VPN

1. Glossario

Come prima cosa vi riporto i principali elementi coinvolti (ed una breve descrizione per ciascuno), molti dei quali dovrebbero già risultarvi familiari. Sì, sono in inglese! Considerando che la maggior parte della documentazione relativa al mondo in IT è, appunto, in inglese, vi chiedo un piccolo sforzo in tal senso.

Encryption Algorithms

Encryption algorithms protect the data so it cannot be read by a third-party while in transit.

1. **AES (Advanced Encryption Standard)** is the strongest encryption algorithm available. AES supports encryption keys of these lengths: 128, 192, or 256 bits. AES is faster than 3DES.
2. **3DES (Triple-DES)**. An encryption algorithm based on DES that uses the DES cipher algorithm three times to encrypt the data. The encryption key is 168-bit. 3DES is slower than AES. The Sweet32 vulnerability affects 3DES.
3. **DES (Data Encryption Standard).** Uses an encryption key that is 56 bits long. DES is the weakest of the three algorithms, and it is considered to be insecure.

Authentication algorithms

Authentication algorithms verify the data integrity and authenticity of a message.

1. **MD5.** It produces a 128-bit (16 byte) message digest, which makes it faster than SHA1 or SHA2. This is the least secure algorithm.

2. **HMAC-SHA1 (Hash Message Authentication Code — Secure Hash Algorithm 1).** SHA1 produces a 160-bit (20 byte) message digest. Although slower than MD5, this larger digest size makes it stronger against brute force attacks. SHA-1 is considered to be mostly insecure because of a vulnerability.

3. **HMAC-SHA2 (Hash Message Authentication Code — Secure Hash Algorithm 2).** SHA2 is stronger than either SHA1 or MD5. Mainly there are 3 variants of SHA2 with different message digest lengths:

    - SHA2-256 — produces a 265-bit (32 byte) message digest;
    - SHA2-384 — produces a 384-bit (48 byte) message digest;
    - SHA2-512 — produces a 512-bit (64 byte) message digest.

HMAC (Hash-based Message Authentication Code)

Is a type of a message authentication code (MAC) that is acquired by executing a cryptographic hash function on the data (that is) to be authenticated and a secret shared key. Like any of the MAC, it is used for both data integrity and authentication.

Diffie-Hellman Key Exchange Algorithm

The Diffie-Hellman (DH) key exchange algorithm is a method used to make a shared encryption key available to two entities without an exchange of the key. The encryption key for the two devices is used as a symmetric

key for encrypting data. Only the two parties involved in the DH key exchange can deduce the shared key, and the key is never sent over the wire.

A Diffie-Hellman key group is a group of integers used for the Diffie-Hellman key exchange.

IKE Protocol

IKE (Internet Key Exchange) is a protocol used to set up security associations for IPSec. These security associations establish shared session secrets from which keys are derived for encryption of tunneled data. IKE is also used to authenticate the two IPSec peers.

Internet Security Association and Key Management Protocol (ISAKMP)

Is used for negotiating, establishing, modification and deletion of SAs and related parameters. It defines the procedures and packet formats for peer authentication creation and management of SAs and techniques for key generation.

Peers

In IPsec, peers are devices or entities that communicate securely either through the exchange of keys or the exchange of digital certificates.

Security Association (SA)

An instance of security policy and keying material applied to a data flow. Both IKE and IPsec use SAs, although SAs are independent of one another.

AH

Defined in RFC 2402. To provide security, AH adds authentication information to the IP datagram. Most VPN tunnels do not use AH because it does not provide encryption.

ESP

Defined in RFC 2406. ESP (Encapsulating Security Payload) provides authentication and encryption of data. ESP takes the original payload of a data packet and replaces it with encrypted data. It adds integrity checks to make sure that the data is not altered in transit, and that the data came from the proper source. ESP is more secure than AH. Mobile VPN with IPSec always uses ESP.

2. Cisco CLI

Procederemo ad una configurazione di tipo **VPN site-to-site** in cui i router di frontiera, posti alle estremità del tunnel, svolgono il ruolo di **IPsec peers**, altresì detti **VPN gateway**. Se pur esista una sottile differenza tra i due termini, di seguito ci riferiremo ad entrambi considerandoli perfettamente sinonimi.

Firmware update

In entrambi i router Cisco coinvolti nel VPN tunnel (R1 e R2) è necessario eseguire l'upgrade delle funzionalità del firmware. A tal proposito, è necessario attivare la licenza **SEC-K9 license** che fornisce, appunto, tutte le features della suite IPSec (algoritmi crittografici, funzioni di hash, protocolli di autenticazione, etc).

Di seguito sono riportati i comandi da eseguire sul router R1. Chiaramente, comandi analoghi devono essere impartiti anche al router R2; prestare dunque massima attenzione alle piccole modifiche richieste.

Come prima cosa modifichiamo l'hostname di entrambi i router, così da poterli facilmente referenziare durante la loro configurazione.

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# end
```

Procediamo adesso alla fase di upgrade del firmware.

```
R1# configure terminal
R1(config)# license boot module c1900 technology-package securityk9

ACCEPT? [yes/no]: yes

R1(config)# end
R1# copy running-config startup-config
Destination filename [startup-config]? # Premere invio

R1# reload
Proceed with reload? [confirm] # Premere invio
```

Configurazione VPN site-to-site router R1

La procedura di negoziazione di un tunnel VPN tra due peers si divide in due fasi e coinvolge, in particolare, il protocollo **IKE.** I principali obiettivi di IKE sono due:

1. autenticare vicendevolmente i due peers;
2. stabilire i protocolli e le chiavi crittografiche da utilizzare nel trasferimento dei dati.

IKE utilizza di norma la porta UDP 500.

Come abbiamo detto, IKE agisce in due fasi successive.

Fase 1

Nella Fase 1 dell'inizializzazione del tunnel, i gateway VPN scambiano le credenziali, si identificano a vicenda e negoziano per definire un insieme comune di impostazioni della Fase 1 da utilizzare.

Qualora la Fase 1 si sia completata con successo, i due gateway VPN dispongono di una associazione sicura (ISAKMP Security Association). Questa SA è valida per un periodo di tempo specificato. Se i due VPN gateway non completano le negoziazioni della Fase 2 prima della scadenza della SA di Fase 1, devono completare nuovamente le negoziazioni della Fase 1.

Dunque, ricapitolando, la Fase 1 di IKE prevede:

- l'autenticazione vicendevole delle identitità dei gateway VPN;
- la creazione di chiavi di sessione, da utilizzare sia durante la negoziazione della stessa Fase 1 che, eventualmente, nella successiva Fase 2.

Di seguito sono riportati i comandi necessari al completamento della Fase 1.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes
R1(config-isakmp)# hash sha
R1(config-isakmp)# group 2
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# exit
R1(config)# crypto isakmp key 123pass address 2.2.2.1
```

Analizziamo nel dettaglio ciascun comando (viene prima riportato il comando e una breve descrizione a seguire).

```
R1(config)# crypto isakmp policy 10
```

Grazie alla keyword **crypto** è possibile accedere alla configurazione del tunnel VPN.

Cos'è **ISAKMP**? Senza addentrarci nei dettagli, il protocollo IKE coopera con il protocollo ISAKMP. ISAKMP definisce le procedure per l'autenticazione e la comunicazione tra i gateway VPN, ed utilizza IKE per la fase di negoziazione delle chiavi crittografiche.

Perché **policy 10**? Due gateway VPN debbono condividere lo stesso set di regole (policy), al fine di concludere con successo la fase di negoziazione del tunnel VPN. In uno stesso peer, quindi, potremmo al contempo definire più di un set di regole (qualora, ad esempio, si abbia la necessità di attivare più tunnel VPN con differenti peers). Ciascuna di esse è identificata con un valore intero nel range [1,10000] che, al contempo, definisce l'ordine. È pratica comune iniziare la numerazione delle policy da 10.

```
R1(config-isakmp)# encryption aes
```

Seleziona l'algoritmo crittografico utilizzato durante la negoziazione della fase 1 (AES).

```
R1(config-isakmp)# hash sha
```

Seleziona la funzione crittografica di hash (SHA-1).

```
R1(config-isakmp)# group 2
```

Definisce quale modulo Diffie-Hellman (DH) utilizzare. L'RFC originale ne definisce due: il group DH 1 utilizza un modulo a 768 bit, mentre il group DH 2 utilizza un modulo a 1024 bit. Maggiore è il valore, più casuale è la chiave, più sicurezza viene garantita. Cisco IOS supporta group 1, group 2 e group 5. La pratica comune è quella di utilizzare il group 2.

```
R1(config-isakmp)# authentication pre-share
```

Quest'ultima opzione della procedura avviata con il comando `crypto isakmp policy 10`, definisce la metodologia di autenticazione: **pre-share**, nel nostro caso, ovvero chiavi precondivise. Sebbene le chiavi precondivise siano il metodo meno sicuro, sono anche le più comunemente utilizzate per autenticare i peer VPN, grazie alla loro facilità e rapidità nella configurazione.

```
R1(config)# crypto isakmp key 123key address 2.2.2.1
```

Quest'ultimo comando della Fase 1 definisce due elementi fondamentali: la chiave precondivisa (123key) e l'indirizzo IP del peer coinvolto nel tunnel VPN (2.2.2.1).

Fase 2

Al termine della Fase 1 è stata correttamente stabilità una ISAKMP Security Association, quindi esiste un canale sicuro tra i due peers, protetto dagli algoritmi crittografici scelti. Inoltre, i due gateway VPN si sono autenticati vicendevolmente.

Utilizzando il canale sicuro ISAKMP, i due host possono ora negoziare la IPSec Security Association, ovvero l'insieme dei parametri che permettono di costruire il tunnel IPSec. Questa fase è detta Quick Mode o Fase 2.

IPSec SA comprende una serie di specifiche che indicano al peer la tipologia di traffico da inviare tramite il tunnel VPN, e come crittografare e autenticare tale traffico.

Di seguito sono riportati i comandi necessari al completamento della Fase 2.

```
R1(config)# crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
R1(config-crypto-map)# description VPN connection to R2
R1(config-crypto-map)# match address 110
R1(config-crypto-map)# set transform-set VPN-SET
R1(config-crypto-map)# set peer 2.2.2.1
R1(config-crypto-map)# exit
```

Analizziamo nel dettaglio ciascun comando.

```
R1(config)# crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
```

Qui entriamo nel vivo delle impostazioni del tunnel IPSec, specificando:

- il protocollo, facente parte la suite IPSec, con cui regolare il flusso comunicativo all'interno del tunnel (ESP);
- l'algoritmo crittografico da applicare al payload (ovvero al pacchetto IP originale), al fine di garantire la confidenzialità dei dati (AES);
- la funzione di hash, sia per autenticare che per verificare l'integrità dei dati (SHA-HMAC).

Più in particolare, a cosa si riferisce `transform-set VPN-SET`?

Un **transform-set** è una combinazione di singole trasformazioni IPSec, progettate per attuare una politica di sicurezza specifica sul flusso di dati che transiterà nel tunnel VPN. Durante la fase di negoziazione della IPSec Security Association, i peers accettano di utilizzare un particolare set di trasformazione che, nel nostro caso risultano essere: protocollo ESP, algoritmo crittografico AES, funzione di hash (SHA-HMAC).

VPN-SET è semplicemente un'etichetta (WORD) con cui è possibile referenziare lo specifico transform-set.

```
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
```

Una mappa crittografica (crypto map) è un'entità di configurazione che svolge tre funzioni principali:

1. seleziona i flussi di dati che richiedono un'elaborazione di sicurezza (referenziando una specifica ACL, nel nostro caso);
2. definisce la politica di sicurezza per questi flussi (ovvero un'associazione con un transform-set);
3. definisce il peer crittografico a cui deve essere indirizzato il traffico (ovvero il gateway VPN remoto coinvolto nel tunnel).

Le tre specifiche sono dettagliate di seguito.

In particolare: `crypto map VPN-MAP` definisce una mappa crittografica denominata VPN-MAP, posizionata nell'elenco delle crypto map disponibili in posizione `10` e che, chiaramente, si poggia sulla suite `ipsec-isakmp`.

```
R1(config-crypto-map)# description VPN connection to R2
```

Definisce una semplice etichetta testuale.

```
R1(config-crypto-map)# match address 110
```

Il flusso di dati interessato dal tunnel VPN è quello referenziato dalla access list numero 110 (si veda di seguito).

```
R1(config-crypto-map)# set transform-set VPN-SET
```

Le trasformazioni di sicurezza applicate al flusso di dati sono quelle definite nel transform-set denominato VPN-SET.

```
R1(config-crypto-map)# set peer 2.2.2.1
```

Indirizzo IP del peer remoto, con cui si intende instaurare il VPN tunnel.

Configurazione interfaccia

Terminata la configurazione delle specifiche IPSec, è necessario applicare il crypto map all'interfaccia outside del router.

```
R1(config)# interface Serial0/1/0
R1(config-if)# crypto map VPN-MAP
```

ACL

Infine, è necessario definire una regola ACL al fine di selezionare i flussi di dati che richiedono l'elaborazione di sicurezza definita da crypto map.

```
R1(config)# access-list 110 permit ip 192.168.0.0 0.0.0.255 192.168.1.0
0.0.0.255
```

Routing

Inizialmente, potremmo erroneamente supporre sia sufficiente definire una regola di instradamento avente le seguenti caratteristiche (ricorda che stiamo parlando del router R1 e che una regola speculare dovrebbe essere definita su R2):

```
- Network: 2.2.2.0
- Mask: 255.255.255.252
- Next Hop: 1.1.1.2
```

Ciò nonostante, una regola così definita, non è esaustiva. Perché? Per rispondere a questa domanda è necessario comprendere più nel dettaglio la logica dei vari moduli che orchestrano il funzionamento di un router Cisco.

Senza entrare nei dettagli, ci è sufficiente sapere che il modulo responsabile delle attività di routing (ovvero quel particolare componente software che, dato un pacchetto IP, stabilisce l'interfaccia di uscita verso cui dovrà essere instradato), viene eseguito **prima** del modulo **crypto**, responsabile delle operazioni di gestione del tunnel VPN IPSec e quindi del transito dei dati al suo interno.

Ciò significa che, affinché sia possibile comunicare con un host della rete privata posta al di là del gateway VPN con cui è stato instaurato il tunnel, è necessario definire una seconda regola di instradamento che stabilisca, appunto, un percorso verso tale rete.

Dovremo dunque aggiungere:

```
- Network: 192.168.1.0
- Mask: 255.255.255.0
- Next Hop: 1.1.1.2
```

In alternativa, è possibile definire una regola di instradamento che comprenda ambedue (regola di default):

```
- Network: 0.0.0.0
- Mask: 0.0.0.0
- Next Hop: 1.1.1.2
```

Troubleshooting e verifica

Visualizza un riepilogo delle specifiche di configurazione dei crypto map e dei transform-set definiti.

```
R1# show crypto map
R1# show crypto isakmp sa
```

Per attivare il tunnel IPSec, è necessario inviare dati significativi alla VPN. Da un host della Sede-1 inviare un semplice ping ad un host della Sede-2 (o viceversa). Poiché la negoziazione IPSec richiede una molteplicità di richieste e risposte (e quindi del tempo), è possibile che i primi pacchetti vadano in timeout.

```
ping 192.168.1.1
```

Al fine di verificare che il tunnel VPN sia stato creato correttamente, devono essere presenti un ISAKMP SA (esito della Fase 1) e un IPSEC SA (esito della Fase 2).

Verificare che il tunnel ISAKMP (Fase 1) sia stato creato con successo.

```
R1# show crypto isakmp sa
```

Verificare che il tunnel IPSec (Fase 2) sia stato creato con successo.

```
R1# show crypto ipsec sa
```