

MICHIGAN STATE
UNIVERSITY

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

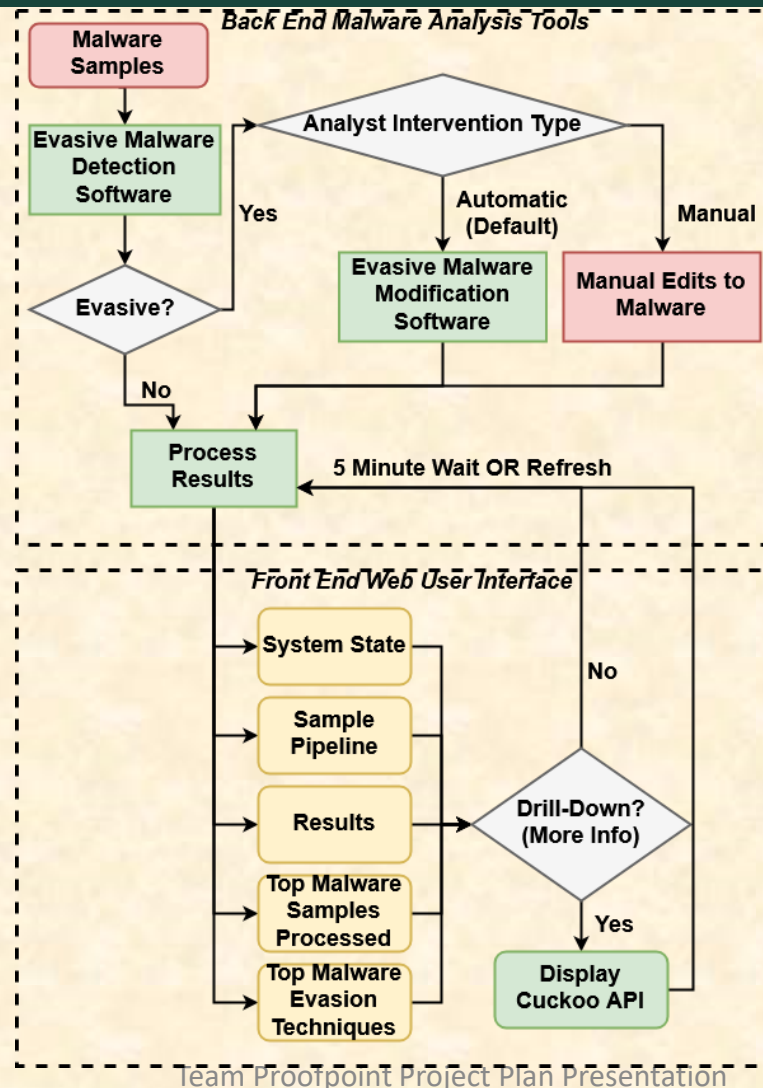


Design Specifications

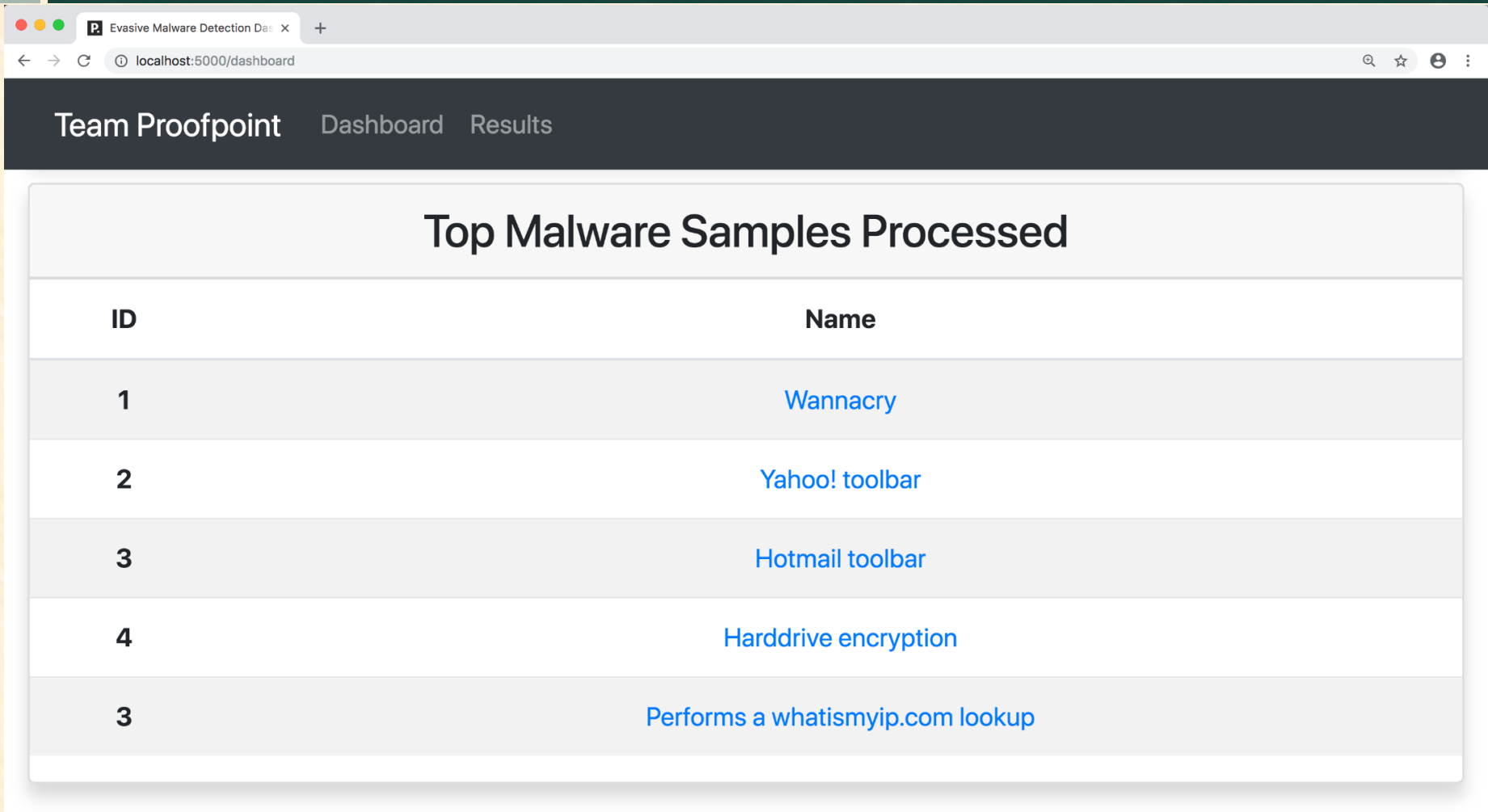
- Evasive Malware Identification
 - Scan for known existing signatures
 - Develop own behavior detection methods
- Malware Modification & Detonation
 - Modify sandbox checks with reverse engineering
 - Forces malware to execute all relevant functions
- Web Interface
 - Top-Level: Displays broad real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



Design Specifications



Screen Mockup: Top Samples

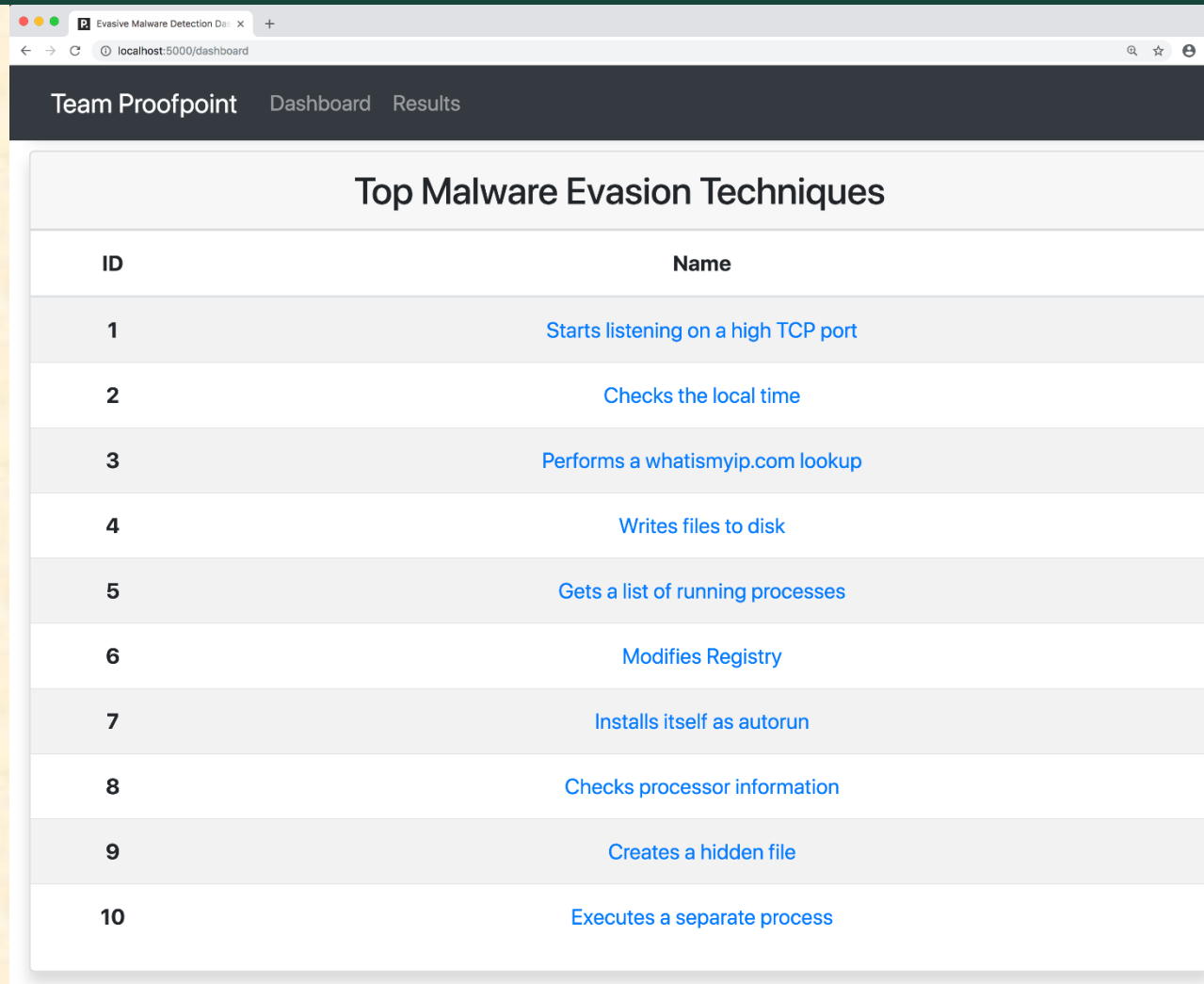


The screenshot shows a web browser window with the address bar at `localhost:5000/dashboard`. The page has a dark header with the text 'Team Proofpoint' and navigation links 'Dashboard' and 'Results'. The main content area is titled 'Top Malware Samples Processed' and contains a table with two columns: 'ID' and 'Name'. The table lists five samples, with the last one having an ID of 3.

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



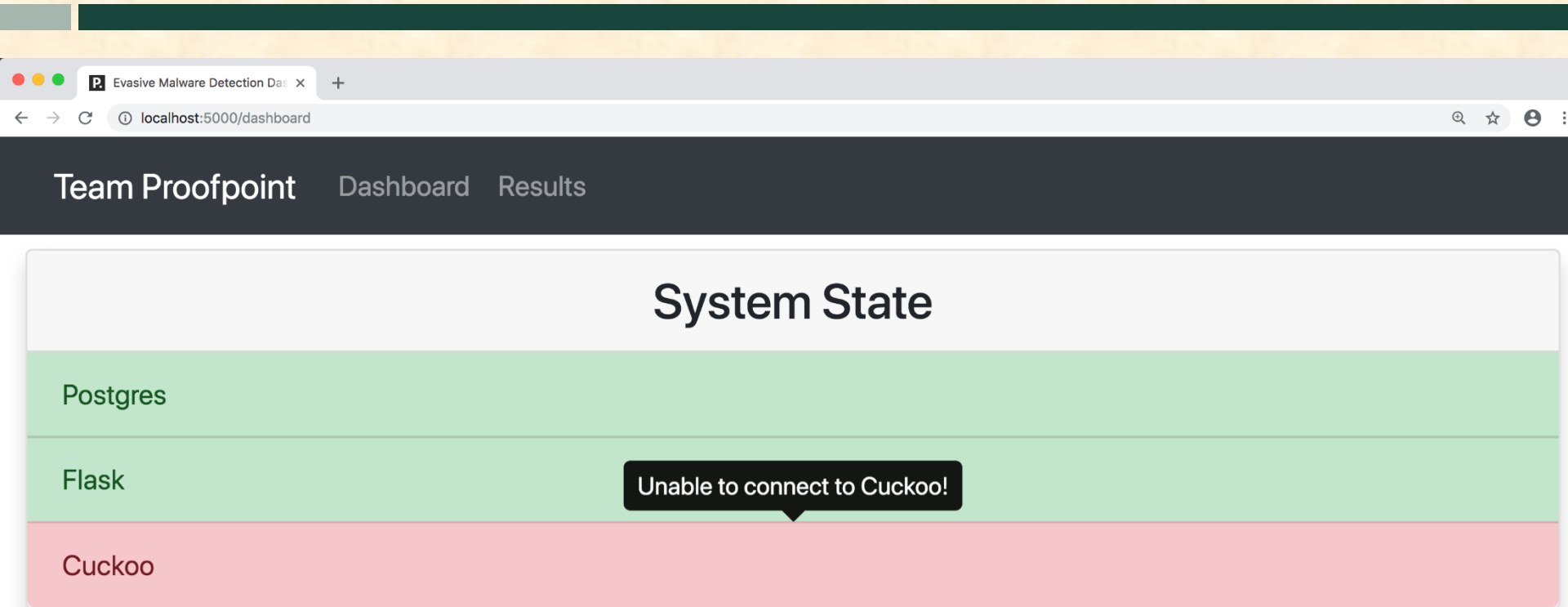
Screen Mockup: Top Techniques



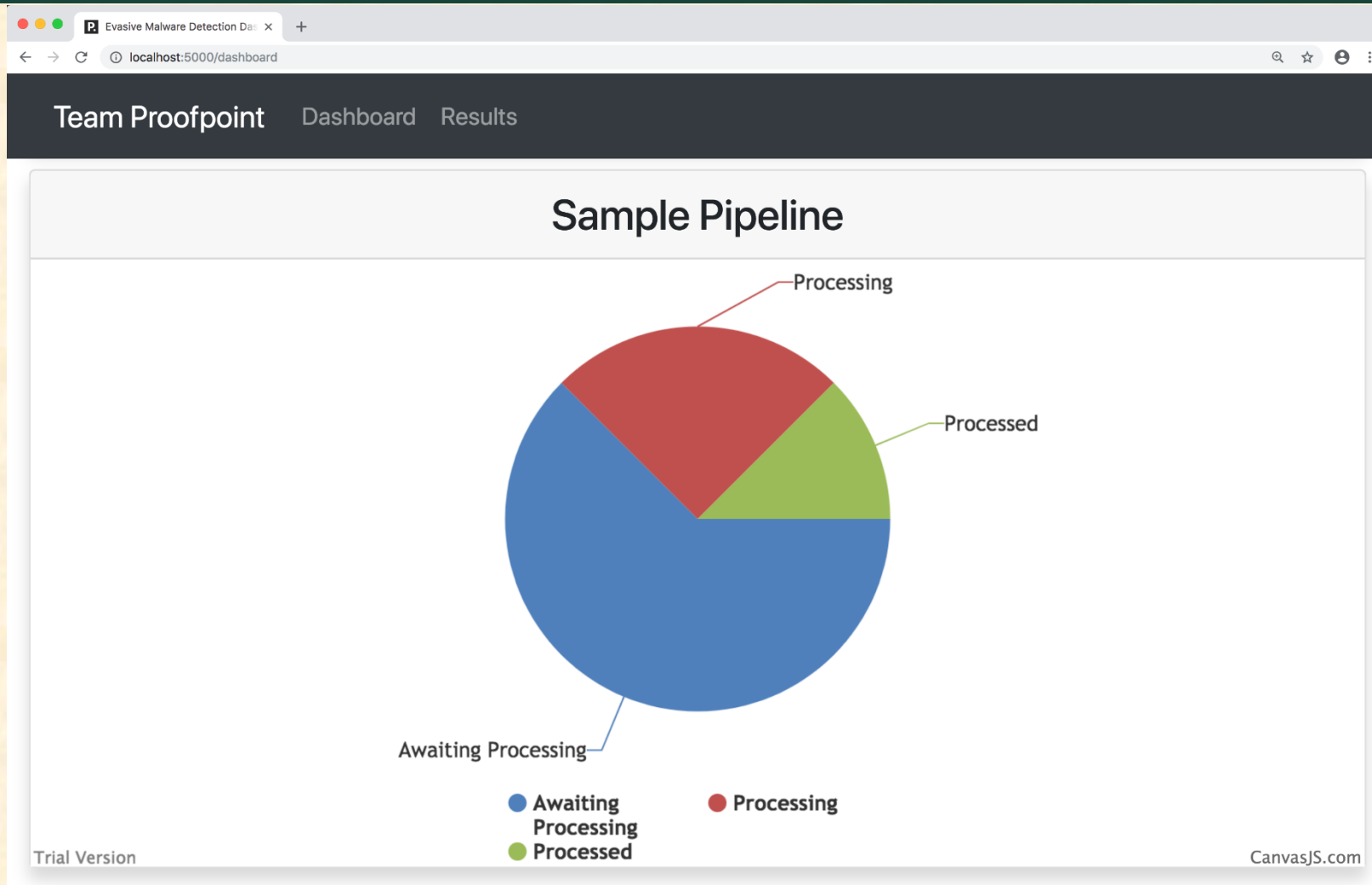
Top Malware Evasion Techniques	
ID	Name
1	Starts listening on a high TCP port
2	Checks the local time
3	Performs a whatismyip.com lookup
4	Writes files to disk
5	Gets a list of running processes
6	Modifies Registry
7	Installs itself as autorun
8	Checks processor information
9	Creates a hidden file
10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



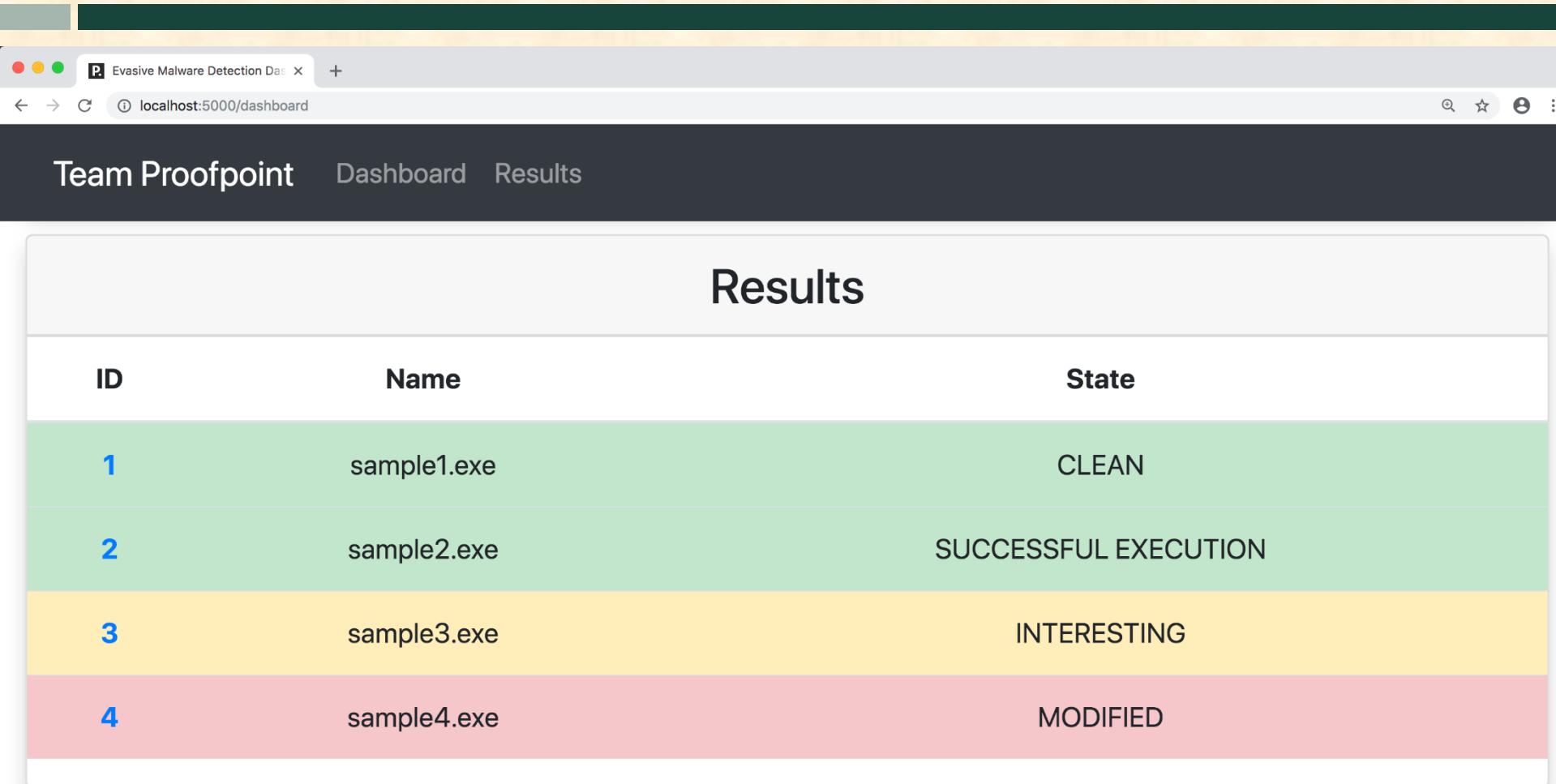
Screen Mockup: System State



Screen Mockup: Sample Queue



Screen Mockup: Results

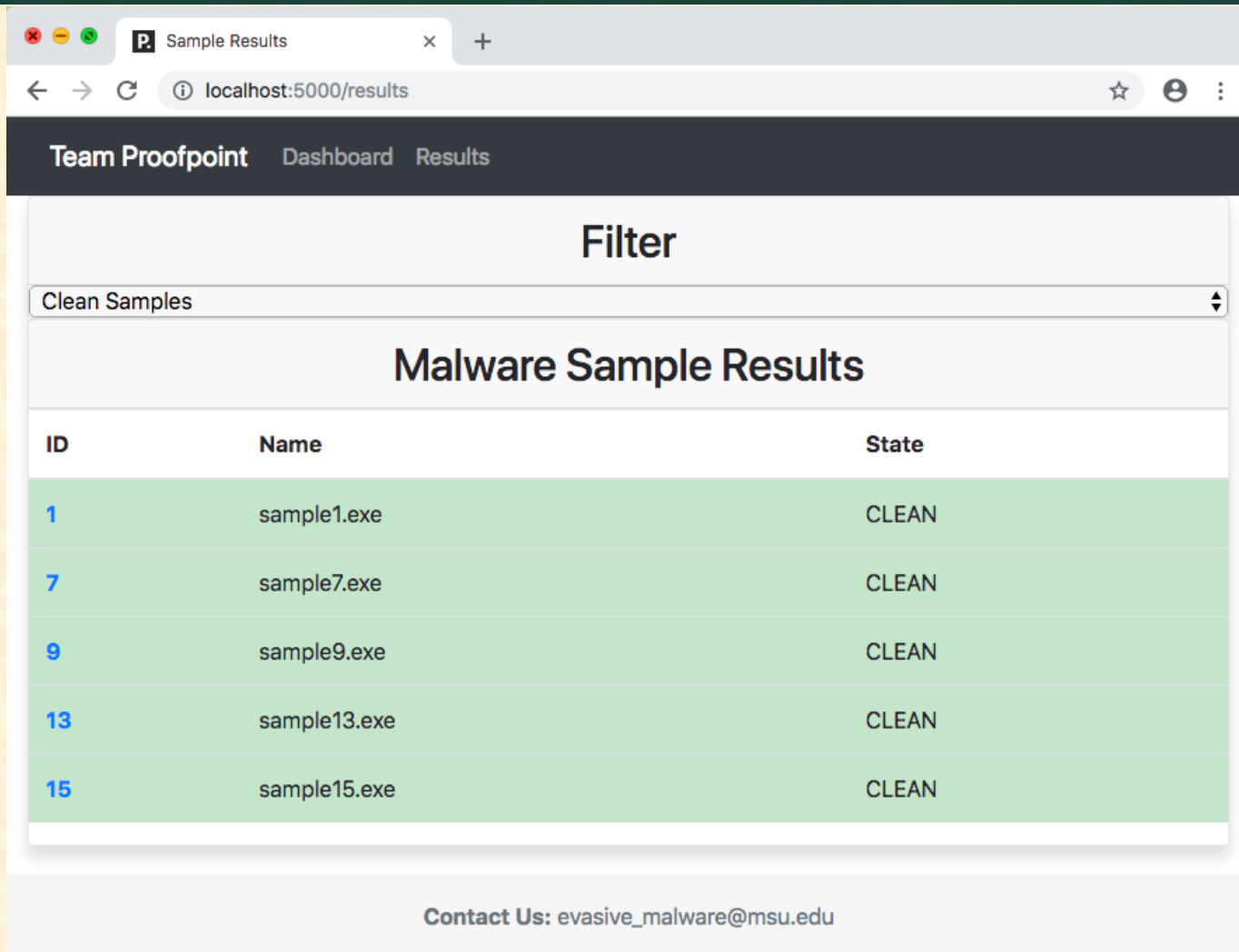


The screenshot shows a web browser window with the address bar displaying 'localhost:5000/dashboard'. The page title is 'Evasive Malware Detection Dashboard'. The navigation bar includes 'Team Proofpoint', 'Dashboard', and 'Results'. The main content area is titled 'Results' and contains a table with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

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 title 'Sample Results'. The address bar displays 'localhost:5000/results'. The navigation bar includes 'Team Proofpoint', 'Dashboard', and 'Results'. A 'Filter' dropdown menu is set to 'Clean Samples'. Below this, the section is titled 'Malware Sample Results'. A table lists five samples, all with a 'CLEAN' state. The table has columns for ID, Name, and State. 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

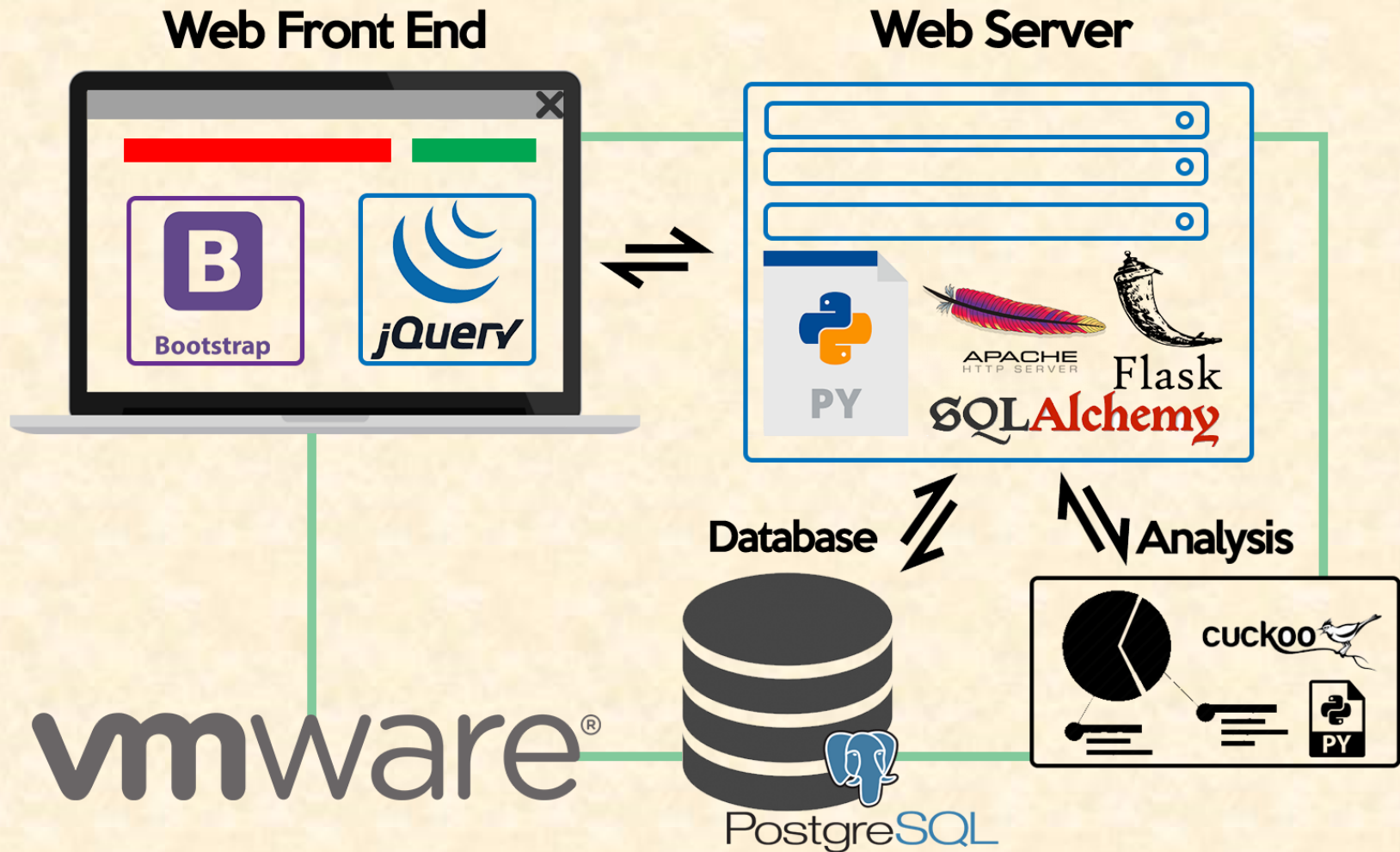


Technical Specifications

- 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?

?

?

?

?

?

?

?

?

?

