ACCESS POINT ANALYTICS

A Data-Aggregation Platform for TCU Network Services

Justis Clark - Hung Doan - Ryan Finnegan - Matt Liddy - Bradley Schoeneweis
Faculty Advisor: Bingyang Wei, Ph.D (COSC); Craig Baugh (TCU IT)

TCU INFORMATION TECHNOLOGY
Technology Services

DEPARTMENT OF
Computer Science

TCU COLLEGE OF SCIENCE & ENGINEERING

AP.ANALYTICA

Introduction

Background

Across TCU's campus, the IT department has over 3,200 Access Points (APs) setup to provide wireless access to students all they way from the Greek Village to the Parking Lots on Sandage Ave. These APs allow student and faculty devices to stay connected seamlessly while on, or traveling across campus. As students roam in between buildings, and connect to a variety of Access Points, the APs transmit real-time information to a service called **Cisco Prime.** Cisco Prime is a software interface that provides TCU Network Services with a massive data pipeline. Through Cisco Prime, Network Services has developed a list of **Key Performance Indicators** that they look for when monitoring the system as a whole.

Problem

As previously mentioned, this data stream is massive. This leads to a lot of sifting around Cisco Prime in order to find actionable insights. Issues can go unnoticed, and this can lead to a longer response time with limited possibility for proactive measures. Network services needs a better way to search for KPIs.

Goal

Our platform aims to provide **proactive** and **reactive** insights to TCU Network Services that are summarized, and actionable. These snapshot insights will be extracted from the massive data stream collected by wireless Access Points all over TCU's campus, and then presented through an internal **Web-Application Interface**.

Example of a

Cisco Wireless Access Point

Key Performance Indicators (KPIs)

Anomaly Detection
Access Point Utilization
Channel Utilization
Client Count
Coverage Holes
Power/Channel Fluctuation
Rogue Access Points

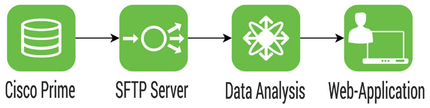
Solution Focuses

Automation — Automatically ingest, analyze, and clean data related to the KPIs on an adjustable schedule

Consolidation — Gather and filter pertinent information for each KPI, while simultaneously performing calculations in real-time to keep the data true and reusable.

Simplification — Present the data to the user through information rich tables and clean, comlex data visualizations.

Solution Overview



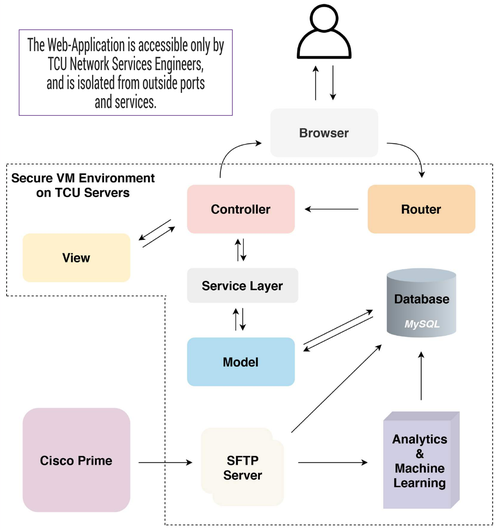
- Data is incrementally imported from the Cisco Prime Server via SFTP onto the secure VM that houses the Web-Application.
- The data is seperated by KPI and undergoes various analyses and calculations, before being cleaned and imported to the MySQL database.
- The aggregated data is then presented to the user through tables and visualizations that are customizable, filterable, and searchable through various means.

Technology Stack

C# · .NET Core · cisco · Chart.js · GitLab · MySQL · JS

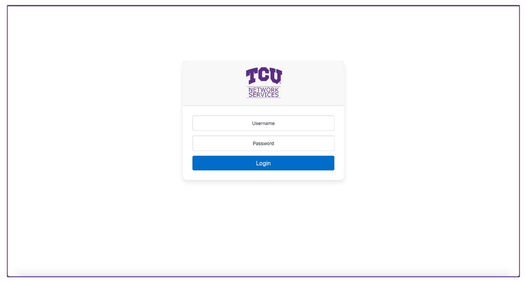
System Architecture

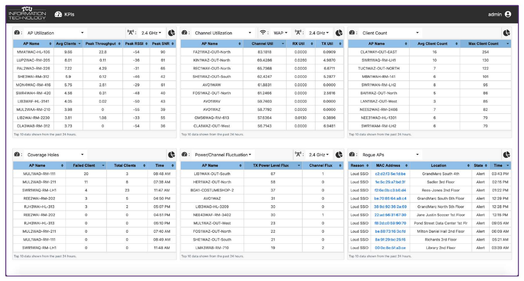
The Web-Application is accessible only by TCU Network Services Engineers, and is isolated from outside ports and services.

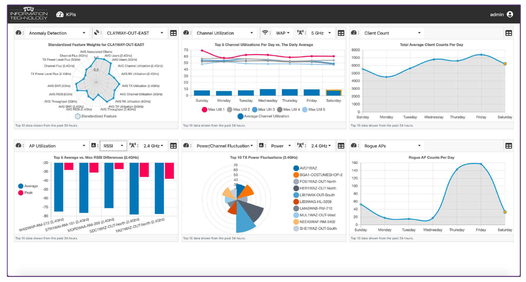


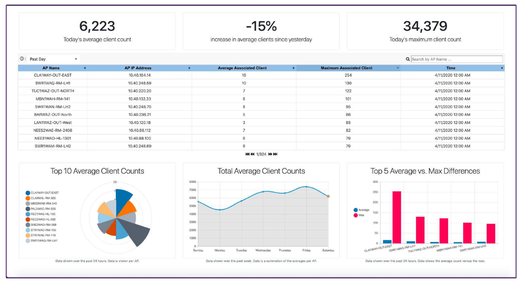
Our solution uses an injectable **Model-View-Controller** (MVC) Architecture that is housed within a secure virtual machine (VM) on a TCU Windows Server. This architecture is portable, secure, and maintainable for future users and developers.

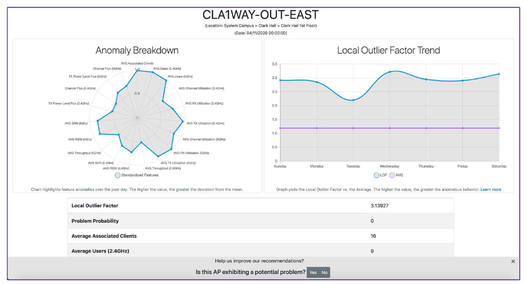
The Web Application

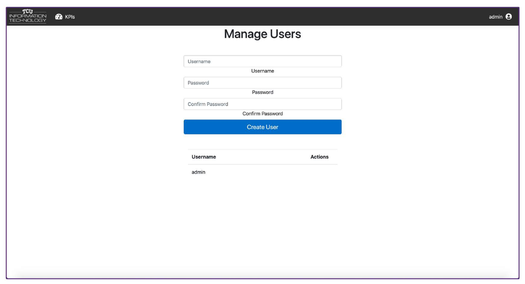












Acknowledgements

Our team would like to thank TCU Network Services, specifically Craig Baugh for letting us be creative and giving us the opportunity to develop this Web-Application, and always being accessible.

We'd also like to thank Dr. Bingyang Wei for his dedication to our class and our team, even with the unorthodox final semester we've had, and the COSC Faculty for supporting us these past 4 years.

References

- .NET Core Documentation: https://docs.microsoft.com/en-us/dotnet/
- jQuery Documentation: https://api.jquery.com/
- Chart.JS Documentation: https://www.chartjs.org/docs/latest/
- jQuery Tablesorter Documentation: https://mottie.github.io/tablesorter/docs/
- Stack Overflow: https://stackoverflow.com/