UNIVERSITÀ DEGLI STUDI ROMA TRE

Dipartimento di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi Di Laurea

Realizzazione di un sistema di cyber-defense: utilizzo delle VPN per un accesso remoto sicuro a risorse interne

Laureando

Filippo Visconti

Matricola 547344

Relatore

Prof. Maurizio Patrignani

Anno Accademico 2021/2022

Questa è la dedica

Ringraziamenti

Questi sono i ringraziamenti.

Introduzione

Questa è l'introduzione.

Pericoli di esporre un server su internet

The Security Architecture of the OSI Reference Model (ISO 7498-2) considers five main classes of security services: authentication, access control, confidentiality, integrity and non-repudiation. These services are defined as follows: The authentication service verifies the supposed identity of a user or a system. The access control service protects the system resources against non-authorized users. The confidentiality service protects the data against non-authorized revelations. The integrity service protects the data against non-authorized modifications, insertions or deletions. The non-repudiation service prevents an entity from denying previous commitments or actions.

Termini di base

Computer networks there are a variety of the following types of computer network based on scope. The scope here is how big the computer network will be built. Based on spaces in scope, a computer network can be distinguished into two, namely [1] : a) Local Area Network (LAN), is a computer network which is built in the room a small scope as a single building or group of buildings. LAN is built in a limited scope and usually owned by organizations that already have the devices installed. An internal data rate of the LAN is usually much greater than the WAN. Wide Area Network (WAN), is a network that covers a large geographic area requires delimiters and rely at least partly on the circuit provided by public operators. Typically, a WAN consists of a number

of switching node interconnects. A transmission from one of the devices is channeled through the internal node to the device purpose. This node (including node limit) does not affect the contents of the data, their goal was to provide a switching facility will move data from node to node until they reach their destination. Traditionally, WAN has been implemented using one of the two technologies: circuit switching and packet switching. Recently, frame relay and ATM networks have assumed the lead role which uses it [2]. 2.2 Virtual Private Network Virtual Private Network (VPN) is a computer network where connections between its nodes utilize public networks (internet/WAN) as it may be in certain cases or conditions do not allow it to build its own infrastructure. When the Connect VPN, the interconnection between the node such as an independent network that has actually created a special line pass through connection or a public network. At every company site, workstations, servers, and databases connected by one or more local area network (LAN) a LAN is under the control of the network manager and can be configured and tuned for cost-effective. The Internet or other public networks can be used to connect the sites, provide cost savings over the use of private networks and reduction of the burden of wide area network traffic to providers of public networks [2].

PILA ISO OSI

TCP is the main protocol in TCP/IP networks. The IP protocol process data packets while TCP allow two hosts to exchange data streams and establish a connection. TCP guarantees that packets will arrive their destination in the same order in which they were sent [7]. UDP provides unreliable, minimum, best-effort, message delivery to upper-layer protocols and applications. UDP do not setup a permanent connection between two end points [8].

The adjustments between TCP and UDP regardless of VPN usage is always said to be the same: Speed is sacrifice for reliability as UDP is connectionless and the server sending the data theoretically does not ensure if it reaches the destination or not. TCP is a connection-oriented protocol, which implies that end-to-end communications is set up using handshaking. Once the connection is established, data can be transferred bi-directionally over the link. UDP is a connectionless protocol and therefore less complex message based when compared to TCP, which includes that the point-to-point

connection is not dedicated and data is transferred uni-directional from the source to its destination without checking whether the receiver is active. TCP regulate retransmission, message acknowledgment, and timeout. TCP deliver lost messages along the way upon multiple attempts. In TCP, there is no missing data, and if ever there are multiple timeouts, the connection is dropped. When a UDP message is sent there is no guarantee that the message will reach its destination; it could get dropped along the way. There is no retransmission, timeout and acknowledgment. When two data packets are sent in sequence, the first message will reach the destination first. When data segments arrive in the wrong order, TCP buffers hold the data until all data are re-ordered before being transmitted; when using UDP the order in which messages arrive cannot be predicted. When TCP packets are transmitted from one end to a remote end across the network, the data packets are reordered in the same sequence created by the sender. The protocol notifies when segments of the data stream have been corrupted, reordered, discarded or duplicated by the network. TCP is a reliable protocol as the sender can retransmit damaged segments. However retransmission creates latency.

Necessità di un'infrastruttura di rete sicura

Ancora del testo. Come si afferma i

Organizzazione dei capitoli

Ancora del testo. Come si afferma i

Indice

Elenco delle figure

Capitolo 1

Requisiti

1.1 Caratteristiche della rete aziendale

La rete su cui siamo stati chiamati a lavorare è illustrata nel diagramma seguente.

1.1.1 Diagramma di rete

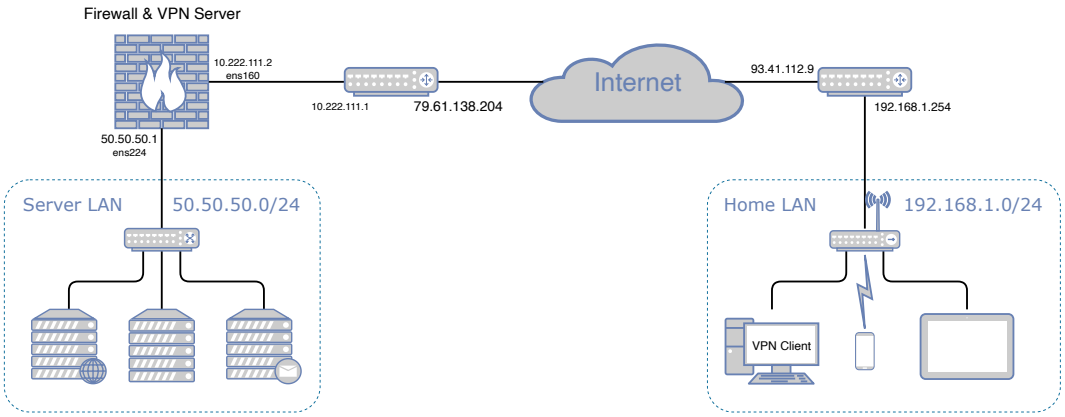

Figura 1.1: Diagramma di rete

La porzione di diagramma che rappresenta l'infrastruttura di rete dell'azienda è quella sinistra della nuvola. La porzione destra raffigura, invece, un'altra sottorete, che per fini di testing immaginiamo sia la rete casalinga di un dipendente dell'azienda. La nuvola sta a rappresentare tutta la rete Internet.

1.1.2 Descrizione dei componenti fondamentali

1.1.2.1 Router

Un componente essenziale all'interno dell'infrastruttura di rete è il router. Il router è un dispositivo di rete che lavora al livello 3 del modello OSI e che permette e gestisce l'instradamento dei pacchetti tra sottoreti diverse. Le tabelle d'instradamento, salvate nella memoria interna del router, contengono informazioni che riguardano come raggiungere gli altri nodi della rete e sono lo strumento che permette al router di instradare correttamente i pacchetti. Queste tabelle associano il prefisso IP [Pos81, RFC0791] e relativa maschera della sottorete di destinazione con il *next-hop*, l'indirizzo IP del prossimo router a cui deve essere destinato al pacchetto affinché si avvicini alla sua vera destinazione, e l'interfaccia di rete del router da cui il pacchetto deve essere inoltrato affinché possa raggiungere il *next-hop*. Generalmente la funzione di routing è svolta da un componente hardware dedicato, che, se di fascia alta, permette di raggiungere prestazioni pari alla velocità della linea - ossia, spedisce i pacchetti alla stessa velocità alla quale li riceve. Tuttavia, è possibile il compito venga svolto da server generici, a patto che siano dotati di un numero adatto di schede di rete, su cui gira un software apposito.

1.1.2.2 Firewall

In entrambi i casi, è comune che il router abbia un firewall [Fre00] integrato. Un firewall è un dispositivo fisico, o un software, che ha come obiettivo la regolazione del traffico di una rete. Ciò avviene applicando una serie di regole che coinvolgono lo stato, la porta e il protocollo dei pacchetti che lo attraversano. L'amministratore di rete ha la responsabilità di inserire regole appropriate al contesto, affinché, tutto ciò che non è strettamente necessario, non venga fatto passare. Un esempio di Firewall software molto conosciuto in ambienti UNIX, è `iptables`. Il seguente comando mostra una regola che *consente* il passaggio di un pacchetto in entrata sul firewall dalla scheda di rete `eth0`, destinato alla porta 22 TCP, e che sia il primo di una comunicazione, o faccia parte di una comunicazione già instaurata.

```
iptables -A INPUT  -p tcp --dport 22 -i eth0 \
```

```
-m state --state NEW,ESTABLISHED -j ACCEPT
```

Il sistema a disposizione avrà un router/firewall installato su un server, che ha come sistema operativo CentOS 7 [Cena] - la versione gratuita di Red Hat Enterprise Linux [Lin].

1.1.2.3 Le LAN utilizzate

Nella configurazione della rete aziendale corrente è presente soltanto una sottorete, denominata LAN dei server, dove risiedono esclusivamente i server che erogano servizi all'esterno dell'azienda. Tutti i servizi per uso interno, destinati a una ulteriore LAN, in questo esempio sono erogati dal router/firewall stesso.

1.1.3 Servizi offerti all'esterno

L'azienda ha necessità di pubblicare

- un sito web, il cui hosting è effettuato sul web server interno;
- un mail server, che si occupa di inviare e ricevere i messaggi di posta elettronica

1.1.3.1 Web servers

Il sito web è servito in HTTP sulla porta 80 e in HTTPS sulla porta 443, e la pubblicazione è affidata a un noto servizio, *Apache Httpd* [Cenb].

1.1.3.2 Mail servers

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.1.4 Servizi offerti all'interno

1.1.4.1 DHCP server

Il Dynamic Host Configuration Protocol [Dro97, RFC2131] è un protocollo ausiliario che permette l'assegnazione automatica degli indirizzi IP e altri parametri di configurazione ai dispositivi connessi alla rete usando una architettura client-server. Il DHCP offre

un servizio non connesso e utilizza UDP come protocollo di trasporto. Il server ascolta le richieste (che saranno broadcast, destinate per convenzione all'IP 255.255.255.255) sulla porta 67 UDP, e inoltra le risposte al client sulla porta 68 UDP. Altri parametri di configurazione che comunemente accompagnano l'appena assegnato indirizzo IP sono i server DNS [Moc87, RFC1034] di default, l'indirizzo IP del default gateway, e la durata per il quale l'IP assegnato è valido.

1.1.4.2 DNS server

Il Domain Name System è il sistema di assegnazione gerarchico e decentralizzato dei nomi che identificano gli host in rete. Un'analogia che aiuta a comprendere la funzione del DNS è quella della rubrica telefonica. Infatti, come nella rubrica telefonica viene mantenuta un'associazione tra un nome - facile da ricordare per una persona - e il relativo numero di telefono - più difficile da ricordare, e facile da confondere -, così il DNS conserva dei *resource records* composti da un nome - `example.com` - associato a un indirizzo IPv4 o IPv6 - `93.130.23.53`. Si tratta di un protocollo di livello 7, che generalmente comunica sulla porta 53 UDP, ma potrebbe sfruttare anche VPN o tunnel, TLS, HTTPS, Tor. È una potenzialità interessante, in quanto le richieste non sono criptate e si potrebbe andare incontro a problemi di sicurezza. Il DNS è in grado di memorizzare anche altre informazioni riguardanti un certo dominio, tra cui:

- i *name servers* che sono autorità per quel dominio - coloro che a loro volta memorizzano i resource records dei vari sottodomini;
- gli indirizzi IP dei mail exchanger di riferimento per quel dominio;
- degli alias, ossia un'associazione tra due nomi di dominio.

Nella configurazione corrente, il server DNS, che gira sullo stesso server del Firewall, lavora come relay e il suo IP viene distribuito a tutti i client della rete interna via DHCP come DNS resolver. Ciò significa che tutti gli host della rete, nel momento in cui devono risolvere un nome, inviano una richiesta al server DNS interno, che si occuperà lui di risolverlo e, una volta ottenuto il risultato, lo restituisce al richiedente. Questo comporta diversi vantaggi, tra cui:

- la comunicazione verso l'esterno per la risoluzione dei nomi avviene da un unico punto della rete;
- si può fare caching, ossia mantenere in memoria per un certo periodo di tempo (che viene specificato nella risposta che il server DNSriceve) le risposte delle varie risoluzioni, cosicché, se di una richiesta si era già trovata la risposta, non dovrà essere fatta di nuovo la risoluzione;
- si possono implementare dei filtri per bloccare la risoluzione di nomi a cui si vuole limitare l'accesso;
- si possono facilmente loggare le varie richieste.

1.1.4.3 Web app interne

All'interno della rete locale dell'azienda, sono accessibili degli applicativi che permettono la gestione di alcuni sistemi, ad esempio degli apparati di rete.

1.1.4.4 File servers

Affinché i dipendenti autorizzati possano collaborare e accedere a file condivisi, è stato predisposto un file server, a cui si può accedere con protocolli quali FTP [PR85, RFC0791] e SFTP. Tuttavia, i file in questione possono contenere dati sensibili. Per questo motivo, è necessario che la risorsa sia adeguatamente protetta, non esposta alla rete esterna e che l'accesso sia regolamentato.

1.1.4.5 Database servers

Similmente al file server, c'è anche un database server a disposizione dei dipendenti, le cui necessità di sicurezza rispecchiano quelle del file server.

1.2 Necessità degli utenti

1.2.1 Accesso ai servizi interni senza esposizione all'esterno

Router, Firewall, IDS, IPS, VPN. Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.2.1.1 Remote work

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.3 Requisiti di sicurezza

1.3.1 Controllo del traffico

1.3.1.1 Proxy interno obbligatorio

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.3.2 Trasmissione sicura dei dati

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.3.2.1 Evitare intercettazioni

Vedi Cina con il Great Firewall

1.3.3 Controllo dei dispositivi

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.3.3.1 Logs

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

1.3.4 Compatibilità

Deve essere compatibile con i 3 OS desktop e i 2 mobile principali

Capitolo 2

Stato dell'arte

2.1 Virtual Private Networks

Una rete privata virtuale consiste in una rete il cui accesso è regolamentato, che si appoggia a un protocollo di trasporto pubblico e condiviso, e che consente di garantire confidenzialità della comunicazione, accesso solo previa autenticazione, integrità dei dati e protezione da alcuni tipi di attacchi, ad esempio Man-in-the-middle o attacco replay.

2.1.1 Architetture disponibili

Una rete VPN può realizzare diversi tipi di collegamenti, per soddisfare esigenze diverse. Nei paragrafi successivi, si andranno ad analizzare i 3 tipi di architetture più comuni:

- Gateway-to-Gateway
- Host-to-Host
- Host-to-Gateway

2.1.1.1 Gateway-to-Gateway

Consiste in una VPN che connette in maniera stabile due reti. Questa configurazione permette ad esempio di estendere una rete privata tra diverse location geograficamente separati e distanti a piacere, oppure di garantire a una serie di uffici un accesso sicuro a un data center.

2.1.1.2 Host-to-Host

Questa configurazione è la meno comune. Consiste nello stabilire una comunicazione diretta tra due host, in cui uno fa da server VPN e l'altro da client VPN. Un caso d'uso potrebbe essere un amministratore di sistema che deve fare gestione remota di un apparecchio.

2.1.1.3 Host-to-Gateway

In questa modalità, il risultato che si ottiene è lo stesso che si avrebbe connettendo un host alla rete locale in cui risiede il server VPN. È usata principalmente per offrire un accesso sicuro da remoto alla rete. Quando l'host vuole instaurare una connessione VPN con il server, gli viene richiesto di autenticarsi.

2.1.2 Perché soddisfano i requisiti

Una VPN in configurazione Host-to-Gateway si prospetta come la soluzione più pratica e funzionale per soddisfare le necessità dell'azienda e dei suoi dipendenti, garantendo loro la possibilità di accedere alle risorse interne attraverso un canale di comunicazione privato, ad accesso controllato, criptato e dove è assicurata l'integrità dei dati.

2.1.3 Soluzioni principali

Tra le soluzioni VPN più comuni troviamo:

- Point-to-Point Tunneling Protocol
- Internet Protocol Security
- OpenVPN
- Wireguard
- Mettere Esempio di Web Based SSL VPN

2.2 Internet Protocol Security

2.2.1 Panoramica

IP Security è una suite di protocolli il cui obiettivo è rendere sicura la comunicazione tra due computer attraverso una rete IP. Contiene protocolli per la mutua autenticazione degli host e per la negoziazione delle chiavi di cifratura da usare durante la sessione. In molti contesti, rendere sicuro il livello di rete (L3 OSI) è una soluzione migliore rispetto a rendere sicuro il livello di trasporto (L4 OSI) o di presentazione (L7 OSI), in quanto offre un ulteriore punto di controllo per gli amministratori e più flessibilità nell'analizzare, e gestire, ogni singolo pacchetto IP. IPSec supporta l'autenticazione a livello di rete, autenticazione del mittente, integrità dei dati, cifratura, e protezione dagli attacchi replay, protezione dall'analisi del traffico e controllo degli accessi.

2.2.2 Transport mode vs Tunnel mode

The IPsec protocols AH and ESP can be implemented in a host-to-host transport mode, as well as in a network tunneling mode.

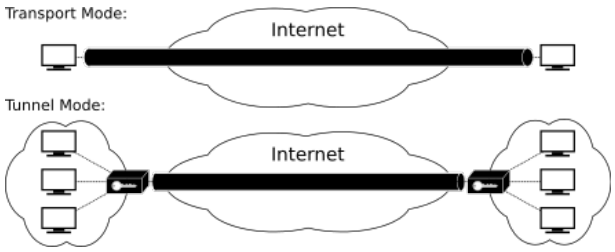

Figura 2.1: Transport mode vs Tunnel mode diagram

2.2.2.1 Transport mode

In transport mode, generalmente solo il payload del pacchetto IP è cifrato o autenticato. L'indirizzamento non cambia, dato che l'header IP non è né modificato né cifrato; tuttavia, quanto si usa il protocollo Authentication Header - approfondito in seguito - l'indirizzo IP non può essere modificato da Network Address Translation, in quanto una modifica al campo invaliderebbe l'hash. Il livello di trasporto e di applicazione sono sempre certificati da un hash, quindi il loro contenuto non può essere modificato

in alcun modo, ad esempio utilizzando una traduzione dei numeri delle porte. Un superamento delle problematiche causate dall'attraversamento di NAT è definito dalle RFC che descrivono il meccanismo NAT-T, ma che va oltre gli scopi di questa tesi.

2.2.2.2 Tunnel mode

In tunnel mode, l'intero pacchetto è cifrato e autenticato. È dunque incapsulato all'interno di un nuovo pacchetto IP con un nuovo header IP. Generando un nuovo header IP, non si incontra nessuna difficoltà nell'attraversamento di NAT.

2.2.3 Protocolli utilizzati

IPSec utilizza i seguenti protocolli per stabilire una connessione sicura. Sia AH che ESP, descritti in seguito, possono lavorare in tunnel mode o in transport mode.

2.2.3.1 Authentication Header

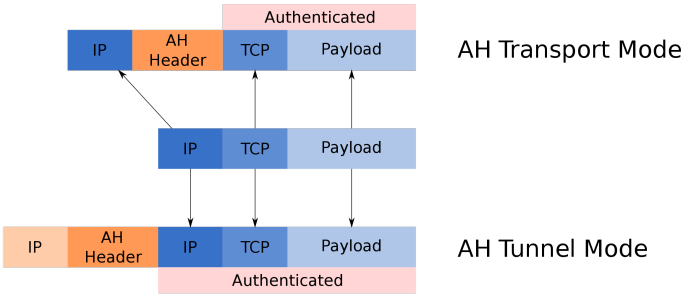

Figura 2.2: Authentication Header packet formats

Authentication Header [Ken05a, RFC4302] garantisce integrità per tutti gli header dei pacchetti, ad eccezione di alcuni campi dell'header IP, e autenticazione del mittente. Se configurato, è anche possibile utilizzarlo per offrire protezione dagli attacchi replay. AH si interfaccia direttamente con IP, utilizzamndo il protocollo IP numero 51. AH autentica l'intero datagramma, ad eccezione dei campi variabili. Tuttavia, le informazioni contenute nel datagramma sono trasferite in chiaro e, dunque, leggibili da uno sniffer. Per questo motivo, AH non soddisfa i requisiti di sicurezza richiesti.

2.2.3.2 Encapsulating Security Payload

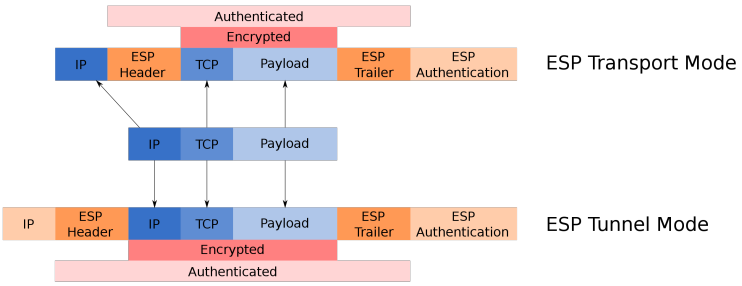

Figura 2.3: Encapsulating Security Payload packet formats

Encapsulating Security Payload [Ken05b, RFC4303] offre confidenzialità dei fati, autenticazione del mittente, controllo di integrità e protezione da attacchi relay. In Transport mode, non autentica né cifra l'header IP: cioè potrebbe esporre le informazioni contenute a potenziali attacchi mentre il pacchetto è in transito. Tuttavia, la Transport mode necessita di meno potenza computazionale, ottenendo un overhead minore della tunnel mode, rinunciando a una maggior sicurezza.

In Tunnel mode, viene creato un nuovo header IP e usato come header esterno del pacchetto, eguito dall'header ESP e poi il pacchetto originale (sia header IP che payload originale). L'ESP Trailer e gli opzionali dati di autenticazione sono aggiunti dopo il payload. Quando si usano cifratura e autenticazione contemporaneamente, ESP protegge completamente il pacchetto orginale, perché diventa il payload del nuovo pacchetto ESP. Da notare è che non viene protetto il nuovo header IP. Un gateway deve necessariamente usare ESP in Tunnel mode.

2.2.3.3 Internet Key Exchange v2

Ancora del testo—IKE è un acronimo per Internet key exchange ed è il protocollo usato per stabilire una security association nella suite di protocolli IPsec. Questo protocollo è definito in RFC 4306. —

Internet Key Exchange [Kau05, RFC4306] è un protocollo che svolge la funzione di negoziazione, gestione e creazione delle Security Associations. Una SA è un insieme di regole necessarie a definire le funzionalità e i sistemi di sicurezza per stabilire una

connessione IPSec. Può essere definita manualmente, anche se non scala dovutamente con VPN di grandi dimensioni. Un metodo più comune è quello di usare una delle cinque possibili modalità di scambio: main, aggressive, quick, informational e group. Le modalità sono differenti per velocità e l'uso di funzioni di cifratura. IKEv2 è la versione più recende di IKE e migliora il protocollo rendendolo più sempice, garantendo affidabilità nel recapito dei messaggi, protezione contro attacchi di tipo DenialOfService e migliora l'uso di IKE su gateways NAT. È un protocollo di livello applicazione e utilizza il protocollo UDP come protocollo di trasporto; la porta su cui viene stabilita la connessione è 500.

2.2.4 Cifratura

IPSec supporta diversi protocolli di cifratura, tra cui AES, Blowfish, Triple DES, Cha-Cha e DES-CBC. Inoltre, usa due tipi di cifratura: simmetrica e asimmetrica. In una codifica simmetrica, una chiave è condivisa tra gli utenti, mentre una asimmetrica fa affidamento su entrambe le chiavi pubbliche e private. La codifica asimmetrica è considerata più sicura: molti utenti condividono la chiave pubblica, ma la sicurezza fa affidamento sulla chiave privata - protetta a tutti costi - che non ha bisogno di essere condivisa con nessuno (a differenza di una chiave simmetrica). IPSec usa la cifratura asimmetrica per instaurare una connessione sicura, per poi sfruttare quella simmetrica per migliorare la velocità di collegamento. Per quello che riguarda il collegamento, è compatibile sia con UDP che con TCP.

2.2.5 Autenticazione

L'autenticazione a chiave pubblica e privata assicura che mittenti e destinatari stiano effettivamente comunicando con il giusto partner. IPSec support molteplici sistemi di autenticazione, tra cui: HMAC-SHA1/SHA2, certificate authorities (CAs), RSA, ECDSA, e pre-shared key (PSK). Ogni tipologia ha i suoi pregi e difetti e casi d'uso in cui è preferibile. Ogni protocollo punta a garantire che i dati rimangano sicuri e affidabili attraverso il loro tragitto.

2.2.6 Implementazioni

StrongSwan è una implementazione open-source di IPSec per Linux. Supporta funzionalità come IPv6, certificati X.509 a chiave pubblica, liste di certificati revocati, storage di chiavi RSA private su smartcard e implementazione completa del protocollo IKEv2.re.

2.2.7 Considerazioni

Ancora del testo

2.3 PPTP

2.3.1 Panoramica

Si tratta di uno dei più vecchi protocolli VPN in uso ancora oggi, ma in quanto tale ha alcune gravi criticità date dall'età. Ad esempio, la crittografia a 128 bit e il protocollo usato per l'autenticazione (MS-CHAP) contenente note vulnerabilità lo rendono ormai un protocollo insicuro, da evitare se le informazioni che transitano sono sensibili. Tuttavia, è estremamente semplice da configurare e il più veloce dal punto di vista prestazionale, il che lo rende ideale per usi quali streaming video o l'utilizzo di VPN su terminali con potenze di calcolo estremamente limitate. È stato sviluppato da Microsoft nel 1999 [HPV+99, RFC2637] e lavora instaurando un canale di controllo tra i due peers sulla porta 1723 TCP e un tunnel GRE su cui transitano effettivamente i dati.

2.3.2 Protocolli utilizzati

2.3.2.1 Generic Routing Encapsulation

GRE è un protocollo di tunneling sviluppato da Cisco Systems che può incapsulare un'ampia varietà di protocolli di livello di rete all'interno di collegamenti Point-to-Point o Point-to-Multipoint virtuali su una rete IP.

2.3.3 Cifratura

PPTP non specifica nessuna

2.3.4 Autenticazione

Ancora del testo

2.3.5 Considerazioni

Ancora del testo

2.4 OpenVPN

2.4.1 Panoramica

OpenVPN è una VPN SSL che permette di incanalare tutto il traffico di una sottorete attraverso una unica porta UDP o TCP, e fa affidamento su OpenSSL. Come le altre soluzioni VPN, OpenVPN servizi essenziali di sicurezza quali autenticazione, cifratura, integrità dei dati e controllo degli accessi. Supporta due modalità di lavoro, routing e bridging:

Routing consiste nell'interconnessione di due sottoreti indipendenti, dove il server VPN (generalmente installato sul router) inoltra i pacchetti all'indirizzo IP specificato in fase di configurazione. Si tratta quindi di un collegamento a livello 3 del modello OSI.

Bridging è una modalità che lavora esclusivamente all'interno di una sottorete; il funzionamento è analogo a quello di uno switch ethernet fisico.

OpenVPN è una soluzione che lavora in user space, dunque l'overhead generato è maggiore in quanto sono necessarie molteplici copie dei pacchetti affiché siano trasferiti dal kernel space allo user space. Supporta l'intero insieme delle funzionalità di TLS, necessitando di una ampia code base, mostrando un maggior potenziale a soffrire di vulnerabilità.

2.4.2 Protocolli utilizzati

Come precedentemente acccennato, OpenVPN usa la libreria di OpenSSL, che implementa il protocollo Transport Layer Security, progettato per offrire una connessione

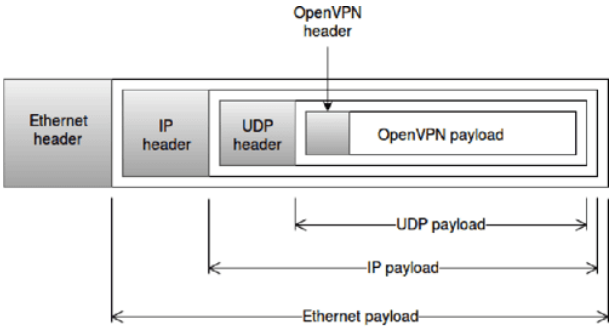

Figura 2.4: Dettaglio di un pacchetto OpenVPN

sicura attraverso una rete non sicura. A differenza del puro TLS, OpenVPN offre all'utente la possibilità di utilizzare una pre-shared key per generare quel che è noto come HMAC firewall, che autentica tutta la sequenza di handshake TLS.

Essendo UDP un protocolllo non connesso, i pacchetti IP criptati e firmati che sono incanalati tramite UDP, non hanno nessuna garanzia di affidabilità. L'affidabilità necessaria per una sicura autenticazione è garantita, però, dal protocollo TLS che utlizza TCP come protocollo di trasporto. È importante notare che il canale dati e il canale di controllo transitano all'interno dello stesso tunnel UDP (o TCP). L'incapsulamento dei pacchetti è descritto dal seguente diagramma.

La struttura mostrata si applica a tutti i pacchetti OpenVPN; tuttavia, differenti pacchetti avranno differenti payloads.

2.4.2.1 Secure Socket Layer/Transport Layer Security VPNs

Il protocollo Transport Layes Security, originariamente noto come Secure Socket Layer, è un protocollo progettato per garantire una connessione sicura attraverso una rete non sicura. TLS permette autenticazione di client e server, integrità dei dati e confidenzialità. Per l'autenticazione usa i certificati X.509 [CSF+08, RFC5280] con una crittografia asimmetrica e si occupa di negoziare una chiave di sessione simmetrica. Un vantaggio delle VPN SSL rispetto a quelle basate su IPSec è che riescono a lavorare anche in reti protette da firewall molto stringenti, in quanto la maggior parte delle aziende non filtra il traffico TCP sulla porta 443, essendo normalmente usato dai dipendenti per accedere a Internet. OpenVPN di default utilizza la porta 1194 UDP, ma, nel caso quella porta

fosse chiusa, può utilizzare la 443 TCP.

2.4.3 TCP vs UDP

Il protocollo TCP utilizza notevoli algoritmi per assicurare un recapito correto dei dati al destinatario. Avere due connessioni TCP una dentro l'altra forzera gli algoritmi di entrambe le connessioni a lavorare in parallelo. Non essendo TCP progettato per lavorare in quella condizione, si potrebbe andare incontro a problemi quali il *retransmission problem*, *TCP meltdown* e doppia ritrasmissione. Questi problemi potrebbero verificarsi nel momento in cui entrambe le connessioni stanno tentando di ritrasmettere pacchetti.

2.4.4 Cifratura

Ancora del testo

2.4.5 Autenticazione

In contrast to pre-share static key mode, TLS mode uses TLS protocol to authen- ticate, establish secure channel and exchange the symmetric tunnel session key between peers. Just like in pre-shared static key mode, session key is used to en- crypt the data tunnel, however, the authentication and symmetric key exchange take place using TLS protocol. This not only provides an automatic and secure way of distributing symmetric keys, but also a way to renew the symmetric key at any point during the communication. The aforementioned aspect of the TLS mode provides the Perfect Forward Secrecy, which is not present in pre-shared static key mode. The structure of the tunnel session key derivation TLS packet, as shown in Wireshark, can be seen in Fig. 4. The transfer of tunnel session keys are encrypted and carried inside the TLS Record layer, so it cannot be decrypted without the proper TLS certificates. The two main steps in this protocol are shown below. 1. Negotiation of the TLS connection. Both sides of the connection are au- thenticated by exchanging certificates and verifying the certificate of the opposing side. If the authentication is successful, the protocol proceeds wit the step two. Otherwise, the connection is terminated. 2. Tunnel session keys are negotiated over the already established secure TLS channel. The tunnel session key derivation TLS packet structure depends on the OpenVPN key method being used. TLS mode

supports two key methods, which are described below. If the first key method is used, then the tunnel session keys are derived from OpenSSL cryptographic library RAND bytes function. The tunnel session key derivation TLS packet structure is shown in Tab. 3. (b) If the second key method is used, (default in the OpenVPN 2.0+), then the tunnel session keys are derived from the RAND bytes func- tion passed through the TLS pseudo-random function (TLS PRF). In order to successfully construct a OpenVPN client, it is important to un- derstand the key differences explained in this chapter between the TLS modes, key methods and their respective packet structures.

2.4.6 Misure di sicurezza aggiuntive

Ancora del testo OpenVPN offers various internal security features. It has up to 256-bit encryption through the OpenSSL library, although some service providers may offer lower rates, effectively providing some of the fastest VPN available to consumers. It runs in userspace instead of requiring IP stack (therefore kernel) operation. OpenVPN has the ability to drop root privileges, use mlockall to prevent swapping sensitive data to disk, enter a chroot jail after initialization, and apply a SELinux context after initialization.

OpenVPN runs a custom security protocol based on SSL and TLS, rather than supporting IKE, IPsec, L2TP or PPTP.

OpenVPN offers support of smart cards via PKCS 11-based cryptographic tokens.

2.4.7 Considerazioni

Ancora del testo

2.5 WireGuard

2.5.1 Panoramica

In Linux, the standard solution for encrypted tunnels is IPsec, which uses the Linux transform ("xfrm") layer. Users fill in a kernel structure determining which ciphersuite and key, or other transforms such as compression, to use for which selector of packets traversing the subsystem. Generally a user space daemon is responsible for updating these data structures based on the results of a key exchange, generally done with IKEv2

[13], itself a complicated protocol with much choice and malleability. The complexity, as well as the sheer amount of code, of this solution is considerable. Administrators have a completely separate set of firewalling semantics and secure labeling for IPsec packets. While separating the key exchange layer from the transport encryption— or transformation—layer is a wise separation from a semantic viewpoint, and similarly while separating the transformation layer from the interface layer is correct from a networking viewpoint, this strictly correct layering approach increases complexity and makes correct implementation and deployment prohibitive. WireGuard does away with these layering separations. Instead of the complexity of IPsec and the xfrm layers, WireGuard simply gives a virtual interface—wg0 for example—which can then be administered using the standard ip(8) and ifconfig(8) utilities. After configuring the interface with a private key (and optionally a pre-shared symmetric key as explained in section 5.2) and the various public keys of peers with whom it will communicate securely, the tunnel simply works. Key exchanges, connections, disconnections, reconnections, discovery, and so forth happen behind the scenes transparently and reliably, and the administrator does not need to worry about these details. In other words, from the perspective of administration, the WireGuard interface appears to be stateless. Firewall rules can then be configured using the ordinary infrastructure for firewalling interfaces, with the guarantee that packets coming from a WireGuard interface will be authenticated and encrypted. Simple and straightforward, WireGuard is much less prone to catastrophic failure and misconfiguration than IPsec. It is important to stress, however, that the layering of IPsec is correct and sound; everything is in the right place with IPsec, to academic perfection. But, as often happens with correctness of abstraction, there is a profound lack of usability, and a verifiably safe implementation is very difficult to achieve. WireGuard, in contrast, starts from the basis of flawed layering violations and then attempts to rectify the issues arising from this conflation using practical engineering solutions and cryptographic techniques that solve real world problems.

For key distribution, WireGuard draws inspiration from OpenSSH, for which common uses include a very simple approach toward key management. Through a diverse set of out-of-band mechanisms, two peers generally exchange their static public keys. Sometimes it is simple as PGP-signed email, and other times it is a complicated key

distribution mechanism using LDAP and certificate authorities. Importantly, for the most part OpenSSH key distribution is entirely agnostic. WireGuard follows suit. Two WireGuard peers exchange their public keys through some unspecified mechanism, and afterward they are able to communicate. In other words, WireGuard's attitude toward key distribution is that this is the wrong layer to address that particular problem, and so the interface is simple enough that any key distribution solution can be used with it. As an additional advantage, public keys are only 32 bytes long and can be easily represented in Base64 encoding in 44 characters, which is useful for transferring keys through a variety of different mediums. Finally, WireGuard is cryptographically opinionated. It intentionally lacks cipher and protocol agility. If holes are found in the underlying primitives, all endpoints will be required to update. As shown by the continuing torrent of SSL/TLS vulnerabilities, cipher agility increases complexity monumentally. WireGuard uses a variant of Trevor Perrin's Noise [23]—which during its development received quite a bit of input from the authors of this paper for the purposes of being used in WireGuard—for a 1-RTT key exchange, with Curve25519 [5] for ECDH, HKDF [15] for expansion of ECDH results, RFC7539 [17]'s construction of ChaCha20 [3] and Poly1305 [8] for authenticated encryption, and BLAKE2s [2] for hashing. It has built-in protection against denial of service attacks, using a new crypto-cookie mechanism for IP address attributability. Similarly opinionated, WireGuard is layer 3-only; as explained below in section 2, this is the cleanest approach for ensuring authenticity and attributability of the packets. The autqwhors believe that layer 3 is the correct way for bridging multiple IP networks, and the imposition of this onto WireGuard allows for many simplifications, 3 resulting in a cleaner and more easily implemented protocol. It supports layer 3 for both IPv4 and IPv6, and can encapsulate v4-in-v6 as well as v6-in-v4. WireGuard puts together these principles, focusing on simplicity and an auditable codebase, while still being extremely high-speed and suitable for a modicum of environments. By combining the key exchange and the layer 3 transport encryption into one mechanism and using a virtual interface rather than a transform layer, WireGuard indeed breaks traditional layering principles, in pursuit of a solid engineering solution that is both more practical and more secure. Along the way, it employs several novel cryptographic and systems solutions to achieve its goals.

2.5.2 Protocolli utilizzati

Ancora del testo

2.5.2.1 Something

Ancora del testo

2.5.3 TCP vs UDP

Ancora del testo

2.5.4 Cifratura

Ancora del testo

2.5.5 Autenticazione

Ancora del testo

2.5.6 Considerazioni

Ancora del testo

Capitolo 3

Realizzazione

3.1 Virtualizzatore

Ancora del testo

3.1.1 Caratteristiche e funzionamento

Ancora del testo

3.1.2 VirtualBox

Ancora del testo

3.1.3 VMWare ESXi

Ancora del testo

3.2 Installazione e configurazione dei servizi VPN

Ancora del testo

3.2.1 IPSec - tunnelmode

Ancora del testo

3.2.1.1 Installazione di stronSwan

Ancora del testo

3.2.1.2 Configurazione del Firewall

Ancora del testo

3.2.2 OpenVPN over TCP

Ancora del testo

3.2.2.1 Installazione di openvpn

Ancora del testo

3.2.2.2 Configurazione del Firewall

Ancora del testo

3.2.3 WireGuard

Ancora del testo

3.2.3.1 Installazione di wireguard

Ancora del testo

3.2.3.2 Configurazione del Firewall

Ancora del testo

Capitolo 4

Testing

4.1 Modalità di esecuzione dei test

Ancora del testo

4.1.1 Panoramica di iperf3

Ancora del testo

4.1.2 Scelta della configurazione di test

Ancora del testo

4.1.3 Criteri di valutazione

Ancora del testo

4.1.3.1 Throughput

Ancora del testo

4.1.3.2 MTU

Ancora del testo

4.1.3.3 Packetloss

Ancora del testo

4.2 Misure senza VPN

Ancora del testo

4.3 Misure con IPSec e IKEv2

Ancora del testo

4.4 Misure con OpenVPN over TCP

Ancora del testo

4.5 Misure con WireGuard

Ancora del testo

4.6 Analisi delle misure

Ancora del testo

Capitolo 5

Security concerns

5.1 Principali problematiche di sicurezza

Ancora del testo

5.2 Attacchi mirati agli utenti

Ancora del testo

5.3 Attacchi mirati al sistema

Ancora del testo

5.4 Multi-factor authentication

Ancora del testo

5.4.0.1 Certificato

Ancora del testo

5.4.0.2 Username e password

Ancora del testo

5.4.0.3 One Time Password

Ancora del testo

Conclusioni e sviluppi futuri

Quale è uscito vincitore

Ancora del testo

Come migliorare le misure

Ancora del testo

Test di VPN Peer-To-Peer

Ancora del testo

Zero Tier

Ancora del testo

Bibliografia

[Cena] CentOS. https://www.centos.org.

[Cenb] CentOS. https://httpd.apache.org.

[CSF+08] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280, RFC Editor, May 2008. http://www.rfc-editor.org/rfc/rfc5280.txt.

[Dro97] Ralph Droms. Dynamic host configuration protocol. RFC 2131, RFC Editor, March 1997. http://www.rfc-editor.org/rfc/rfc2131.txt.

[Fre00] N. Freed. Behavior of and requirements for internet firewalls. RFC 2979, RFC Editor, October 2000.

[HPV+99] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-point tunneling protocol (pptp). RFC 2637, RFC Editor, July 1999.

[JS96] Trevor H. Jones and Il-Yeol Song. Analysis of binary/ternary cardinality combinations in entity-relationship modeling. *Data Knowledge Engineering*, 19(1):39–64, 1996.

[Kau05] C. Kaufman. Internet key exchange (ikev2) protocol. RFC 4306, RFC Editor, December 2005. http://www.rfc-editor.org/rfc/rfc4306.txt.

[Ken05a] S. Kent. Ip authentication header. RFC 4302, RFC Editor, December 2005. http://www.rfc-editor.org/rfc/rfc4302.txt.

[Ken05b] S. Kent. Ip encapsulating security payload (esp). RFC 4303, RFC Editor, December 2005. http://www.rfc-editor.org/rfc/rfc4303.txt.

[Lin] Red Hat Enterprise Linux. https://www.redhat.com/en.

[Moc87] P. Mockapetris. Domain names - concepts and facilities. STD 13, RFC Editor, November 1987. http://www.rfc-editor.org/rfc/rfc1034.txt.

[Pos81] Jon Postel. Internet protocol. STD 5, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc791.txt.

[PR85] J. Postel and J. Reynolds. File transfer protocol. STD 9, RFC Editor, October 1985. http://www.rfc-editor.org/rfc/rfc959.txt.