

MICHIGAN STATE

U N I V E R S I T Y

Project Plan

Improved Detonation of Evasive Malware

The Capstone Experience

Team Proofpoint

Ian Murray

Ryan Gallant

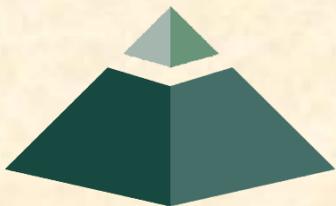
Jack Mansueti

Sean Joseph

Tae Park

Department of Computer Science and Engineering
Michigan State University

Fall 2018



*From Students...
...to Professionals*

Functional Specifications

- Sandbox is essential for malware analysis
- New evasive techniques hinder quarantine
- **Fundamental Solution:** Flag malware whose execution deviates in sandboxes.
- **Auxiliary Solution:** Support autonomous code modification to remove the ability to avoid sandbox execution
- Display in intuitive web UI

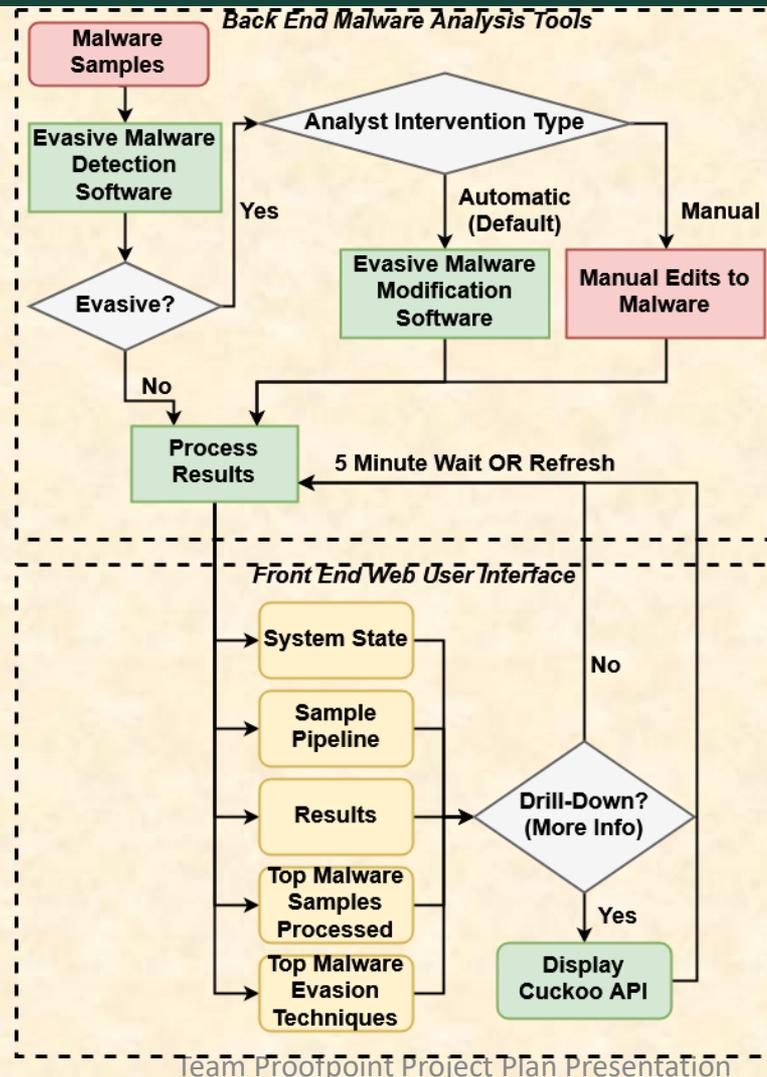


Design Specifications

- Evasive Malware Identification
 - Scan for known existing signatures
 - Develop own behavior detection methods
- Malware Modification & Detonation
 - Modify sandbox checks with reverse engineering
 - Forces malware to execute all relevant functions
- Web Interface
 - Top-Level: Displays broad real time data
 - Drill-Downs: Widgets, enters more detailed reports



Design Specifications



Screen Mockup: Top Samples

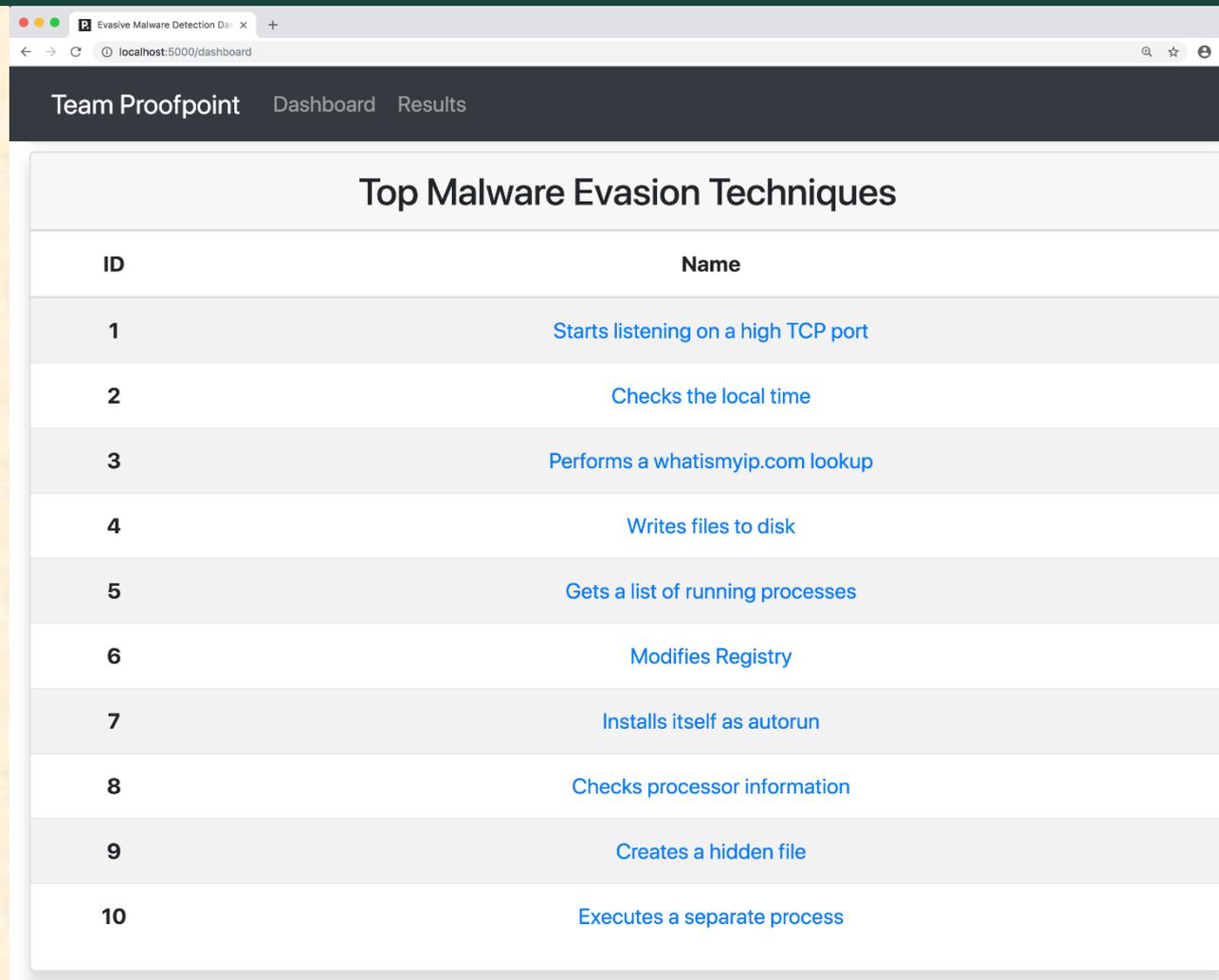
The screenshot shows a web browser window with the following details:

- Browser tabs: Evasive Malware Detection Dashboard
- Address bar: localhost:5000/dashboard
- Navigation: Team Proofpoint | Dashboard | Results
- Section Header: Top Malware Samples Processed
- Table Data:

ID	Name
1	Wannacry
2	Yahoo! toolbar
3	Hotmail toolbar
4	Harddrive encryption
3	Performs a whatismyip.com lookup



Screen Mockup: Top Techniques

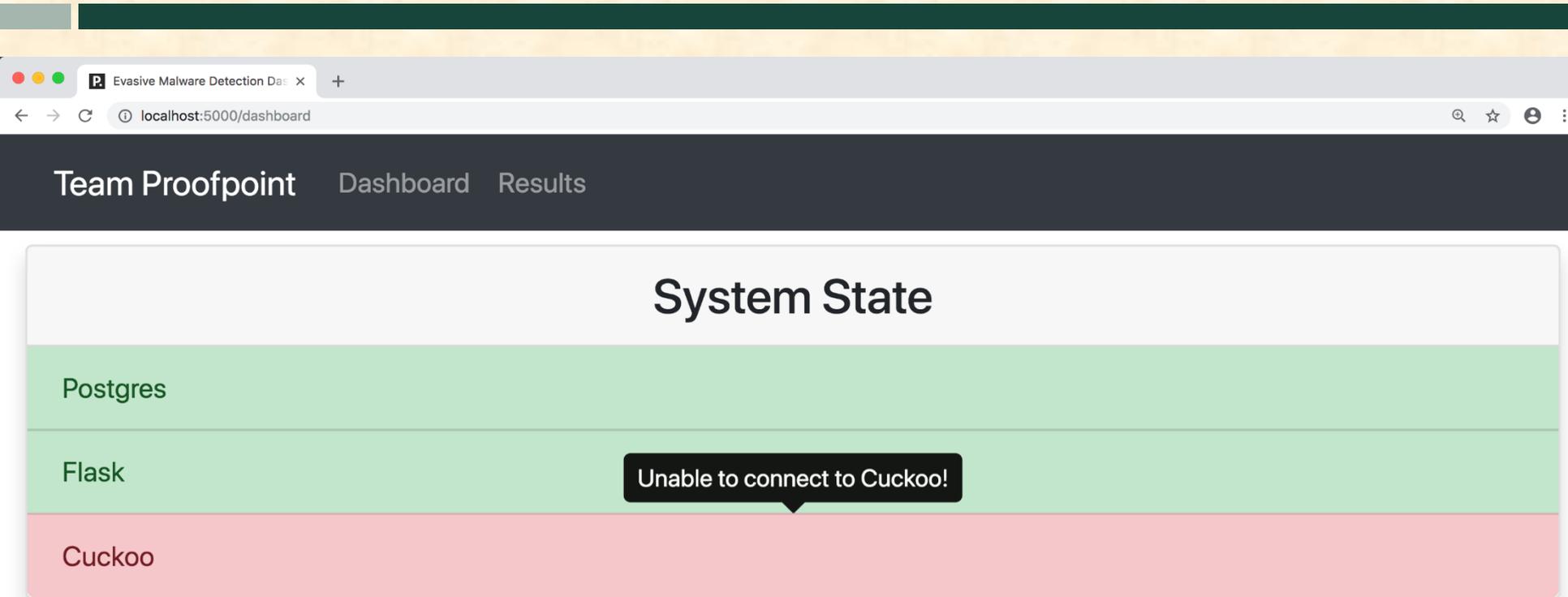


The screenshot shows a web browser window with the address bar at localhost:5000/dashboard. The page has a dark header with 'Team Proofpoint', 'Dashboard', and 'Results' links. The main content area features a table with the title 'Top Malware Evasion Techniques'. The table has two columns: 'ID' and 'Name'. It lists 10 techniques, each with a unique ID and a description of the evasion method.

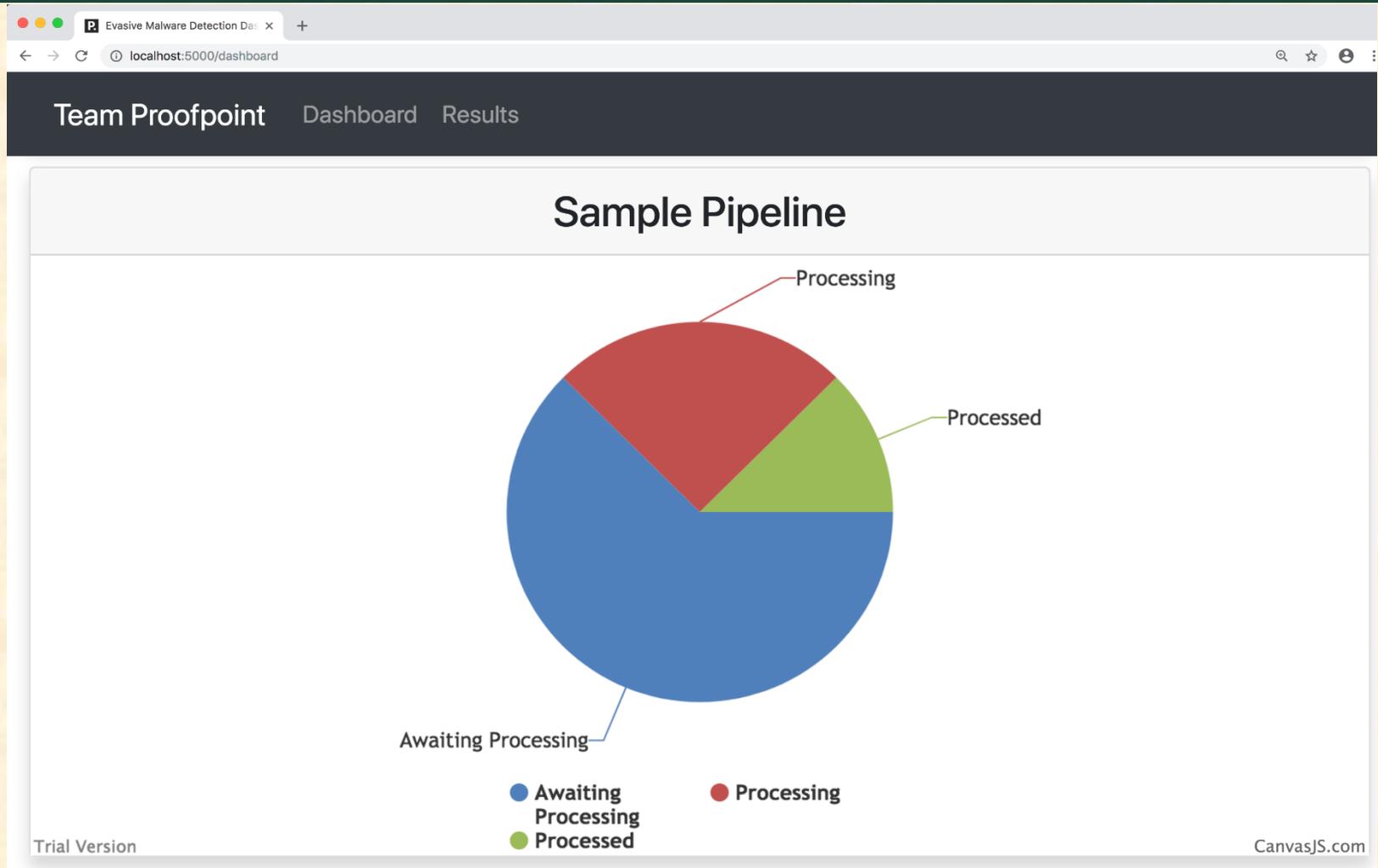
ID	Name
1	Starts listening on a high TCP port
2	Checks the local time
3	Performs a whatismyip.com lookup
4	Writes files to disk
5	Gets a list of running processes
6	Modifies Registry
7	Installs itself as autorun
8	Checks processor information
9	Creates a hidden file
10	Executes a separate process



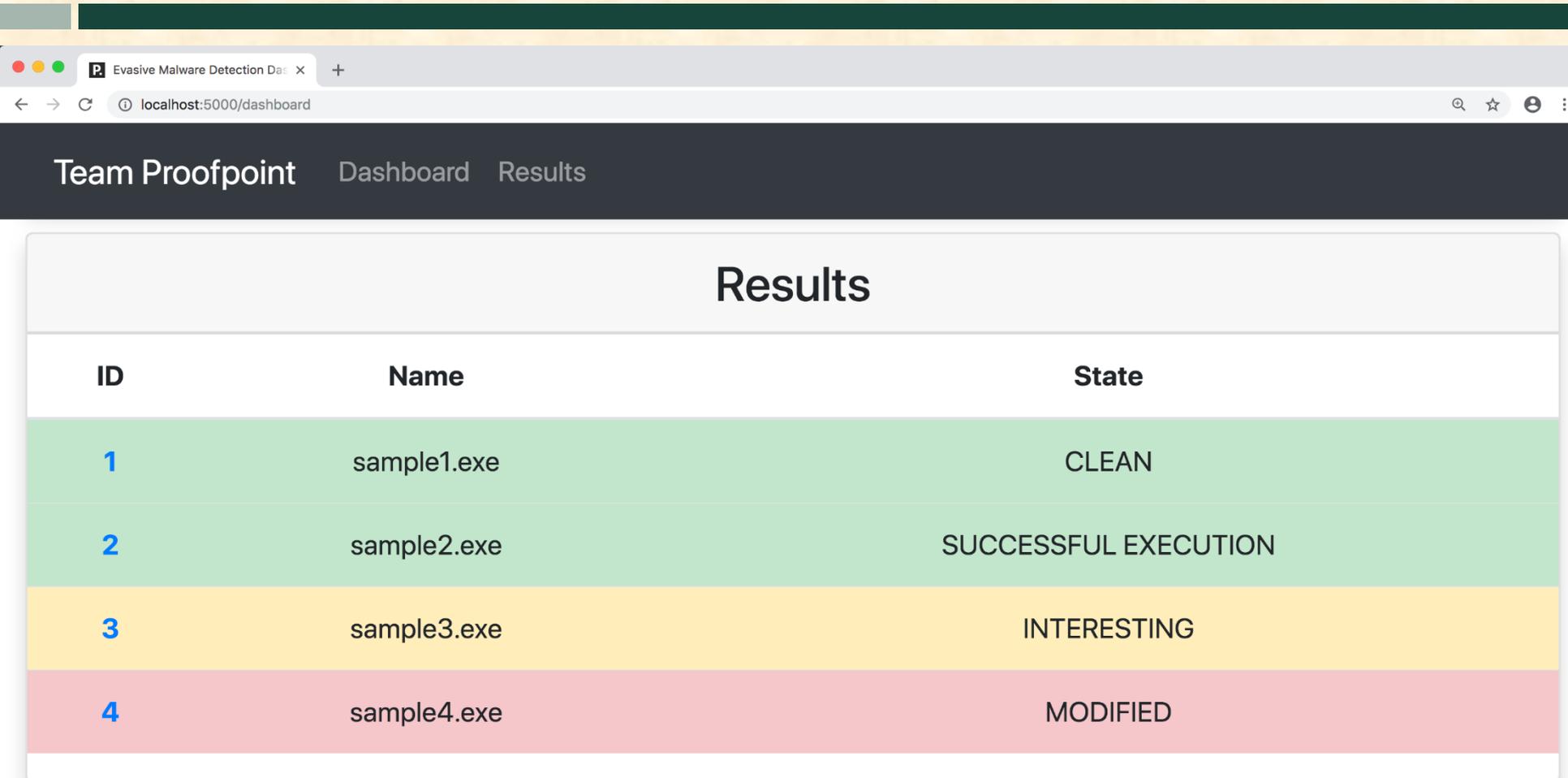
Screen Mockup: System State



Screen Mockup: Sample Queue



Screen Mockup: Results

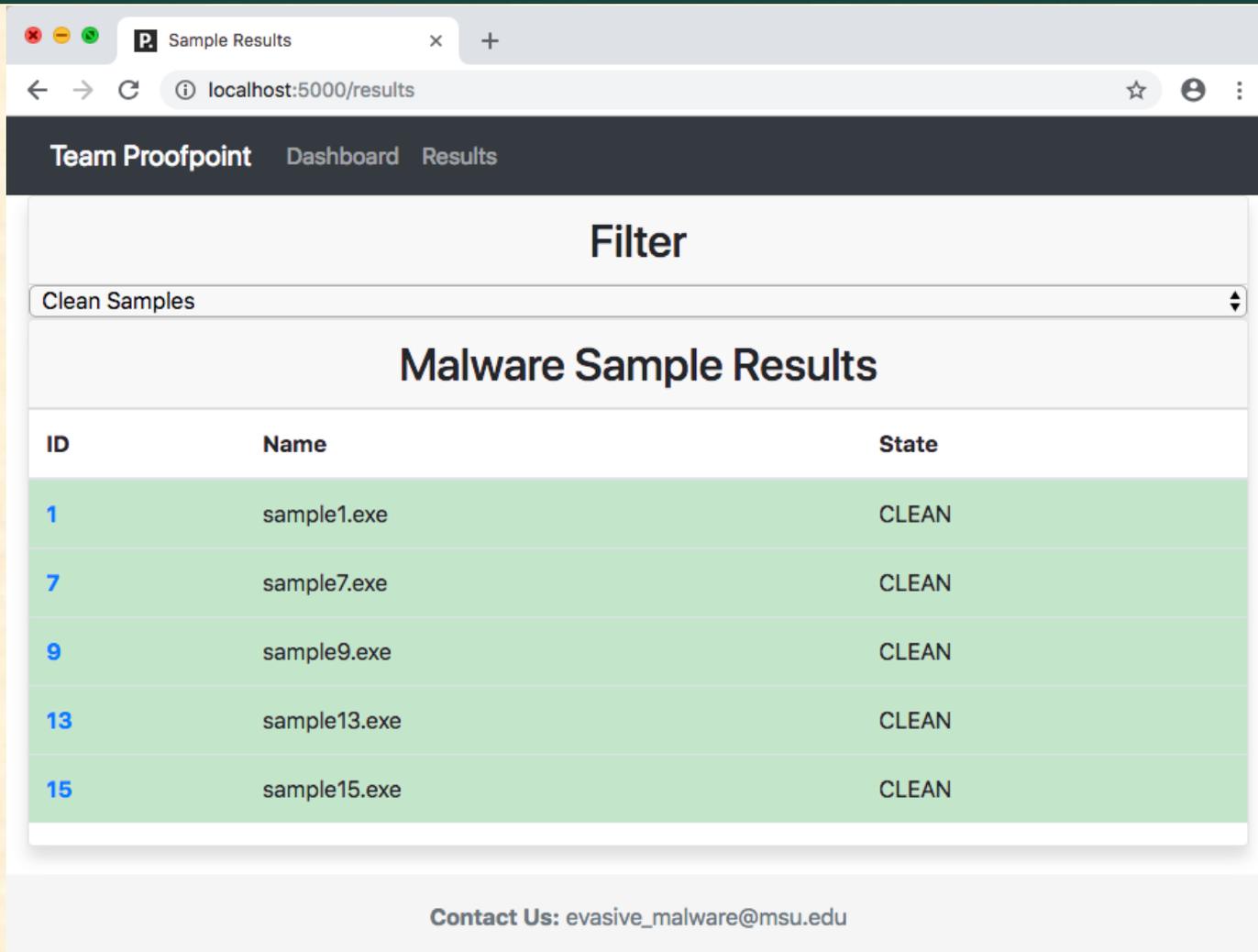


The screenshot shows a web browser window with the address bar displaying 'localhost:5000/dashboard'. The page title is 'Evasive Malware Detection Dashboard'. The navigation bar includes 'Team Proofpoint', 'Dashboard', and 'Results'. The main content area is titled 'Results' and contains a table with the following data:

ID	Name	State
1	sample1.exe	CLEAN
2	sample2.exe	SUCCESSFUL EXECUTION
3	sample3.exe	INTERESTING
4	sample4.exe	MODIFIED



Screen Mockup: Results w/ Filter



The screenshot shows a web browser window with the address bar at localhost:5000/results. The page has a dark navigation bar with 'Team Proofpoint', 'Dashboard', and 'Results'. Below this is a 'Filter' section with a dropdown menu set to 'Clean Samples'. The main content area is titled 'Malware Sample Results' and contains a table with three columns: ID, Name, and State. The table lists five samples, all with a state of 'CLEAN'. The footer contains the contact information 'Contact Us: evasive_malware@msu.edu'.

ID	Name	State
1	sample1.exe	CLEAN
7	sample7.exe	CLEAN
9	sample9.exe	CLEAN
13	sample13.exe	CLEAN
15	sample15.exe	CLEAN

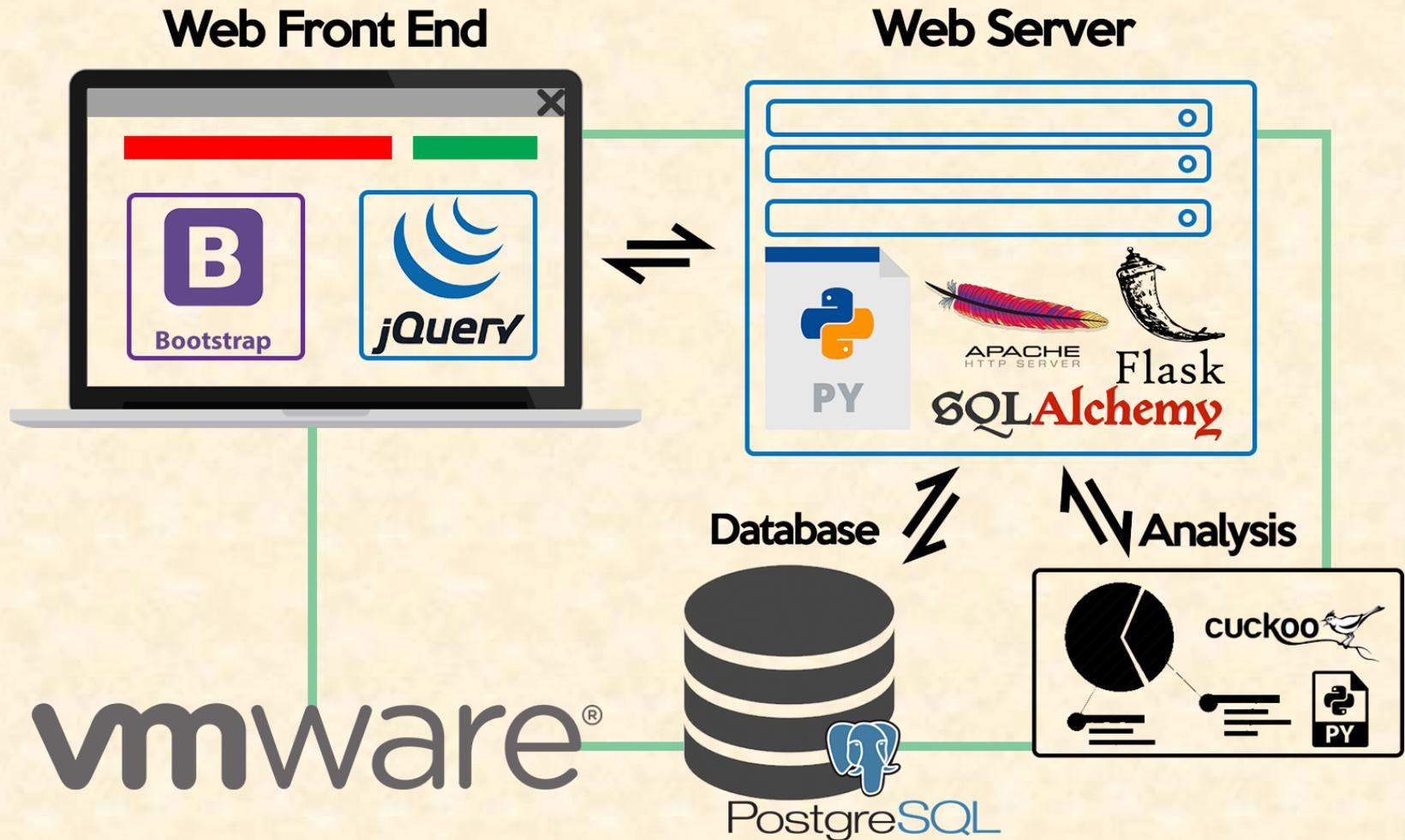


Technical Specifications

- Front End UI
 - Bootstrap, jQuery, HTML5, and CSS3 are used to effectively present users with appropriate data from the malware detonation system.
- Web Application
 - Apache, Flask, and Python are used to serve our web application.
 - PostgreSQL is used for data storage outside the data Cuckoo's API provides.
 - SQLAlchemy is used for mapping Python Objects to PostgreSQL statements and schema.
- Backend Malware Analysis
 - Cuckoo and Suricata are used for detonation and classification, Python is used to disassemble and modify malware samples classified as evasive.



System Architecture



System Components

- Software Platforms / Technologies
 - Front End
 - Python 3.6
 - HTML & CSS3
 - Bootstrap CSS
 - Cuckoo API
 - Flask
 - jQuery
 - Back End
 - Python 2.7
 - Cuckoo
 - Suricata
 - PostgreSQL
 - SQLAlchemy
 - Apache
 - VMWare



Risks

- Reverse Engineering Difficulty
 - Malware samples are rarely available as readable code.
 - Variety of tools for disassembly.
- Multiple Language Proficiency
 - Malware comes in variety of languages.
 - Limit analysis to a subset of the greater universe of languages.
- Navigating Proofpoint's Lab
 - Unknown how customizable Proofpoint's lab environment is.
 - Client runs samples the team uploads via Secureshare.
- Malware Samples Evade through Unknown Means
 - Unknown how a sample determines the difference between a live machine and a sandbox.
 - Proofpoint has identified several evasive malware for the team to examine.



Questions?

?

?

?

?

?

?

?

?

?

