TechNative

TRACKLIB VPN

Summary of discussion about VPN improvements

opdrachtgever:	Tracklib
door:	Ronald Uitermark
datum:	October 31 2022
report:	TNT02
versie:	1

Inhoudsopgave

1 Introduction

This document contains a summary of the discussion about current VPN setup and possible improvements.

2 Current softwarestack: Wireguard

Newest VPN technology that has recently been developed. I consider it a successor of IPSEC which is another successor of OpenVPN. Recent developments have moved towards more static setups that allow kernel acceleration of common VPN operations thereby improving performance and reducing context switchtes.

OpenVPN has been considered because it's mostly software defined and has, from my point of view, 2 major advantages:

- Most stable clients on a wide set of operating systems.
- Capable of pushing routes and DNS from the server if the client is able to support this. Clients must support these routes combined with the different OS systems and their own configurations.

Decision: We will not be exploring the possibility of using OpenVPN. The fact that Wireguard is already implemented gives the benefits of this new technology and the drawbacks have been no concern during roll-out. Also reimplementing a new VPN across the Tracklib organization gives considerable costs.

3 Issue connecting to remote services

José mentioned that the most common operational issue is connecting to services that use non fixed IP addresses. This is the primary issue to be resolved. We inquired about which services where involved and Tracklib confirmed that they were using ECS. We are assuming that:

- Applications that run on EC2 machines will always be reachable using the same private IP even if the instance stops and starts. This is not the case for public IPs. I don't consider this

applicable but this situation can be remediated with AWS CloudMap and a private hosted zone.

- Another more plausible explanation is the fact that you are talking to ECS services. ECS private ips change whenever a task (Kubernetes: pod) is rebooted. This can be remediated by either using a private loadbalancer in front of ECS or using CloudMap.
- The private loadbalancer has more stable private ips that should not change unless the loadbalancer has interruptions. This solution does not require changing the DNS on the client. The most simple solution is using host entries with fixed ip addresses.
- The CloudMap solution requires that the client uses AWS DNS to resolve addresses from the AWS private hosted zones.

3.1 Setting up new users is a repeating and cumbersome task

Just like any other VPN solution you must setup a new user and hand out credentials in the form of (encrypted) keysets. This could at least be improved by automating this process:

- Allow each user in AWS to use a self-service system (e.g. Lambda) to generate their own keys and let the Lambda register the user in the Wireguard system. This solution is 'safe' as the Lambda permissions can be wider than the user permissions.
  - You should consider the revoking process: If a user is revoked from AWS then they might still have the keyset obtained which is still valid. There are several solutions:
    * A seperate administration to do whitelisting or blacklisting. Generally this only works if the software has hooks to check these (I know OpenVPN has this). WireGuard does not have this but there is an unofficial implementation that exists: https://www.eduvpn.org/taming-wireguard-in-eduvpn/
- Use a generic keyset and let the user authenticate with AWS again somehow when connecting over VPN. OpenVPN has LDAP for this which is also not available for WireGuard.

- WireGuard approach would be to build a small tool that (re)generates wg0.conf on request and reimplements a new list for clients (which also includes the whitelist of clients). This could be triggered by an IAM User delete event on EventBridge (in another account if needed).