

Language Based Technology for Security

Cybersecurity LM-66

Samuele Padula

Aprile 2021

1 Introduzione

È stata ampliata una porzione dell'interprete scritto in OCaml per il progetto di Programmazione 2. L'interprete implementa la funzionalità di Stack Inspection e l'intero progetto è contenuto in un unico file sorgente "stack_inspection.ml". L'interprete visto a lezione è stato esteso con la possibilità di chiamare funzioni privilegiate come la Read, Write o Open per le quali a livello di linguaggio sono stati definiti i permessi minimi richiesti (es: la Open richiede i permessi Pread e Pwrite solamente per un motivo di test didattico). Per eseguire la batteria di test (sono stati effettuati dei test con una singola funzione e con più funzioni innestate che richiedono liste diverse di permessi) si può usare il comando ocaml. Il tutto è stato simulato e sviluppato in ambiente Linux Ubuntu 20.04 ocaml 4.05):

```
$ocaml stack_inspection.ml
```

2 Possibili sviluppi (Set)

Dato che la stack inspection, a run time, concettualmente effettua una operazione di intersezione tra gli insiemi contenenti ciascuno i permessi di ogni funzione presente sullo stack, si può rendere il codice più snello implementando nativamente il tipo Set nell'AST oppure usando la classe Set di Ocaml, così da creare un insieme di permessi per ogni funzione ed effettuare poi l'intersezione. In questo modo è possibile verificare in modo più compatto (evitando l'uso delle tre funzioni: superPrivilegesVerifier, demandPermission e verify_singlePermission) se negare o meno l'operazione richiesta.