VauLSMorg

02 SPPC - Security, Protection, Privacy & C
Aneka Soal Ujian Sistem Operasi
Rahmat M. Samik-Ibrahim et.al.

1. **2016-1**

   Circle or cross: "**T**" if True – "**F**" if False.

   T / F Principle of least privilege: programs, users and systems should be given unlimited privileges to perform their tasks.

   T / F Computer system objects may be hardware or software.

   T / F Breach of confidentiality involves unauthorized reading of data.

   T / F Breach of integrity involves preventing legitimate use of the system.

   T / F Breach of availability involves unauthorized destruction of data.

   T / F An attack is always malicious and never accidental.

   T / F Script kiddies are persons who write scripts or codes to crack into computers.

2. **2016-2**

   Circle or cross: "T" if True – "F" if False.

```
$ ls -al
total 12
drwxr-xr-x 3 demo demo 4096 Oct 17 17:05 .
drwxrwxrwt 8 root root 4096 Oct 17 17:04 ..
dr-x--x--x 2 demo demo 4096 Oct 17 17:06 tmp
```

   T / F All users can enter directory "`tmp/`".

   T / F Only user "demo" can read directory "`tmp/`".

   T / F A cyber breach occurs when someone accesses a database through an insufficiently secured network connection.

   T / F A physical breach occurs when an unauthorized person is able to physically access a piece of equipment.

   T / F "**Security**" is an internal problem. On the other hand, "**protection**" also requires consideration of the external environment.

   T / F A backdoor is a method of bypassing normal authentication.

   T / F A trojan horse is an example of a backdoor.

   T / F A Keylogger is the action of recording (covertly) a keyboard.

3. **2017-1**

Circle or cross: "T" if True – "F" if False.

T / F Security is a mechanism for controlling processes or users to resources (Yakoob et. al.).

T / F Operating Systems automatically apply permissions to files and folder, however users can manually apply them too (Yakoob et. al.).

T / F Symmetric cryptography is much faster than asymmetric one.

T / F Protection is strictly an internal problem. On the other hand, security is strictly an external problem.

T / F The security mechanisms control access to a system. On the other hand, protection system prevents unauthorized access.

T / F The three aspects to a protection mechanism are authentication, authorization, and access enforcement.

T / F In GNU/Linux, users can be organized into groups, with a single Access Control List (ACL) for an entire group.

T / F Trojan horses are often computer games software infected with viruses.

T / F An access list is a list of objects and the operations allowed on those objects for each domain (OSC9).

T / F If users are allowed to perform their own I/O operation, system integrity will be guaranteed (OSC9).

C Programing

```
001 /*
002  * (c) 2017 Rahmat M. Samik-Ibrahim
003  * This is free software.
004  * REV01 Thu Mar 30 17:32:33 WIB 2017
005  * START Thu Mar 30 12:13:58 WIB 2017
006  */
007
008 #include <stdio.h>
```

```
010 int tambah(int ii, int jj) {
011    return ii + jj;
012 }
013
014 void main(void) {
015    int ii = 4;
016    printf("The return of tambah is %d\n", tambah(1,ii));
017 }
```

Program Output (Line 016):

4. **2017-2**

Principle of least (01) dictates that programs, users, and even systems be given just enough privileges to perform their tasks (OSC9). (02) is strictly an internal problem (OSC9). (03) requires also consideration of the external environment within which the system operates (OSC9). A system is (04) if its resources are used and accessed as intended under all circumstances (OSC9). Security is often deployed for (05) against external threats (OSC9). Breach of (06) involves unauthorized reading of data (OSC9). Breach of (07) involves unauthorized modification of data (OSC9). Breach of (08) involves unauthorized destruction of data (OSC9). (09) of service involves unauthorized use of resources (OSC9).

(10) of service involves preventing legitimate use of the system (OSC9). (11) is when one participant in a communication pretends to be someone else (OSC9). In a session (12), an active communication session is intercepted (OSC9). A code segment that misuses its environment is called a (13) (OSC9). (14) are self-replicating and are designed to infect other programs (OSC9). A (15) is a process that uses the spawn mechanism to duplicate itself (OSC9). In a (16) encryption algorithm, the same key is used to encrypt and to decrypt (OSC9). In an (17) encryption algorithm, there are different encryption and decryption keys (OSC9). (18) are very useful in that they enable anyone to verify the authenticity of the message (OSC9). (19) is the ability of an individual or group to seclude themselves, or information about themselves, and thereby express themselves selectively (WIKI).

Match the number of the sentence above with these following phrases:

[ ] Asymmetric	[ ] Availability	[ ] Confidentiality	[ ] Denial
[ ] Digital Signatures	[ ] Hijacking	[ ] Integrity	[ ] Masquerading
[ ] Privacy	[ ] Privilege	[ ] Protection	[ ] Protection
[ ] Secure	[ ] Security	[ ] Symmetric	[ ] Theft
[ ] Trojan Horse	[ ] Viruses	[ ] Worm	

C Programing

```
001 /*
002  * (c) 2017 Rahmat M. Samik-Ibrahim
003  * http://rahmatm.samik-ibrahim.vlsm.org/
004  * This is free software.
005  * REV00 Mon Oct 16 21:15:03 WIB 2017
006  * START Mon Oct 16 21:15:03 WIB 2017
007  */
008
009 #include <stdio.h>
010
011 char globalChar='a';

012
013 char* getGlobal(void) {
014    char* charPTR=&globalChar;
015    printf("getGlobal1 %c\n", globalChar);
016    *charPTR='b';
017    printf("getGlobal2 %c\n", *charPTR);
018    return charPTR;
019 }
021 void main (void) {
022    char localChar='c';
023    printf("==== main1 %c\n", localChar);
024    localChar=*getGlobal();
025    printf("==== main2 %c\n", localChar);
026 }
```

Program Output:

5. **2018-1**

An (01) list is a list for each object consisting of the domains with a nonempty set of access rights for that object. A (02) list is a list of objects and the operations allowed on those objects for each domain. Proper access to the hardware is necessary for system (03). It will be difficult to (04) a system if users are allowed to access the hardware. The (05) principle is useful in limiting the amount of damage from a faulty process. Typically, a breach of confidentiality is the goal of an (06). Breach of integrity can result in passing of (07) to an innocent party. (08) is a common example of breach of availability. Theft of service involves (09) use of resources. (10) is not an attack but rather a means for a cracker to detect a system's vulnerabilities to attack.

Match the number of the sentence above with these following phrases:

[ ] access [ ] capability [ ] integrity [ ] intruder

[ ] liability [ ] need-to-know [ ] Port scanning [ ] protect

[ ] unauthorized [ ] Website defacement

What is the output of this following program:

```
001 /* (c) 2018 This is a free program */
002 /* Rahmat M. Samik-Ibrahim         */
003
004 #include <stdio.h>
005
006 void main(void) {
007    char string[]="HALLO";
008    printf("START\n");
009    printf("%s\n",  string);
010    printf("%c\n", *string);
011    printf("%c\n",  string[1]);
012    printf("STOP\n");
013 }
```

6. **2018-2 (79%)**

(01) is a measure of confidence that the integrity will be preserved. (02) is the set of access control mechanisms. A system is (03) if its resources are used and accessed as intended. A (04) resource can defend against use or misuse. A (05) is the potential for a security violation, whereas an (06) is an attempt to break security. (07) is when a participant in a communication pretends to be someone else. Mechanisms determine (08) something will be done; policies decide (09) will be done. A list of objects together with the operations allowed on those objects is known as (10) list.

Match the number of the sentence above with these following phrases:

[ ] attack (100%) [ ] capability (90%) [ ] how (90%) [ ] Masquerading (100%) [ ] protected (70%)

[ ] Protection (70%) [ ] secure(60%) [ ] Security (70%) [ ] threat(100%) [ ] what(90%)

What is the output of this following program (76%):

```
001 /* (c) 2018 This is free software *
002  * NOTE: ASCII 61H = a; 62H = b   */
003 #include <stdio.h>
004 void main(void) {
005    unsigned int   ii='a';
006    unsigned char  ch='b';
007    unsigned char* st="dcba";
008    printf("START\n");
009    printf(" ii    = %X or %c\n",   ii,  ii);
010    printf(" ch    = %X or %c\n",   ch,  ch);
011    printf("*st    = %X or %c\n",  *st, *st);
012    printf(" st[2] = %X or %c\n", st[2], st[2]);
013    printf("STOP\n");
014 }
```

7. **2019-1 (81.0%)** (Ref: Schilberschatz et.al.)

   (01) ensures the authentication of system users to protect the integrity as well as the physical.

   The (02) mechanism must provide a means for specifying the controls to be imposed.

   A(n) (03) is an attempt to break security.

   A(n) (04) is the potential for a security violation

   (05) involves unauthorized destruction of data.

   (06) involves unauthorized use of resources.

   (07) is pretending to be someone one is not.

   Computer attacks such as [08] require human interaction, while [09] are self-perpetuating.

   (10) is capturing data as it is transmitted over a network.

   (11) attacks are launched from multiple sites at once, toward a common target.

   A (12) is a token that gives the system permission to access an object.

   Match the number(s) in the sentence above with these following phrases:

[ ] ATTACK (97%)	[ ] CAPABILITY (79%)	[ ] BREACH OF AVAILABILITY (87%)	[ ] DISTRIBUTED DENIAL-OF-SERVICE (72%)
[ ] MASQUERADING (93%)	[ ] PROTECTION (64%)	[ ] SECURITY (63%)	[ ] SNIFFING (79%)
[ ] THEFT OF SERVICE (88%)	[ ] THREAT (95%)	[ ] VIRUSES (44%)	[ ] WORMS (49%)

   What is the output of this following program (89%):

```
001 // (c) 2019 This is Free Software R01
002 // Rahmat M. Samik-Ibrahim 20190324-234700
003 // Clue: ASCII 'a' is 0x61.
004 #include <stdio.h>
005 void main (void) {
006    unsigned char ch1='a', ch2='y', ch3='z';
007    printf("START\n");
008    printf("1) ch1 = %c or ASCII %#X\n", ch1, ch1);
009    ch1 = ch1 + ch3 - ch2;
010    printf("2) ch1 = %c or ASCII %#X\n", ch1, ch1);
011    printf("STOP\n");
012 }
```

8. **2019-2 (60%)**

(01) ensures the authentication of system users to protect the integrity as well as the physical. The (02) mechanism must provide a means for specifying the controls to be imposed. Encryption limits the domain of (3) of data, while authentication limits the domain of (4). (05) involves unauthorized destruction of data. (06) involves unauthorized use of resources. A (07) acts in a clandestine or malicious manner rather than simply performing its stated function. Computer attacks such as [08] require human interaction, while [09] are self-perpetuating. (10) is capturing data as it is transmitted over a network. (11) attacks are launched from multiple sites at once, toward a common target. A (12) is a token that gives the system permission to access an object.

Match the number(s) in the sentence above with these following phrases:

[ ] CAPABILITY (74%)	[ ] BREACH OF AVAILABILITY (87%)	[ ] DISTRIBUTED DENIAL-OF-SERVICE (63%)	[ ] RECEIVERS (43%)
[ ] TROJAN HORSE (67%)	[ ] PROTECTION (79%)	[ ] SECURITY (89%)	[ ] SNIFFING (70%)
[ ] THEFT OF SERVICE (87%)	[ ] SENDERS (46%)	[ ] VIRUSES (49%)	[ ] WORMS (63%)

What is the output of this following program (52%):

```
001 // (c) 2019 This is Free Software R00
002 // Rahmat M. Samik-Ibrahim 20191022-1854
003 #include <stdio.h>
004 int aa=0;
005 int* function(int* bb) {
006    return bb;
007 }
008 void main (void) {
009    int cc=aa++;
010    printf("START\n");
011    printf("1. aa = %d\n", aa);
012    printf("2. *function()=%d\n", *function(&cc));
013    printf("3. cc = %d\n", ++cc);
014    printf("STOP\n");
015 }
```

9. **2020-1**

Define/explain briefly (maximum two sentences):

(a) "Personally Identifying Information (PII)" or "Personal Data" or "Personal Information":

(b) "Password Manager":

(c) "Strong Password":

(d) "Two-Factor Authentication":

What is the output of this following program:

```
001 // (c) 2020 This is Free Software R01
002 // Rahmat M. Samik-Ibrahim 2020 0310Tue1501
003 #include <stdio.h>
004 int returnInt(int ii) {
005    return ii;
006 }
007 char returnChar(char cc) {
008    return cc;
009 }
010 void main(void) {
011     int ii=0x41424344;
012     printf("returnChar=%c\n",
013             returnChar((char) ii));
014     printf("returnChar=%#x\n",
015       (int) returnChar((char) ii));
016     printf("returnInt==%c\n",
017      (char) returnInt(ii));
018     printf("returnInt==%#x\n",
019             returnInt(ii));
020 }
```

HINT#1: ASCII '0x41' = 'A'
HINT#2: This is a Little Endian system.

Program Output: