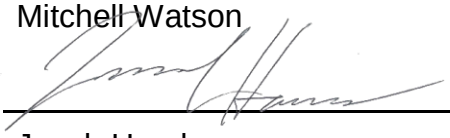
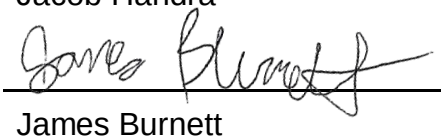
Operating System Customs

By

Mitchell Watson, Jacob Handra, James Burnett

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

	4/15/2019
Mitchell Watson	Date
	4/15/2019
Jacob Handra	Date
	4/15/2019
James Burnett	Date
Bogdan Vykhovanyuk	4/15/2019
Bogdan Vykhovanyuk, Faculty Advisor	Date

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

April 29, 2019

OPERATING SYSTEM Customs

Prepared by
Mitchell Watson, Jacob Handra, James Burnett

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

April 29, 2019

Table of Contents

Tables and Figures

Abstract

“In 2016, Researchers from the University of Illinois left 297 unlabeled USB flash drives around the university campus to see what would happen. 98% of the dropped drives were picked up by staff and students, and at least half were plugged into a computer in order to view the content. For a hacker trying to infect a computer network, those are irresistible odds” (Kaspersky Labs). Plug-and-play (PNP) technologies have become very common in today’s world, but there’s no reliable way to ensure that everything plugged into a computer will play nicely. Operating System Customs (OSC) aims to maintain the “plug-and-play” name by introducing a new layer to ensuring that everything plugged in is verified as 100% safe. This project ensures your computer’s safety as well as your network’s, identifying your external medium and scanning it to verify if a virus/malware was installed without your knowledge. After the scan, it’s either verified to be safe or dangerous and lists off any discovered corrupt files. OSC allows users to operate in a safe network and grant system administrators peace of mind.

1. **Problem Statement:**

Introduction

External media is a common way of network penetration that is often overlooked in companies of large and small scale alike, with infected USB sticks often being the most common method of infection. Companies will often accommodate this by implementing a form of USB security where only preemptively approved USB sticks assigned to users can be plugged into any computer at any given time. While this is effective in some cases, it doesn't prevent those company-approved USB sticks from being infected from other external forces (such as an off-site customer with an infected computer or an acquaintance's infected laptop). There are small, cumbersome workarounds, but none that offer quicker and easier solutions that don't leave room for human error.

Problem

Viruses are ubiquitous in this modern world, and they can cost companies dearly, literally billions of dollars. They can also jeopardize the safety and well-being of individuals. Beyond that, they are multiplying in number at an exponential rate. "More than 317 million new pieces of malware -- computer

viruses or other malicious software -- were created last year. That means nearly one million new threats were released each day." (Harrison, Pagliery). This article was compiled in 2014, since then, the situation has become vastly more dire, and will only become worse and worse as time goes on.

One of the most common ways a private network may get infected is at the fault of external media being simply plugged into a network. There are ways to prevent this from accidentally happening, but at the cost of being a cumbersome approval system and an almost "too foolproof" ruleset that becomes an annoying roadblock for users where they need media approved by their IT department when that cumbersome system is in play.

One of the greatest threats to the well-being of individuals and companies in terms of information security, is the PNP device. "Nearly one in five people who found a random USB stick in a public setting proceeded to use the drive in ways that posed cybersecurity risks to their personal devices and information and potentially, that of their employer, a recent experiment conducted on behalf of CompTIA, the IT industry association, revealed" (Downers).

Solution

The solution that we offer provides a method that not only allows the end user to not be bothered with gaining approval by the IT department, but also ensures the safety of a company network by preventing potentially threatening devices from gaining access to them. With our software, users can plug anything into their company/personal computer and not have to worry about accidentally infecting their network/computer. With the constantly updated database that OSC is based on, our application will ensure that your users are accessing safe data and working as efficiently as possible.

Description

The software developed acts as a sort of airport customs for any external media. Once the medium has been inserted, the software will detect the device's connection and conduct multiple different scans. These scans will be conducted by the currently integrated antivirus software and other third-party scanners that will search for Javascript-embedded (or any other disguised executable) files and other types of infection types.

If all scans come back negative for viruses and/or malicious software, then the external media is granted access to the computer and vice versa. If the

scans come back positive for viruses and/or malicious software, the corrupted files will be presented to the user and ask whether they want to delete the listed files or not. A prompt will inform the user of these conditions and will leave any further options to the discretion of the user.

Methodology/Technical Approach

Figure 1. User Profile was used to represent the scope of what users we expected to utilize our product.

APPLICATION: Operating System Customs – a software for making certain that all flash drives, disks, and other media attached to system are secure
POTENTIAL USERS: · Business professionals · Everyday home computer users
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This software is extremely hands-on and easy to use, with almost no input or prior knowledge required for optimal functionality. Whenever a disk or flash drive is plugged in, the software automatically detects any malware or potential threats, alerting the user to their existence immediately. This will allow anyone to successfully secure their system using our software

EXPERIENCE WITH SIMILAR APPLICATIONS:

- McAfee
- Symantec Endpoint Protection
- EgoSecure
- Windows Defender

TASK EXPERIENCE:

- Using the Windows operating system
- Previous use of similar Anti-virus tools, although this is not mandatory
- Prior knowledge of the safety concerns of USB plug-and-play

FREQUENCY OF USE:

Constantly running as a background process that takes up very minimal hardware resources

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Quickly find and eliminate all potential threats with ease, before they can manifest on a user's machine

· User friendliness, you will not have to be an IT professional to use our program

· Takes up little resources

Figure 1. User Profile

2. Project Management

Objectives and Deliverables

The early objective of this project was to deliver an application that would block any and all traffic between a computer and a newly inserted external medium. The application would have a way to completely isolate the data stored on it while still being able to scan it. Once the scan was complete, the block would be lifted, and data stored within the medium would become accessible again.

We wanted an integration that allowed compatibility with any popular antivirus software available at the time. This included, but was not limited to: Symantec, Sophos, Norton, McAfee, ESET, Kaspersky, and Avast.

We also looked for an integration of third-party or internally developed scanning codes that would scan for Javascript-integrated documents that would be disguised as a common filetype. Scanning codes included scanning for any executables that made any sort of changes upon immediate plugin.

The user would be notified when the medium has been plugged in and that it's currently undergoing a scan. They would then be prompted that it's either safe, or malicious and needs to remain in quarantine. This is represented in ***Figure 2. Use Case Diagram***, which shows both the architecture of OSC and what we wished to accomplish from the perspective of the user.

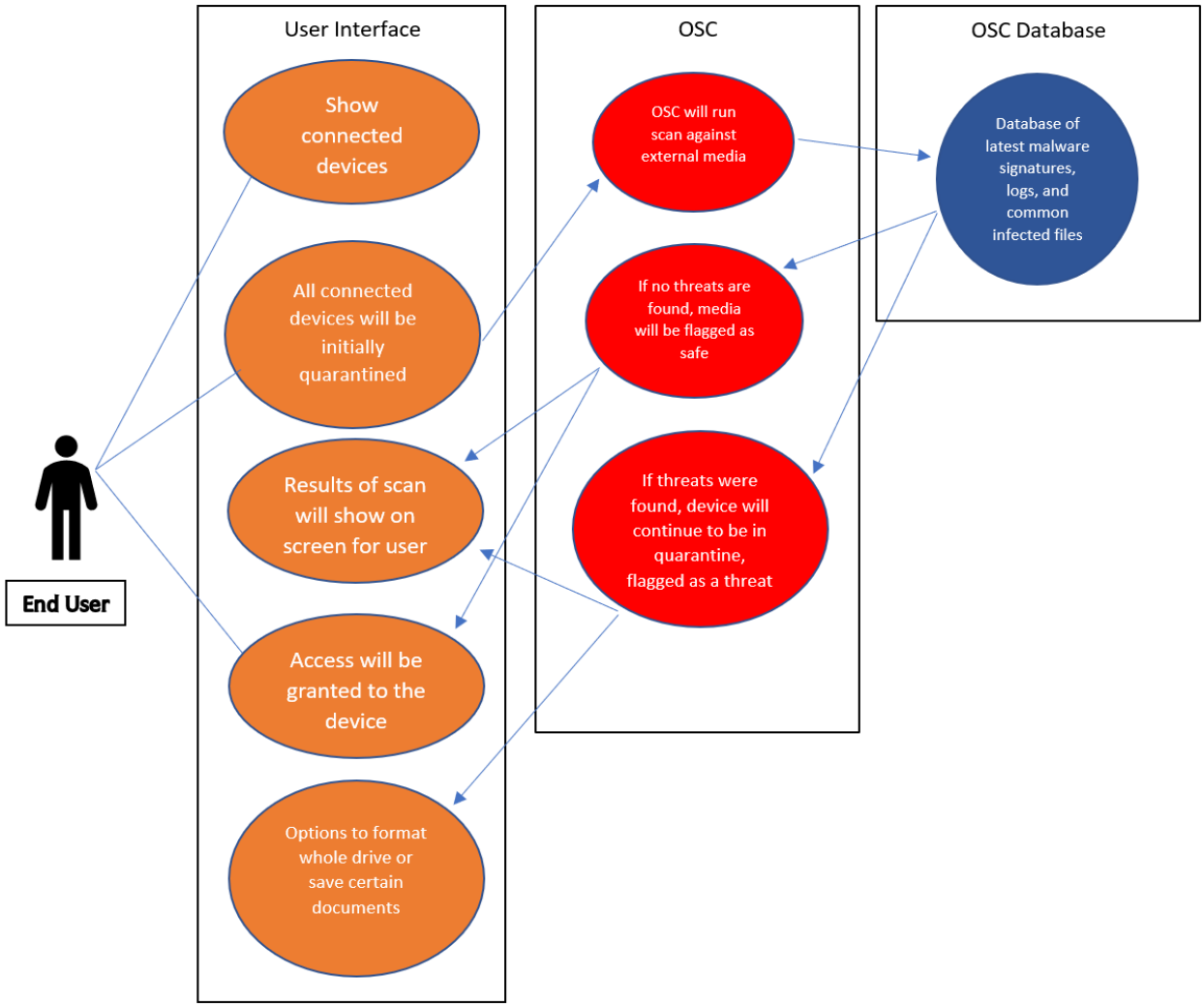


Figure 2. Use Case Diagram

Fall Semester Schedule

August 27, 2018 – September 17, 2018

- Form group and establish project idea
- Create contract lay foundation for the project
- Establish work schedule, WBS, and Gantt chart

September 18, 2018 – October 15, 2018

- Begin research on software development
- Decide on development platforms (programming environment, language, and test environment)
- Develop Use Case Diagram and User Profile

October 16, 2018 – November 19, 2018

- Continue research and development
    - Specify research regarding external media quarantining
- Begin writing and finalizing report for the Fall Semester
- Present project to class

November 20, 2018 – January 7, 2019

- Group goes on Winter Break – reconvene a week after the New Year

Spring Semester Schedule

January 8, 2019 – February 11, 2019

- Resume research and development phase
- Begin testing phase
- Submit Test Plan and Report to Instructor

February 12, 2019 – March 18, 2019

- Personal deadline for final product: March 18th
- Revise abstract and create poster for IT Expo
- Fix bugs found in testing phase

March 19, 2019 – April 29, 2019

- Finish project's development
- Present Final Project and submit Final Report
- Attend IT Expo

Gantt Chart and Work Breakdown Structure

Figure 3. Fall Semester 2018 Gantt Chart represents the schedule we created in the beginning of the Fall semester. We laid out the different phases of development and based it on deliverables that were required for the class. ***Figure 4. Spring Semester 2019 Gantt Chart*** represents similar goals, but instead for the following Spring semester. This was intended to be updated accordingly to where we were at the time during development of the project. ***Figure 5. Work Breakdown Structure*** represents the entirety of the project planning and scope.

Figure 3. Fall Semester 2018 Gantt Chart

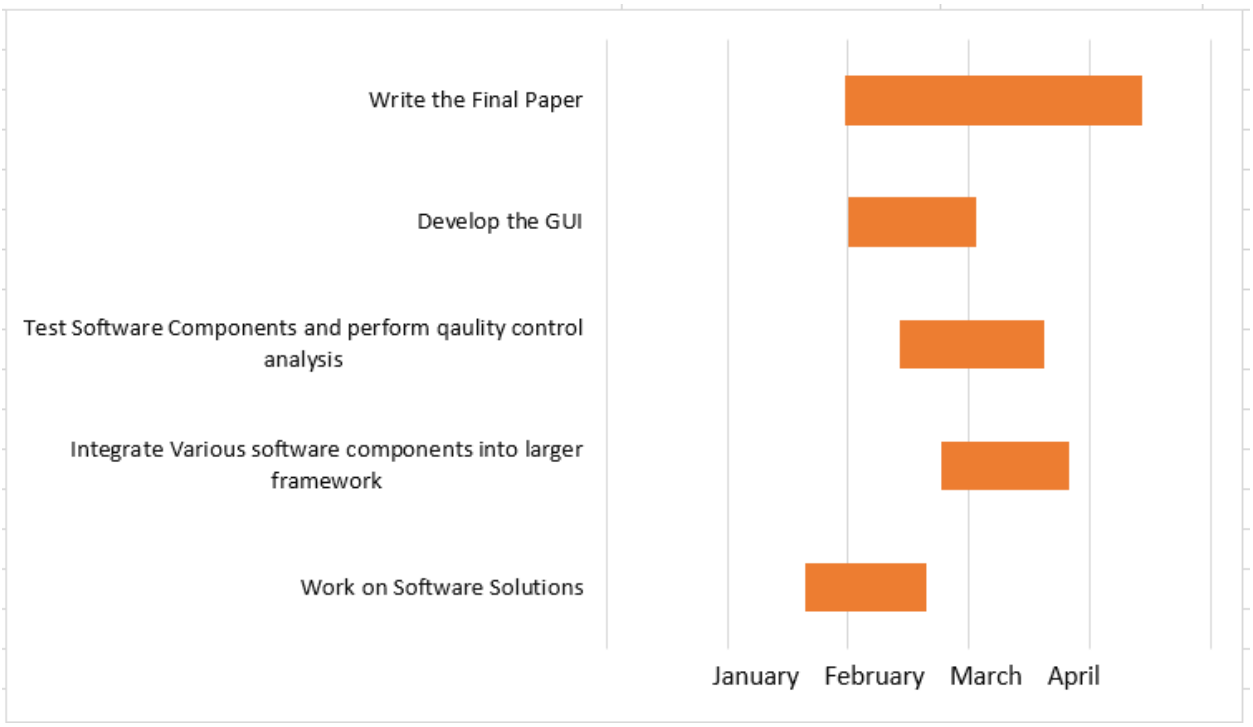


Figure 4. Spring Semester 2019 Gantt Chart

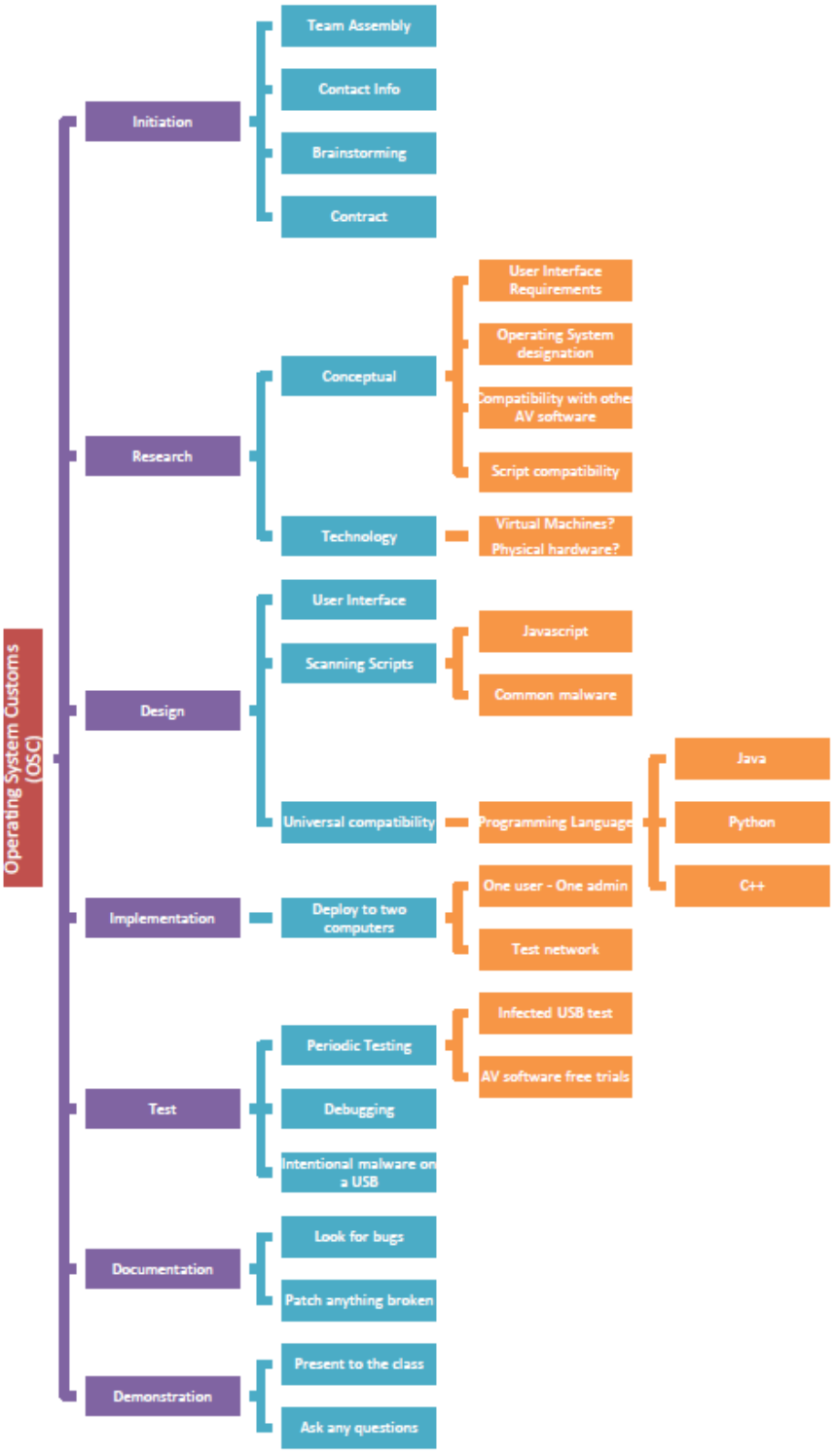


Figure 5. Work Breakdown Structure

Budget

Everything that was used to develop this application was either previously owned or free. This includes all computers, USB devices, and software. The only cost that we would have would be labour for development. ***Figure 6. Budget*** shows all costs associated with OSC.

No.	Item	Hours	Price Per Hour	Total Cost
1	Labor	150	$30	$4,500
0	Equipment	0	0	0
				$4,500

Figure 6. Budget

3. Technical Elements

Technical Architecture

OSC is primarily built with Powershell including the GUI, while also tapping into the compatibility of Python within Windows. Powershell really opens the door for enterprise environments as it is built into every Windows machine. This gives us longevity with our program because Powershell is slowly taking over for Command Prompt. As mentioned above, we utilized Windows' compatibility with Python to run some of our vulnerability scripts. We utilized Duck Hunter, PeePDF, and PFTriage on top of Windows Defender to have an added layer of protection on the client. OSC was built with ease of use in mind from the start of development. It will run in the background, and only prompt with any suspected security concerns or when a PNP device is connected. It will prompt the user with an easy to understand message and make certain the user is aware of the threat, and what can be done to mitigate it. This way even those without deep knowledge of computers and network security can perform their jobs without fear of infection. The GUI we have created with PowerShell code is easy to navigate, light on resources, and most pertinent information can be found immediately.

Application Screenshots

“Potential threats” is the only section users will have to note, however, safe files is a category showing files with no detected malware, and there is a technical section above that with more advanced information pertaining to the scan. ***Figure 7. Select Drive dropdown when starting OSC*** is a screenshot of what the users will see once they open OSC and select what drive they would like to scan.

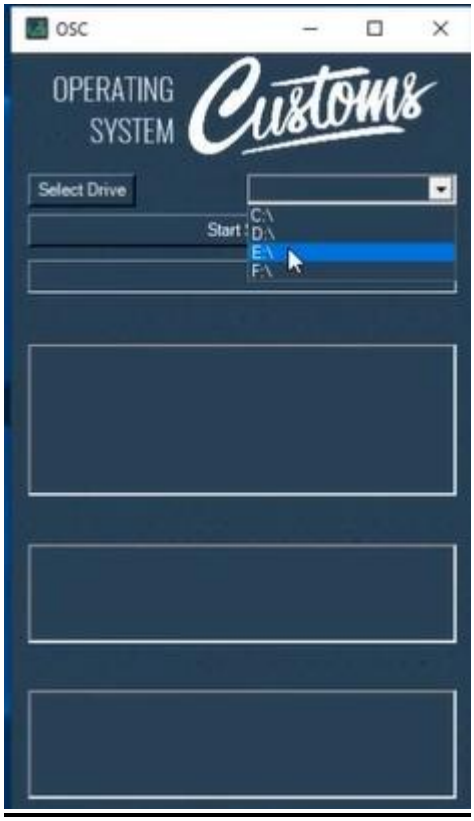


Figure 7. Select Drive dropdown when starting OSC

Figure 8. Completed Scan with no threats found is a screenshot of what a completed scan looks like when there aren't any threats detected on the drive.

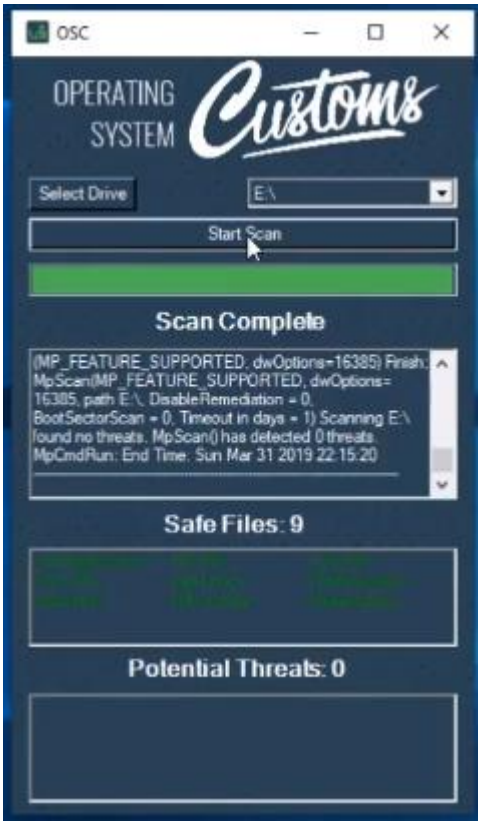


Figure 8. Completed Scan with no threats found

Figure 9. Potential Threat found with Error Message is what a user would see if OSC finds a potential threat on their drive. They must click either "Yes" or "No" to continue to the next screen or do anything else on their computer.

Figure 9. Potential Threat found with Error Message

4. Test Plan

Overview

This section will explain the testing methodology for the Operating System Customs application and should be used as a guide. The application will run solely on Windows-operated computers, so the following types of individuals should observe this section of the document:

- Developers
- Network Administrators
- Helpdesk Personnel
- IT Managers
- Security Analysts

Scope of Testing

The scope of testing is to test the overall operational side of OSC. We aim to test the overall functionality and make notes of any bugs that need to be fixed.

Objectives

The objective of testing OSC is to verify that its overall functionality is up to par where we need it to be in the current stage of development. This should allow any user to see how effective our application is when plugging in a USB device with a known harmful file within the device. If a USB is clear of harmful files, then the user should simply be notified of no risks being detected.

Logging Test and Procedures

We'll have various colleagues load up different malwares onto their designated USB devices and then plug each of them in one at a time to test the overall detection of our software. We will also be looking for any bugs, which will then be patched out in our final product.

Pass/Fail Conditions

It's expected that OSC will catch any and all malware that would be embedded into files located on the newly inserted USB device. Each malware detected will then be logged and reported to the user, granting them the option to delete said malware. If one single malware can penetrate this defense or if the malware isn't listed in the final scan report, then the application will have failed and will need to be patched. ***Figure 10. Test Results*** represents our findings whilst testing our product from January to March.

Tester	Date	Item#	Expected	Actual	Pass/Fail	Bug
Colleague	1/25/2019	1.1 Program Stability	Program runs consistently for 5 minutes whilst plugging and unplugging USB devices	Program ran consistently for 5 minutes	Pass	NA
Colleague	1/25/2019	1.2 Program Accuracy	Program accurately finds malware-infected file	Malware was detected	Pass	NA

			hidden in USB device			
Colleague	1/25/2019	1.2 Program Accuracy	Program accurately finds both malware-infected files hidden in USB device	One malware was detected, other was not	Fail	The scanning software meant to detect that filetype was not functioning properly
Colleague	1/25/2019	1.3 Program Results	Program accurately displays desired results: “malware.pdf”	Malware was properly displayed to user	Pass	NA
Colleague	1/25/2019	1.3 Program Results	Program accurately displays desired results: “malware.pdf” “malware.jpg”	Only one malware was reported, the other wasn’t	Fail	The filetype wasn’t detected in the scan, therefore it wasn’t reported

Colleague	3/5/2019	2.1 Program Stability	OSC runs consistently for 5 minutes after Duck Hunter was implemented.	Program ran consistently for 5 minutes	Pass	NA
Colleague	3/5/2019	2.2 Program Stability	OSC runs consistently for 5 minutes after PEEPDF was implemented.	Program ran consistently for 5 minutes	Pass	NA
Colleague	3/5/2019	2.3 Program Stability	OSC runs consistently for 5 minutes after PFTriage was implemented.	Program crashed after 10 seconds of running.	Fail	Space in print statement in PFTriage code.
Colleague	3/5/2019	2.4 Program Stability	OSC runs consistently for 5 minutes after PFTriage was implemented.	Program ran consistently for 5 minutes	Pass	N/A
Colleague	3/5/2019	2.5 Program Stability	OSC successfully deletes suspected file	Suspected file will be deleted.	Pass	

			after user gives consent.			

Figure 10. Test Results

What We Learned From Testing

To ensure that certain file types are properly scanned, we must ensure the code accounts for said file types. This also brought to question if there's a filetype that our application is not familiar with. We decided that if there is such an application type (such as: .n64 which is exclusively for Nintendo 64 emulation files) then OSC will see this, blacklist them, and then notify the user that they have been blacklisted and prompt if they'd like it to keep it this way.

Overall, our program is on the right path, but there is some room for improvement. We will be troubleshooting and testing this until our personal deadline date.

4. Conclusion

A lot of hardships were endured during development for OSC. One of the biggest roadblocks we had to work around was our lack of programming skills; all three of us were inexperienced in coding so we had to find ways to adapt to this. One commonality amongst us was familiarity with PowerShell and desire to learn about Python scripting. We also bit off a little more than we could chew by saying we would absolutely quarantine the USB device in order to run a scan through the same Operating System. The quarantining stage of the project required a plethora of coding within the Windows OS, and a lot of risks that would've taken a lot of time to work around and get right.

When gathering different Python scripts to integrate into our project, we had a hard time getting them to run within a PowerShell environment. We also wanted reliable tools that were able to catch anything we threw its way, and we were able to find those within GitHub.

A lot of adaptations were required to be made to the overall scope of the project, but we ended up very satisfied with what we accomplished. The response we received at IT Expo was relieving and we were proud of what we had accomplished. We also got a lot of compliments on our poster which is displayed in ***Appendix B. IT Expo Poster***.

Appendix

Appendix A

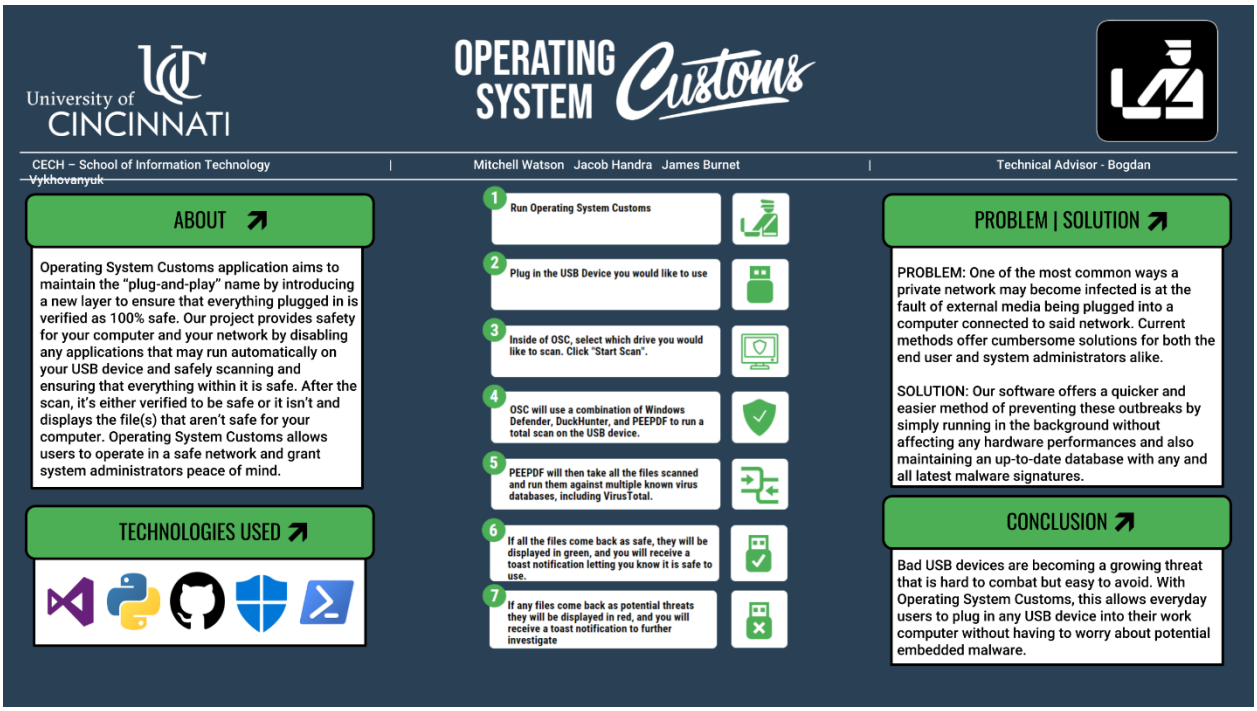
Downers, Grover, III. "Press Releases." CompTIA Information Technology. October 26, 2015. Accessed April 29th, 2019. https://www.comptia.org/about-us/newsroom/press-releases/2015/10/26/find-a-flash-drive-pick-it-up-experiment-shows-how-lack-of-cybersecurity-knowledge-can-impact-organizations.

Github. https://github.com/

Harrison, Virginia, and Pagliery, Jose. "Nearly 1 Million New Malware Threats Released Every Day." CNNMoney. April 14, 2015. Accessed April 30, 2019. https://money.cnn.com/2015/04/14/technology/security/cyber-attack-hacks-security/index.html

Kaspersky Labs. "USB Threats from Malware to Miners." Securelist English. September 25, 2018. Accessed April 30, 2019. https://securelist.com/usb-threats-from-malware-to-miners/87989/.

Appendix B


University of CINCINNATI
OPERATING SYSTEM Customs
CECH – School of Information Technology
Vykhovanyuk
Mitchell Watson Jacob Handra James Burnet
Technical Advisor - Bogdan
ABOUT
Operating System Customs application aims to maintain the "plug-and-play" name by introducing a new layer to ensure that everything plugged in is verified as 100% safe. Our project provides safety for your computer and your network by disabling any applications that may run automatically on your USB device and safely scanning and ensuring that everything within it is safe. After the scan, it's either verified to be safe or it isn't and displays the file(s) that aren't safe for your computer. Operating System Customs allows users to operate in a safe network and grant system administrators peace of mind.
TECHNOLOGIES USED
1 Run Operating System Customs
2 Plug in the USB Device you would like to use
3 Inside of OSC, select which drive you would like to scan. Click "Start Scan".
4 OSC will use a combination of Windows Defender, DuckHunter, and PEEPDF to run a total scan on the USB device.
5 PEEPDF will then take all the files scanned and run them against multiple known virus databases, including VirusTotal.
6 If all the files come back as safe, they will be displayed in green, and you will receive a toast notification letting you know it is safe to use.
7 If any files come back as potential threats they will be displayed in red, and you will receive a toast notification to further investigate
PROBLEM | SOLUTION
PROBLEM: One of the most common ways a private network may become infected is at the fault of external media being plugged into a computer connected to said network. Current methods offer cumbersome solutions for both the end user and system administrators alike.
SOLUTION: Our software offers a quicker and easier method of preventing these outbreaks by simply running in the background without affecting any hardware performances and also maintaining an up-to-date database with any and all latest malware signatures.
CONCLUSION
Bad USB devices are becoming a growing threat that is hard to combat but easy to avoid. With Operating System Customs, this allows everyday users to plug in any USB device into their work computer without having to worry about potential embedded malware.