Stimare i consumi di gas nell'esecuzione di Smart Contract in Ethereum

Melania Ghelli

10 ottobre 2019

Introduzione

Negli ultimi anni la blockchain ha riscosso molto successo. Questa nuova tecnologia permette l'esecuzione di programmi in modo distribuito e sicuro, senza la necessitá di un ente centrale che faccia da garante. Il paradigma trova applicazioni nei settori piú disparati, offrendo innovazione grazie alla possibilitá di fare a meno di banche o istituzioni pubbliche.
Tra i sistemi nati grazie a questa tecnologia troviamo Ethereum, una piattaforma che mette a disposizione un linguaggio di programmazione di alto livello. Questo linguaggio puó essere utilizzato dagli utenti per implementare dei programmi, i cosí detti smart contract.
Gli smart contract possono essere eseguiti sulla rete di Ethereum solo al fronte di un pagamento anticipato. Per ragioni di sicurezza a ciascuna istruzione di basso livello é associato un costo monetario. Dunque eseguire un programma costerá tanto quante sono le istruzioni che lo compongono.
Il costo di ciascuna istruzione é espresso in termini di gas, una sorta di carburante che viene pagato in ether, la crittovaluta di Ethereum.
Dal momento che il gas viene pagato in anticipo, potrebbe accadere che l'esecuzione di un programma ecceda la quantitá messa a disposizione. In questi casi la computazione non giunge al termine, risultando nella perdita delle risorse investite dall'utente. Oltre a questo comportamento indesiderato l'esaurimento del gas disponibile puó avere conseguenze pericolose.
Un programma che non gestisce correttamente queste situazioni viene etichettato come vulnerabile. La conseguenza piú diretta é il blocco del contratto, che puó essere anche permanente. La pericolositá peró risiede nel fatto che questo tipo di programmi diventano un bersaglio facile per attacchi malevoli. Vedremo come queste vulnerabilitá possono essere sfruttate per ottenere comportamenti dannosi per la rete.
Dato il valore monetario associato agli smart contract il rischio che si corre in caso di attacchi informatici é una perdita di denaro. Per questo motivo é necessario individuare possibili criticitá nel codice prima della sua esecuzione. In questo contesto l'analisi statica dei programmi costituisce un potente strumento di prevenzione.

All'interno di questo elaborato ci concentreremo solo sulle tecniche di analisi dei consumi di gas. Poter conoscere a priori quest'informazione permetterebbe non solo un investimento adeguato da parte degli utenti, ma anche uno strumento di prevenzione da possibili attacchi.
Attualmente non esistono strumenti in grado di calcolare con precisione la quantitá di gas richiesto durante una computazione. Cercheremo di capirne le ragioni ma soprattutto di individuare dei margini di miglioramento.