CS338: Computer Security
Professor: Jeff Ondich
11/13/24

OpenVPN protocol - a brief introduction

Prompt Eua-anant

First of all….

What is a VPN?

In a nutshell, a VPN establishes a **network** of devices where communication between devices on this network is **private,** but the medium of communication is not exclusive to the network.[1]

That's a lot to take in. Let's break that down:

- **Network** - a network is basically a group of devices (laptops, PCs, printers, whatever) that can communicate among themselves.
- **Private** - while this can mean different things in different contexts, private here means that access to communicated contents is limited to a defined set of devices, or put simply, our "friends." More than that, private also means that non-friends are completely unaware of the private nature of the communication.
- The medium of communication is not exclusive to the network - you are still using the internet from your internet service provider, which does not exclusively provide service to your network.

So, to put all of that together, a VPN is a group of devices that can communicate with each other **privately** despite using a public medium of communication, eg. the global Internet.

What is OpenVPN?

It is many things:[2]

- A software that lets you establish a VPN
- The name of a company that offers the software and other products
- One of the most popular **protocol** for establishing a VPN

In this document, OpenVPN refers to the **protocol** for establishing a VPN, which is used by the OpenVPN software and other popular VPNs such as NordVPN and expressVPN. The protocol

[1] http://sol.te.net.ua/www/nanog/vpn.pdf
[2] https://nordvpn.com/blog/what-is-openvpn/

is not part of an RfC yet - they are still working on it.[3] Meanwhile, we can read what OpenVPN makers have provided us thus far.

How does it work

Say you want to send your banking info to cs338.jeffondich.com via OpenVPN protocol. How does that work?

Short version:

1. The user connects to an openVPN server by exchanging **control channel packets** with the server. This sets up how further communication to and from the VPN server will be encrypted.
2. Suppose the user wants to send banking information to cs338.jeffondich.com.
3. The user first sends a **data channel packet** to the VPN server, which goes something like “I want to send [banking info] to cs338.jeffondich.com”
4. The VPN server relays the request to cs338.jeffondich.com, using an IP address that’s different from the user
5. cs338.jeffondich.com responds to the VPN server
6. The VPN server relays the response to the user

Longer version:

1. User and VPN server performs a TCP handshake
2. The user and the VPN server establish a TLS session so that **control channel packets** are sent privately[4]
3. The user sends a **control channel packet** to configure the VPN session[5].

—----------QUICK & FUN DETOUR ON CONTROL CHANNEL PACKETS—----------

Each **control channel packet** consists of the following[6]

Opcode

- [5 bits] packet type
    - Is the packet a data or control channel packet? Also, is it an acknowledgement of a received control channel packet (**P_ACK_V1**), or a normal control packet (**P_CONTROL_V1**)?
- [3 bits] key_id

---

[3] https://github.com/openvpn/openvpn-rfc?tab=readme-ov-file
[4]Keijser, Jan Just, and Crist, Eric F. *Mastering OpenVPN : Master Building and Integrating Secure Private Networks Using OpenVPN*, Packt Publishing, Limited, 2015, page 8.
[5] https://build.openvpn.net/doxygen/network_protocol.html
[6] https://github.com/openvpn/openvpn-rfc?tab=readme-ov-file

    - Refers to an already negotiated TLS session

Identification stuff

- [64 bits] user session_id - to identify the TLS session.
- [128-512] bits - HMAC info, if HMAC authentication is used
- [64 bits] replay packet ID - used for protecting against packet replay attacks
- [8 bits] acked_pktid_len - how many packets have been acknowledged by the remote server
- [32 * n bits] acked_pktid_list - a list containing the packet_ids that have been acknowledged by the remote server
- [64 bits] peer session id
- [32 bits] packet_id

And last but not least, the **payload**, which tells the VPN server what encryption key should be used for the **data channel packets**

The payload consists of[7]…

- Literal 0 (4 bytes).
- Key method (1 byte)
- key_source structure
    - Consists of random numbers used for generating keys for encrypting data channel packets
- Options string length, including null (2 bytes).
- Options string
    - This is to check whether the client and server's configurations are compatible
- [The username/password data below is optional, record can end at this point.]
- Username string length, including null (2 bytes).
- Username string (n bytes, null terminated).
- Password string length, including null (2 bytes).
- Password string (n bytes, null terminated)

----------DETOUR FINISHED----------

4. The VPN server decrypts the control channel packet, read it, and decided what keys should be used for encrypting data channel packets
5. The user sends an encrypted **data channel packet** to the VPN server, which consists of…

    Opcode (see above)
    Payload (in this case the banking info intended for cs338.jeffondich.com)

[7] https://build.openvpn.net/doxygen/network_protocol.html

6. The VPN server decrypts the packet and relays it to cs338.jeffondich.com. This uses the IP address of the VPN server, making it as if the packet originates from the VPN server itself.
7. cs338.jeffondich.com sends a response to the VPN server
8. VPN server encrypts the response using the same method established earlier, and sends it to the user
9. User decrypts the response.

Upshot

Do VPNs increase your privacy?

Now, you may ask, is using VPN useless for privacy since most websites nowadays use HTTPS? For those who don't know, HTTPS is an addition to the HTTP protocol that (1) encrypts all communication between client and server, and (2) the server needs to send a legitimate certificate to the client and prove their identity before the client can send information to the server. That's absolutely true, but a VPN increases your privacy in addition to HTTPS:

VPNs hide info about what server you are connecting to

Using a VPN prevents eavesdroppers from figuring out the DNS of the server you connect to. A DNS is the server's IP address and hostname, like 172.233.221.124 and cs338.jeffondich.com. The eavesdropper only sees that you send packets to a VPN, but doens't know that your packets are relayed to cs38.jeffondich.com. In contrast, HTTPS doesn't hide the DNS, or the domain name and IP address of the server you are connecting to. So even if your banking information is encrypted, eavesdroppers or your ISP can still determine which websites you send data packets to.

VPN hides your IP address from the server you are communicating with

Remember step 4 from the short version? The VPN server relays your packet to the server using its own IP address. So, from the server's side, it doesn't see your IP at all. This means that your identity and geographical location won't be revealed to the server, adding a layer of privacy in addition to HTTPS.

Avoid geographical restrictions

Some websites, like Netflix, restrict what content the user can access based on the user's geographical location.[8] For example, some films may have different release dates for various countries. If you want to view a film before it becomes available in your country, you can connect to a VPN server in some location where the film is available. Then, when you request for a film,

[8] https://help.netflix.com/en/node/118959

the Netflix server sees that the incoming IP address is from a valid location, sends the film back to the VPN server, which passes it on to you.

Limitations

Encryption is only guaranteed between VPN user and VPN server

Suppose cs338.jeffondich.com uses HTTP protocol without encryption. The data packets sent from the user to cs338.jeffondich.com is visible to the internet service provider, as well as eavesdroppers with their packet sniffers.

But will using a VPN make it any better? Not really. Yes, the data packets intended for cs338.jeffondich.com are encrypted, but when it reaches the VPN server, it gets decrypted and relayed to cs338.jeffondich.com, which does not enforce any encryption protocol. An eavesdropper can still intercept and read the packets sent from the VPN server to cs338.jeffondich.com.

Data packets can be read by the VPN server

If the server you want to send data to doesnt use any encryption protocol (eg. HTTP servers), the banking info you send to the VPN server are encrypted according to the terms you established with the VPN server. Thus, the VPN server can decrypt the data packets and voila, the VPN server has your banking info.

Your browsing pattern and IP address is visible to the VPN server

Needless to say, the VPN server needs to know what server you intended to send the packets to. More than that, it needs to know your IP address so that it can send the server's response to your device. Now, does this mean that VPN means nothing for privacy? Not at all, if the VPN server keeps those info private, ie. not sharing it with other parties. The question is whether you can trust the VPN server with that.

HTTP/1.1 302 FOUND
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 15 Nov 2024 16:36:37 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 197
Connection: keep-alive

Location: /fdf/
Vary: Cookie
Set-Cookie:
session=.eJwtzrsRwzAIANBdVKcAfUB4GR8gOKe14yqX3ZMib4L3LnuecR1le513PMr-XGUrtUp
ADuvMcxqmhWN3WTJqqNIYjRw5UHxUE8RZgcVGAikARRPH6YzDMNLSOAwBOVMqixJhCE
XobAgEE5v3hQu8urI27drLL3Jfcf435fMFnOsu_A.Zzd4lQ.39wZ1oWNUGxiUi8UwTXtxOevusA;
Path=/