Project 1: Anonymity low-latency System

Tecnologie utilizzate

Il progetto è stato realizzato in linguaggio Python. Per poter compilare il progetto è necessaria la libreria Pycryptodome (https://pycryptodome.readthedocs.io/en/latest/src/introduction.html) liberamente scaricabile e installabile attraverso pip, il package manager di Python. La libreria è stata utilizzata per fornire le funzioni di crittazione asimmetrica RSA e simmetrica AES e per la generazione di chiavi pubbliche e private.

I file presenti

Nella cartella del progetto sono presenti diversi file:

- Client
- Server
- AES
- KeyThread
- Proxy 1
- Proxy 2
- Proxy 3

Si è scelto per comodità realizzativa di implementare una rete di "solamente" tre proxy, ma il programma può funzionare tranquillamente con un numero molto superiore di proxy con piccoli accorgimenti.

Chiaramente, al momento del deploy dell'applicativo andranno eseguiti e avviati prima i file server e la rete di proxy e solo in un secondo momento il file client.

Il programma simula una rete di comunicazione ma per motivi pratici tutti i proxy, il client e il server girano sulla stessa macchina e sullo stesso indirizzo fisico, semplicemente su porte diverse.

Implementazione

Il server genera la propria chiave pubblica e la salva su un file con estensione .pem. Questo processo avviene in un thread separato per non rallentare le operazioni di comunicazioni. Il file verrà letto in seguito dal client e cripterà il messaggio segreto che vuole spedire con la chiave pubblica del server in modo RSA.

Viene stabilita una password comune che utilizzeranno i proxy e i client per criptare in modo AES la lista di porte dei proxy e del server finale al quale dovrà arrivare il messaggio.

Inizialmente il client sceglie casualmente una porta dalla lista dei proxy (si suppone che il client conosca già la lista dei nodi della rete di proxy o che gli sia stata fornita da un server esterno fidato) e ne instaura una connessione. La prima comunicazione sarà il messaggio segreto criptato in RSA, mentre la seconda comunicazione sarà la lista delle porte dei proxy e del server di destinazione criptata in AES.

La lista è rappresentata sotto forma di array strutturato nel seguente modo: il primo elemento corrisponderà alla porta del server di destinazione, gli altri elementi saranno tutte le porte della rete di proxy.

Il funzionamento operativo

Il primo proxy scelto decripterà solo la lista delle porte dei proxy. In modo randomico sceglierà una porta casualmente in un range che va da 1 a LUNGHEZZA_LISTA in modo da evitare sempre di scegliere casualmente la porta del server. Dopo di che si eliminerà dalla stessa lista. A questo punto cripta di nuovo la lista in AES e manda in una prima comunicazione il messaggio originario criptato in RSA e nella successiva comunicazione la lista dei proxy rimanenti più la porta del server alla porta del proxy scelto. I passaggi si ripetono fino all'ultimo proxy della lista che si accorgerà di essere l'ultimo perché nella lista rimarrà solo la propria porta oltre all'elemento di testa (porta del server). A questo punto l'ultimo proxy manda solo il messaggio originario al server, prendendo l'elemento l'indirizzo dall'elemento di testa dell'array. Il server a questo punto potrà decriptarlo possedendone la chiave privata.

Questo meccanismo presuppone che la rete di proxy sia fidata. I proxy saranno a conoscenza del server di arrivo e della password comune condivisa fra client e proxy per criptare il contenuto della lista per il percorso. Nessun proxy però può scoprire il contenuto del messaggio segreto e il percorso precedente o successivo ad esso in quanto generato in modo randomico. Garantisce anonimato e confidenzialità dei dati.