MICHIGAN STATE UNIVERSITY

Project Plan AMAP: Automated Malware Analysis Platform

The Capstone Experience

Team Accenture

Teng Xu, Hefei
Griffin Metevia
Julian Ellis
Andrew Mitchell
Sam Kling

Department of Computer Science and Engineering
Michigan State University

Spring 2018

From Students…
…to Professionals

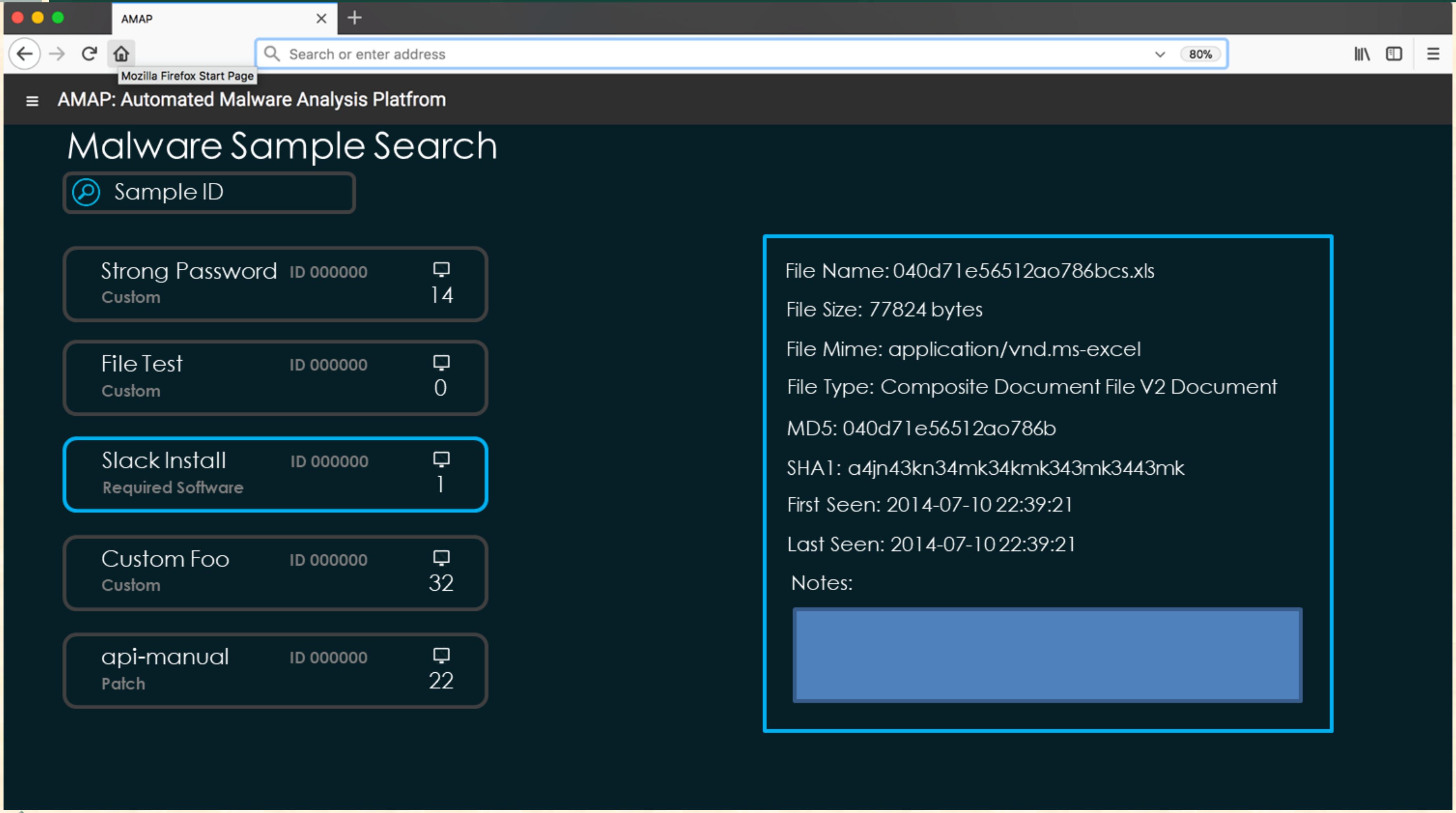
Functional Specifications

- Automate the process of practically analyzing malware samples
- Perform analysis on a large volume of malware samples
- Focus on basic static and basic dynamic analysis
- Record results of analysis and display information to a dashboard

Design Specifications

- Series of modules used to analysis each malware sample
- Status dashboard displays information on the state of the AMAP system
- Malware search page allows users to see specific malware sample information
- Wizard-style UI to add, edit, or remove modules

Screen Mockup: Malware Search



AMAP: Automated Malware Analysis Platfrom

Malware Sample Search

Sample ID

Strong Password	Custom	ID 000000	14
File Test	Custom	ID 000000	0
Slack Install	Required Software	ID 000000	1
Custom Foo	Custom	ID 000000	32
api-manual	Patch	ID 000000	22

File Name: 040d71e56512ao786bcs.xls

File Size: 77824 bytes

File Mime: application/vnd.ms-excel

File Type: Composite Document File V2 Document

MD5: 040d71e56512ao786b

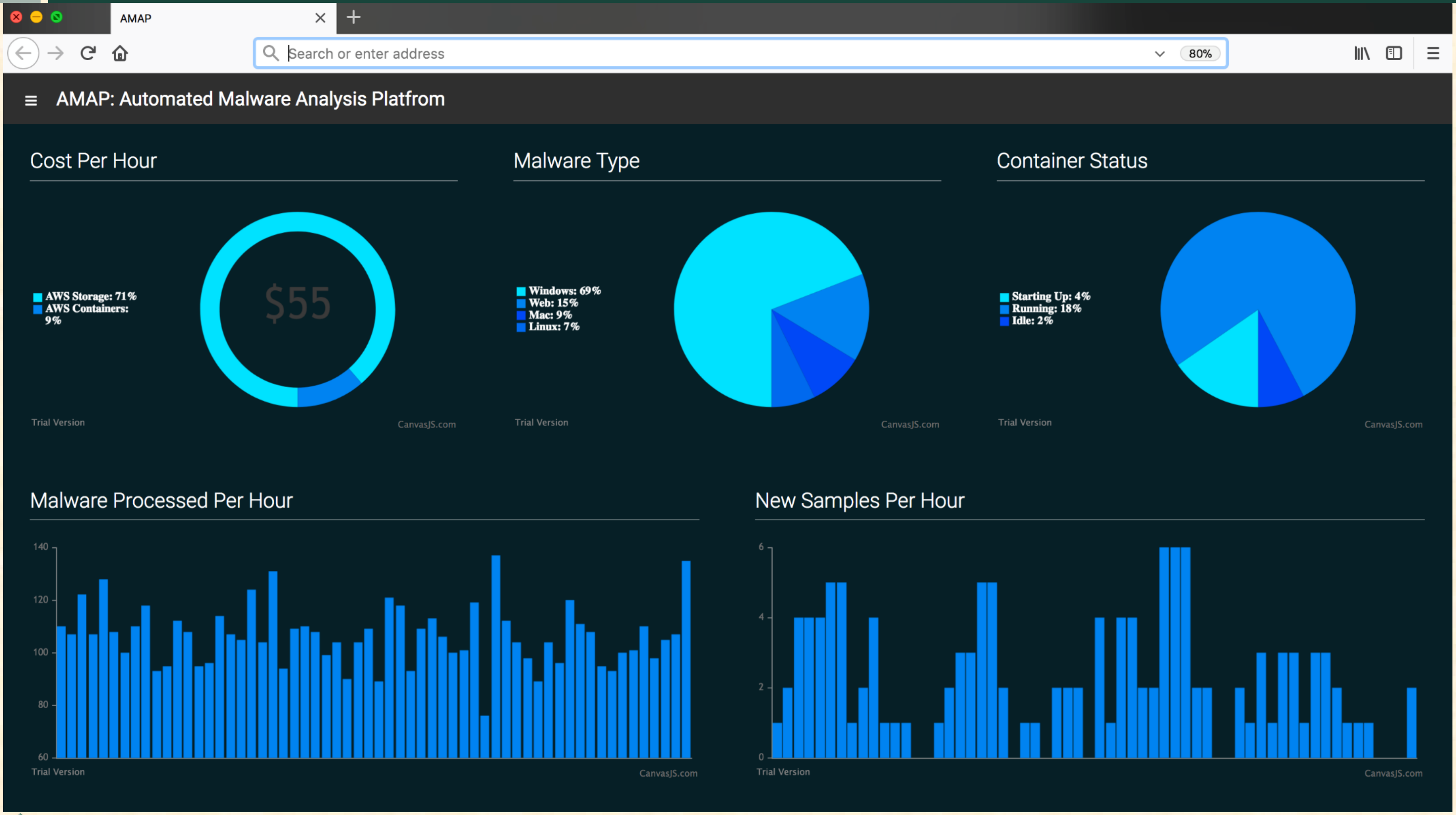
SHA1: a4jn43kn34mk34mk343mk3443mk

First Seen: 2014-07-10 22:39:21

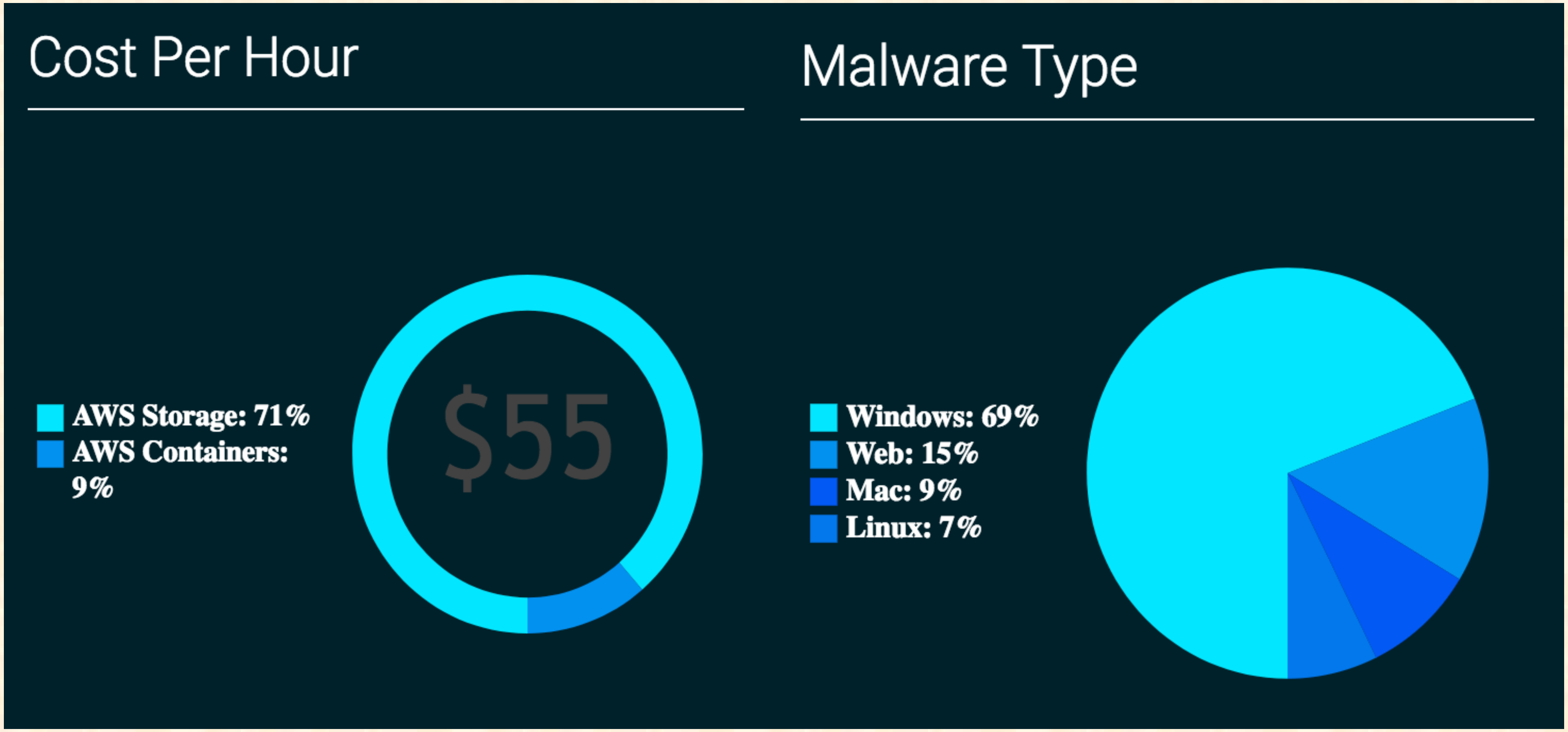
Last Seen: 2014-07-10 22:39:21

Notes:

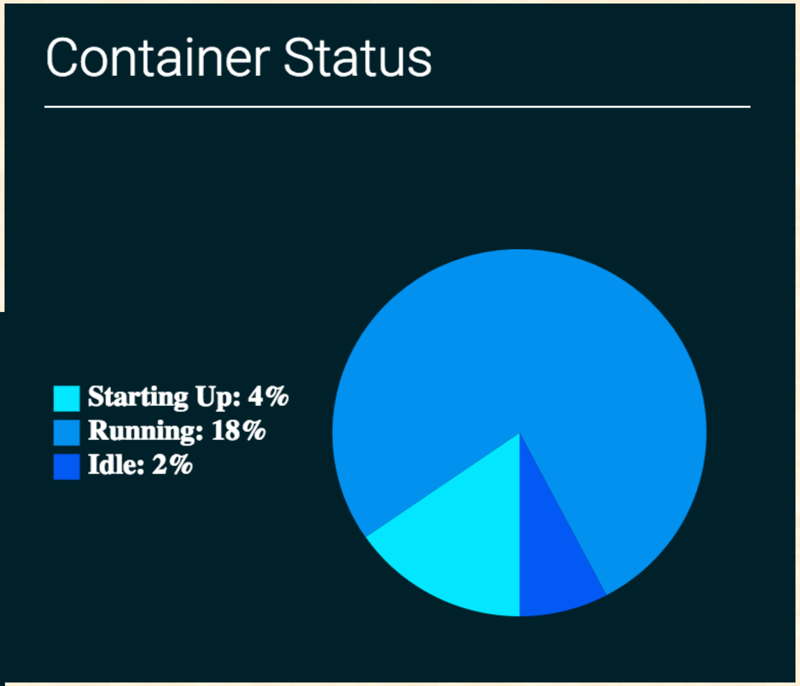
Screen Mockup: AMAP Dashboard

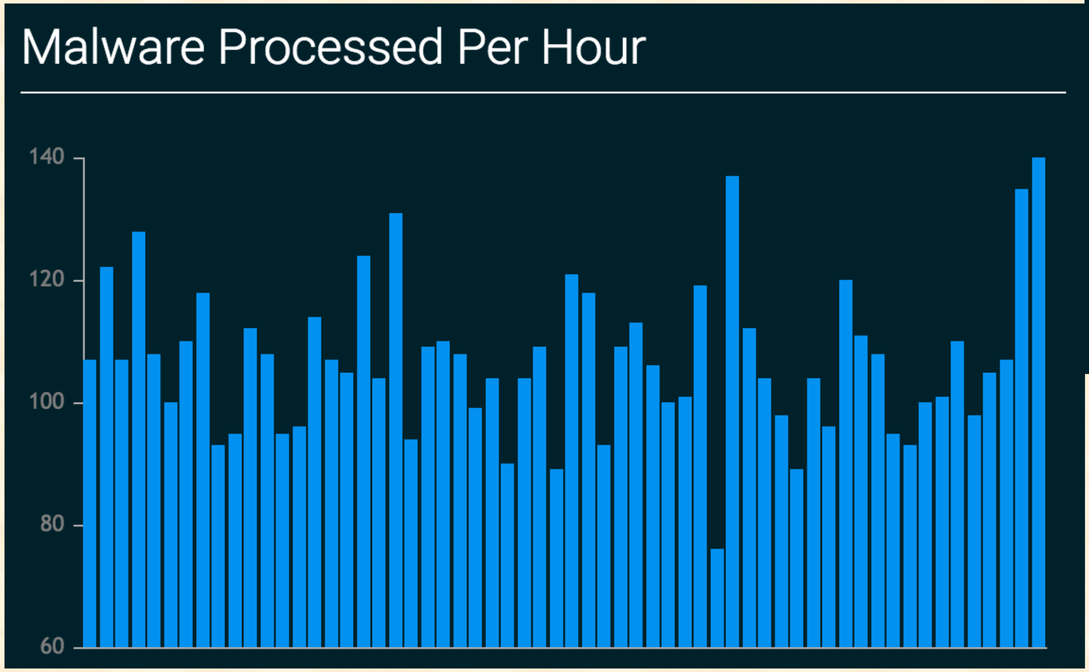


Screen Mockup: AMAP Dashboard



Screen Mockup: AMAP Dashboard

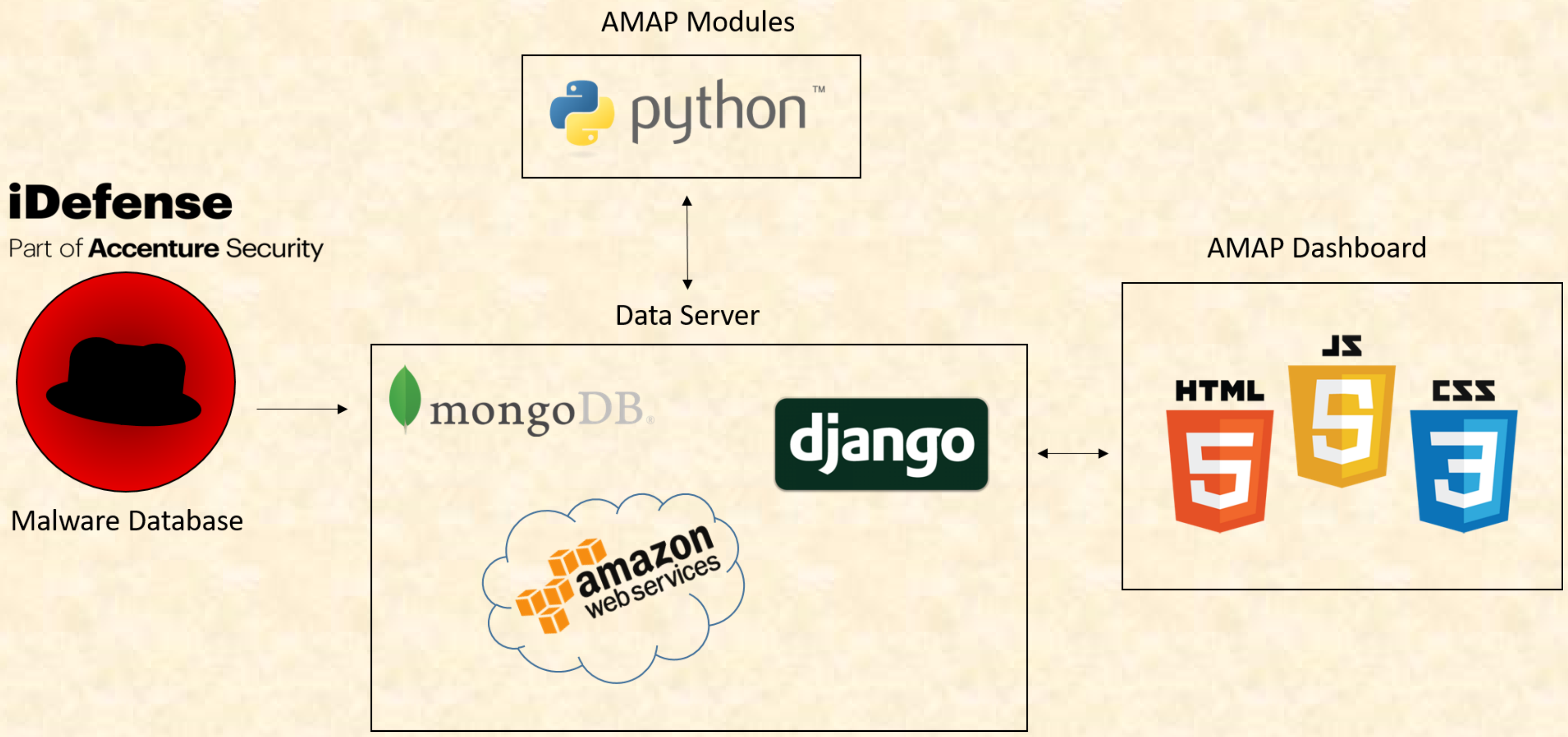




Technical Specifications

- Basic static analysis
  - Provides information about functionality
  - Produce simple network signatures
  - Ineffective against sophisticated malware
- Basic dynamic analysis
  - Observe malware behavior after executing on system such as encrypting files or changing file names
  - Takes place in a controlled environment such as VM or sandbox

System Architecture



System Components

- Software Platforms / Technologies
  - iDefense IntelGraph API
  - iDefense Malware Repository
  - mongoDB hosted on AWS
  - Python/Django -- PyCharm
  - HTML, CSS, JavaScript

Risks

- Processing a large quantity of samples
  - System needs to handle an average of 300 thousand per day
  - Using multithreading to allow many modules to be run concurrently
- Categorizing malware based on type
  - Malware must be classified based on detection signatures, byte patterns, and other information
  - Undergoing training from the client to learn how to categorize malware based on these criteria
- Getting information from dynamic analysis
  - Malware samples are executed in a VM or sandbox environment and information about their effects must be recorded
  - The client has extensive knowledge about how to perform this method of malware analysis
- Determining when a sample is finished processing
  - Malware analysis can sometimes produce as a result encoded payloads that require further analysis
  - Client can provide information about when this situation occurs and small scale testing can be used to determine what kinds of samples might cause this

Questions?

? ? ? ?

? ?

? ? ?