

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

Nicola Bena

12 Ottobre 2018

Attività di **Security Assessment** e **Security Assurance** sono fondamentali in sistemi IT sempre più **complessi** (es: cloud)

- dovrebbero essere **continue**
- erogate **as-a-service** ridurrebbero i costi
- per ispezionare una cloud pubblica si possono sfruttare gli **hook** messi a disposizione dal cloud provider
- in una rete *classica* o cloud privata non sono disponibili
  - necessità di passare attraverso **firewall** e **NAT**

Sviluppare una soluzione che consentisse di:

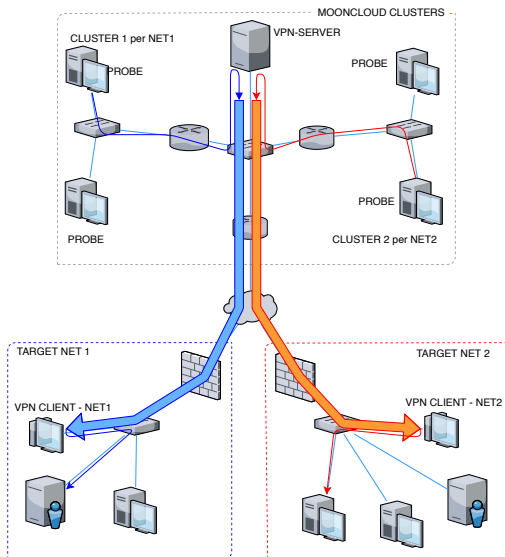
- fare **ispezioni** in reti e cloud private per valutare lo **stato effettivo** della sicurezza
- usando un paradigma **as-a-service**
- **configuration-free**
- garantire alto livello di sicurezza

- Framework per la **valutazione** ed il **monitoraggio continuo** di servizi cloud
- valutazione che certe proprietà (non solo di sicurezza) siano rispettate mediante **raccolta continua di evidenze**
- per l'utente finale MoonCloud è offerto **as-a-service**
  - inserisce informazioni sul target
  - MoonCloud effettua valutazione
  - mostra risultati

Utilizzare un collegamento **VPN** (*ponte*) tra MoonCloud e la rete da analizzare

- device **Linux** portato nella rete target che fa da VPN client
- in MoonCloud i **VPN server**
- **OpenVPN** per il collegamento VPN
- **nftables** (successore di **iptables**) per risolvere problemi di configurazione
- installazione di **VPN non standard**

# Soluzione (2)



# NAT al contrario

- IP sorgente dei pacchetti MoonCloud diretti alla rete target appartiene alla rete MoonCloud
- la rete target deve inviare le risposte al VPN client, ma senza rotte configurate le invierebbe al proprio default gateway

**NAT al contrario:** tutti i pacchetti provenienti dalla VPN vengono immessi nella rete target usando come IP sorgente quello del client VPN

- stesso NET ID della rete target
- quindi le risposte possono tornargli senza problemi
- realizzato con **nftables**

*“Ogni rete connessa alla VPN deve stare in reti IP diverse”<sup>1</sup>*

- si vuole che un server gestisca il maggior numero di reti target diverse
- alta probabilità che due reti abbiano lo stesso NET ID

**IP mapping:** *mappare* ogni rete target in una nuova rete **garantita univoca** perché scelta da MoonCloud

- tutta MoonCloud conosce solo indirizzi mappati quindi unici

---

<sup>1</sup><https://openvpn.net>



# IP mapping (2)

- ① Quando si registra un nuovo cliente le sue reti vengono **mappate** in reti nuove ed univoche
- ② il cliente specifica il target dell'analisi usando l'indirizzo IP reale
- ③ MoonCloud ne ottiene la **versione mappata** in maniera **trasparente**: è il l'IP dst dell'analisi
- ④ l'analisi parte, nel **VPN client**
  - richieste MoonCloud → host target:
    - ① modifica IP dst mappato → IP originale
    - ② applica *NAT al contrario* ed invia ai target
  - risposte target → MoonCloud:
    - ① applica inverso di *NAT al contrario*
    - ② modifica IP src originale → IP mappato

**Microservizio** integrato in MoonCloud per gestire la soluzione VPN

- creazione file di **configurazione** per **OpenVPN**
  - **trasferimento** via SSH ai server
- gestione **certificati** di client e server
  - creazione
  - revoca e rinnovo mediante **CRL**
- gestione **IP mapping**
  - assegnazione nuove reti ai client
  - creazione file di **configurazione** per **nftables**
  - dato un IP originale ritornare quello mappato

Il device VPN viene portato in una rete **non trusted** quindi occorre **proteggere MoonCloud**:

- **regole di firewalling** sui VPN server che consentono alle **sole richieste e risposte** da/per MoonCloud di transitare
- si utilizza **nftables**

Grazie all'architettura presentata è possibile applicare gli stessi efficaci approcci di **Security Assessment** e **Security Assurance** in precedenza disponibili solo per cloud pubbliche

- mantenendo un paradigma **as-a-service**
- garantendo elevati livelli di sicurezza