Project Plan

Endpoint Data Monitoring and Analysis Agent

The Capstone Experience

Team Rook

Bohao Gao
Andrew Gilbertson
Jeremy Specht
Vikram Thakur
Jared Clark

Department of Computer Science and Engineering
Michigan State University
Spring 2018
Functional Specifications

• Endpoint Agent Log Collection
  ▪ Cross Platform Compatible

• Communication Channels for data
  ▪ Configurable

• Web Application to Analyze Agent
  ▪ Configuration
    o Log paths, storage location
  ▪ Health Analysis
    o Alert priorities
Design Specifications

• Agent
  ▪ Background Process
  ▪ Limited client interaction

• Web Application
  ▪ Extends Current Force Platform
    o Display Log History
    o Current Client Host Health Status
    o Configuration
Screen Mockup: System Health

Specify range of time

Select individual logs

System Health

Log Source Info

Alerts
Screen Mockup: Configuration

Add/Remove Source Paths

Select New Storage Location

Agent Dashboard

Configure Agent

System Health

Agent Version: 1.02.07

Current Paths

Path 1
Path 2
Path 3
Path 4
Path 5
Path 6
Path 7
Path 8

Current Storage Location: Amazon S3 Bucket
Screen Mockup: Storage Location

Change Default Storage Option

Button to bring up window
Technical Specifications

• Storage System
  ▪ AWS S3, Django RESTful endpoint, Elastic Search

• Log Collection Agents using Go
  ▪ Easy Cross Platform Design
  ▪ Local storage capability

• Web Interface
  ▪ ReactJS, Redux, HTML, CSS
  ▪ Backend leverages Django
System Architecture

- Lambda
- S3 API
- REST API
- Elasticsearch
- React JS
- Django
- Agents
- Local Storage
System Components

• Hardware Platforms
  ▪ Ubuntu Django Server
  ▪ AWS S3 Buckets
  ▪ AWS Elastic Search

• Software Platforms / Technologies
  ▪ GoLand IDE
  ▪ Django REST Framework
  ▪ OS Specific Log Collection Interaction
  ▪ AWS API for Go
Risks

• Developing Cross Platform Software
  ▪ Streamlining Log Collection Process for all OS
  ▪ Collect in each OS and build overarching process

• Health Metrics
  ▪ Making sure what is being analyzed is useful
  ▪ Constant communication with Rook

• Integration of Current Platform
  ▪ Cannot compromise integrity of existing platform
  ▪ Iterative Process including Rook Analysts’ Feedback

• Effective Testing
  ▪ Replicate realistic traffic on agent and web application
  ▪ Use mock data provided by rook and bench testing
Questions?