VPN Decentralizzate

Jacopo Federici

December 17, 2018

Abstract

1 Introduzione

cosa è una VPN, cosa è una VPN decentralizzata, principali differenze e perchè si è sviluppata ora e non prima, descrizione delle nuove tecnologie che consentono al sua realizzazione

1.1 concetti introduttivi

- blockchain in linea di massima
- ethereum in linea di massima (con rif a erc20) e descrizione degli smart contracts
- proof of work, proof of stake
- reputazione in una net (Eigentrust)
- - cercare applicazione pratica di Eigentrust
- descrizione di BFT
- - Tendermint e quindi Cosmos

2 Players

Chi sono i player, come e quando nascono, dettagli sui tempi di realizzazione, sul modello di business.

2.1 Mysterium

Mysterium Network è un progetto open source per la realizzazione di un network P2P decentralizzato che usa multi-hops, relays e micropagamenti integrati basati su token.

2.1.1 Contesto

Nel contesto che si vuole descrivere si identificano due entità: un Client A che usa il servizio ed un Agent B che lo fornisce.

2.1.2 Gestione dei pagamenti

Mysterium utilizza un Service Provider, basato su blackchain ethereum, che registra le intenzioni del Client A di pagare l'Agent B per il servizio che B offre ad A.

2.1.3 Introduzione del concetto di promessa

Una promessa è una quadrupla contenente i seguenti elementi:

- identificatore Issuer (A)
- serial number della promessa (SN)
- identificatore Benefiter (B)
- somma MYST

Quando A paga B viene creata, o se già esistente aggiornata, una promessa P.

2.1.4 Caratteristiche di una promessa P

- P è firmato da A
- Una promessa P può essere aggiornata in caso di prosecuzione del servizio
- B può bloccare l'aggiornamento della promessa P
- P è valida se è firmata da A e se non è stata aggiornata dopo essere stata bloccata
- L'attributo SN incrementa ad ogni modifica
- Per identificare uno stato non valido basta verificare il SN se è maggiore del SN nel momento del blocco
- La chiusura di un contratto P, con P valido, termina con il pagamento della somma MYST da A a B
- Ogni transazione ethereum per il pagamento di MYST, generata a seguito della chiusura di un contratto, comporta il pagamento di fees e gas
- Si crea un equilibrio tra trasferimento di poco denaro ma frequente e trasferimento di molto denaro ma poco frequente
  - Il primo comporta un totale di fees maggiore rispetto al secondo ma corrisponde ad una più alta garanzia di pagamento rispetto al secondo (se un client A non paga, non paga poco denaro, invece che tanto) [merita di essere analizzato meglio]
- Le promesse possono essere condivise con più Agents B senza perdita di valore

  - La condivisione consente di evitare situazioni in cui un Client A con pochi denari instauri molteplici promesse (false) con diversi Agents B: [DA CAPIRE MEGLIO]

- Gli Agents B possono decidere il grado di rischio a cui esporsi (ovvero al rischio che i Clients non paghino)

2.1.5 Gestione dei Rischi

La promessa non garantisce, di per sé, il pagamento: infatti se A non ha abbastanza MYST nell'account per il pagamento, la sua promessa non viene mantenuta e la sua identità, in successive promesse, viene compromessa. Si utilizza quindi un registro, chiamato Identity Registry (IR) che tiene traccia delle identità [capire meglio che dati dell'identità vengono usati] che usano il servizio, attraverso il quale scoraggiare l'emissione di promesse non mantenibili.

2.2 Privatix

Privatix utilizza due tipi diversi di smart contract con due funzionalità diverse.

- PTC: Privatix Token Contract è usato per scambiare token, upgrade o nuovi contratti di servizio

- PSC: Privatix Service Contract è usato per immagazzinare balance (depositare e ritirare) e per la gestione del channel...

I coin PRIX possono essere acquistati e venduti sono con PTC. I servizi Privatix, invece, possono essere pagati solo usando PSC. L'idea quindi è quella di usare lo smart contract PSC per le operazioni interne e lo smart contract PTC per le operazioni con l'esterno, con il fine ultimo di aumentare la sicurezza.

2.2.1 Attacchi Sybil

Attualmente non esistono tecnologie per mitigare l'attacco che non prevedano l'introduzione di qualche svantaggio: così dicono loro. Separano l'attacco in due categorie, in base all'entità malevola.

2.2.2 Agents Malevoli

A tutti gli agenti è richiesto di registrare il servizio offerto nella blockchain Ethereum e di depositare una somma di denaro che è possibile ritirare in seguito così da diminuire la presenza di Smart Operation (SO) fasulle. Il deposito è proporzionale alla quantità di servizio offerto. Se l'Agent malevolo posiziona SO fasulle, gli sarà richiesto di posizionare la stessa quantità di denaro che il Client ha posizionato quando ha accettato l'offerta dell'Agent. Le SO hanno associata una età e quindi gli Agents devono mantenere la loro SO viva, altrimenti i Clients la considereranno irrilevante. Ad ogni ri-notifica degli Agents, essi devono bloccare (posizionare) nuovamente la stessa quantità di denaro.

Quando un Client crea uno state channel con un Agent e non riceve il servizio accordato, allora chiuderà lo state channel con lo stato Uncooperative. In questo caso l'Agent sarà notificato da un evento blockchain così da intaccare la sua reputazione e forzare l'Agent malevolo a creare una nuova identità.
Se l'Agent malevolo opera sia come Client che come Agent per incrementare il numero operazioni Cooperative, si troverà di fronte al pagamento di fees ed ogni volta che opera come Client diminuirà la quantità di servizio a disposizione come Agent.

2.2.3 Client Malevoli

Un Client può creare uno state channel e non inviare la prova di possesso della quantità di denaro necessaria a pagare il servizio. Ma al Client è richiesto di depositare quella cifra che sarà bloccata per tutto il periodo del channel. Per ottenere il denaro indietro è necessario chiudere il channel, ma la transazione consuma Ethere e quindi costa. Quindi se un Client vuole danneggiare un Agent le sue possibilità sono ridotte e costose, inoltre sarà segnata Uncooperative, a suo sfavore, una chiusura di un channel.
Sia i Client che gli Agent ottengono la chiusura del channel in modo Cooperative o Uncooperative e la chiusura viene registrata in blockchain con associati gli indirizzi di entrambi. Il sistema blockchain consente di analizzare le chiusure Cooperative e Uncooperative di tutti così da verificare la reputazione a cascata.

2.2.4 Sommario

Fondamentalmente non hanno trovato un modo per eliminare al 100% attacchi Sybil ma cercano, con i sistemi sopra descritti, di limitare al massimo le possibilità di azione e la loro efficacia.

2.3 Sentinel

2.3.1 Architettura

- Identity Chain: Anonymous User ID (AUID) creati e memorizzati in una blockchain ad hoc.
- Service Chain: Rete sicura su cui si appoggiamo le operazioni di gestione e sono usati i token $SENT/$SST per le transazioni ($SENT sono usati esternamente e sono usati comprare $SST che sono usati internamente, rapporto 1:1)
- Transaction Chain: gestisce i pagamenti ed i relativi processi di invio delle transazioni al Sentinel Transaction Pool.

L'unico punto di accesso ai servizi di Sentinel è l'AUID.
Il sistema è basato sulla reputazione tale per cui più è alta e maggiore è l'accesso ai servizi. C'è un meccanismo di guadagno di denaro in funzione della reputazione.

Il consenso distribuito riconosce e mitiga velocemente cattivi attori.
La soluzione adottata per mitigare il problema di attori cattivi sul nodo di uscita è la stessa adottata dalle reti come TOR, ovvero l'uso di un tecniche di routing e ritrasmissione dei pacchetti per assicurare che sia la sorgente che la destinazione non siano rilevati.
Si sta attualmente sviluppando la rete di ritrasmissione nella quale i partecipanti posso decidere il ruolo (perché?).

2.3.2 Meccanismo del consenso + hybrid packet routing

Il percorso del pacchetto sarà deciso dal gestore dei nodi in base a parametri tra cui:

- numero di hop richiesti dall'utente
- reputazione e latenza dei relays...

2.3.3 Idee di Sentinel

Al contrario delle altre soluzioni loro vogliono creare una piattaforma su cui appoggiare una serie di prodotti e servizi oltre alla dVPN: dChat, dVoIP, dFiles, dCDN, dDNS, dCompute, chiamate dAPPs. Hanno intenzione di realizzare un internet box hardware da posizionare tra l'accesso ad internet (gateway) ed il router.
Attualmente Sentinel si appoggia a Cosmos, piattaforma che consente la creazione di una blockchain decentralizzata di criptovaluta. Cosmos è costruito su Tendermin che fornisce il livello networking e consensus di una blockchain (quindi Cosmos è l'application di cryptocurrencies)
Quindi, per quanto riguarda Sentinel, il problema del consenso BFT è risolto dalla piattaforma su cui si appoggia.

2.3.4 Tendermint

Tendermint è un software per la replicazione in modo sicuro e consistente di una applicazione su una moltitudine di macchine. Con in modo sicuro si intende che Tendermint funziona anche se 1/3 delle macchine fallisce arbitrariamente. https://tendermint.com/docs/introduction/what-is-tendermint.html

2.4 Substratum

2.5 caratteristiche comuni

2.6 BTF

2.7 Attacchi di tipo...

3 problemi individuati

1. valutazione della reputazione dell'agent da parte del client

2. valutazione della reputazione dell'agent da parte della Net

4 soluzioni proposta

4.1 Proposte per la valutazione della reputazione dell'agent da parte del client

Le seguenti proposte sono metodi di verifica del corretto comportamento ed hanno come scopo finale la valutazione della reputazione degli agent

- Dati civetta: modifiche al client che con scadenze richiede una o più risorse dal valore noto e verifica che siano integre. Le scadenze possono essere regolari/random/all'inizio frequenti/dipendenti dalla reputazione dell'agent. È necessario avere delle risorse distribuite e disponibili: potrebbero essere i nodi stessi della rete (altri agent).

- Dati duplicati: modifiche al client che implementa la possibilità di connessione con più agent. Dopo la richiesta il client compara i dati ottenuti dalle due fonti

  - Hash *: (soluzione aggiuntiva) I nodi sono generalmente in posizioni migliori dei client in termini di velocità. L'idea è quella di far generare un hash dei dati che il client richiede ad un altro nodo fuori dal servizio primario, così da comparare gli hash e non tutto il dato. Inoltre, se la richiesta la faccio a molti più nodi posso intrinsicamente verificare quali modificano i dati nel network.

- Applicazione di modelli di reputazione (Eigentrust): attualmente non studiato

4.2 Proposte per la valutazione della reputazione dell'agent da parte della Net

- idea: basarsi sul concetto di proof of work e far lavorare ogni nuovo nodo un pò prima di fargli guadagnare della reputazione. Il lavoro consiste nella verifica di affidabilità dei nodi della rete con metodi descritti sopra (dati civetta/hash dati divetta/ dati duplicati).
  Alla fine del lavoro lui ha contribuito alla verifica della reputazione degli altri nodi migliorando la qualità della net. (sia che scropra o meno che alcuni nodi modificato i dati (in caso si modifica la loro reputazione: come?? booo, vedere sistema che mystirium usa, magari semplicemente chiude come uncooperative il channel).
  Alla fine del lavoro lui ha consumato delle energie che lo abilitano all'uso della rete (probabile livello aggiuntivo opzionale per pagare meno la fee di ingresso)

4.3 Impostazione del lavoro

parlare di:

- docker (cosa è, comandi base per fare cosa)

5 conclusioni