ReapRE

by

Claire Church, Michael Ferguson, Matthew Granitto

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Claire Church	**4/13/20**
Claire Church	Date
Michael Ferguson	**4/13/20**
Michael Ferguson	Date
Matthew Granitto	**4/13/20**
Matthew Granitto	Date
Bogdan Vykhovanyuk	**4/13/20**
Bogdan Vykhovanyuk, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2020

ReapRE

Prepared by
Claire Church, Michael Ferguson, Matthew Granitto

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services
School of Information Technology

April 13, 2020

TABLE OF CONTENTS

List of Illustrations ii
TABLES ii
FIGURES ii

ACRONYMS AND ABBREVIATIONS iii

ABSTRACT 1

PROBLEM STATEMENT 2
Introduction 2
Problem 2
Solution 2
Project Description 3
User Profile 5
Use Case Diagram 7
Testing 7
Test Results 10

PROJECT MANAGEMENT 12
Budget 12
Objectives/Deliverables 14
Project Schedule 14

TECHNICAL ELEMENTS 18
Network 18
Application 18
Database 19

USER INTERFACE 20

PROBLEMS ENCOUNTERED 22

FUTURE RECOMMENDATIONS 22
CONCLUSION 24

APPENDIX A. ReapRE Poster 25
APPENDIX B. REFERENCES 26

List of Illustrations

TABLES

No.		Page
Table 1.	Schedule of User Testing	10
Table 2.	Schedule of Team Member Testing	10
Table 3.	User Test Results	11
Table 4.	Team Member Test Results	12
Table 5.	Initial Budget	13
Table 6.	Final Budget	13
Table 7.	Work Breakdown Structure	15

FIGURES

No.		Page
Figure 1.	User Profile	5
Figure 2.	Use Case Diagram	7
Figure 3.	Fall Semester 2019 Gantt Chart	14
Figure 4.	Spring Semester 2020 Gantt Chart	14
Figure 5.	Application Architecture	19
Figure 6.	Home Page Screenshot	21
Figure 7.	Results Page Screenshot	21
Figure 8.	Report Screenshot	21

ACRONYMS AND ABBREVIATIONS

IOCs	Indicators of Compromise
MISP	Malware Information Sharing Platform
API	Application Programming Interface
AWS	Amazon Web Service
IT	Information Technology
A/V	Anti-Virus
PII	Personally Identifiable Information
URL	Uniform Resource Locator
GUI	Graphical User Interface

ABSTRACT

IT security teams at small companies are often understaffed and overworked, according to a survey published by Sophos (June 2019). ReapRE is an automated malware analysis tool designed to assist analysts at small businesses. Unlike other costly solutions, the analyst will not have to worry about potential confidential data leakage resulting from a 3rd party application breach, as everything will be hosted internally. A potentially malicious document will be uploaded by an analyst into a locally hosted instance of our application, where it will then be run against a stack of malware analysis tools. A summarized report will be given to the analyst based on its findings and a list of indicators of compromise (hashes, IPs, URLs, domains, etc.). After review, the analyst is able to import the IOCs into a local database, Malware Information Sharing Platform (MISP).

PROBLEM STATEMENT

Introduction

Problem

As companies and technology continues to expand and cybersecurity become more complex, IT security teams in small businesses are often overworked and understaffed, as indicated by a Sophos survey taken by more than 3000 cyber security managers. 86% of respondents said that they need greater skills in their organization. 50% of respondents claimed that phishing emails are their main security risk. According to a report by CriticalStart, 70% of analysts investigate more than 10 incidents every day and 35% investigate more than 20. 78% of those surveyed stated they spend over 10 minutes on each incident. Most small businesses simply do not have the resources to build and train a large team of analysts, seeing as the average salary for a single analyst is $71,000+ (According to PayScale).

IT security teams simply do not have the resources available for detailed document analysis. They will rely on A/V solutions and block any malicious indicators of compromise, IOCs. Many IT teams will utilize open source sandboxes, which poses a threat to PII exposure.

Solution

ReapRE will help security analysts at small businesses automate the analysis of potentially malicious documents. Users can upload files into our application, where it will be run against various malware analysis tools and modules to scan and analyze

documents, outputting a report based on its findings. The tools will include the LaikaBoss object scanner, OLETOOLS, Yara rule sets, and more. Based on this report, the user will then be presented with a list of IOCs, including hashes, domains, IPs, URLs, etc. which will be put into a threat intelligence database (MISP) where they can feed them into firewall blacklists and other security devices. Analysts do not have to worry about data leakage, as ReapRE will be hosted locally. This will help smaller businesses to be able to better manage their security and will allow analysts to effectively automate and assess malicious threats and security risks.

There are similar products and services that assist in the analysis process, however none that meet our requirements and scope. VirusTotal Enterprise starts at $10,000 per year, but the price drastically increases based on usage. Additionally, you upload the files into their application, where they can be accessed by anyone. Anti-virus solutions range from free (Windows Defender) to approximately $70 per device (Malwarebytes). Anti-virus solutions are not adequate, as they scan documents against known malicious patterns which are simple to bypass by simply modifying portions of the malicious payload.

Project Concept / Solution

To put it simply, ReapRE is to assist cybersecurity analysts in small businesses and smaller IT teams in automating the process of analyzing malicious documents. With our project, we wanted to not only help with the analysis process but also give teams the resources to a less expensive application, as well as a way to keep sensitive information private.

Matthew and Michael both work at a security company as Cyber Threat Analysts, where they analyze and reverse engineer malicious documents. They both use a variety of command line tools to manually remove malicious code embedded in different types of documents. They also monitor network traffic, and create packet captures of potentially hazardous traffic, to further investigate, and determine its validity. Based on this we wanted to create our own tool that could automatically perform these steps.

Design Objectives

The goal of our project was to develop an application that will automate document analysis. ReapRE allows a user to upload a suspicious document to a simple and intuitive application and run it through a framework of tools designed for analysis. Once finished, the user is presented with a report that includes a list of IOCs, including domain names, URLs, emails, IP addresses, etc. The report will tell the user if the document is benign, suspicious, or malicious. The user is then be able to add any relevant IOCs into their firewall/security tools. Other goals and features of ReapRE include file upload, analysis summary report including IOC list, and malicious macro extractor. Once finished, ReapRE will also search file hashes against known threat database, display charts and graphs containing uploaded file statistics, and perform Yara file scans.

The first thought for our project started with an initial interest in network and cyber security. We had a few ideas, however, an IDS system to detect malicious activity and documents in the home was the beginning of our ideas. After this was proposed and further researched, we decided that switching our focus towards small businesses and IT

teams would be a better option to carry out what we wanted to accomplish with our project.

Methodology / Technical Approach

We decided on a design that would be easy to use and aesthetically pleasing to the user. We wanted to ensure that the user would have no issues locating or using the features of our project. Our goal was to make a clear and user-friendly design that would be the most effective in assisting analysts. We wanted it to be streamlined and simple, keeping in mind our target audience of small business analysts

User Profile

As listed below in the User Profile (Figure 1), the primary user for ReapRE is a security analyst at a small business, required to determine if a file is benign or malicious. The frequency of use will largely be dependent on each individual user. The user interface is simple and intuitive. The User Profile (Figure 1) below outlines this further.

User Profile Form
PROJECT: ReapRE will allow security analysts at small businesses to automate the analysis of potentially malicious documents. Users will be able to upload files into our application, where it will be run against various malware analysis tools and modules to scan and analyze documents, outputting a report based on its findings.
POTENTIAL USERS: -IT security teams needing to analyze potentially malicious documents -Cybersecurity analysts

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

ReapRE is aimed at personnel with some document analysis experience and command-line interfaces (if setting up a shared database). These users will likely understand the basics of cyber security and fundamentals of reverse engineering, as well as have strong critical thinking skills. They should be comfortable reading and ingesting potentially malicious traits and IOCs to validate the maliciousness of uploaded files. The user must be trusted to import IOCs into firewall block lists and other security tools the company might utilize.

EXPERIENCE WITH SIMILAR APPLICATIONS:

-Open source sandboxing
-Command line interface
-Basic understanding of security concepts

TASK EXPERIENCE:

-Standing up MISP (threat intelligence sharing platform)
-Using a web browser application to upload files
-Comfortable with interpreting the IOC report

FREQUENCY OF USE:

Frequency of use will vary based on use case. Users will upload files whenever they need to analyze a potentially malicious document.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

-Simple, intuitive GUI
-Quickly analyze the document
-Easy to read report
-Simple button to add indicators to MISP

Figure 1. User Profile

Use Case Diagram

In Figure 2 below, our Use Case Diagram will display the type of user, and how they will interact with our application, ReapRE.

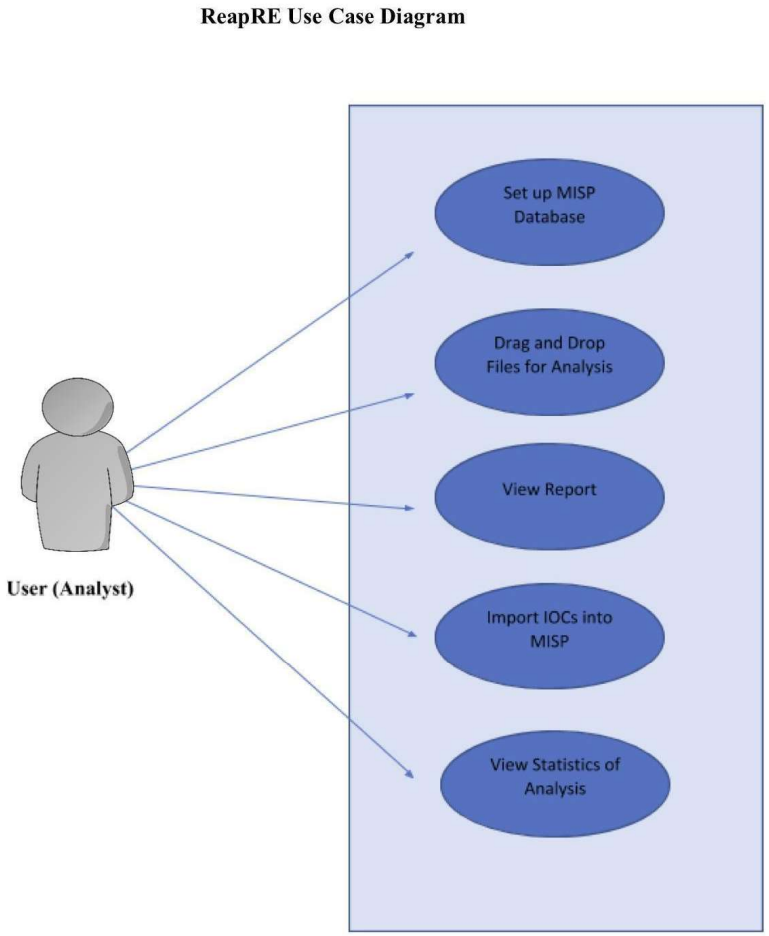


Figure 2. Use Case Diagram

Testing

Overview and Methodology

We will use three different manual approaches for our testing process:

1. Unit testing each module
2. User Acceptance Testing with documented use cases
3. Exploratory testing

The unit testing will consist of testing each module as they are completed. For instance, importing a file, analyzing the file and generating a report, importing the file into MISP. The developer will overview the test to answer any questions and record the results. They will be pass/fail and if it fails, the developer will write down any errors and the steps taken. The tests will be repeated to ensure accuracy of the results.

The user acceptance testing will consist of use cases created from the business requirements. We will recruit security analysts to test the functionality of the application. We will give them various malicious and non-malicious files to test. We will document the steps and record the results as pass/fail.

The exploratory testing will consist of 3 separate sessions where we will run through the application steps. We will be creative and attempt to break functionality of the application.

Scope of Testing

Our testing will cover the complete analysis life cycle, including importing a file, analyzing a file, generating a report, and importing IOCs into MISP. Attribution of the indicators is out of scope, as that is entirely environment dependent.

Testing Objective

Our testing must accomplish the following:

1. All intended features must work 100% of the time
2. Use cases tested must cover every step of the analyst process
3. All failed test cases must result in logging a “bug”
4. All bugs must be resolved before the IT EXPO

Logging Test and Procedures

Each test case will be recorded step-by-step by a team member using the Test Report. If the test fails or the user encounters an error, the team member will record as much information as possible about the error. They will then submit a bug and bring it to the attention of the other team members. Table 1. Schedule of User Testing, and Table 2. Schedule of Team Member Testing, which are both located below, will give the time ranges of when both sets of tests will be conducted.

Test Cases

1. Submit File
    a. Steps:
        i. Navigate to ReapRE home page.
        ii. Upload File
        iii. Click Submit
    b. Expected Outcome
        i. User is presented with links to Report and IOC csv
2. Analyzing File
    a. Steps
        i. User clicks on Report link
        ii. User/Analyst verifies findings
        iii. User Navigates back to Results Page
        iv. User clicks IOC csv link
    b. Expected Outcome
        i. The Report is generated with proper findings
        ii. The IOC csv contains all expected IOCs from uploaded file

3. Importing IOCs into MISP
    a. Steps
        i. User clicks import in MISP
        ii. User uploads csv file
        iii. User clicks Submit
    b. Expected Outcome
        i. IOCs are successfully imported into MISP

Schedule of User Testing

Members	Timeline	How Often?
Analysts	2/1/2020 to 3/28/2020	Weekly
Students	2/1/2020 to 3/28/2020	Weekly

Table 1. Schedule of User Testing

Schedule of Team Member Testing

Members	Timeline	How Often?
Developer	1/1/2020 to 3/28/2020	Weekly
Project Manager	1/1/2020 to 3/28/2020	Weekly

Table 2. Schedule of Team Member Testing

Test Results

Table 3 below shows the results of the User testing. These tests were taken by other students and security analysts. Table 4 below shows the results from the tests taken by the team members.

Date	Tester	Test Case	Input	Expected Output	Actual Output	Pass/Fail
2/1/2020	Student	#1 - Uploading test files	Click "Browse" Select proper test files Click "Submit"	Successful file upload Redirected to "Results" landing page	File uploaded successfully User is redirected to "Results" Landing page	Pass
2/1/2020	Analyst	#1 - Uploading test files	Click "Browse" Select proper test files Click "Submit"	Successful file upload Redirected to "Results" landing page	File uploaded successfully User is redirected to "Results" Landing page	Pass
2/8/2020	Student	#1 and 2 - Uploading and Analyzing TXT	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/8/2020	Analyst	#1 and 2 - Uploading and Analyzing TXT	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/15/2020	Student	#1 and 2 - Uploading and Analyzing PDF	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/15/2020	Analyst	#1 and 2 - Uploading and Analyzing PDF	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/22/2020	Student	#1 and 2 - Uploading and Analyzing DOC	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/22/2020	Analyst	#1 and 2 - Uploading and Analyzing DOC	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/29/2020	Student	#1 and 2 - Uploading and Analyzing EXE	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/29/2020	Analyst	#1 and 2 - Uploading and Analyzing EXE	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
3/7/2020	Student	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
3/7/2020	Analyst	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
3/14/2020	Student	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
3/14/2020	Analyst	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
3/21/2020	Student	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis/Report IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/21/2020	Analyst	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/28/2020	Student	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/28/2020	Analyst	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass

Table 3. User Test Results

Date	Tester	Test Case	Input	Expected Output	Actual Output	Pass/Fail
1/3/2020	Team Member	#1 - Uploading test files	Click "Browse" Select proper test files Click "Submit"	Successful file upload Redirected to "Results" landing page	File uploaded successfully User is redirected to "Results" Landing page	Pass
1/10/2020	Team Member	#1 - Uploading test files	Click "Browse" Select proper test files Click "Submit"	Successful file upload Redirected to "Results" landing page	File uploaded successfully User is redirected to "Results" Landing page	Pass
1/17/2020	Team Member	#1 and 2 - Uploading and Analyzing TXT	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
1/24/2020	Team Member	#1 and 2 - Uploading and Analyzing PDF	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
1/31/2020	Team Member	#1 and 2 - Uploading and Analyzing DOC	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/7/2020	Team Member	#1 and 2 - Uploading and Analyzing EXE	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/14/2020	Team Member	#1 and 2 - Uploading and Analyzing File	Submit file Click on "Results" and "IOC.CSV" Verify results	Successful file upload Report is generated correctly IOCs are extracted and listed	File uploaded successfully Report and IOC list are correct	Pass
2/21/2020	Team Member	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
2/28/2020	Team Member	#3 - Importing IOCs into MISP	User imports test IOCs into MISP	User imports IOCs User Verifies MISP event	IOCs imported successfully MISP event verified	Pass
3/6/2020	Team Member	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis/Report IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/13/2020	Team Member	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/20/2020	Team Member	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass
3/27/2020	Team Member	#1, 2, and 3	Submit file Verify results Import IOCs into MISP	Successful file upload Successful Analysis IOCs imported into MISP	File uploaded successfully Analysis/Report verified IOCs imported	Pass

Table 4. Team Member Test Results

Budget

Table 5: Initial Project Budget outlines the budget for this project. The total amount of money that will be needed to create this project is zero. Any software that will be used in this project is open source. The MISP database will be run on an old computer. Since this is a project, we will estimate that our labor rates would be equal with an industry standard, calculated at $20 an hour. Assuming 180 hours of labor, the estimated

total labor cost is $3,600.

Item	Estimated Cost	Actual Cost
Hardware	$0.00	N/A
Software	$0.00	N/A
Server	$0.00	N/A
Labor	$3,600.00	$0.00

Table 5. Initial Project Budget

Our final budget, which is located below in Table 6, is exactly the same as our initial budget. As previously stated, all the software we used was open source, and any hardware that was used, was repurposed from old equipment. The MISP database was housed on a virtual machine, as well as our application. Once again, using industry standards, we used $20/hr. as our labor rate, for 180 hours, giving us a total labor cost of $3,600.

Item	Estimated Cost	Actual Cost
Hardware	$0.00	N/A
Software	$0.00	N/A
Server	$0.00	N/A
Labor	$3,600.00	$0.00

Table 6. Final Project Budget

Objectives/Deliverables

Project Schedule

Figure 3: Fall Semester 2017 Gantt Chart, Figure 4: Spring Semester 2018 Gantt Chart and Table 7: Work Breakdown Structure outline the projected schedule for completion of this project.

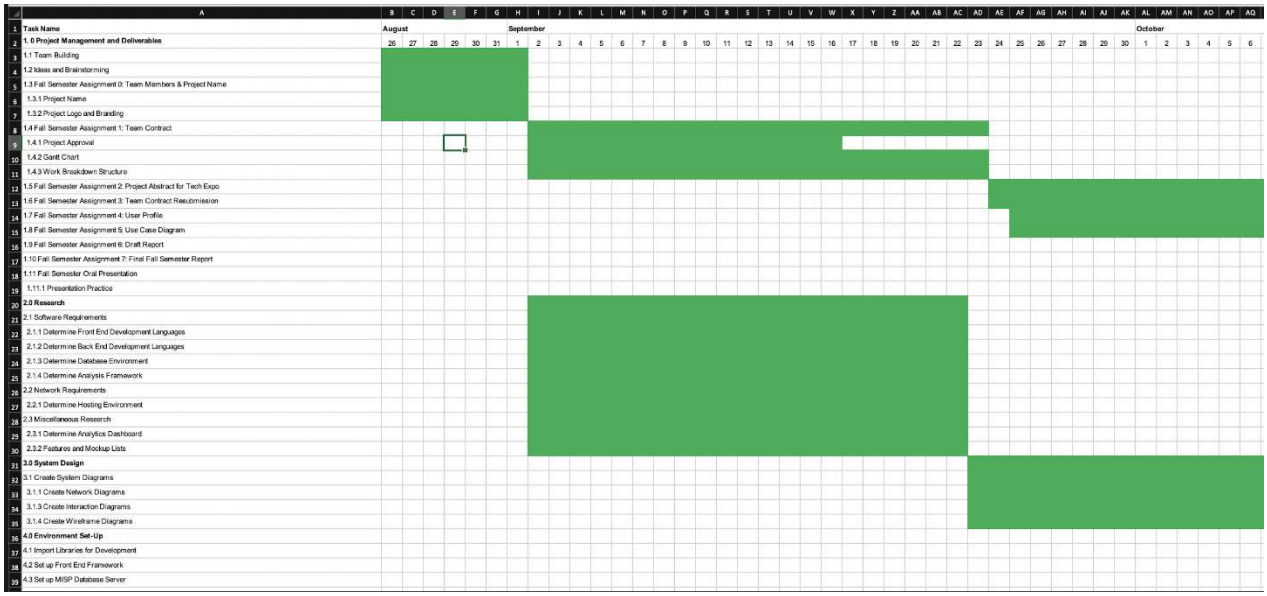


Figure 3. Fall Semester 2019 Gantt Chart

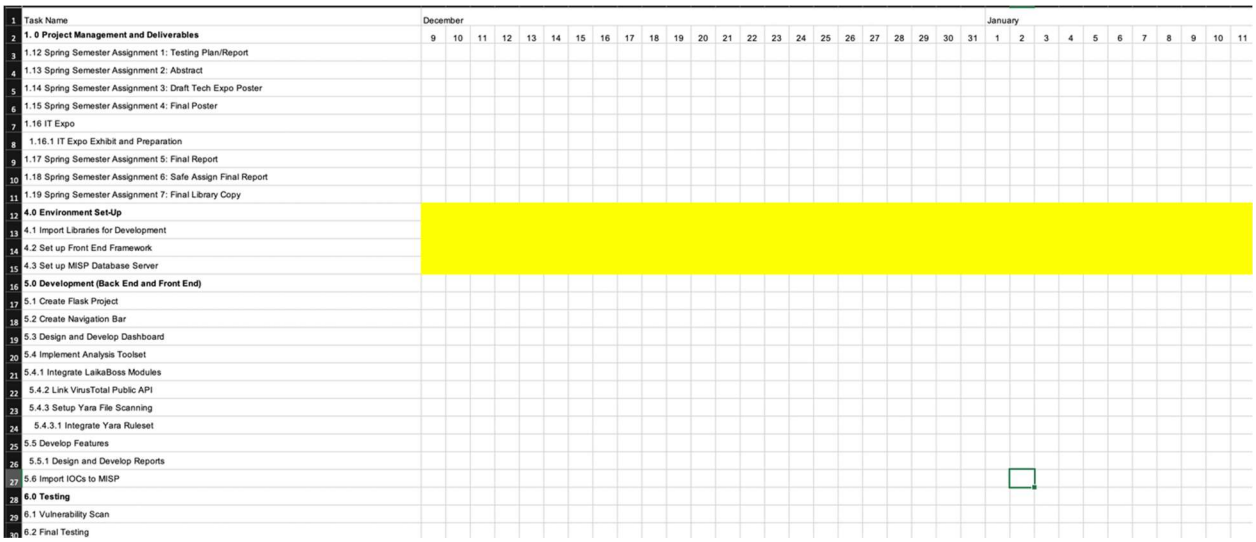


Figure 4. Spring Semester 2020 Gantt Chart

ReapRE WBS			
Task Name	**Duration (Days)**	**Start Date**	**End Date**
1. 0 Project Management and Deliverables	232	8/26/19	4/14/20
1.1 Team Building	6	8/26/19	9/1/19
1.2 Ideas and Brainstorming	6	8/26/19	9/1/19
1.3 Fall Semester Assignment 0: Team Members & Project Name	6	8/26/19	9/1/19
1.3.1 Project Name	6	8/26/19	9/1/19
1.3.2 Project Logo and Branding	6	8/26/19	9/1/19
1.4 Fall Semester Assignment 1: Team Contract	21	9/2/19	9/23/19
1.4.1 Project Approval	14	9/2/19	9/16/19
1.4.2 Gantt Chart	21	9/2/19	9/23/19
1.4.3 Work Breakdown Structure	21	9/2/19	9/23/19
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	20	9/24/19	10/14/19
1.6 Fall Semester Assignment 3: Team Contract Resubmission	20	9/24/19	10/14/19
1.7 Fall Semester Assignment 4: User Profile	26	9/25/19	10/21/19
1.8 Fall Semester Assignment 5: Use Case Diagram	26	9/25/19	10/21/19
1.9 Fall Semester Assignment 6: Draft Report	13	10/22/19	11/4/19
1.10 Fall Semester Assignment 7: Final Fall Semester Report	20	11/5/19	11/25/19
1.11 Fall Semester Oral Presentation	6	11/26/19	12/2/19
1.11.1 Presentation Practice	6	11/26/19	12/2/19
1.12 Spring Semester Assignment 1: Testing Plan/Report	28	1/13/20	2/10/20
1.13 Spring Semester Assignment 2: Abstract	6	2/11/20	2/17/20

1.14 Spring Semester Assignment 3: Draft Tech Expo Poster	13	2/18/20	3/2/20
1.15 Spring Semester Assignment 4: Final Poster	6	3/3/20	3/9/20
1.16 IT Expo	0	4/14/20	4/14/20
1.16.1 IT Expo Exhibit and Preparation	1	4/13/20	4/14/20
1.17 Spring Semester Assignment 5: Final Report	5	4/1/20	4/6/20
1.18 Spring Semester Assignment 6: Safe Assign Final Report	5	4/1/20	4/6/20
1.19 Spring Semester Assignment 7: Final Library Copy	13	4/7/20	4/20/20
2.0 Research	20	9/2/19	9/22/19
2.1 Software Requirements	20	9/2/19	9/22/19
2.1.1 Determine Front End Development Languages	20	9/2/19	9/22/19
2.1.2 Determine Back End Development Languages	20	9/2/19	9/22/19
2.1.3 Determine Database Environment	10	9/12/19	9/22/19
2.1.4 Determine Analysis Framework	20	9/2/19	9/22/19
2.2 Network Requirements	20	9/2/19	9/22/19
2.2.1 Determine Hosting Environment	20	9/2/19	9/22/19
2.3 Miscellaneous Research	20	9/2/19	9/22/19
2.3.1 Determine Analytics Dashboard	20	9/2/19	9/22/19
2.3.2 Features and Mockup Lists	20	9/2/19	9/22/19
3.0 System Design	76	9/23/19	12/8/19
3.1 Create System Diagrams	76	9/23/19	12/8/19
3.1.1 Create Network Diagrams	76	9/23/19	12/8/19
3.1.3 Create Interaction Diagrams	76	9/23/19	12/8/19

3.1.4 Create Wireframe Diagrams	76	9/23/19	12/8/19
4.0 Environment Set-Up	58	12/9/19	2/5/20
4.1 Import Libraries for Development	58	12/9/19	2/5/20
4.2 Setup Front End Framework	58	12/9/19	2/5/20
4.3 Set up MISP Database Server	58	12/9/19	2/5/20
5.0 Development (Back End and Front End)	55	2/6/20	4/1/20
5.1 Create Flask Project	55	2/6/20	4/1/20
5.2 Create Navigation Bar	55	2/6/20	4/1/20
5.3 Design and Develop Dashboard	55	2/6/20	4/1/20
5.4 Implement Analysis Toolset	55	2/6/20	4/1/20
5.4.1 Integrate LaikaBoss Modules	55	2/6/20	4/1/20
5.4.2 Link VirusTotal Public API	55	2/6/20	4/1/20
5.4.3 Setup Yara File Scanning	55	2/6/20	4/1/20
5.4.3.1 Integrate Yara Ruleset	55	2/6/20	4/1/20
5.5 Develop Features	55	2/6/20	4/1/20
5.5.1 Design and Develop Reports	55	2/6/20	4/1/20
5.6 Import IOCs to MISP	55	2/6/20	4/1/20
6.0 Testing	10	4/2/20	4/12/20
6.1 Vulnerability Scan	10	4/2/20	4/12/20
6.2 Final Testing	10	4/2/20	4/12/20

Table 7. Work Breakdown Structure

TECHNICAL ELEMENTS

Network

ReapRE will be hosted locally on an analyst's computer instead of on an external server. This ensures that analysts will be able to utilize this tool when not connected to the internet. It is an extremely cost-effective solution as each analyst has their own personal work computer. The MISP database will be hosted on a local server, easily accessible by the analysts on the same network.

Application

Users will access ReapRE through a simple web application hosted locally. The front end will be built using HTML, bootstrap for core CSS elements, and jQuery. The back end will be written in python utilizing the Laikaboss framework. The web application is built on a Flask server. When the file is put through the application, it is scanned using Laikaboss. ReapRE will scan the document against open source yara rules. Users can import their own yara rules by pointing to the location of the rules in the configuration file. Depending on the type of file submitted, ReapRE will scan it against various additional analysis modules. If the file type is determined to be a Microsoft Office document or an OLE (Object Linking and Embedding), it will be sent to the oletools module. This module will attempt to extract any embedded objects and pull out macros for the user to review in the report. The user will be presented with a report containing the analysis results and a CSV containing the extracted IOCs. The IOCs will be imported to the MISP database based on the user's discretion. Figure 5 (below) outlines the application architecture.

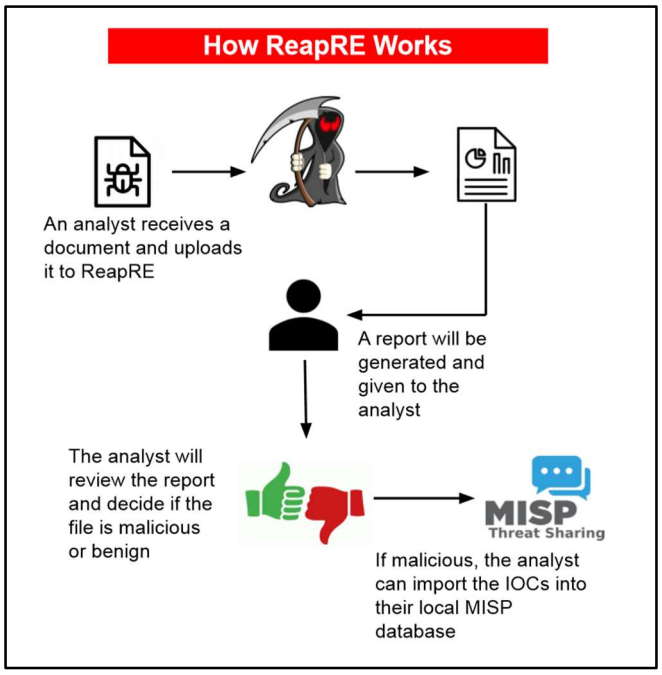


Figure 5. Application Architecture

Database

MISP will be the database used to store any IOCs generated by ReapRE. An analyst will selectively import IOCs from ReapRE's analysis report. Using MISP's automatic correlation feature, once IOCs are entered into the database, an analyst will be able to attribute IOCs based on such factors as the threat actor and the method obtained. MISP's intuitive GUI will also allow an analyst the ability to easily navigate between an event and any possible correlations, such as a particular attack campaign.

MISP is designed to easily incorporate IOCs into external security devices, such as firewalls and block lists to prevent future cyber-attacks. With MISP's API(Application Programming Interface), an analyst will have the ability to search against previously collected IOCs when analyzing a new document. Each MISP instance has the ability to be customized based on an organization's individual environment need, with features such as the ability to make custom tags for IOC labeling purposes.

With MISP's feed import feature, an analyst/organization will have the ability to import verified IOCs from a variety of sources, which can be filtered on attributes such as industry or location. An organization will also have the ability to share their IOCs with others, based on the same filtering features. Analysts will be able to label and group indicators relating to similar threats/targets, write notes, and have the ability to link other events for future references.

USER INTERFACE

The user interface was designed with HTML and bootstrap CSS templates on top of a Flask server. The home page contains a button where the analyst can upload a file for analysis. The minimalistic design ensures fast response and loading times. Once the file has gone through the analysis framework, the user is dynamically presented with a hyper-link taking them to the analysis report. Figure 6 shows the Home Page, where the user is presented with the option to browse their device for a file they want to upload. After clicking the submit button, the user is taken to the results page (Figure 7). On this page, the user is presented with options to view the analysis report or download a .CSV containing the extracted IOCs. Figure 8 shows the report that the user can view after clicking the "Click here for results" link located on the results page.

SCREENSHOTS

Home Page:

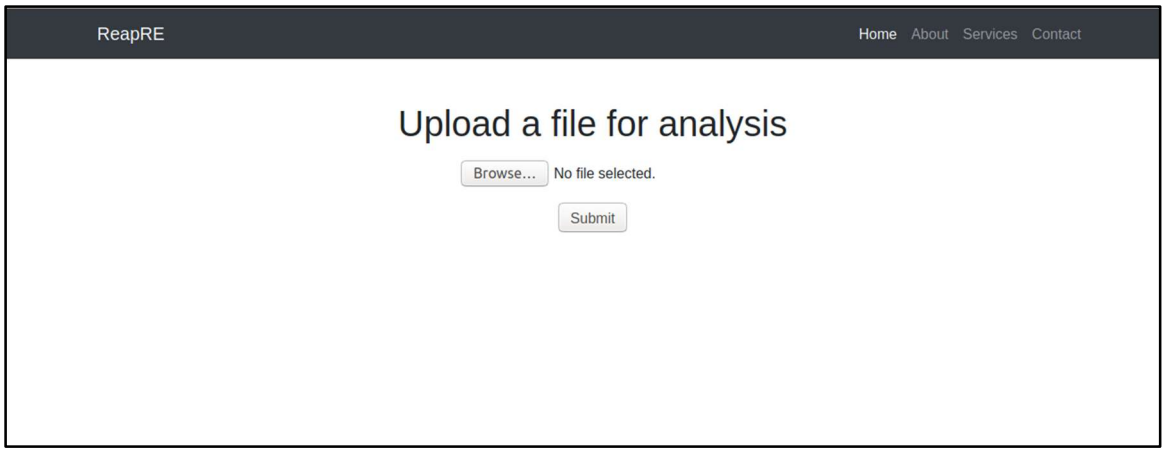


Figure 6. Home Page Screenshot

Results Page:

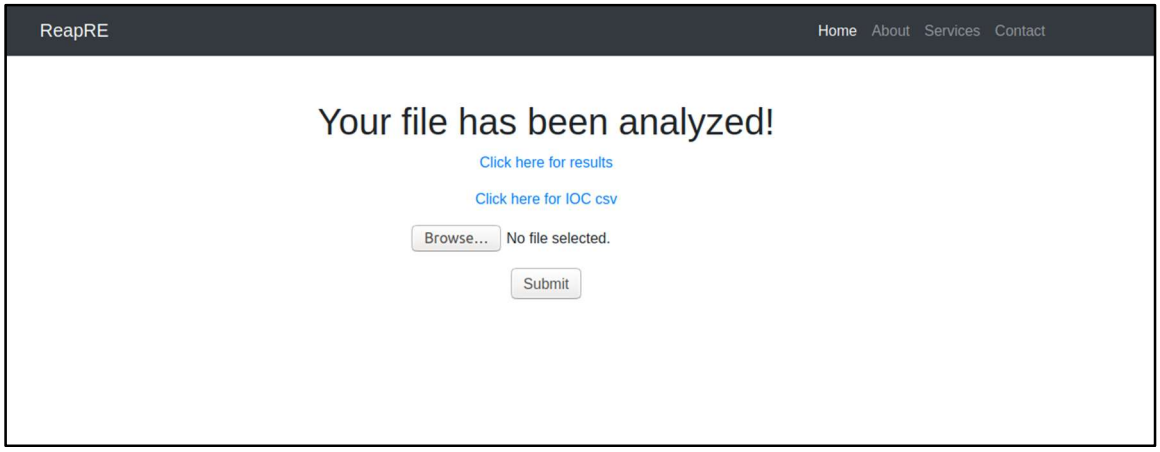


Figure 7. Results Page Screenshot

Report:

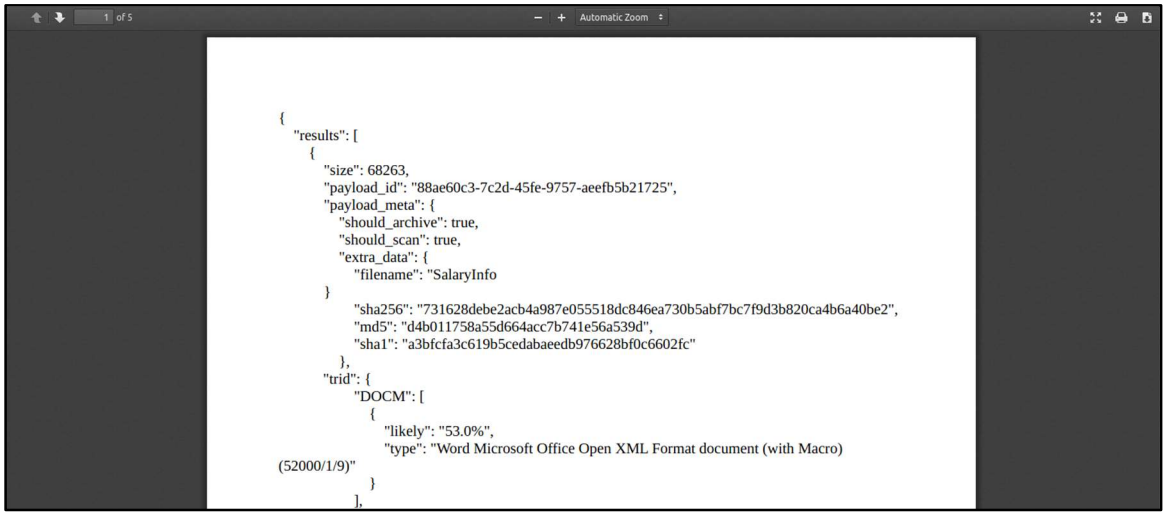


Figure 8. Report Screenshot

PROBLEMS ENCOUNTERED

The first problem we encountered was deciding on our project idea. We originally wanted to create a home network intrusion detection sensor. After meeting with our advisor, we decided to scrap the idea as we ran into several issues. We did not think that the average home user would be able to understand the technical aspects of an IDS. After brainstorming for a few days, Michael and Matt thought about issues they commonly face during their daily work and decided on creating an application to assist in analyzing potentially malicious documents.

The second problem we encountered was implementing the analysis framework. We often found one of the modules failing, causing the results to throw an error. This was mostly resolved through reading the documentation and GitHub pages of the various tools. We then would make small changes to the configuration files and perform tests to ensure the issues were resolved.

Future Recommendations

Our recommendation for ourselves is to spend less time researching the perfect solution. It would have been better to find a suitable solution and begin development right away. If we started development early on in the Fall semester, we may have been able to implement ReapRE in front of a mail server to actively stop malicious documents from being delivered to end users, as was discussed in early meetings with our advisor, and is currently our plan for the future.

A second recommendation is redesigning the front end. The application would look better if we added more color and visuals. It would benefit the analyst to have graphs of recent trends and statistics, including number and types of files analyzed,

number of malicious documents, and a map of where each IP/domain is hosted.

CONCLUSION

Fall Semester 2019

We began the semester with the idea to build a home network intrusion detection sensor but changed our minds to an analysis tool to help small businesses and IT departments. The main goal and progress of the first semester was to learn about different methods to create a simple front-end web application. We narrowed down our choices and settled on using flask and bootstrap. This has been a challenge, as these tools are new to our group. We will be spending the next few weeks reading the documentation to get a firm understanding of how best to utilize these tools. We spent a large amount of time narrowing our scope and creating a list of document analysis tools we would like to implement into ReapRE. We also started researching the most optimal method of incorporating the tools decided.

Spring Semester 2020

The majority of this semester was spent getting our project up and running. There was a lot of research done to get all the tools to work seamlessly together. The MISP database we used was hosted on a virtual machine. We spent a majority of the semester writing programs so that Laika boss, oletools, and yara were able to function as needed in extracting the IOCs from the documents. Testing and refining our project was also an important part during this semester. A lot of time was also spent formatting how our final report and board would be displayed, as well as the user interface design of our application.

APPENDIX A. ReapRE Poster

University of CINCINNATI

ReapRE

College of Education, Criminal Justice, and Human Services
School of Information Technology

Team 12: Matthew Granitto, Claire Church & Michael Ferguson | Advisor: Bo Vykhovanyuk

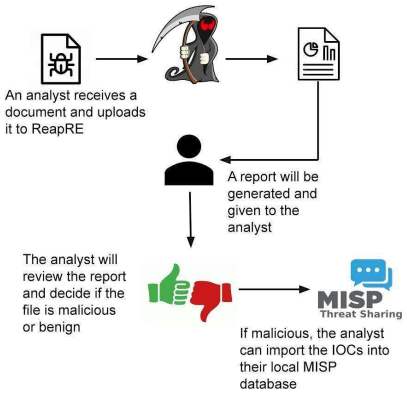
Problem

- IT Security teams are often overworked and understaffed
- According to a survey taken by Sophos (2018), 86% of respondents said they need greater skills in their organization and 50% of respondents claimed phishing emails are their main risk
- Small businesses do not have the resources for detailed document analysis

Our Solution

ReapRE is an automated malware analysis tool designed to assist analysts at small businesses. Analysts can upload potentially malicious documents into our application where it will be run against several malware analysis tools and modules.

How ReapRE Works



Technology

LAIKA BOSS
Flask
MISP Threat Sharing
Oletools
/* YARA */

APPENDIX B. REFERENCES

Sophos. “The Impossible Puzzle of Cybersecurity” Sophos June 2019. Accessed November 10, 2019 https://www.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/sophos-impossible-puzzle-of-cybersecurity-wp.pdf

Critical Start. “The Impact of Security Alert Overload” Critical Start 2019. Accessed November 10, 2019 https://www.criticalstart.com/wp-content/uploads/CS_MDR_Survey_Report.pdf