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

Prova di testo di capitolo.

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].

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

Introduzione **iv**

Pericoli di esporre un server su internet iv

Termini di base iv

Necessità di un'infrastruttura di rete sicura v

Organizzazione dei capitoli v

Indice **vi**

Elenco delle figure **x**

1 Requisiti **1**

1.1 Caratteristiche della rete aziendale 1

1.1.1 Diagramma di rete 1

1.1.2 Descrizione dei componenti fondamentali 2

1.1.3 Servizi offerti all'esterno 3

1.1.4 Servizi offerti all'interno 3

1.2 Necessità degli utenti 5

1.2.1 Accessi ai servizi interni senza esposizione all'esterno 5

1.3 Requisiti di sicurezza 6

1.3.1 Controllo del traffico 6

1.3.2 Trasmissione sicura dei dati 6

1.3.3 Controllo dei dispositivi 6

1.3.4 Compatibilità 6

2 Stato dell'arte **7**

2.1 Virtual Private Networks 7
2.1.1 Concetti fondamentali 7
2.1.2 Architetture disponibili 7
2.1.3 Soluzioni principali 8
2.1.4 Perché soddisfano i requisiti 9
2.2 IPSec 9
2.2.1 Panoramica 9
2.2.2 Protocolli utilizzati 10
2.2.3 Transport mode vs Tunnel mode 11
2.2.4 Cifratura 11
2.2.5 Autenticazione 11
2.2.6 Implementazioni 11
2.2.7 Considerazioni 11
2.3 PPTP 11
2.3.1 Panoramica 11
2.3.2 Protocolli utilizzati 12
2.3.3 TCP vs UDP 12
2.3.4 Cifratura 12
2.3.5 Autenticazione 12
2.3.6 Considerazioni 12
2.4 OpenVPN 12
2.4.1 Panoramica 12
2.4.2 Protocolli utilizzati 13
2.4.3 TCP vs UDP 14
2.4.4 Cifratura 16
2.4.5 Autenticazione 16
2.4.6 Misure di sicurezza aggiuntive 17
2.4.7 Considerazioni 17
2.5 WireGuard 17
2.5.1 Panoramica 17

2.5.2 Protocolli utilizzati 20
2.5.3 TCP vs UDP 20
2.5.4 Cifratura 20
2.5.5 Autenticazione 20
2.5.6 Considerazioni 20

3 Realizzazione **21**
3.1 Virtualizzatore 21
3.1.1 Caratteristiche e funzionamento 21
3.1.2 VirtualBox 21
3.1.3 VMWare ESXi 21
3.2 Installazione e configurazione dei servizi VPN 21
3.2.1 IPSec - tunnelmode 21
3.2.2 OpenVPN over TCP 22
3.2.3 WireGuard 22

4 Testing **23**
4.1 Modalità di esecuzione dei test 23
4.1.1 Panoramica di iperf3 23
4.1.2 Scelta della configurazione di test 23
4.1.3 Criteri di valutazione 23
4.2 Misure senza VPN 24
4.3 Misure con IPSec e IKEv2 24
4.4 Misure con OpenVPN over TCP 24
4.5 Misure con WireGuard 24
4.6 Analisi delle misure 24

5 Security concerns **25**
5.1 Principali problematiche di sicurezza 25
5.2 Attacchi mirati agli utenti 25
5.3 Attacchi mirati al sistema 25
5.4 Multi-factor authentication 25

Conclusioni e sviluppi futuri **27**

Quale è uscito vincitore 27

Come migliorare le misure 27

Test di VPN Peer-To-Peer 27

Zero Tier 27

Bibliografia **28**

Elenco delle figure

1.1 Diagramma di rete 1

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

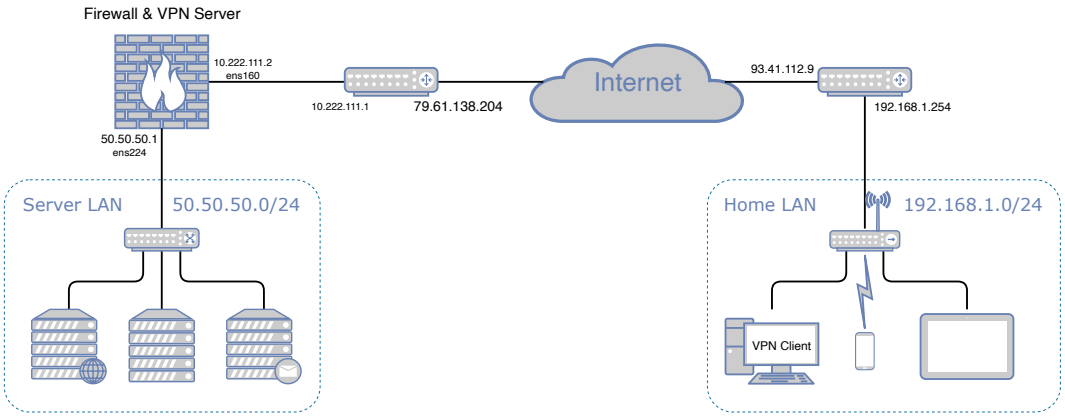

Figura 1.1: Diagramma di rete

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 Accessi 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 dagli attacchi, ad esempio di tipo Man-in-the-middle o replay.

2.1.1 Concetti fondamentali

Ancora del testo

2.1.2 Architetture disponibili

In all three architecture types, in order to connect to the host, typically must authenticate itself. This is usually done either by gateway itself of by consulting a dedicated authentication server.

2.1.2.1 Gateway-to-Gateway

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

2.1.2.2 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.

2.1.2.3 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.

2.1.3 Soluzioni principali

Tra le soluzioni VPN più comuni troviamo:

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

2.1.3.1 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.

One advantage of SSL VPNs over an IPsec VPNs is that SSL VPNs can connect to more restricted environments where NAT (Network Address Translation) or strict

firewall rules are used. The reason for this is because most organizations do not filter the traffic on the TCP port 443, as it is usually used for employees to securely access Internet. For instance, OpenVPN, SSL- based VPN, uses UDP Port 1194 for secure data transmission. However, in the case that port is filtered, OpenVPN can also make use TCP port 443. The advantages of using UDP as a transport protocol are discussed at the beginning of the Chapter 4. There are two primary types of SSL VPNs, namely SSL portal VPNs and SSL tunnel VPNs [15]. • SSL portal VPN works over a single network port, namely TCP 443 and acts as a Transport layer VPN. It allows users to connect with most web browsers to access web related content. • SSL tunnel VPN is used to access multiple network services through a tunnel that is running under SSL. The main difference from portal SSL VPNs is that tunnel SSL VPNs allow accessing multiple network services, including applications and protocols that are not web-based. The require- ment for SSL tunnel VPN is that it must be able to handle different active content like Java, JavaScript, Flash, ActiveX [15]. Being able to use more services, this tunnel has more capabilities compared to a portal type VPNs, but it may prevent some users from being able to connect to VPN. The SSL VPN tunnels are created in SSL, but just like in IPsec tunnels, IP traffic is fully protected by the tunnel [15].

2.1.4 Perché soddisfano i requisiti

Prova di testo di capitolo. Vorrei citare qui tutta l'opera omnia di

2.2 IPSec

2.2.1 Panoramica

IPsec is a framework of open standards for ensuring private communications over IP networks which has become the most commonly used network layer security control [11]. IPsec is based on securing Network layer of TCP/IP model. In many environments securing Network layer is a better solution than securing higher Transport or Application layers. It makes a way for network administrators to enforce certain security policies, and also provides a more flexible way in protecting IP information for each packet [11]. Depending on the implementation IPsec can provide a combination of following secu-

rity measures: confidentiality, integrity, peer authentication, replay protection, traffic analysis protection and access control.

2.2.2 Protocolli utilizzati

As noted earlier, IPsec uses multiple additional protocols to establish a secure connection [11]

2.2.2.1 Authentication Header

Authentication Header (AH) , defined in RFC 4302 [12], provides integrity protection for all packet headers (except few IP header fields) and user authen- tication. Optionally it can provide replay and access protection. AH is not able to encrypt data.

2.2.2.2 Encapsulating Security Payload

Encapsulating Security Payload (ESP) , defined in RFC 4303 [13], has two modes: tunnel and transport. Tunnel mode can provide encryption and integrity protection for an encapsulated IP packet as well as authentication for ESP header, while transport mode can provide encryption and integrity protection for the payload of an IP packet and integrity protection for the ESP header.

2.2.2.3 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. 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. —

Internet Key Exchange (IKE) is used to negotiate, create and manage Secu- rity Associations (SA) [11]. SA is a set of rules needed to define the features and security mechanisms for the establishing a IPsec connection. It can be defined manually, however it does not scale well with large-scale VPNs. A more common method is using one of the five possible IKE exchange modes - main, aggres- sive, quick, informational or group. The modes differ in speed and the usage of their cryptographic primitives for establishing

a secure connection. IKEv2 is the newest version of IKE, and it improves the protocol in the following ar- eas: clearly defined RFC (RFC 5996 [14]), simplicity, reliable message delivery, protection against Denial of Service (DoS) attacks, and improved usage of IKE with Network Address Translation (NAT) gateways [11].

2.2.3 Transport mode vs Tunnel mode

Ancora del testo

2.2.4 Cifratura

Ancora del testo

2.2.5 Autenticazione

Ancora del testo

2.2.6 Implementazioni

StrongSwan is an open source IPsec implementation for the Linux oper- ating system [18]. Maintained by Andreas Steffen, strongSwan supports features, such as IPv6, Android 4+, X.509 public key certificates, certifi- cate revocation lists, RSA private key storage on smartcards, ability to interoperate with various MS Windows and Mac OS X VPN clients, full implementation of IKEv2 protocol, and much more.

2.2.7 Considerazioni

Ancora del testo

2.3 PPTP

2.3.1 Panoramica

Point-to-Point Tunneling Protocol (PPTP) is a virtual private network imple- mentation method which uses TCP control channel and a Generic Routing En- capsulation (GRE) tunnel to encapsulate Point-to-Point Protocol (PPP) [16] packets and send them over

TCP/IP links. The protocol was developed by a vendor consortium and documented in RFC 2637 [17]. PPTP encapsulated virtual network packets inside the PPP packets, which are then encapsulated

3 Background 13 inside the GRE packets and these encapsulated inside the TCP control channel. Everything is then sent over IP network on TCP port 1723.

2.3.2 Protocolli utilizzati

Ancora del testo

2.3.2.1 Something

Ancora del testo

2.3.3 TCP vs UDP

Ancora del testo

2.3.4 Cifratura

NESSUNA

2.3.5 Autenticazione

Ancora del testo

2.3.6 Considerazioni

Ancora del testo

2.4 OpenVPN

2.4.1 Panoramica

OpenVPN is a SSL VPN which implements the OSI layer 2 and 3 secure network extension [19]. It allows any IP subnetwork being tunneled over a single UDP or TCP port

and completely relies on the security of OpenSSL. The high-level client-server communication is shown in Fig. 1. Just like other VPNs, the OpenVPN provides the essential security services, such as authentication, encryption, integrity protection, and access control. As noted in Chapter 3.1, understanding and selecting the correct networking components largely important. Therefore, in this next chapter, two different OpenVPN networking possibilities are presented. The OpenVPN supports two networking modes for connecting networks, namely routing and bridging [31]. • Routing is essentially an interconnection of independent subnets. The router is a Level 3 (OSI model) device which forwards the packet according to a specified IP address. • Bridging is a much simpler infrastructure because it operates only on the same local network (subnet). Bridge is a Layer 2 (OSI model) device which forwards packets based on the physical MAC address instead of IP address.

On the other end of the spectrum is OpenVPN, a user space TUN/TAP based solution that uses TLS. By virtue of it being in user space, it has very poor performance—since packets must be copied multiple times between kernel space and user space—and a long-lived daemon is required; OpenVPN appears far from stateless to an administrator. While TUN/TAP interfaces (say, tun0) have similar wg0-like benefits as described above, OpenVPN is also enormously complex, supporting the entire plethora of TLS functionality, which exposes quite a bit of code to potential vulnerabilities. OpenVPN is right to be implemented in user space, since ASN.1 and x509 parsers in the kernel have historically been quite problematic (CVE-2008-1673, CVE-2016-2053), and adding a TLS stack would only make that issue worse. TLS also brings with it an enormous state machine, as well as a less clear association between source IP addresses and public keys.

2.4.2 Protocolli utilizzati

The OpenVPN utilizes the OpenSSL encryption library which itself imple- ments TLS. Originally known as Secure Socket Layer (SSL), TLS is a protocol designed to provide a secure connection over an insecure network. TLS uses X.509 digital certificates4 with asymmetric cryptography to authenticate coun- terparty and negotiate a symmetric tunnel session key. TLS is most widely known for being used over Hypertext Transfer

Protocol (HTTP) to provide en- cryption and authentication, which led to the naming of HTTPS, which stands for HTTP Secure. To establish a secure connection, TLS uses Handshake, ChangeCipherSpec, and Alert subprotocols. Multiple different implementations of TLS, including OpenSSL, were fuzzed and tested by de. Ruiter [6] and therefore this paper is not concerned about testing security of TLS itself. Unlike a vanilla TLS, the OpenVPN gives the user an opportunity to use a static key (or pre-shared passphrase) to generate a what is known as HMAC firewall, which authenticates the whole TLS handshake sequence. The details of HMAC firewall, as well as the security features it provides are explained in Chapter 4.5. The OpenVPN multiplexes the TLS session used for authentication and key exchange (Control Channel) with the actual encryption tunnel data stream (Data Channel)(Fig. 2) [3]. As UDP is connectionless protocol, the encrypted and signed IP packets are tunneled over UDP without any reliability guarantee. The reliability needed for secure authentication is provided by the TLS protocol which uses TCP as its reliability layer. It is important to note that control and data channels are explained separately, however, they are inside the same UDP (or TCP) tunnel hence on the same network layer. The general structure of the OpenVPN packet and ensuing encapsulation can be seen in Fig. 3. It is explained in detail further in section 4.1.1. As shown, the OpenVPN data is encapsulated inside the UDP (or less commonly TCP) layer. The structure shown in Fig. 3 applies to all OpenVPN packets, however, different packets will have different OpenVPN payloads. For instance, P CONTROL V1 packets will have TLS protocol data encapsulated inside the OpenVPN payload.

2.4.2.1 Something

Ancora del testo

2.4.3 TCP vs UDP

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]. IV. COMPARISON BETWEEN TCP AND UDP 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 des- tination or not. TCP is a connection-oriented protocol [9], 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 connec- tionless protocol and therefore less complex message based [10] 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 [11]. 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 drop 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. The TCP protocol has extensive algorithms to ensure correct delivery of the data. Having two TCP connections stacked together will thus force the algorithms of both TCP con- nections to work in parallel [12]. TCP was not designed to work this way and problems are likely to occur in different situations. The retransmission problems, TCP meltdown and double retransmit, are problems caused by tunneling TCP in TCP. The problems can occur when both of the stacked con- nections are retransmitting packets. In previous work, related to TCP in TCP tunneling, it is not entirely clear, how severe the retransmission problems really are. TCP protocol suite allows automatic recovery of any dropped or lost data even if a host goes down in the network. 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 [11]. TCP is a reliable protocol as the sender can retransmit damaged segments. However retransmission creates latency. TCP was designed to make an efficient protocol with little overhead, a protocol set having a basic amount of 'extra' data being transferred. This extra data is named as overhead, and they package the data that needs to be transferred. TCP tunnel is a technology that combines and transmits packets sent between end hosts as a single TCP connection. When using a TCP tunnel, the fairness among aggregated flows can be improved and several protocols can be transparently transmitted through a firewall

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.

[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.

[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.