代號：41440
頁次：4－1

115年公務人員特種考試司法人員、法務部調查局調查人員及海岸巡防人員考試試題

考 試 別：調查人員
等　　別：三等考試
類 科 組：資訊科學組
科　　目：作業系統及系統程式
考試時間：2小時　　座號：＿＿＿＿＿＿

※注意：㈠禁止使用電子計算器。
㈡不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。
㈢本科目除專門名詞或數理公式外，應使用本國文字作答。

一、某政府機關建置線上申辦系統，採用三層式分散式系統架構（Three-tier Distributed Architecture），其中 Web Server、Application Server 與 Database Server 部署於虛擬化平台（Virtualization Platform）之虛擬機（Virtual Machine, VM），部分應用程式運用容器（Container）與微服務（Microservices）架構部署，透過 API 進行服務間通訊。近期系統發生異常事件，系統管理者需於不中斷主要服務的情況下完成維護及安全處置。

- 部分 Web Server CPU 使用率持續偏高，導致系統回應時間明顯增加。
- 某一微服務無法正常存取資料庫，造成部分功能無法使用。
- 某容器遭惡意程式入侵，疑似透過橫向移動（Lateral Movement）攻擊其他容器或虛擬機。

㈠對於上述系統異常事件，請列舉可能發生之二項原因；並請提出三項系統管理或資訊安全措施，以降低攻擊影響並維持服務可用性，並說明各項措施之目的。（15分）

㈡請製表比較虛擬機（Virtual Machine, VM）、容器（Container）與微服務（Microservices）三種技術於系統部署、資源管理、擴充性及故障隔離四方面之差異；並說明在本案例中如何整合上述技術，以提升系統之可用性、擴充性及資訊安全。（10分）

二、下列 Linux 程式使用 fork()建立子行程，並於父、子行程中分別建立執行緒。請閱讀下列 C 程式碼並回答問題。假設所有系統呼叫皆成功；PID 與 Thread ID 的實際數值不固定，輸出順序則依 pthread_join()與 wait()所形成的先後關係判斷。

```
01  // 省略標頭檔
02  int value = 10;
03  void *worker(void *arg){
04        int *local = (int *)arg;
05        value = value + (*local);
06        printf("PID=%d, TID=%lu, value=%d\n",
07                    getpid(), (unsigned long)pthread_self(), value);
08        return NULL;
09  }
10  int main(void){
11        int x = 5;
12        pthread_t tid;
13        pid_t pid = fork();
14        if (pid == 0) {
15              x = 20;
16              value = value + x;
17              pthread_create(&tid, NULL, worker, &x);
18              pthread_join(tid, NULL);
19              printf("C_PID=%d, value=%d, x=%d\n", getpid(), value, x);
20        } else {
21              x = 30;
22              value = value + x;
23              pthread_create(&tid, NULL, worker, &x);
24              pthread_join(tid, NULL);
25              wait(NULL);
26              printf("P_PID=%d, value=%d, x=%d\n", getpid(), value, x);
27        }
28        return 0;
29  }
```

㈠請分析上述程式碼之執行結果，分別說明父行程與子行程中變數 value 與 x 的最終值，並列出可能的輸出內容。（10 分）

㈡請從作業系統行程與執行緒管理的角度回答下列問題。（15 分）

1. 請說明 fork()執行後，父行程與子行程分別擁有之獨立或共享資源，包含虛擬位址空間、程式碼、全域變數、堆疊、檔案描述符及行程識別碼之關係。
2. 請說明同一行程內的多個執行緒共享及各自獨立擁有之資源，包含程式碼區、資料區、Heap、Stack、暫存器與 Thread ID。
3. 若同一行程中建立多個 worker 執行緒，且這些執行緒可能同時執行 value = value + (*local);，請說明可能發生的同步問題，以及如何利用 Mutex 避免該問題。

三、針對 MIPS 指令及架構的組合語言程式，假設 X 與 Y 均為每個元素占用 4 Bytes 的一維整數陣列，變數 a、b、c、d、e 分別配置於暫存器$s0、$s1、$s2、$s3、$s4，而陣列 X 與 Y 的起始位址分別儲存於暫存器$s6 與$s7。

```
I1  addi  $t0, $s6, 36
I2  lw    $t0, 0($t0)
I3  sll   $t0, $t0, 3
I4  sll   $t1, $s4, 2
I5  add   $t1, $t1, $s7
I6  lw    $t1, 0($t1)
I7  add   $t0, $t0, $t1
I8  add   $s3, $t0, $s2
```

㈠請說明上述 MIPS 組合語言程式（I1）~（I6）的運作邏輯，並將整個程式轉換為等效的 C 語言程式碼。（10 分）

㈡假設上述程式在單一發射、循序執行的典型五階段 MIPS 管線處理器中執行，管線階段依序為 IF、ID、EX、MEM 與 WB。處理器具有由 EX/MEM 與 MEM/WB 管線暫存器至 EX 階段的完整資料轉傳路徑，且 lw 指令載入的資料須至 MEM 階段結束後才能使用。請找出程式中可能發生的資料危障（Data Hazard），指出具有 RAW 資料相依關係的指令及相依暫存器，並說明那些危障可透過資料轉傳解決，以及那些載入使用危障（Load-use Hazard）仍須插入停頓週期（Stall Cycle）。（15 分）

四、某政府機關 Linux Web Server 疑似發生資安問題，管理者發現系統 CPU 使用率異常偏高，且有未知程序持續對外連線。請依下列要求進行系統安全檢查與數位鑑識。

㈠請利用 Linux 指令完成下列工作。（8 分）

- 列出系統中所有執行中的程序，並依 CPU 使用率由高至低排序。
- 找出目前已建立 TCP 連線之程序，顯示本機位址、遠端位址、連接埠、程序名稱及 PID。
- 顯示系統目前所有處於 Listening 狀態之 TCP 與 UDP Port，並找出其對應程序。
- 建立並保存系統記憶體映像檔（Memory Image），以供後續數位鑑識分析使用。

㈡若已取得該主機之記憶體映像檔（Memory Image），請說明如何利用 Volatility 工具進行下列分析，並寫出適當之 Volatility 分析模組（Plugin）名稱。（17 分）

- 分析記憶體中的網路連線資訊，找出本機與遠端 IP、Port、連線狀態及所屬 PID，並判斷是否存在連往可疑外部 IP 或異常 Port 之程序。
- 分析程序是否存在程式碼注入（Code Injection）、異常可執行記憶體區段（Executable Memory Region）、可疑共享物件（Shared Object）載入或可疑記憶體區域（Memory Region）等異常行為，並說明其可能造成之安全風險。
- 利用 pslist（Process Listing）與 psscan（Process Scanning）等不同方式交叉驗證程序資訊，判斷是否存在遭 Rootkit 隱藏、已終止但程序結構仍殘留，或未出現在正常程序鏈結中的程序，並分析其可能原因。
- 若發現可疑程序，請說明如何利用 Volatility 擷取其程序記憶體內容或可復原之可執行檔，計算其 SHA-256 雜湊值，並說明該雜湊值於驗證證據完整性、與已知惡意程式雜湊資料庫進行比對，以及數位鑑識證據保存之用途。