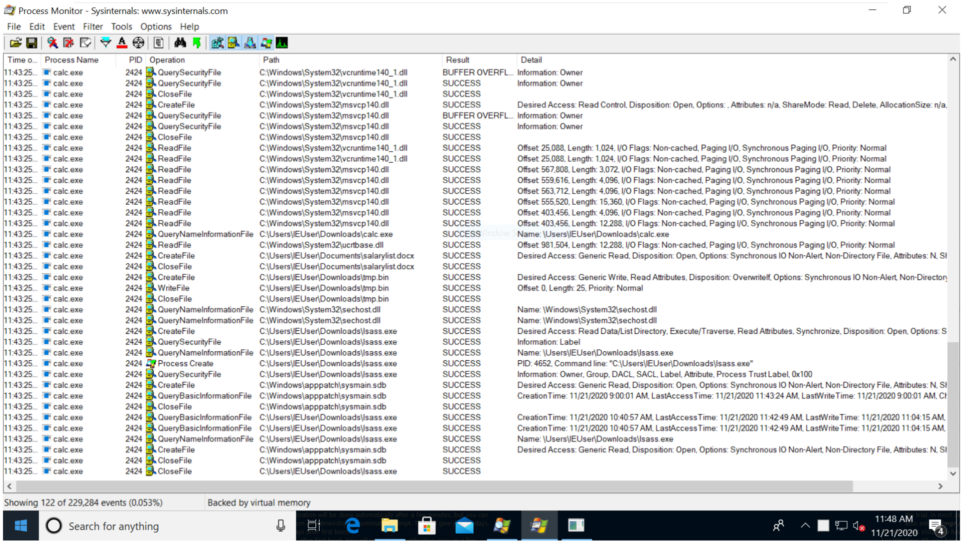
An employee has found a suspicious file calc.exe on his computer. You are tasked to investigate this file. During your analysis you get these two screenshots.



Process Explorer - Sysinternals: www.sysinternals.com [MSEDGEWIN10\IEUser]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Path	User Name
svchost.exe		3,084 K	14,552 K	4944	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		1,780 K	7,492 K	5028	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
ctfmon.exe		20,616 K	19,524 K	5084			[Access is denied.]	MSEDGEWIN10\IEUser
svchost.exe		3,092 K	17,696 K	5236	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	MSEDGEWIN10\IEUser
svchost.exe		2,000 K	9,360 K	5012	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		1,660 K	7,072 K	5152	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		3,824 K	16,968 K	6384	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		2,320 K	10,400 K	7108	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		2,184 K	12,284 K	7984	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		3,924 K	8,604 K	8036	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		2,888 K	10,964 K	6536	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	MSEDGEWIN10\IEUser
svchost.exe		1,612 K	10,648 K	7728	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		1,856 K	7,768 K	2144	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		1,360 K	6,076 K	12296	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
svchost.exe		1,556 K	7,176 K	11584	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	MSEDGEWIN10\IEUser
svchost.exe		2,272 K	7,176 K	9304	Host Process for Windows S...	Microsoft Corporation	C:\Windows\System32\svchost.exe	<access denied>
lsass.exe		9,148 K	18,044 K	632	Local Security Authority Proc...	Microsoft Corporation	C:\Windows\System32\lsass.exe	<access denied>
fontdrvhost.exe		1,496 K	3,080 K	724			[Error opening process]	<access denied>
csrss.exe	0.10	1,896 K	5,264 K	504			[Error opening process]	<access denied>
winlogon.exe		3,120 K	10,628 K	584			[Error opening process]	<access denied>
fontdrvhost.exe		6,512 K	13,164 K	732			[Error opening process]	<access denied>
dwm.exe	0.31	61,024 K	93,588 K	1004			[Error opening process]	<access denied>
explorer.exe	0.03	109,192 K	154,636 K	556	Windows Explorer	Microsoft Corporation	C:\Windows\explorer.exe	MSEDGEWIN10\IEUser
SecurityHealthSystray.exe		1,708 K	8,800 K	4924	Windows Security notification ...	Microsoft Corporation	C:\Windows\System32\SecurityHealthSystray.exe	MSEDGEWIN10\IEUser
vm3dservice.exe		1,744 K	7,588 K	2180	VMware SVGA Helper Service	VMware, Inc.	C:\Windows\System32\vm3dservice.exe	MSEDGEWIN10\IEUser
vmtoolsd.exe	0.06	23,036 K	36,708 K	1492	VMware Tools Core Service	VMware, Inc.	C:\Program Files\VMware\VMware Tools\vmtoolsd.exe	MSEDGEWIN10\IEUser
OneDrive.exe		24,752 K	56,036 K	7244	Microsoft OneDrive	Microsoft Corporation	C:\Users\IEUser\AppData\Local\Microsoft\OneDrive\OneDrive.exe	MSEDGEWIN10\IEUser
procexp64.exe	0.70	23,920 K	48,048 K	12364	Sysinternals Process Explorer	Sysinternals - www.sysinter...	C:\local\procexp64.exe	MSEDGEWIN10\IEUser
Procmon64.exe		2,800 K	16,848 K	5620	Process Monitor	Sysinternals - www.sysinter...	C:\local\Procmon64.exe	MSEDGEWIN10\IEUser
Procmon64.exe		17,336 K	37,716 K	11880			[Access is denied.]	MSEDGEWIN10\IEUser
calc.exe		640 K	3,304 K	2424			C:\Users\IEUser\Downloads\calc.exe	MSEDGEWIN10\IEUser
conhost.exe		7,212 K	20,352 K	264	Console Window Host	Microsoft Corporation	C:\Windows\System32\conhost.exe	MSEDGEWIN10\IEUser
lsass.exe		540 K	2,432 K	4652			C:\Users\IEUser\Downloads\lsass.exe	MSEDGEWIN10\IEUser
MpCmdRun.exe		3,256 K	11,860 K	9368			[Error opening process]	<access denied>

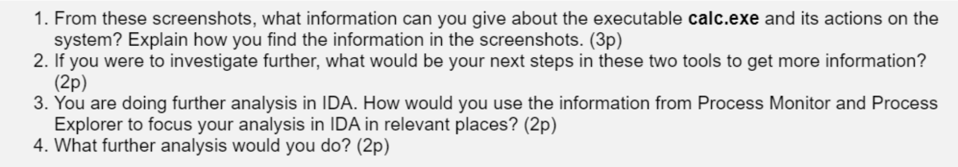
CPU Usage: 3.06% Commit Charge: 37.14% Processes: 130 Physical Usage: 43.28%



Process Monitor - Sysinternals: www.sysinternals.com

Time o...	Process Name	PID	Operation	Path	Result	Detail
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Windows\System32\vcruntime140_1.dll	BUFFER OVERFL...	Information: Owner
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Windows\System32\vcruntime140_1.dll	SUCCESS	Information: Owner
11:43:25...	calc.exe	2424	CloseFile	C:\Windows\System32\vcruntime140_1.dll	SUCCESS	
11:43:25...	calc.exe	2424	CreateFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Desired Access: Read Control, Disposition: Open, Options: , Attributes: n/a, ShareMode: Read, Delete, AllocationSize: n/a,
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Windows\System32\msvcp140.dll	BUFFER OVERFL...	Information: Owner
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Information: Owner
11:43:25...	calc.exe	2424	CloseFile	C:\Windows\System32\msvcp140.dll	SUCCESS	
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\vcruntime140_1.dll	SUCCESS	Offset: 25,088, Length: 1,024, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\vcruntime140_1.dll	SUCCESS	Offset: 25,088, Length: 1,024, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 567,808, Length: 3,072, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 559,616, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 563,712, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 555,520, Length: 15,360, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 403,456, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\msvcp140.dll	SUCCESS	Offset: 403,456, Length: 12,288, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	QueryNameInformationFile	C:\Users\IEUser\Downloads\calc.exe	SUCCESS	Name: \Users\IEUser\Downloads\calc.exe
11:43:25...	calc.exe	2424	ReadFile	C:\Windows\System32\ucrtbase.dll	SUCCESS	Offset: 981,504, Length: 12,288, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority: Normal
11:43:25...	calc.exe	2424	CreateFile	C:\Users\IEUser\Documents\salarylist.docx	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, Sh
11:43:25...	calc.exe	2424	CloseFile	C:\Users\IEUser\Documents\salarylist.docx	SUCCESS	
11:43:25...	calc.exe	2424	CreateFile	C:\Users\IEUser\Downloads\tmp.bin	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: OverwriteIf, Options: Synchronous IO Non-Alert, Non-Director
11:43:25...	calc.exe	2424	WriteFile	C:\Users\IEUser\Downloads\tmp.bin	SUCCESS	Offset: 0, Length: 25, Priority: Normal
11:43:25...	calc.exe	2424	CloseFile	C:\Users\IEUser\Downloads\tmp.bin	SUCCESS	
11:43:25...	calc.exe	2424	QueryNameInformationFile	C:\Windows\System32\sechost.dll	SUCCESS	Name: \Windows\System32\sechost.dll
11:43:25...	calc.exe	2424	QueryNameInformationFile	C:\Windows\System32\sechost.dll	SUCCESS	Name: \Windows\System32\sechost.dll
11:43:25...	calc.exe	2424	CreateFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Read Attributes, Synchronize, Disposition: Open, Options: S
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	Information: Label
11:43:25...	calc.exe	2424	QueryNameInformationFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	Name: \Users\IEUser\Downloads\lsass.exe
11:43:25...	calc.exe	2424	Process Create	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	PID: 4652, Command line: "C:\Users\IEUser\Downloads\lsass.exe"
11:43:25...	calc.exe	2424	QuerySecurityFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	Information: Owner, Group, DACL, SACL, Label, Attribute, Process Trust Label, 0x100
11:43:25...	calc.exe	2424	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, Sh
11:43:25...	calc.exe	2424	QueryBasicInformationFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	CreationTime: 11/21/2020 9:00:01 AM, LastAccessTime: 11/21/2020 11:43:24 AM, LastWriteTime: 11/21/2020 9:00:01 AM, Ch
11:43:25...	calc.exe	2424	CloseFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	
11:43:25...	calc.exe	2424	QueryBasicInformationFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	CreationTime: 11/21/2020 10:40:57 AM, LastAccessTime: 11/21/2020 11:42:49 AM, LastWriteTime: 11/21/2020 11:04:15 AM,
11:43:25...	calc.exe	2424	QueryBasicInformationFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	CreationTime: 11/21/2020 10:40:57 AM, LastAccessTime: 11/21/2020 11:42:49 AM, LastWriteTime: 11/21/2020 11:04:15 AM,
11:43:25...	calc.exe	2424	QueryNameInformationFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	Name: \Users\IEUser\Downloads\lsass.exe
11:43:25...	calc.exe	2424	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, Sh
11:43:25...	calc.exe	2424	CloseFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	
11:43:25...	calc.exe	2424	CloseFile	C:\Users\IEUser\Downloads\lsass.exe	SUCCESS	

Showing 122 of 229,284 events (0.053%) Backed by virtual memory



1. From these screenshots, what information can you give about the executable **calc.exe** and its actions on the system? Explain how you find the information in the screenshots. (3p)
2. If you were to investigate further, what would be your next steps in these two tools to get more information? (2p)
3. You are doing further analysis in IDA. How would you use the information from Process Monitor and Process Explorer to focus your analysis in IDA in relevant places? (2p)
4. What further analysis would you do? (2p)

1. 1p for reads salarylist.docx. 1p for writes tmp.bin. 1p for starts process lsass.exe with PID 4652.

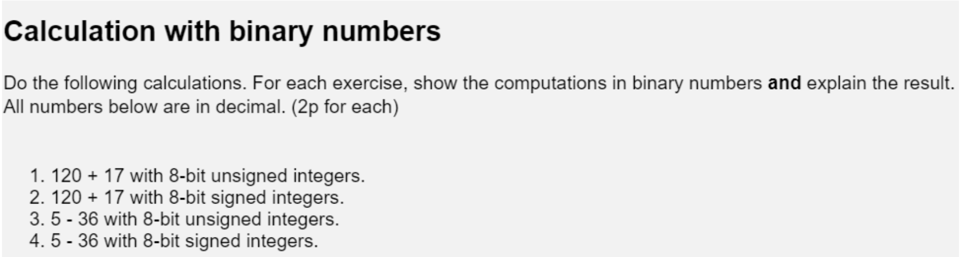
2. 1p for each of: Check for network activity in procmon, investigate the process lsass.exe in procmon, look for registry activities in procmon, or other reasonable actions. Max 2p.

3. 1p for each of these: (Other reasonable ways to locate in IDA where salarylist.docx, tmp.bin and lsass.exe are accessed are accepted)

   - Locate where in calc.exe the files salarylist.docx and tmp.bin are accessed, either by finding the file paths/names or look for the operations CreateFile/WriteFile you saw accessing them in procmon.
   - Locate where lsass.exe is started, either by finding the file path/name or look for the operations Process Create you saw in procmon.

4. Max 2p. Possible answers: locating the file lsass.exe and analyze it in IDA, check what tmp.bin contains, check for network activity by both lsass.exe and calc.exe, ... Make sure to see this question in relation to the previous answers. Extensive answers earlier, which have exceeded the requirements, can be given points here.

NOTE: Full score to the entire exercise should not be given without pointing out that lsass.exe with PID 4652 is another file than the legitimate lsass.exe with PID 632.



Calculation with binary numbers

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. 120 + 17 with 8-bit unsigned integers.
2. 120 + 17 with 8-bit signed integers.
3. 5 - 36 with 8-bit unsigned integers.
4. 5 - 36 with 8-bit signed integers.

1p for each correct answer, 1p for each correct binary calculation with explanation.

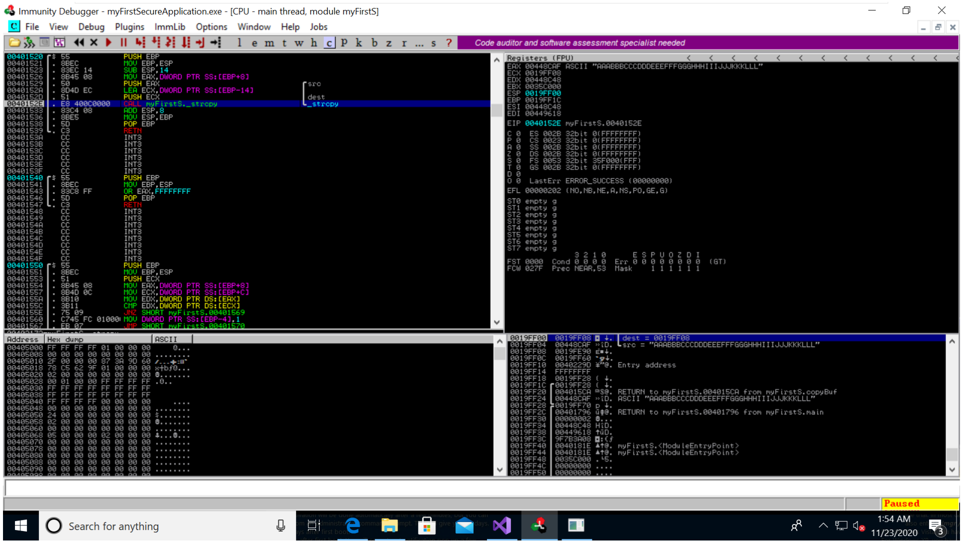
1. Unsigned: $120 + 17 = 01111000_2 + 10001_2 = 10001001_2 = 137$. No problems.

2. Signed: $120 + 17 = 01111000_2 + 11_2 = 10001001_2 = -119$. Signed 8-bit integers go from -128 to 127 before they wrap around.

3. Unsigned: $5 - 36 = 101_2 - 100100_2 = 11100001_2 = 225$ Unsigned 8-bit integers go from 0 to 255, so they wrap around at 0.

4. Signed: $5 - 36 = 101_2 - 100100_2 = 11100001_2 = -31$. No problems.

4 Debugging a process



Immunity Debugger - myFirstSecureApplication.exe - [CPU - main thread, module myFirstS]

File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?

Code auditor and software assessment specialist needed

```
00401520 ┌$ 55             PUSH EBP
00401521 │. 8BEC           MOV EBP,ESP
00401523 │. 83EC 14        SUB ESP,14
00401526 │. 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
00401529 │. 50             PUSH EAX                                   ┌src
0040152A │. 8D4D EC        LEA ECX,DWORD PTR SS:[EBP-14]              │
0040152D │. 51             PUSH ECX                                   │dest
0040152E │. E8 400C0000    CALL myFirstS._strcpy                      └_strcpy
00401533 │. 83C4 08        ADD ESP,8
00401536 │. 8BE5           MOV ESP,EBP
00401538 │. 5D             POP EBP
00401539 └. C3             RETN
0040153A    CC             INT3
0040153B    CC             INT3
0040153C    CC             INT3
0040153D    CC             INT3
0040153E    CC             INT3
0040153F    CC             INT3
00401540 ┌$ 55             PUSH EBP
00401541 │. 8BEC           MOV EBP,ESP
00401543 │. 83C8 FF        OR EAX,FFFFFFFF
00401546 │. 5D             POP EBP
00401547 └. C3             RETN
00401548    CC             INT3
00401549    CC             INT3
0040154A    CC             INT3
0040154B    CC             INT3
0040154C    CC             INT3
0040154D    CC             INT3
0040154E    CC             INT3
0040154F    CC             INT3
00401550 ┌$ 55             PUSH EBP
00401551 │. 8BEC           MOV EBP,ESP
00401553 │. 51             PUSH ECX
00401554 │. 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
00401557 │. 8B4D 0C        MOV ECX,DWORD PTR SS:[EBP+C]
0040155A │. 8B10           MOV EDX,DWORD PTR DS:[EAX]
0040155C │. 3B11           CMP EDX,DWORD PTR DS:[ECX]
0040155E │. 75 09          JNZ SHORT myFirstS.00401569
00401560 │. C745 FC 01000  MOV DWORD PTR SS:[EBP-4],1
00401567 │. EB 07          JMP SHORT myFirstS.00401570
```

```
Registers (FPU)
EAX 00448CAF ASCII "AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLL"
ECX 0019FF08
EDX 00448C48
EBX 0035C000
ESP 0019FF00
EBP 0019FF1C
ESI 00448C48
EDI 00449618

EIP 0040152E myFirstS.0040152E

C 0  ES 002B 32bit 0(FFFFFFFF)
P 0  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 35F000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address   Hex dump                  ASCII
00405000  FF FF FF FF 01 00 00 00       ☺...
00405008  00 00 00 00 00 00 00 00   ........
00405010  2F 00 00 00 87 3A 9D 60   /...ç:¥`
00405018  78 C5 62 9F 01 00 00 00   x┼bƒ☺...
00405020  02 00 00 00 00 00 00 00   ☻.......
00405028  00 01 00 00 FF FF FF FF   .☺..
00405030  FF FF FF FF FF FF FF FF
00405038  FF FF FF FF FF FF FF FF
00405040  FF FF FF FF 00 00 00 00       ....
00405048  00 00 00 00 00 00 00 00   ........
00405050  24 00 00 00 00 00 00 00   $.......
00405058  02 00 00 00 00 00 00 00   ☻.......
00405060  00 00 00 00 00 00 00 00   ........
00405068  05 00 00 00 02 00 00 00   ♣...☻...
00405070  00 00 00 00 00 00 00 00   ........
00405078  00 00 00 00 00 00 00 00   ........
00405080  00 00 00 00 00 00 00 00   ........
00405088  00 00 00 00 00 00 00 00   ........
00405090  00 00 00 00 00 00 00 00   ........
```

```
0019FF00   0019FF08  ■ ↓.  ┌dest = 0019FF08
0019FF04   00448CAF  »îD.  └src = "AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLL"
0019FF08   0019FE90  É■↓.
0019FF0C   0019FF60  `■↓.
0019FF10   0040229D  ¥"@.  Entry address
0019FF14   FFFFFFFF
0019FF18   0019FF28  ( ↓.
0019FF1C  ┌0019FF28  ( ↓.
0019FF20  │004015CA  ╩§@.  RETURN to myFirstS.004015CA from myFirstS.copyBuf
0019FF24  │00448CAF  »îD.  ASCII "AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLL"
0019FF28  ┌0019FF70  p ↓.
0019FF2C  │00401796  û↨@.  RETURN to myFirstS.00401796 from myFirstS.main
0019FF30  │00000002  ☻...
0019FF34  │00448C48  HîD.
0019FF38  │00449618  ↑ûD.
0019FF3C  │9F7B3A08  ■:{ƒ
0019FF40  │0040181E  ▲↑@.  myFirstS.<ModuleEntryPoint>
0019FF44  │0040181E  ▲↑@.  myFirstS.<ModuleEntryPoint>
0019FF48  │0035C000  .└5.
0019FF4C  │00000000  ....
0019FF50  │00000000  ....
```

Paused

```
void copyBuf(char* str) {
    char buf[20];
    strcpy(buf, str);
}
int main(int argc, char** argv) {
    std::cout << "Hello Secure Application!\n";
    copyBuf(argv[1]);
    return 0;
}
```

Examine the screenshot and the source code.

1. What is happening here? (3p)
2. Identify the relevant entries in the stack. What are the relevant entries, and what are their values? (4p)
3. How will the stack look after returning from strcpy? Which relevant entries are changed, and what are the consequences? (3p)
4. Explain what ASLR, DEP and stack cookies are and whether they are relevant here. (3p)
5. Would you recommend the programmer to improve the code? How? (2p)

1. 1p for stack buffer overflow. 1p for missing length check of input. 1p for seeing that the input argument in the screenshot is larger than the local buffer.
2. Half score if value is missing. Max 4:
   - 1p for: Strcpy takes two arguments, first argument: destination (value: 0019ff08), second argument: source (value: 00448caf, the address to the string “AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLL”, but not the string itself)
   - 1p for: Local variables (buf): The value is what is located at 0019ff08 and 20 bytes further, but what is there now is trash

from earlier. Will be overwritten. (values: 0019fe90.....) Here an explanation is more relevant than the actual values

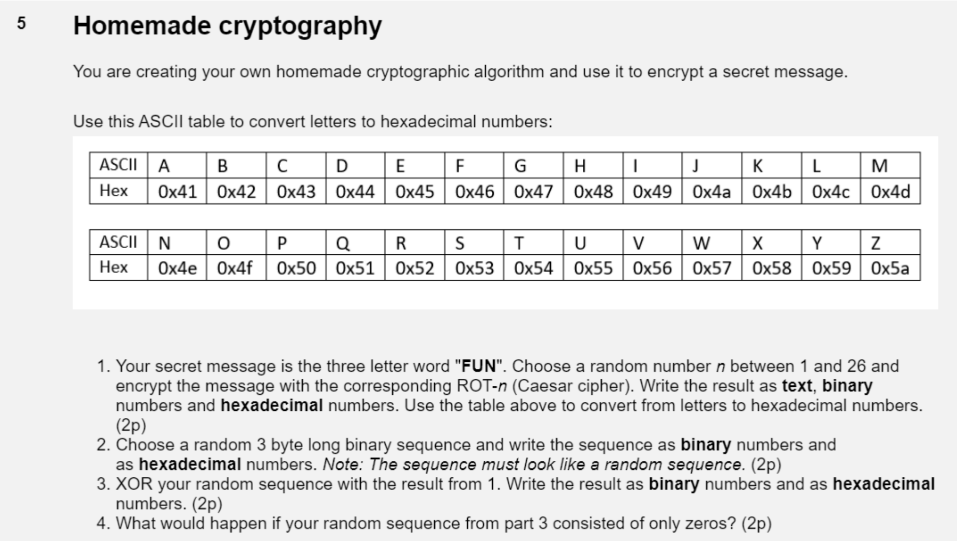
- 1p for: Saved EBP (value: 0019ff28)
- 1p for: Saved EIP (return address) (value: 004015ca)
- Not so relevant: Argument to copyBuf: value 00448caf. May give a point for identifying this if not 4 points are reached.

3. 3p for an explanation like: Starting from address 0019ff08 (destination, address of local variable buf), the string “AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLL” is written over what was there, resulting in:
   0019ff08 AAAB
   0019ff0c BBCC
   0019ff10 CDDD
   0019ff14 EEEF
   0019ff18 FFGG
   0019ff1c GHHH Saved EBP
   0019ff20 IIIJ Saved EIP
   0019ff24 JJKK
   0019ff28 KLLL
   The contents of the local variable buf is changed, and the values at the positions for saved EBP and saved EIP are changed. The process will finally return to EIP=4a494949 (=JIII). The reason for not returning to IIIJ is endianness.

4. Short explanations are sufficient. Half score for explaining each concept, half for arguing whether relevant here.

   - 1p for: ASLR is address space layout randomization, which causes executables and libraries to be loaded at unpredictable addresses. It would not prevent the overflow, but make it harder to predict where to jump to. Could be solved through an infoleak or if one library does not have ASLR enabled.
   - 1p for: DEP is data execution prevention, which prevents code in certain parts of memory to be executed. If an attacker wanted to put executable code (shellcode) on the stack and jump right to it, that would not work if DEP made the stack nonexecutable. Could be bypassed by ROP.
   - 1p for: Stack cookie is a random value placed on the stack after the local variables before the function starts, and it is checked before returning. If an overflow has occurred such that the value has changed, it is detected. The overflow would not have been prevented, but it is detected before an attacker is able to return to the overwritten EIP.

5. 2p for something like: Yes, the programmer should in copyBuf check that the length of the received string is not longer than buf. Suggesting use of strncpy (correctly) is also accepted. Only suggesting a length check without saying anything about where is not sufficient for full score.



5

Homemade cryptography

You are creating your own homemade cryptographic algorithm and use it to encrypt a secret message.

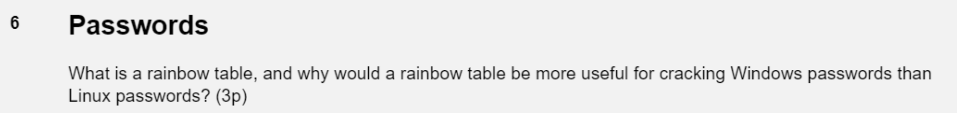
Use this ASCII table to convert letters to hexadecimal numbers:

ASCII	A	B	C	D	E	F	G	H	I	J	K	L	M
Hex	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4a	0x4b	0x4c	0x4d

ASCII	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Hex	0x4e	0x4f	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5a

1. Your secret message is the three letter word "**FUN**". Choose a random number *n* between 1 and 26 and encrypt the message with the corresponding ROT-*n* (Caesar cipher). Write the result as **text**, **binary** numbers and **hexadecimal** numbers. Use the table above to convert from letters to hexadecimal numbers. (2p)
2. Choose a random 3 byte long binary sequence and write the sequence as **binary** numbers and as **hexadecimal** numbers. *Note: The sequence must look like a random sequence.* (2p)
3. XOR your random sequence with the result from 1. Write the result as **binary** numbers and as **hexadecimal** numbers. (2p)
4. What would happen if your random sequence from part 3 consisted of only zeros? (2p)

1. 1p for correct ROT-n cipher, 1p for correct conversion to binary and hexadecimal numbers.
2. 1p for writing a sequence of correct length that appears random, 1p for correct conversion.
3. 1p for correct XOR, 1p for correct conversion.
4. 2p for: Any number XOR-ed with zero is unchanged. It would result in only a ROT-n cipher, as the XOR step would not change the data.



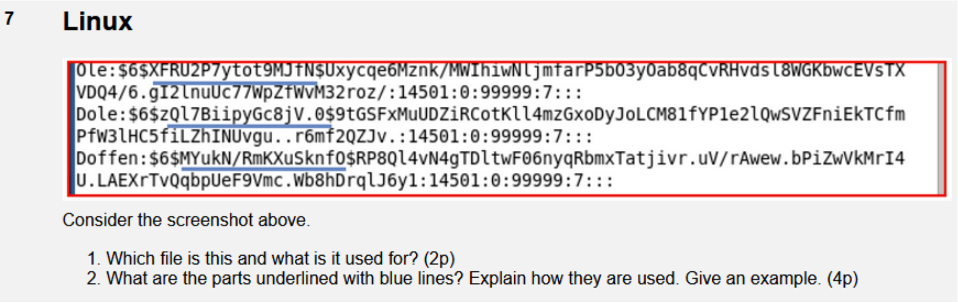
6

Passwords

What is a rainbow table, and why would a rainbow table be more useful for cracking Windows passwords than Linux passwords? (3p)

1p for explaining what a rainbow table is. From the slides:
A rainbow table is a precomputed table of passwords and their hashes, using clever methods for reducing the storage space required. Using rainbow tables, the attacker can just lookup the stored password hash in the table and get back the password.
To achieve full score (1p) no more details are needed, but the answer should not be an exact copy of the slides.

2p for: Windows passwords are not salted before hashing, and the same hashing algorithm is always used. Linux passwords are salted before hashing and one would therefore need one rainbow table per possible salt, which is not very feasible.



7 **Linux**

```
Ole:$6$XFRU2P7ytot9MJfN$Uxycqe6Mznk/MWIhiwNljmfarP5bO3yOab8qCvRHvdsl8WGKbwcEVsTX
VDQ4/6.gI2lnuUc77WpZfWvM32roz/:14501:0:99999:7:::
Dole:$6$zQl7BiipyGc8jV.0$9tGSFxMuUDZiRCotKll4mzGxoDyJoLCM81fYP1e2lQwSVZFniEkTCfm
PfW3lHC5fiLZhINUvgu..r6mf2QZJv.:14501:0:99999:7:::
Doffen:$6$MYukN/RmKXuSknfO$RP8Ql4vN4gTDltwF06nyqRbmxTatjivr.uV/rAwew.bPiZwVkMrI4
U.LAEXrTvQqbpUeF9Vmc.Wb8hDrqlJ6y1:14501:0:99999:7:::
```

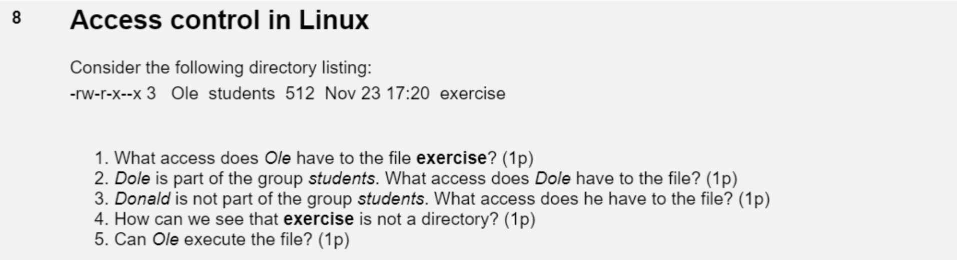
Consider the screenshot above.

1. Which file is this and what is it used for? (2p)
2. What are the parts underlined with blue lines? Explain how they are used. Give an example. (4p)

1. 1p for /etc/shadow, 1p for storing the hashed passwords and related information

2. 2p for: The underlined part is salt, it is concatenated with the password before it is hashed.

   2p for an example like: Let salt= abc and password=Passw0rd! Then the stored hash value in /etc/shadow would be the result of sha-512(abcPassw0rd!), if sha-512 is the used hashing algorithm. The example is also accepted if the salt is appended to the password since both possibilities have been mentioned in the course.



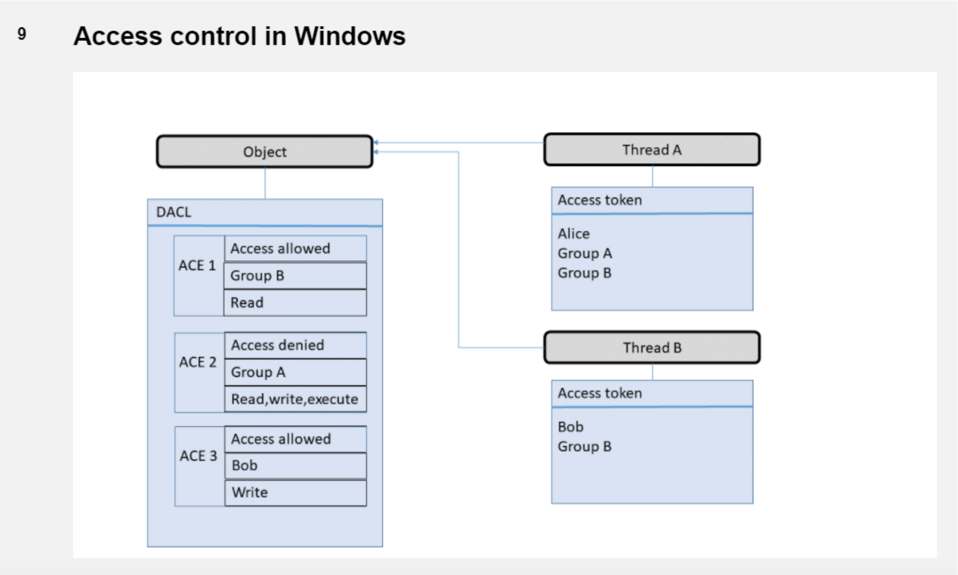
8 Access control in Linux

Consider the following directory listing:

```
-rw-r-x--x 3   Ole  students  512  Nov 23 17:20  exercise
```

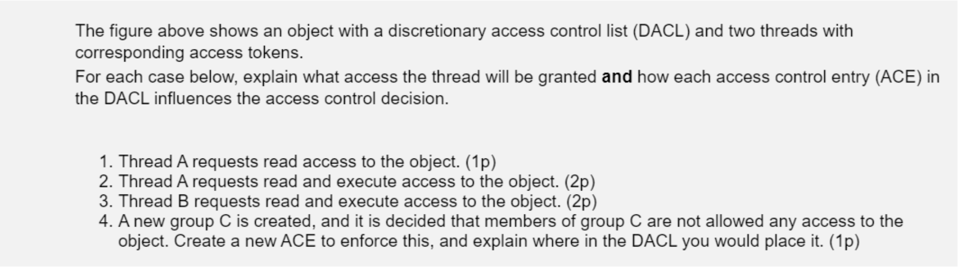
1. What access does *Ole* have to the file **exercise**? (1p)
2. *Dole* is part of the group *students*. What access does *Dole* have to the file? (1p)
3. *Donald* is not part of the group *students*. What access does he have to the file? (1p)
4. How can we see that **exercise** is not a directory? (1p)
5. Can *Ole* execute the file? (1p)

1. 1p for Ole has read and write access to the file.
2. 1p for Dole has read and execute access to the file.
3. 1p for Donald has execute access to the file (as part of others)
4. 1p for If it were a directory, the first entry would be a d, not a -.
5. 1p for Ole cannot execute the file directly, but as the file owner he can give himself execute access and thereafter execute the file.



9

Access control in Windows



The figure above shows an object with a discretionary access control list (DACL) and two threads with corresponding access tokens.
For each case below, explain what access the thread will be granted **and** how each access control entry (ACE) in the DACL influences the access control decision.

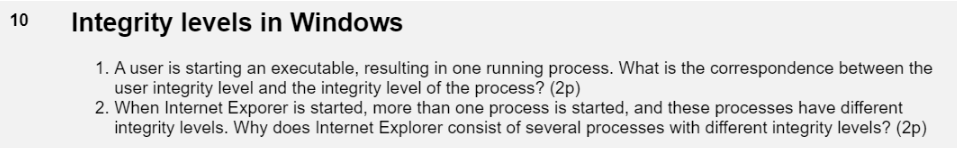
1. Thread A requests read access to the object. (1p)
2. Thread A requests read and execute access to the object. (2p)
3. Thread B requests read and execute access to the object. (2p)
4. A new group C is created, and it is decided that members of group C are not allowed any access to the object. Create a new ACE to enforce this, and explain where in the DACL you would place it. (1p)

1. 1p for Thread A is granted read access from ACE 1. The rest of the list is not checked, as all requested access is granted.

2. 2p for Thread A is denied access. From 1 we know that ACE 1 grants read access, but since ACE 2 denies execute access, no access is granted. The rest of the list is not checked.

3. 2p for Thread B is denied access. ACE 1 grants read access, ACE 2 does not apply because of wrong group, ACE 3 does not apply because write access is not requested. The end of the list is reached before all requested access is granted, and no access is granted.

4. 1p for: We add the following ACE at the top of the list:
   - Access denied
   - Group C
   - Read, write, execute

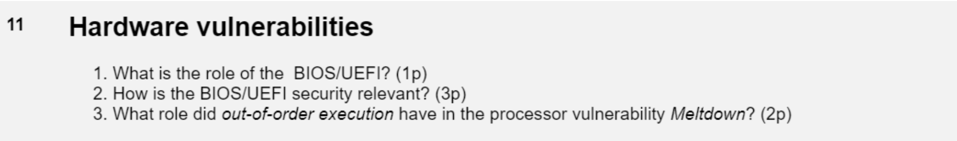


10 Integrity levels in Windows

1. A user is starting an executable, resulting in one running process. What is the correspondence between the user integrity level and the integrity level of the process? (2p)
2. When Internet Exporer is started, more than one process is started, and these processes have different integrity levels. Why does Internet Explorer consist of several processes with different integrity levels? (2p)

1. 2p for The integrity level of the started process is the minimum of the integrity level of the user and of the executable file. Alternative formulation: Processes a user starts receive the user's integrity level (medium if started normally or high if started as administrator) or low if the executable file's level is low.

2. 2p for: Internet Explorer starts one process with medium integrity level and one process per tab with low integrity level. This works as a sandboxing mechanism. The tabs may open malicious web pages, but if an attacker gets control over the process, it will only be as a low integrity process, making it harder to do damage or move further.



11 Hardware vulnerabilities

1. What is the role of the BIOS/UEFI? (1p)
2. How is the BIOS/UEFI security relevant? (3p)
3. What role did *out-of-order execution* have in the processor vulnerability *Meltdown*? (2p)

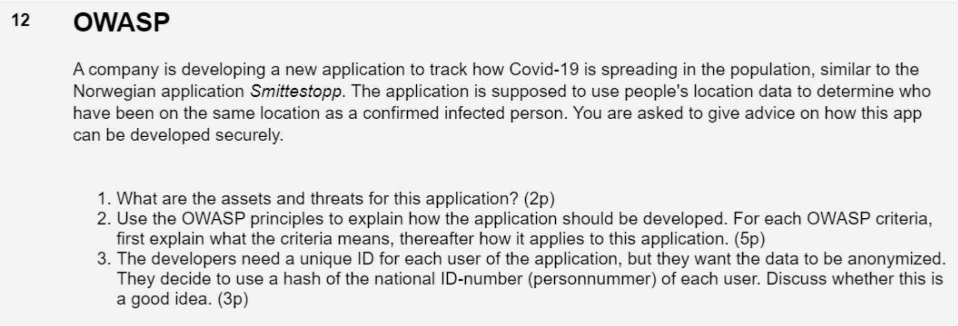
The first two answers below are copies from the slides. The answers should not be direct copies.

1. 1p for: The BIOS/UEFI is responsible for initializing the system when it boots, but also provides runtime services that are still accessible while the operating system is running.

2p 3p for:

- The BIOS is the first code that runs on the processor
  - ∗ can maliciously modify the OS image that it will load
- The BIOS has privileged access to all hardware
  - ∗ can talk to and reprogram all devices
- The BIOS provides code for runtime services (SMM) that will keep running below the operating system
  - ∗ a good place for malicious rootkits

3. 2p for: A process attempts to read something without having access. Since instructions are executed tentatively (out-of-order), the read instruction can be executed before it is discovered that necessary read access is missing, resulting in the interesting data being loaded to an internal cache/storage unit in the processor.



12 **OWASP**

A company is developing a new application to track how Covid-19 is spreading in the population, similar to the Norwegian application *Smittestopp*. The application is supposed to use people's location data to determine who have been on the same location as a confirmed infected person. You are asked to give advice on how this app can be developed securely.

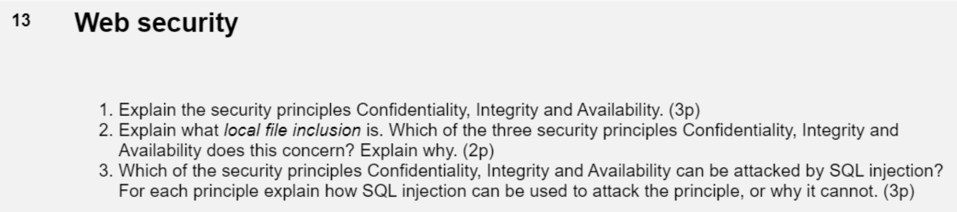
1. What are the assets and threats for this application? (2p)
2. Use the OWASP principles to explain how the application should be developed. For each OWASP criteria, first explain what the criteria means, thereafter how it applies to this application. (5p)
3. The developers need a unique ID for each user of the application, but they want the data to be anonymized. They decide to use a hash of the national ID-number (personnummer) of each user. Discuss whether this is a good idea. (3p)

1. Many possible answers. 1p for an answer showing reasonable understanding of assets and 1p for an answer showing reasonable answer of threats. Possible ideas: Assets: The location data of the users, and possibly other personal information stored in the app. Threats: Personal data fall in the wrong hands. Databases are tampered with or the app is silently malfunctioning resulting in the app giving wrong indications of possible close contacts.

2. **0.5p** 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

3. Max 3p. The answer should include that hash functions are hard (impossible) to reverse directly, but that it would be possible to create a table of all personal ID numbers with the corresponding hash values and use the table to look up the ID number corresponding to a given hash. It would therefore not be very anonymized.



13 **Web security**

1. Explain the security principles Confidentiality, Integrity and Availability. (3p)
2. Explain what *local file inclusion* is. Which of the three security principles Confidentiality, Integrity and Availability does this concern? Explain why. (2p)
3. Which of the security principles Confidentiality, Integrity and Availability can be attacked by SQL injection? For each principle explain how SQL injection can be used to attack the principle, or why it cannot. (3p)

1. 1p for each of: Confidentiality is about preventing unauthorized disclosure of information (unauthorized reading), Integrity is about preventing unauthorized modification of information (unauthorized writing), Availability is the property of being accessible and usable upon demand by an authorized entity.

2. 1p for: The attacker can include a local file of the webserver using the webpage and thereby gain access to it. 1p for Confidentiality, unauthorized reading.

3. SQL injection may attack all three principles. 1p for each principle with a reasonable explanation. Example: Avalability: If an attack deletes information from the database, the data is no longer available to authorized users.

The following is copied from a relevant slide, and answers should not be direct copies:

- Confidentiality: Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL injection vulnerabilities.

- Authentication: If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.

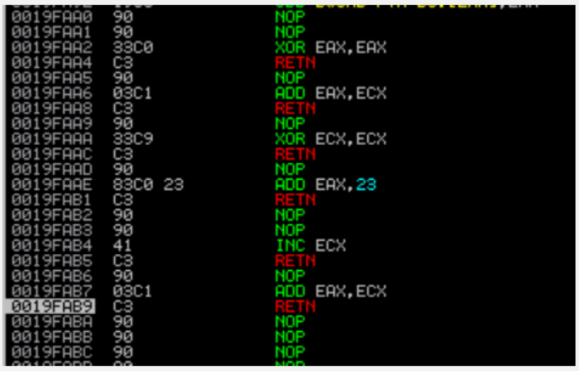
- Integrity: Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL injection attack.

14

ROP

1. What is ROP and what is it typically used for? (2p)
2. Given the somewhat constructed memory layout in the screenshot, what is the result of the following ROP chain? Explain each step, and the contents of relevant registers after returning from the last gadget. (4p)

ROP chain:
0019FAA2
0019FAAA
0019FAAE
0019FAB4
0019FAB4
0019FAB7



```
0019FAA0  90          NOP
0019FAA1  90          NOP
0019FAA2  33C0        XOR EAX,EAX
0019FAA4  C3          RETN
0019FAA5  90          NOP
0019FAA6  03C1        ADD EAX,ECX
0019FAA8  C3          RETN
0019FAA9  90          NOP
0019FAAA  33C9        XOR ECX,ECX
0019FAAC  C3          RETN
0019FAAD  90          NOP
0019FAAE  83C0 23     ADD EAX,23
0019FAB1  C3          RETN
0019FAB2  90          NOP
0019FAB3  90          NOP
0019FAB4  41          INC ECX
0019FAB5  C3          RETN
0019FAB6  90          NOP
0019FAB7  03C1        ADD EAX,ECX
0019FAB9  C3          RETN
0019FABA  90          NOP
0019FABB  90          NOP
0019FABC  90          NOP
```

1. 1p for what ROP is, 1p for what it is typically used for.

   From the solution to a previous exam: 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 chain do not need to be stored in executable memory to work, but the addresses must point to executable memory.

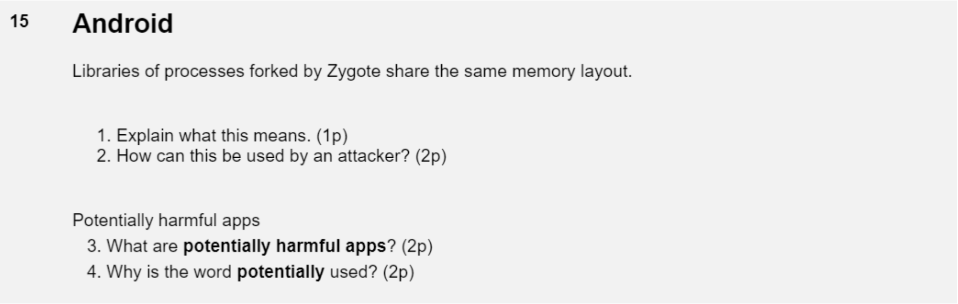
   Typically ROP chains are used to bypass DEP to make a piece of memory (where the attacker puts his code) executable.

2. 2p for explaining the steps, 2p for the values of the registers at the end:

   0019FAA2 xor eax,eax;retn //Set eax to zero
   0019FAAA xor ecx,ecx;retn //Set ecx to zero
   0019FAAE add eax, 23;retn //Add 23 to eax. eax = 0+23 = 23.

```
0019FAB4 inc ecx; retn //Add 1 to ecx. ecx = 0+1 = 1.
0019FAB4 inc ecx; retn //Add 1 to ecx. ecx = 1+1 = 2.
0019FAB7 add eax,ecx;retn // eax = eax+ecx = 23+2 = 25
```

ecx=2, eax = 25



15 **Android**

Libraries of processes forked by Zygote share the same memory layout.

1. Explain what this means. (1p)
2. How can this be used by an attacker? (2p)

Potentially harmful apps

3. What are **potentially harmful apps**? (2p)
4. Why is the word **potentially** used? (2p)

1. 1p for: Libraries are loaded at the same addresses in these processes.
2. 2p for: A memory leak in one application can give the attacker knowledge of addresses that can be used in an attack on another app.
3. 2p for: Apps that disobey Google Play's guidelines. Could be called malware.
4. 2p for: The word potentially is used since some apps, like apps rooting a device, are classified as potentially harmful apps, but these apps may still be wanted by some users. A user wanting to root a device would not consider an app doing so a harmful app, but rather a helpful app. Also, some malware are targeted at users in some parts of the world and will only do malicious actions in the phones in these parts of the world, while they will do no harm to users in other parts of the world.