ITEC5102F Progress Presentation

Georges Ankenmann

Jeff Bailey

Fahmida Haque

Table of Contents

- Background
- Problem Statement
- Work Plan
- Metrics
- Software that will be used
- Experimentation Process
- Lab setup
    - What has been completed
    - What needs to be completed
- Quick Demo and Questions

Background

Background

- Suggest ways to implement simple cross network secure service networking for IoT and legacy products
- Have a way to encrypt traffic at a service level without changing the service itself
- Missing "drop in security" for legacy applications
- Remove the need for VPNs
- Remove the need to open ports
- Less management
- Service based tunnels
- Not too many papers exists on this topic

Experimentation Success

- What counts as a success?
    - Data is encrypted
    - Latency is “reasonably” different

Problem Statement

Problem Statement

- Overlay networks must be implemented at the network level rather than the service level
- A comparison of existing frameworks and the proposed framework that offers security as the preliminary design consideration to:
    - Demonstrate improved performance
    - Show lower maintenance and management requirements

Work Plan

Work Plan

- Proposal Phase – *Completed*
- Research Phase – *Completed*
  - Finding and reading papers (concise literature review)
  - Determine metrics
  - Find software
  - Determine best course of action
- Experimenting Phase – *Current Phase*
  - Setup software
  - Prepare experimentation
  - Gather metrics and data
- Reporting Phase
  - Report on metrics and data
- Presentation Phase
  - Present reported metrics and data

Metrics

Metrics

- To determine the true value of this idea it must be compared to the typical technique it aims to replace – VPNs
- Two systems will be implemented and compared:
  - The proposed system that is security as a service, without changing the service itself (our proposed idea).
  - A docker VPN container that will act as proxy to other containers (services) needing to access the internet (typical implementation)
- Thus the following will be compared across both systems
  - Latency
  - Throughput
  - Number of encrypted packets
  - Number of clear text packets
  - Resource utilization of host

Metrics

- To gather metrics a few techniques are used
  - Elasticsearch (Kibana) is used to visualize and log packet data between server and client services
  - Latency will be determined by timestamping packets (time sent and time received)
  - Wireshark is used to capture and inspect packets (encrypted vs unencrypted)
  - Resource utilization information can be returned for each docker container which in a system of containers can be summed to return total utilization
- These metrics will be recorded and reported on for both the proposed “security as a service” and typical container VPN techniques

Software that will be used

Software that will be used

- Elasticsearch
- Kibana
- Docker Engine
- Docker Compose
- Libvirt – QEMU/KVM
- Shadowsocks
- Wireguard
- Consul
- Wireshark
- Python
- MQTT
- Stunnel

Lab Setup

What has been completed?

- Base lab setup
- Docker Images
- MQTT Server
- MQTT Client
- HTTP Server
- HTTP Client
- Manual Wireshark packet capture

What must be completed?

- Encryption tunnels
- Kibana Graphs
- Automatic Wireshark packet capture
- Reporting on metrics
- Reporting on data

Quick Demo and Questions