CYENG 312: Trusted Operating System (OS) – Fall 2022: Final Exam

Instructor: Dr. Shayan (Sean) Taheri

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All your responses to the assignment questions should be included in a single compressed file to be uploaded in the Gannon University (GU) – Blackboard Learn environment.

Question 1-A. Specify usages of Authentication, Access Control, and Auditing.

Question 1-B. Provide definitions for Authentication and Access Control.

Question 1-C. Mention different variations of passwords as well as the threats to them.

Question 1-D. Explain Password Entropy, how it is estimated, how its measurement is improved, and how it is balanced.

Question 1-E. Provide a discussion on the Spoofing and the Keylogging attacks along with their related defenses.

Question 1-F. Discuss how passwords are protected over an insecure channel.

Question 2-A. Explain similarities and differences of symmetric and asymmetric encryption algorithms.

Question 2-B. Provide how public key encryption and public key authentication are combined using descriptive, algorithmic, and system architecture representations.

Question 2-C. Show the general format of the system architecture of block ciphers. Discuss how it can be more secure through usage of multiple of keys.

Question 2-D. Explain different modes of block ciphers with formulas and figures.

Question 2-E. Show the general components in a block cipher as well as the ones in the AES cryptographic algorithm with provision of explanations.

Question 2-F. Mention the goals and the applications of Secure Hash Functions.

Question 2-G. Explain the computations in ElGamal Signature Algorithm.

Question 2-H. Specify the layers and their associated components in the SSH Layered Architecture.

Question 2-I. Provide definitions for Galois Fields.

Question 3-A. Explain the area of Computer Forensics with provision of its example applications and users.

Question 3-B. Specify types of reasons for Forensic Evidences with five examples for each of them.

Question 3-C. Provide system architecture for the steps of computer forensics with providing related examples.

Question 3-D. Discuss the function of "Comparison Against Known Data" with having a figure for the process. Specify the new technologies that are used in this process and the governmental entity that provides it.

Question 3-E. Explain Forensic and Anti-Forensic based on an Investigation-Response System with providing a figure for the processes.

Question 3-F. Mention methods of hiding and detecting/recovering data in the area of computer forensic.

Question 4. Explain the computations and the outputs of the following four codes.

Q4 – Code 1.

```
#include <stdio.h>
#include <string.h>
void pr(char *s, char *p) {
int i = strcmp(s,p);
if ( i == 0 )
puts("strcmp == 0");
else if ( i < 0 )
puts("strcmp < 0");
else
puts("strcmp > 0");
}
int main() {
char s[9] = "abc-78";
char *p = "abf-192";
pr(s, p); pr( s + 3, p + 5);
return 0;
}
```

Q4 – Code 2.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
int main(int argc, char *argv[])
```

```
{
   printf("I am: %d\n", (int) getpid());

   pid_t pid = fork();
   printf("fork returned: %d\n", (int) pid);

   if (pid < 0) { /* error occurred */
     perror("Fork failed");
   }
   if (pid == 0) { /* child process */
     printf("I am the child with pid %d\n", (int) getpid());
               printf("Child process is exiting\n");
               exit(0);
       }
   /* parent process */
   printf("I am the parent waiting for the child process to end\n");
       wait(NULL);
       printf("parent process is exiting\n");
       return(0);
}
```

Q4 – Code 3.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int tmp=0;
void my_function(int count) {
 tmp=tmp+1;
 printf("%d: Value= %d\n", count, tmp);
}
int main(void)
{
 int f=1, i;
 for (i=0; i<3; i++) {
 if (f>0)
 f=fork();
 if (f==-1) {
 printf("fork error....\n");
 exit(-1);
 }
 if (f==0)
 break;
 }
 if (f == 0) {
 my_function(i);
 }
 else {
 printf("Main: Created %d procs.\n", i);
 }
 return 0;
}
```

Q4 – Code 4.

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
int tmp=0;
void *my_function(void *threadid) {
 tmp=tmp+1;
 printf("%d: Value= %d\n", threadid, tmp);
 pthread_exit(NULL);
}
int main(void)
{
 pthread_t threads[3];
 int rc, i;

 for(i=0; i<3;i++) {
 rc = pthread_create(&threads[i], NULL,
 my_function, (void *)i);
 if (rc) {
 printf("thread creation error ...\n");
 exit(-1);
 }
 }
 printf("main(): Created %d threads.\n", i);
 pthread_exit(NULL);
 return 0;
}
```