Studio ed implementazione di una architettura avanzata basata su VPN per Security Assessment

Nicola Bena (matricola 870103)

RELATORE
Prof. Marco Anisetti

CORRELATORI
Prof. Claudio A. Ardagna
Dott. Filippo Gaudenzi

Nell'ambito della mia tesi ho collaborato a MoonCloud, spinoff di questa università, il quale è un prodotto in grado di fornire valutazione e monitoraggio continuo di servizi cloud. In particolare, MoonCloud offre *Security Assessment* e *Security Assurance*, cioè la capacità di valutare, mediante la raccolta continua di evidenze, lo stato *effettivo* della sicurezza di un sistema.

Il target iniziale di MoonCloud sono i sistemi cloud, tutttavia si è voluto espandere tale target anche ai sistemi IT tradizionali (*reti target*), ovvero sistemi confinati all'interno di una classica rete aziendale. Se, per quanto riguarda target nella cloud, ci si appoggia a degli *hook* forniti dal cloud provider, per analizzare sistemi IT tradizionali è stato necessario ripensare a come MoonCloud possa dialogare con tali sistemi.

MoonCloud viene attualmenete erogato come un servizio, ovvero l'utente finale non deve installare niente per analizzare i propri assert nella cloud, e nella transizione verso sistemi IT tradizionali si è cercato di mantenere questa caratteristica.

La soluzione che si è trovata, e che è stata il punto di partenza della mia tesi, è stata quella di utilizzare una VPN per collegare MoonCloud alla rete da analizzare; il collegamento sarebbe avvenuto mediante un device VPN portato fisicamente nella rete, il quale si sarebbe collegato alla VPN di MoonCloud dando accesso a quest'ultima all'intera rete aziendale, il tutto senza che l'utente debba operare alcuna configurazione sulla propria rete (es: aprire porte del firewall, ecc...), se non sul VPN client. Quest'ultimo aspetto, ovvero una soluzione *configuration-free* è molto importante.

Il lavoro che ho svolto per arrivare alla tesi si può articolare in tre parti principali:

1. uno studio sulle tecnologie VPN esistenti al fine di valutare quali fossero le più adatte per il problema
2. una volta scelta la tecnologia VPN, si è trattato di capire come configurarla al meglio, risolvendo numerosi problemi che in una installazione VPN classica non si hanno. In particolare, in questa seconda fase si sono introdotti alcune soluzioni particolarmente innovative.
3. la creazione di un microservizio da integrare in MoonCloud per supportare la nuova architettura.

Durante la prima fase ho studiato le maggiori tecnologie VPN disponibili, tenendo presente che si richiedeva una VPN molto flessibile, anche a costo di non essere la più performante in assoluto. Naturalmente doveva

essere molto sicura. La scelta è infine ricaduta su *OpenVPN*, una soluzione open-source ampiamente diffusa, sicura, e, soprattutto, flessibile, grazie alle sua numerose opzioni di configurazione.

A questo punto ho affrontato il problema di configurare OpenVPN per MoonCloud. Posto che i server VPN verranno installati in MoonCloud, è stato fondamentale capire quali problemi tale soluzione presenta. In particolare, se ne sono individuati tre.

Staticità della configurazione OpenVPN si configura mediante file di testo, che vengono letti solo all'avvio, tuttavia vi è la necessità di aggiungere dinamicamente dei nuovi client e configurare rotte, sia interne ad OpenVPN, sia nel kernel. Per risolvere questo problema si sono sfruttati degli *hook* messi a disposizione da OpenVPN, che consentono di eseguire degli script quando si compiono certe azioni (es: quando un client si connette, aggiungi una rotta al sistema operativo).

Configuration-free I pacchetti inviati da MoonCloud alla rete target, una volta ricevuti nel target, hanno come indirizzo IP sorgente un IP appartenente alla rete interna MoonCloud, e le risposte ad essi devono quindi passare per il VPN client, anziché essere inviate al default gateway della rete target. Tuttavia non è pensabile di dover chiedere al cliente di effettuare tali configurazioni, pertanto si utilizza il cosidetto *NAT al contrario*: il VPN client invia i pacchetti sulla rete target facendo del NAT specificando come IP sorgente il proprio. Poiché esso appartiene alla rete target, per definizione si trova nello stesso spazio di indirizzi, e quindi le risposte tornano ad esso anziché essere inviate al default gateway.

Conflitti di IP Si vuole far sì che un VPN possa gestire il maggior numero di VPN client, tuttavia bisogna rispettare un vincolo fondamentale: ogni rete che partecipa alla VPN deve avere un NET ID diverso. E' ragionevole presumere che le reti target utilizzino degli indirizzi IP privati, che sono limitati, e che prima o poi vi sarà un conflitto. Per risolvere questo problema alla fonte, si è introdotto il concetto dell'*IP mapping*. Quando si registra un nuovo VPN client, ad ogni rete target che esso raggiunge si assegna una nuova rete garantita univoca per il server a cui il client è connesso, e tutta MoonCloud conoscerà solo queste reti mappate, anziché le originali. Il VPN client è responsabile di fare l'operazione inversa, quando riceve pacchetti che hanno per destinazione la rete mappata, deve modificarli specificano come destinazione l'IP reale, viceversa per le risposte. Capire come fare ciò è stata la parte più difficile dell'intera tesi, alla fine si è deciso di utilizzare *nftables*, successore di *iptables*.

Come ultima parte, ho realizzato un microservizio in Python, linguaggio che non conoscevo, a supporto di questo nuova soluzione. In particolare, il microservizio mette a disposizione delle API REST che assolvono ai seguenti compiti:

- creazione di tutti i file di configurazione per la VPN
- gestione dei certificati mediante i quali avviene l'autenticazione nella VPN
- gestione dell'*IP mapping*: assegnare correttamente le reti ai client e, dato un indirizzo IP originale, ritornare la sua versione mappata
- trasferimento dei file di configurazione per i server verso i server stessi.

Oltre all'architettura per MoonCloud basata su VPN, che consente di mantenerla *as-a Service*, vi sono ulteriori evoluzioni in fase di studio, tra cui la possibilità di portare parte dei componenti software di MoonCloud presso i clienti.