Domande (e risposte) frequenti di SW sicuro

1) Modelli di ciclo di vita per i processi incrementali

E' un modello di ciclo di vita capace di adattarsi ai cambiamenti dei requisiti, della specifica e del design. Permettono il riciclo: prima modifichi il progetto e poi cambi il codice. E' possibile applicare i cambiamenti a tutti i documenti. Consente validazione e verifica. Esistono in molte forme: prototipazione, modello a fasi di release, modello a spirale

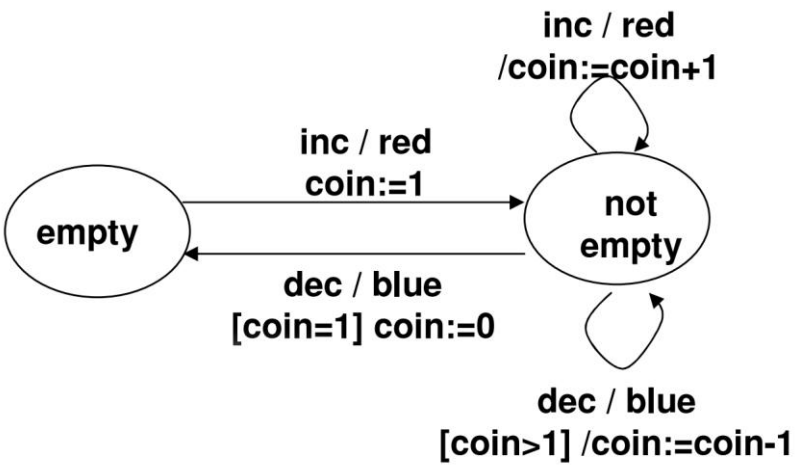
2) Dare la definizione di una FSM estesa con esempio (sulle slide salvadanaio elettronico)

Le EFSM (Extended Finite State Machines) estendono le FSM con il concetto di variabile. Una EFSM è una tupla (S, I, O, V, T) dove:

- S: insieme finito di stati
- I: insieme finito di eventi di input
- O: insieme finito di eventi di output
- V: insieme finito di variabili
- T: insieme finito di transizioni. Una transizione è una tupla (s, i, o, g, a, s'):
  - s: stato sorgente
  - i: evento di input
  - o: evento di output
  - g: predicato sulle variabili in V, detto Guardia
  - a: assegnamento ad una variabile V, detto Azione
  - s': stato target

Esempio: salvadanaio elettronico:

- inserendo monete (inc) il valore di coin viene incrementato
- emettendo monete (dec) il valore di coin viene decrementato
- la luce diventa red quando si inseriscono monete, diventa blue quando si richiedono monete
- la macchina non restituisce monete quando è vuota
  - S={empty, not-empty}
  - I= {inc, dec}
  - O={red, blue}
  - V={coin}
  - T={(empty, inc, red, coin:=1, not-empty),(not-empty, dec, blue, coin=1, coin:=0, empty)…}



3) Stato globale, transizione globale, grafo raggiungibilità di una FSM
   - Stato globale: Per una EFSM (S, I, O, V, T) una coppia (s, σ) è detta stato globale se s è uno stato e σ è una valutazione su V; esempio: (empty, coin=0), (not-empty, coin=1)

   - Transizione globale: per una EFSM (S, I, O, V, T) una tupla ((s, σ), i, o, (s', σ')) è detta transizione globale se esiste una transizione (s, i, o, g, a, s') tale che σ soddisfa la guardia g e σ'(v)= σ(exp) dove l'azione a è v:=exp; esempio: ((empty, coin=0), inc, red, (not-empty, coin=1))

   - Grafo di raggiungibilità: Il grafo di raggiungibilità di una EFSM (S, I, O, V, T) è un grafo direzionato in cui: i nodi sono gli stati globali e gli archi sono le transizioni globali. Se le variabili hanno un range finito, il grafo di raggiungibilità di una EFSM è una FSM

4) Criterio di test: affidabile, valido, ideale
   - Affidabile: se per ogni coppia di test set T1 e T2 adeguati secondo il criterio C, se T1 individua un malfunzionamento, allora anche T2 e viceversa
   - Valido: un criterio C è valido se, qualora il programma P non sia corretto, esiste almeno un test set T che soddisfa C che è in grado di individuare il difetto
   - Ideale: test set T che è selezionato con un criterio affidabile e valido

5) Teorema di Goodenough e Gerhart:
   Dato un criterio C, un programma P se:
   - C è verificabile per P -
   - C è valido per P - C è ideale
   - T test suite selezionato con C
   - T non trova malfunzionamenti in P

   Ok (P, T) ->OK (P), T è ideale

6) Elementi che costituiscono un contratto SW secondo il principio del design by contract
   L'interfaccia di un modulo definisce un contratto. Un contratto è un accordo tra cliente e fornitore che:
   - Lega le due (o più) parti: fornitore e cliente
   - È esplicito (scritto)
   - Specifica gli obblighi e i benefici delle due parti
   - Normalmente mappa gli obblighi di una parte come benefici dell'altra parte
   - Non contiene clausole nascoste: gli obblighi sono quelli dichiarati

7) Software affidabile, robusto, corretto, efficiente:
   - Affidabile: se si comporta come previsto
   - Robusto: se si comporta in modo ragionevole in circostanze non previste
   - Corretto: se rispecchia le specifiche funzionali di progetto
   - Efficiente: se usa intelligentemente le risorse di calcolo

8) Cosa sono artifatti e stakeholders?
   - Artifatti: insieme di documenti riguardanti i requisiti e l'architettura, modelli, pezzi di codice esistente ecc.
   - Stakeholders: persone coinvolte al progetto (chi ci lavora ma anche i clienti)
9) Architetture sicure, principio di auditability:

Deve essere possibile ricostruire la sequenza di eventi che hanno condotto a certe azioni chiave (es: cambio di dati). Sono necessari gli audit log.

10) Fase di specifica (o analisi dei requisiti), classificazione dei requisiti:
La fase ha lo scopo di determinare le funzionalità e le proprietà del SW in termini di performance, facilità d'uso, portabilità, facilità di manutenzione ecc. Classificazione requisiti:
    - Funzionali: cosa deve fare l'applicazione e come deve farlo
    - Non funzionali: dove deve essere usato il SW e su che architettura
    - Requisiti del processo di manutenzione: quando e per quanto tempo verrà usato il SW

11) 3 modalità di manutenzione:
    - Correttiva: per correggere i difetti
    - Adattiva: per adattare il SW a diverse esigenze
    - Perfettiva: per migliorare il SW negli aspetti che già svolge comunque correttamente

12) Test-set adeguato per il criterio di copertura delle istruzioni:
Un test set T è adeguato secondi il criterio visto sopra se per ogni istruzione s di P esiste un caso di test in T che esegue s. Ogni istruzione viene eseguita almeno una volta.

13) Processo di valutazione per ottenere la certificazione di SW sicuro in base ai common criteria. Es:
    a. Wall Street produce un profilo di protezione per il firewall usato per proteggere informazioni sensibili
    b. Il profilo viene valutato in base alle common evalutation methodology per garantire che sia completo
    c. Ottenuta la valutazione, il profilo viene pubblicato (target di sicurezza)
    d. Un vendor realizza una sua versione (target di valutazione) di firewall dotato del profilo di protezione definito da wall street
    e. Il target di valutazione viene inviato ad un istituto accreditato per la valutazione rispetto al target di sicurezza

14) La politica di sicurezza in java 2 (java sandbox)
Java 2 introduce in controllo degli accessi a livello più fine, basato su security policies e permessi. Controllo di accesso flessibile: alcune applet possono avere accesso a risorse al di fuori della sandbox

15) I problemi di sicurezza del linguaggio C
    - Deferenziazione del null
    - Type cast non controllato
    - Pointer arithmetic
    - Accesso alla memoria non valida (violazione spaziale [out of bound] o temporale [dangling pointers])

16) Processo SW, le sue qualità (3)
    - Produttività: misura dell'efficienza del processo in termini di velocità di consegna
    - Tempestività: misura la capacità del processo di rispettare i tempi di consegna
    - Trasparenza: misura la capacità del processo di capire il suo stato attuale e tutti i suoi passi

17) Tipi di test: accettazione, conformità, integrazione, regressione
    - Accettazione: il comportamento del SW è confrontato con i requisiti dell'utente finale

- Conformità: il comportamento del SW nella sua interezza è confrontato con le specifiche dei requisiti
- Sistema: controlla il comportamento dell'intero sistema (hw + sw) come monolitico
- Integrazione: controllo sul modo di cooperazione delle unità (come previsto dal progetto)
- Unità: test di comportamento delle singole unità
- Regressione: test del comportamento di release successive

18) Principi guida di un'architettura sicura che riguarda malfunzionamenti e errori di sistema (3): graceful degradation, appropiate error handling, fault safety
    - Graceful degradation: quando si verifica un guasto il sistema non si ferma ma continua ad operare con funzionalità ridotte
    - Fallimento sicuro: in caso di fallimento un sistema deve terminare in una configurazione sicura
    - Misure di sicurezza adeguate all'utente: principio di accettabilità psicologica

19) 3 principi importanti per le architetture sicure
    - Inizia facendo domande
    - Decidi che livello di sicurezza è sufficiente
    - Progetta con in mente il nemico
    - Identifica le assunzioni
    - Conosci e rispetta la chain of trust
    - Definisci i privilegi adeguati
    - Testa le azioni possibili contro la policy
    - …

20) Qualità esterne di un prodotto SW sicuro
    Sono quelle qualità percepite da un osservatore esterno che esamina il prodotto come se fosse una black box. Alcune qualità esterne:
    - Correttezza
    - Affidabilità
    - Efficienza
    - Usabilità
    - Portabilità
    - Interoperabilità
    - Robustezza

21) Tipi di testing: funzionale e di sicurezza
    - Funzionale: comporta il mettere alla prova un sistema per determinare se fa ciò che si suppone debba fare in circostanze normali o critiche
    - Di sicurezza: comporta il mettere alla prova un sistema allo stesso modo in cui potrebbe farlo un malintenzionato

22) 3 attacchi in fase di implementazione
    - Buffer overflow: accetta più caratteri di quanti sono effettivamente memorizzabili
    - Back door: porta sul retro del sw
    - Parsing error: input senza controllare i dati

23) Macchine di Mealy, Macchine di Moore

Una FSM (S, I, sigma) con output è detta macchina di

- Mealey: se è una FSM che produce un output per ciascuna transizione
- Moore se è una FSM che produce un output per ciascuno stato

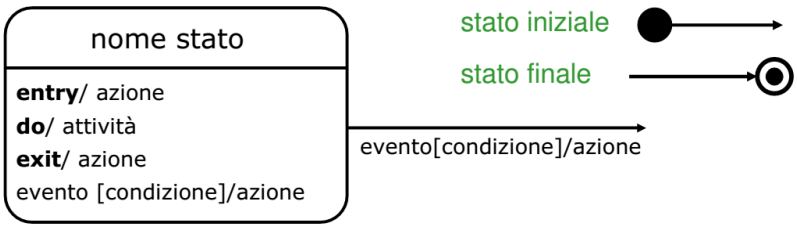
24) Criteri di copertura delle decisioni e delle condizioni

- Decisioni: predicato (espressione booleana) guardia di una istruzione condizionale (if) o iterativa (ciclo). Es: if(x>0 | y>0)
- Condizioni: espressione booleana atomica che appare in una decisione. Nell'esempio sopra: "x>0" e "y>0"

Criteri di copertura delle

- decisioni: Copre ogni decisione e la sua negazione. Ogni arco nel grafo del flusso viene percorso. Un test-set T è adeguato per testare un programma P secondo il criterio di copertura delle decisioni se per ogni decisione di P esiste un caso di test T in cui la decisione è presa e un caso di test T in cui la decisione non è presa
- condizioni: Copre ogni condizione e la sua negazione. Un test-set T è adeguato per testare un programma P secondo il criterio di copertura delle condizioni se per ogni condizione di P esiste un caso di test T in cui la condizione è vera e un caso di test T in cui la condizione è falsa.

25) UML



Parallela, con stati di sincronizzazione e sbarre di sincronizzazione: