Proposal: Cross Network Secure Service Networking

Georges Ankenmann - 100935237
School Of Information and Technology
Carleton University
Ottawa, Canada
georgesankenmann@cmail.carleton.ca

Fahmida Haque - 101137370
School Of Information and Technology
Carleton University
Ottawa, Canada
fahmidahaque@cmail.carleton.ca

Jeff Bailey - 101020551
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
jeffbailey@cmail.carleton.ca

Abstract—Cross network secure service networking is starting to be a more common requirement these days where certain services must be accessible but proper security controls must also be implemented to prevent unauthorized access. Traditionally this can be accomplished by means of a virtual private network (VPN), but this does not always work with services or applications that can't run the required VPN software. Cross network secure service networking allows for services to communicate with each other across networks by establishing an encrypted and authenticated overlay network that allows for secure communication at the service level. This proposal will compare existing frameworks, and propose a new framework that offers security as the preliminary design consideration.

I. Introduction

Cross network secure service networking is starting to be a more common requirement these days where certain services must be accessible but proper security controls must also be implemented to prevent unauthorized access.

VPN are present in a modern connected world. One instance is in corporate environments which allows secure access to protected services from untrusted environments, such as the public internet. Secure access to corporate services can be accomplished by means of a client to site or a site to site virtual private network (VPN). Generally with a site to site VPN, the tunnel are usually established by dedicated hardware. With a client to site VPN, the tunnel is usually established by software on a client device [?] [?].

Overlay networks are common in modern data centres where they allow abstraction of certain network functions that can run on top of physical networks [?]. A common network overlay is Virtual eXtensible Local Area Network (VXLAN) [?]. This framework allows data centres to offer tenants with their own isolated network domains without deploying additional infrastructure. Network encryption is available for VXLAN by means of IPSec.

Cross network secure service networking allows for services to communicate with each other across networks by establishing an encrypted and authenticated overlay network that allows for secure communication at the service level. Cross network secure service networking can also be leveraged by legacy applications that don't offer in transit data encryption by means of a bastion host [?] that can functions as a proxy which would take in unencrypted data then pipes it to a service or client over an encrypted tunnel. A common example of such a device are terminal servers [?] that connect to supervisory control and data acquisition (SCADA) and legacy equipment that only have serial connection or telent, and offer the user a secure session by means of a SSH session or VPN.

Multiple commercial offerings exist for software applications, notably Consul and Google Compute Engine (GCE) Identity-Aware Proxy (IAP). They offer software that enables cross network secure service networking. Consul can run on almost any operating system, while IAP is Google Cloud specific where it allows a identity aware proxy that allows secure access to non public services.

Consul is software that is majorly used for discovery and configuration for a range of applications and services, it provides an up to date outlook on services in an infrastructure. Consul was mainly built to manage services in a distributed system. One of the main challenges faced by many large scale industries, is the heightened use of distributed systems and service discovery has come as a blessing to the challenges faced by the use of distributed systems. Consul is made user-friendly with the use of a flexible and powerful interface. Consul requires a data plane and it runs with a built-in proxy. All the services discovered by consul in an infrastructure is stored in a single registry so that the services can find each other by storing the information of IP addresses or other location information, this makes consul a centralized service registry. Clients can find services that are registered with consul with the help of DNS or HTTP interfaces which leads to an easy procedure to discover resources [?].

Identity-Aware Proxy (IAP), is a service from Google which provides a "central authorization layer for applications accessed by HTTPS" [?]. IAP works by only

granting users or so-called “members” access when they posses the correct role. This allows for access control policies to be specific to resources and applications [?]. IAP works by intercepting web requests, authenticating the user making the request, and only granting access when the user is in fact authorized. This technique of implementing authentication and authorization remotely removes the need for a VPN [?]. Google’s goal with IAP is to create a method that users may connect to untrusted networks and access services without needing a VPN [?] – this directly mitigates the issues surrounding VPN usage, such as reduced speed. As well, users of VPN’s must either have full access or no access to the network, and service granularity is extremely limited – IAP solves this by providing authentication on an application by application basis.

As the previously mentioned Consul run in software while Google IAP runs only within Google Cloud. Consul is not designed to operate with physical equipment.

In this project we will be experimenting with available products along with a custom solution that we will design that will meet our requirements. We will be using Docker Engine with Docker Compose scripts to experiment with the software along with creating a consistent setup for a more accurate comparison.

II. Problem Statement

The issue you have with overlay networks is that it must be implemented at the network level, it is not implemented at the service level. This is where cross network secure service networking comes into play. This proposal will compare existing frameworks, and propose a new framework that offers security as the preliminary design consideration.

III. Lab Setup

We have setup a lab environment that is consistent and easy to use. We used Docker Engine with Docker Compose scripts to allow for a easy lab setup. A Github repository has been provided with all our research and scripts [?].

After installing Docker Engine and Docker Compose on our machines, we then started to experiment with designing scenarios on how to best implement the software we found.

IV. Metrics

One of the first tasks we had to complete was to determine what metrics we were looking for to determine the ”best” solution. We settled on the following metrics.

- Latency
- Throughput
- Resource utilization of host and containers
- Packets dropped

V. Software

After many hours of research we settled on the following software.

A. Backend Software (out of scope of this project)

For this project we used some software in the backend to make our lives much easier to analyze the data and provide meaningful results. The following list of software is not within the scope of this project as it does not actually relate to the problem statement, but it offers the possibility to accomplish research relating to the problem statement.

- Kibana
- Elasticsearch
- Mosquitto MQTT Broker

B. Lab Software (in scope of this project)

For this project we used some software in our lab that we determined that would be best for the project.

- Docker Engine
- Docker Compose
- Shadowsocks
- stunnel
- Wireguard
- Wireshark
- NGINX

VI. Lab Setup

A. Base Lab Setup

B. Wireguard Lab Setup

C. Stunnel Lab Setup

D. Shadowsocks Lab Setup

VII. Results

A. Base Results

B. Wireguard Results

C. Stunnel Results

D. SHadowsocks Results

References

[1] Norton, What is a VPN?, Norton, 2020. Accessed on: September 20, 2020. [Online]. Available: https://us.norton.com/internetsecurity-privacy-what-is-a-vpn.html

[2] Raritan, Serial Console Server - Your Next Generation Solution, Raritan, 2020. Accessed on: September 20, 2020. [Online]. Available: https://www.raritan.com/products/kvm-serial/serial-console-servers/serial-over-ip-console-server

[3] J. Tyson, C. Pollette, S. Crawford, How a VPN (Virtual Private Network) Works, How Stuff Works, 2019. Accessed on: September 20, 2020. [Online]. Available: https://computer.howstuffworks.com/vpn.htm

[4] J. Kozlowicz, What’s a Jumpbox or Bastion Host, Anyway?, Green House Data, 2019. Accessed on: September 20, 2020. [Online]. Available: https://www.greenhousedata.com/blog/whats-a-jumpbox-or-bastion-host-anyway

[5] SDxCentral, What are Network Overlays, SDxCentral, 2015. Accessed on: September 20, 2020. [Online]. Available: https://www.sdxcentral.com/networking/virtualization/definitions/get-on-top-of-network-overlays/

[6] IETF, Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, IETF, 2014. Accessed on: September 20, 2020. [Online]. Available: https://tools.ietf.org/html/rfc7348

[7] Consul, Consul by HashiCorp, HashiCorp, 2020. Accessed on: September 20, 2020. [Online] https://www.consul.io/

[8] Google Cloud, Identity-Aware Proxy (IAP), Google, 2020. Accessed on: September 20, 2020. [Online] Available: https://cloud.google.com/iap
[9] Google Cloud, Identity-Aware Proxy: Documentatio , Google, 2020. Accessed on: October 4, 2020. [Online] Available: https://codelabs.developers.google.com/codelabs/user-auth-with-iap/index.html?index=..%2F..index#0
[10] T. Treat, API Authentication with GCP Identity-Aware Proxy, Medium.com, January 25th, 2019. Accessed: October 4, 2020. [Online]. Available: https://blog.realkinetic.com/api-authentication-with-gcp-identity-aware-proxy-3a4147b30770
[11] G. Ankenmann, J. Bailey, F. Haque ITEC5102 Project Repository, Github.com. [Online]. Available: https://github.com/devagent42/ITEC5102F-Project