UNIVERSITÀ DI PISA

Language-Based Technology for Security
Homework 1

Monica Amico *Matricola: 516801*

1 Svolgimento

1.1 Permessi e Operazioni

I permessi riguardano la lettura, la scrittura e l'esecuzione di un file. Due permessi sono uguali se si riferiscono allo stesso file. Due operazioni sono uguali se richiedono la scrittura/lettura/esecuzione dello stesso file.

Il tipo `permission` è stato definito come:

```
type permission =
  | ReadPerm of string
  | WritePerm of string
  | ExecutePerm of string
```

1.2 Nuove strutture utilizzate a Runtime

- `Stack`: di tipo textttstring list, rappresenta lo stack delle chiamate di funzione.
- `Permission Table`: di tipo `(string * permission list) list`, è una lista di coppie che tiene traccia dei permessi di ogni funzione.

1.3 Simulazione della Stack Inspection

E' stata modificata la funzione `ieval`, in modo tale da poter tenere conto anche dello stack e della tabella dei permessi durante la valutazione di un espressione. Ad ogni funzione, durante la dichiarazione, viene associata una lista di permessi, che sono quelli che la funzione possiede. E' stata quindi modificata anche la Chiusura, in modo tale che contenga la lista dei permessi, che verranno poi controllati prima di effettuare un'operazione di lettura/scrittura/esecuzione.

Ad ogni chiamata di funzione vengono modificati lo stack e la tabella dei permessi, nel modo seguente:

- viene aggiunta la funzione chiamata in testa nello stack, in modo tale da tener traccia di tutte le funzioni chiamate. Per semplicità le funzioni sono identificate univocamente con il loro parametro. Se la funzione contiene nel Body un' ulteriore funzione verrà ricorsivamente fatta la stessa cosa, l'ultima funzione chiamata sarà quindi in testa nello stack.
- viene aggiunta nella tabella dei permessi una coppia contenente il nome della funzione chiamata e la lista dei suoi permessi, come nel caso

precedente, se il Body contiene una funzione verranno aggiunti anche i permessi di tale funzione, grazie alla ricorsione.

Prima della valutazione di un espressione privilegiata, `Read` o `Execute` o `Write`, viene chiamata la funzione `check` (di tipo `stack -> permissionTable -> permission -> bool`), la quale controlla che tutte le funzioni presenti nello stack (partendo dalla testa della lista, quindi dall'ultima funzione chiamata) posseggano il permesso richiesto dall'operazione. Per fare questo, costruisce una lista di tutte le liste di permessi delle funzioni presenti nello stack, dopodichè controlla che in tutte le liste sia presente il permesso richiesto dall'operazione; sé si, restituisce `true`, altrimenti `false`, negando l'esecuzione dell'operazione richiesta.