SOFTWARE PROPOSAL DOCUMENTATION:
PROJECT APATE

(Version 1.0)

Lead Developer: Jeff Benson

08.10.2020

APATE - ASSET PROTECTION & ADVERSARIAL TRACKING ENVIRONMENT

Current information environments require constant monitoring to keep up with emerging adversarial threats to company infrastructure and client data.

Security Operations Centers (SOCs) often find themselves task-saturated or understaffed.

Threats come in the form of DDOS attacks, Social Engineering Attempts, or even Malicious Insiders, but an often-overlooked threat is simply one of lost or stolen equipment.

WHAT PROBLEM DOES PROJECT APATE SOLVE?

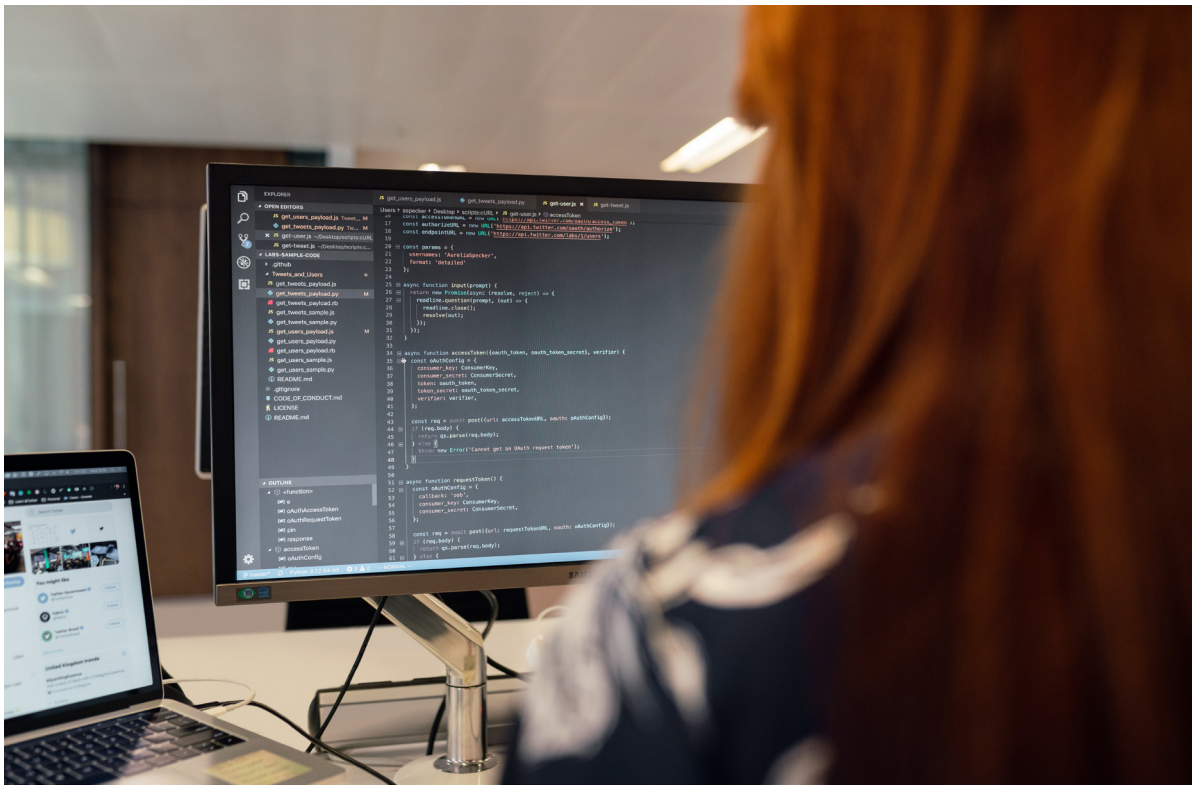
APATE takes the strain of tracking lost or stolen corporately owned assets and simplifies it so that SOCs can do what they do best: defeating threats.

APATE is a simple software service installed on newly provisioned devices that runs at startup.

After report of a lost or stolen asset, the SOC team members can simply log into a premade dashboard to view the last time the device was connected to the Internet.

With this data, the SOC team can attempt to locate the device, or turn the data over to local authorities for police reports.

This service saves companies time by efficiency, money by recovering assets, and protects potential customer data leakage by controlling the asset remotely.

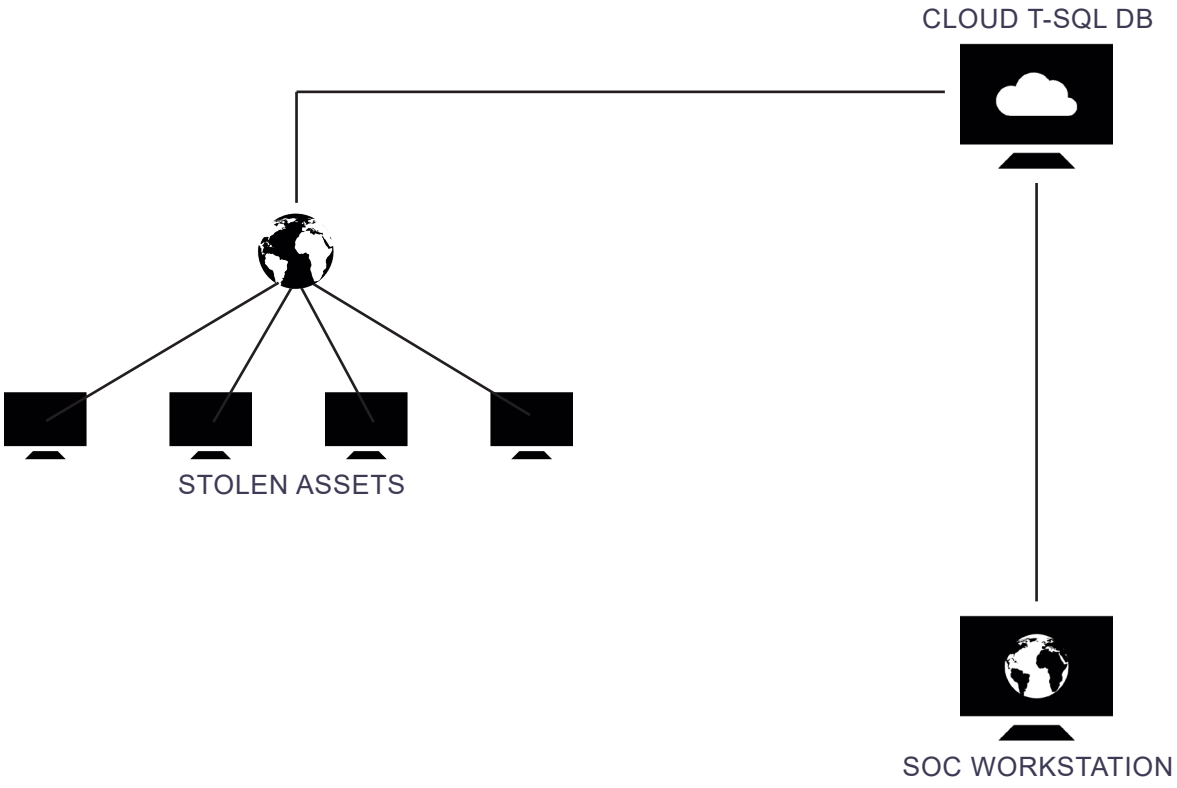


FUNCTIONALITY

APATE is installed with other corporate software installations. It runs at every startup with administrator privileges and requires no end-user interaction. **APATE** is always running and updating a backend SQL server at set intervals.
Once company field technician loses possession of device, the SOC can pull the last known geolocation, public IP, private IP and the hostname of the computer.

With this ever-updating information, the SOC can attempt to geolocate a device based off its various Public IP addresses over time, therefore establishing patterns and timelines for law enforcement. SOC can issue remote commands to wipe sensitive data and remove login password to ensure that the adversary is able to get a network connection for callback to SQL backend.

Because of a flaw in the way Wi-Fi protocols work, the SOC can also remotely insert commonly trusted public SSIDs (such as Starbucks Free Wi-Fi) into settings to force a connection through KARMA vulnerabilities.



REFERENCES & COMPONENTS

APATE is a combination of three pieces of software:

- It uses a T-SQL backend that will always be running. It only requires building the database, tables and setting security settings.

- It requires the actual apateservice.exe that will run on the device.

- It will require an ASP.NET web application to visualize the T-SQL backend.

For the purposes of this project, the ASP.NET web app will be the dashboard of choice. But in the future, this system could easily be integrated into a pre-existing Kibana (ELK Stack) environment.

- Microsoft Socket Programming Documentation

https://docs.microsoft.com/en-us/dotnet/framework/network-programming/socket-code-examples

- "Gray Hat C#: A Hacker's Guide to Creating and Automating Security Tools" (Book)

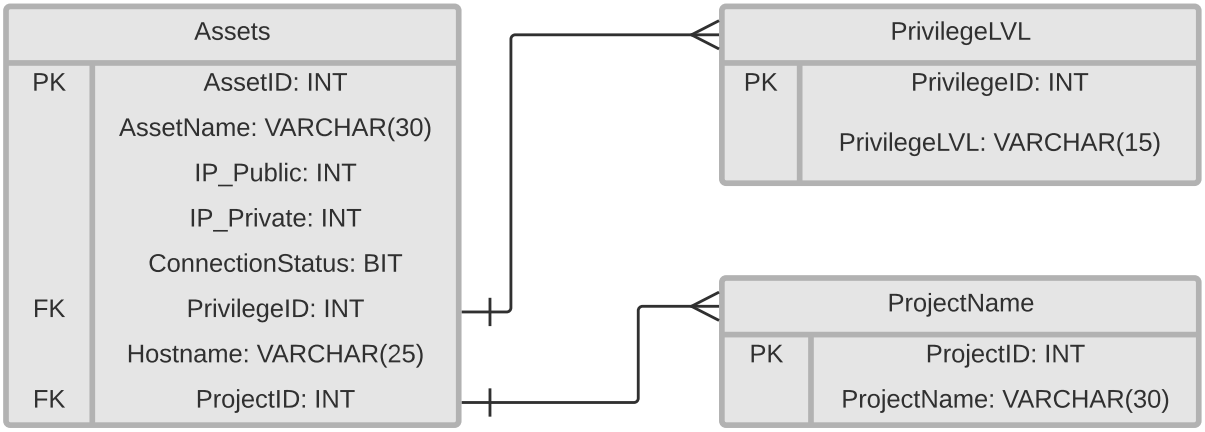
ISBN-13: 978-1593277598

ISBN-10: 1593277598

- T-SQL Fundamentals - Microsoft

- Microsoft Visual C# Step by Step – Microsoft

Entity Relationship Diagram (ERD)



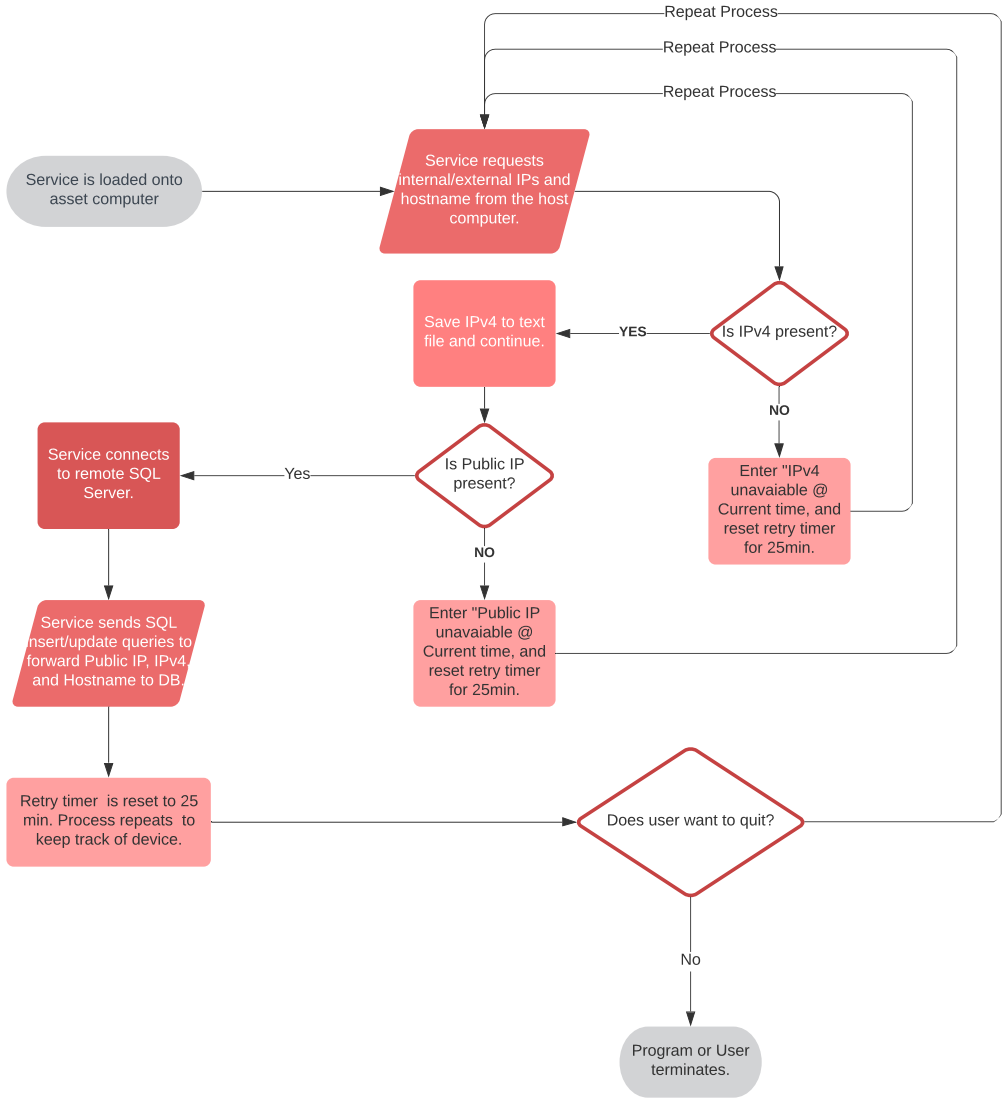
SIMILAR PRODUCTS

Currently there are other similar projects but with a lack opensource code, risks and vulnerabilities inherent in other's code are introduced into production environments.

Having control of proprietary code will enable adaptable and agile security operations that adjust to changing needs.

An example of a similar project would be https://preyproject.com/.

Flowchart



Threats, Countermeasures & Defensive Tactics

SQL Injection

- Parameterized SQL Queries
- Parameterized Stored Procedures
- Escape Routines for Special Characters
- Database Least Privilege
- Access Levels
- Input Constraints at DB and Client level (Length & Format)

XSS Attacks

- Anti-XSS Libray through NuGet
- Output Encoding (Characters treated as Data)
- HTML Encode Methods
- URL Encode Methods

Directory Traversal

- Path.Combine
- Absolute Paths
- Content Security Policy (Prevent usage of data from outside the domain)

Open Redirect Exploitation

- Cross Origin Request Sharing (CORS)
- Set Default Homepage
- Disable Directtory Browsing

Other Countermeasures:

- Sanitize Input by Casting
- Secure Static Files

PRODUCTION ORDER

1. Conceptual, Logical, and Physical Entity Relationship Diagrams
2. Flowchart for apate.exe and T-SQL backend
3. T-SQL DB and tables
4. Installer and apate.exe developed as console app for local testing
5. UI Wireframes
6. UI Mockup
7. ASP.NET code developed
8. Functional Prototype developed
9. White/Blackbox UX testing
10. Functional Revisions
11. Final Marketing Established
12. Project Complete

CONCLUSION

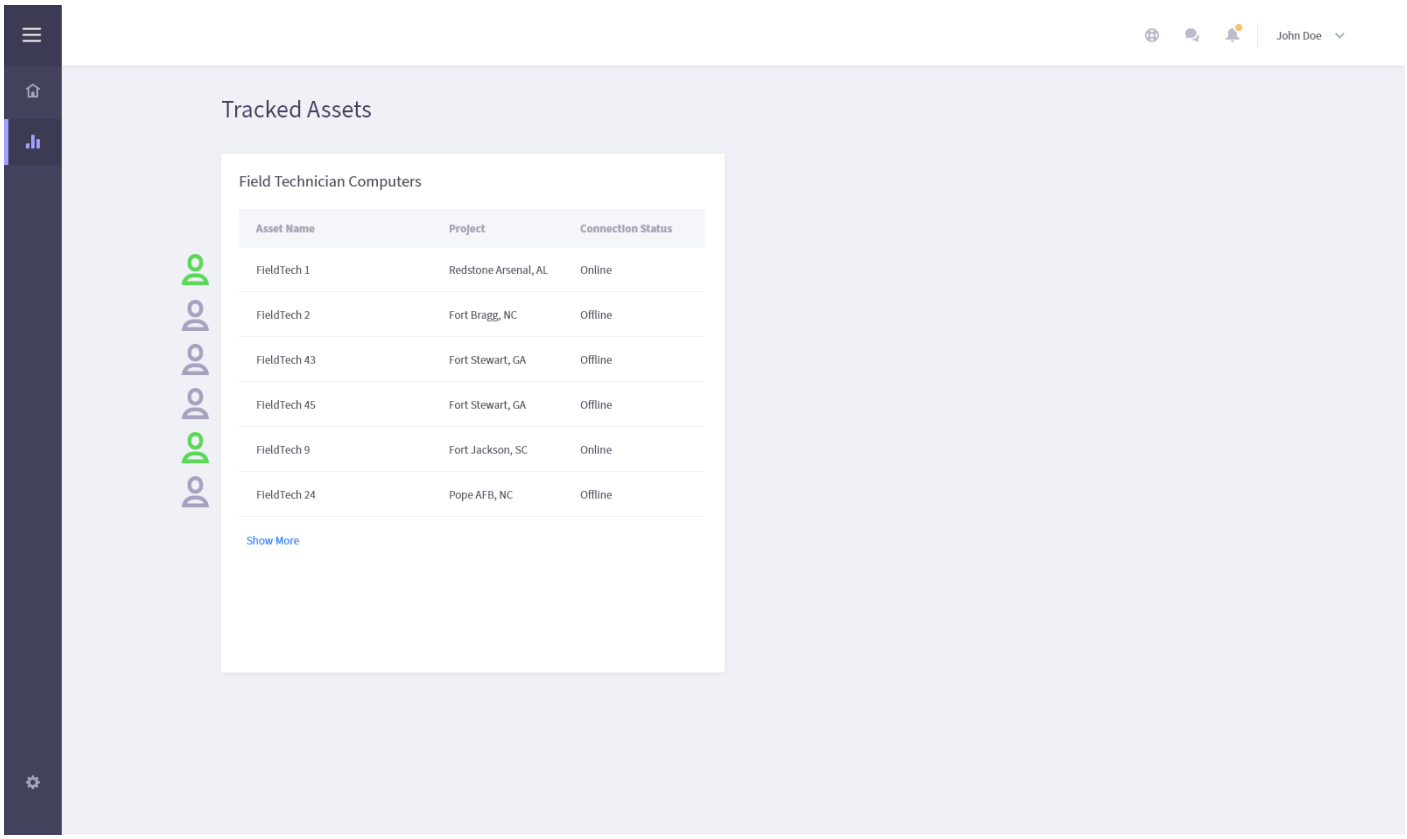
Adversaries have many tricks at their disposal.

They do not follow the same laws and regulations that corporate entities follow.

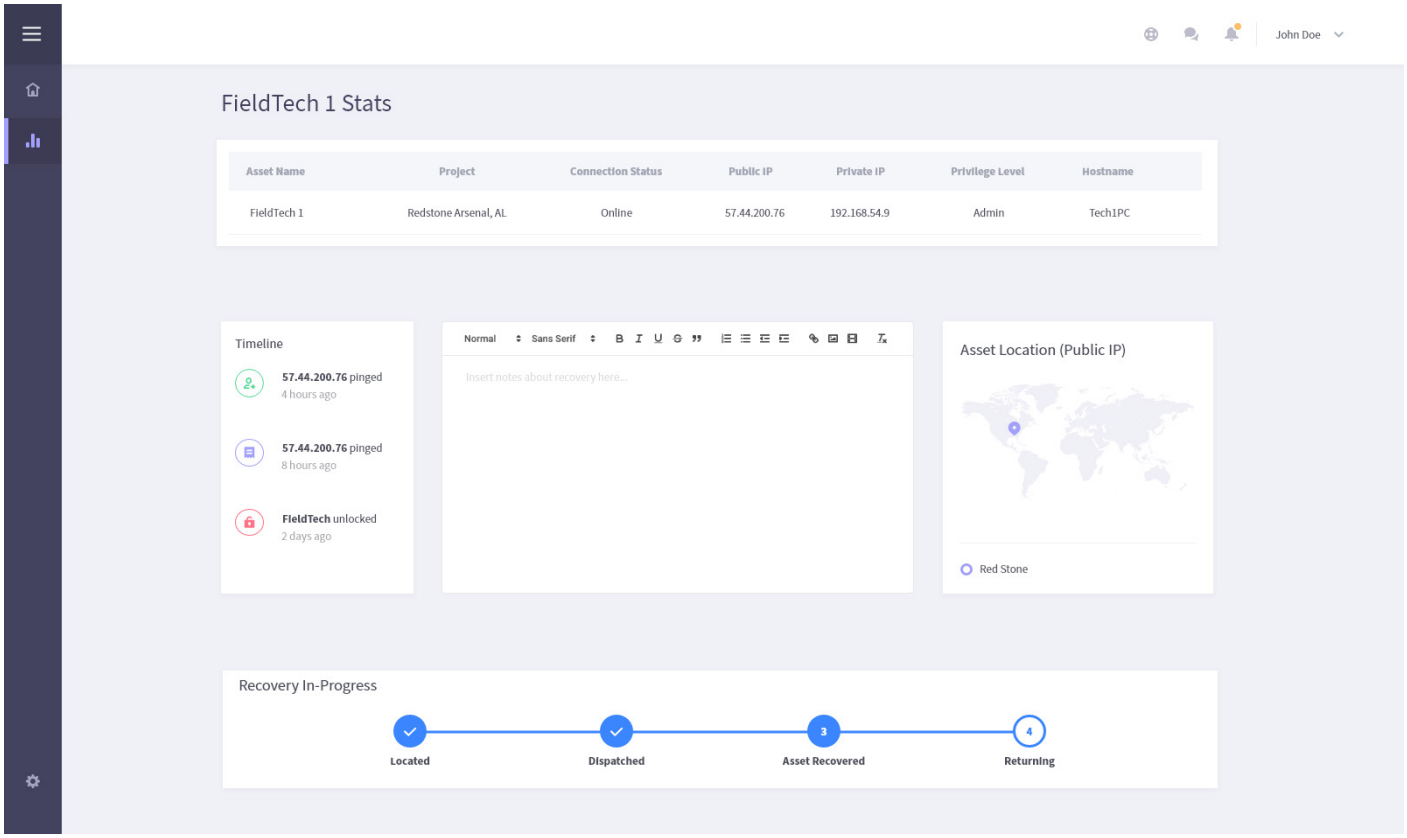
Security personnel have enough to contend with.

Make their lives easier. Secure company assets with **APATE**.

Asset Overview Dashboard Mockup



Individual Asset Dashboard Mockup



Notes:

Lead Developer: Jeff Benson

08.10.2020