Citrix Tactical Tool Belt - A Kroger Utility

by

Jeff Ayers, John Hibbs, Eric Burden

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

Jeffery Ayers	*4/28/2019*
Jeffrey Ayers	Date
John Hibbs	*4/28/2019*
John Hibbs	Date
Eric Burden	*4/28/2019*
Eric Burden	Date
Jim Scott	*4/28/2019*
Jim Scott, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
May 2019

Table of Contents

List of Figures

FIGURES

TABLES

ABSTRACT

With over 250,000 unique logins per day, Kroger Citrix requires significant administrative resources. Ticket counts periodically exceed the capacity of the Kroger Support Center (KSC) - most issues are simple but have time consuming fixes. To reduce overhead, this team intends to implement a tool for KSC members that will streamline functions such as: (1) resetting Citrix users' profiles, (2) verifying Citrix entitlements, and (3) viewing user Active Directory information. An additional goal is to enable KSC members to resolve an increased number of tickets without the need for escalation. This team predicts a recovery of approximately ten to fifteen minutes per incident, providing significant cost and time savings. Reduced overhead will allow more resources for investigating potential access violations and improving account controls.

1. PROBLEM STATEMENT

1.1 Introduction

A Fortune Twenty-Five company, Kroger has a yearly revenue of over 101 billion dollars employing approximately 449,000 employees. In order to service this large workforce, Kroger uses Citrix to deploy WorkCentre resources to thin clients and knowledge worker machines. On average, the data center receives over 225,000 unique logins to Citrix per day. Considering the large potential for problem tickets, the Citrix Tactical Tool Belt is designed to streamline troubleshooting efforts to reduce escalation and improve lead time. By automating simple troubleshooting tasks, the initial ticket recipient in the Kroger Support Center can handle most resolution tasks normally escalated to Citrix Engineers.

1.2 Project Description

The Citrix Tactical Tool Belt is a support tool help desk analyst can use to automate troubleshooting tasks normally conducted by Level 2 engineers. Users can access the tool by logging in through a Kroger domain site, then enter a unique user ID to request relevant information on the user and the status of their Citrix sessions. In conjunction with Citrix Director, there will be functions available that automate troubleshooting processes for various issues.

1.3 Problem

Citrix is used by almost every Kroger employee in some capacity. From the instore associates to the salaried office workers, Kroger's infrastructure handles over 225,000 unique logins every day. Desktop support teams often have limited access and knowledge of Citrix components, leading to most tickets escalating from Level 1 support directly to Level 2 and Level 3, teams that have significantly less manpower to handle the volume.

1.4 Solution

To mitigate escalations and reduce lead time, we will develop and deliver an application that will contain a suite of tools to handle everyday issues that occur in the Kroger Citrix environment. This application will be available to Level 1 Desktop Support to assist and streamline troubleshooting, as well as any additional teams that may benefit from the functionality therein. This will primarily be a utility to access troubleshooting features for Citrix issues including:

- Users unable to access applications
- Repairing permission issues related to files or access
- Adding, removing, and troubleshooting printer functions in Citrix
- Viewing User AD groups
- Viewing a user's entitled applications
- Enabling a quick and clear understanding and format user logs to quickly assess issues
- Viewing User sessions with tools for logging off and terminating the sessions
- Providing a function to reset Citrix profiles

1.5 Project Goals

This application should be easily understood by end users, designed to be modular, well documented, and easily modified by future Citrix Engineers. Typically, Kroger engineers have an understanding of Linux scripting and Microsoft PowerShell; this tool will be developed on the same platforms. Dependent on future testing, the program GUI will be built in PowerShell with a web-based application as a potential alternative. A "user-toolkit" section will contain functionality that automates common resolution tasks for issues such as: hung/stuck Citrix sessions, repairing missing printers, and clearing printer queues. In addition to the previous, the

tool will function as a quick AD and Citrix resource lookup tool where a user's access can be verified, ensuring they have the correct entitlements for the resources their role requires in Citrix.

1.6 User Profile

The Citrix Tactical Tool belt (CTTB) Team will produce a Citrix Utility that ensures the optimum virtual desktop experience is available to empower Kroger employees to provide customers with the highest level of customer service and products available.

1.6.1 Project Title

Citrix Tactical Tool Belt - A Kroger Utility

1.6.2 Potential Users

- Level 1, Desktop Support
- Level 2, Systems Administrators
- Level 3, Network Engineers

1.6.3 Software and Interface Experience

The Citrix Tactical Tool Belt (CTTB) is a suite of administrative tools for managing and correcting issues that occur in the Citrix environment. These features include repairing User Profiles, requesting Active Directory data, and capturing User Session logs. The CTTB is intended for frontline Level 1 Desktop Support: users are not expected to have advanced knowledge of the Windows Active Directory environment and Citrix's functions. This tool will allow less experienced support staff to resolve issues more quickly and with less escalations, automating most resolutions and reducing the workload Level 2 support currently performs.

1.6.4 Experience with Similar Applications

While there are no off-the-shelf solutions or applications available on the market today, technicians expect and desire useful features to be available and seamlessly implemented for the Citrix platform. This network-based application is designed to be simple, robust, and quickly apparent to potential users, simplifying workflows and creating value through reduced lead time.

1.6.5 Task Experience

- Manage Active Directory profiles
- Administrate User access to the Kroger Network
- Reset User credentials
- Communicate with Users to determine potential solutions
- Troubleshoot User Accounts and printer functions
- Establish entitled application access
- Provide logs and information collected to Level 2 with escalated tickets to reduce redundant work

1.6.6 Frequency of Use

An estimated 1-3% of 200,000 daily logins are prone to issues with a potential range of 2,000 to 6,000 tickets per day. Active monitoring of changes and lead time will provide metrics to evaluate the effectiveness of the tool and extrapolate cost savings.

1.6.7 Key Interface Design Requirements That the Profile Suggests

This tool should be available whenever a ticket is submitted for Citrix profiles. This could happen daily, weekly, monthly, or yearly, depending on frequency.

- Repair permission issues related to files or access
- View a User's Active Directory groups
- Add, remove, and troubleshoot printer functions in Citrix

- View User's entitled applications
- Reset Citrix Profiles
- Logout of all sessions

1.7 Use Case Diagram

Figure 1: The project use case diagram. This utility will assist Level 1 (Helpdesk) complete most tasks independent of Level 2 and 3 (Citrix Engineer).

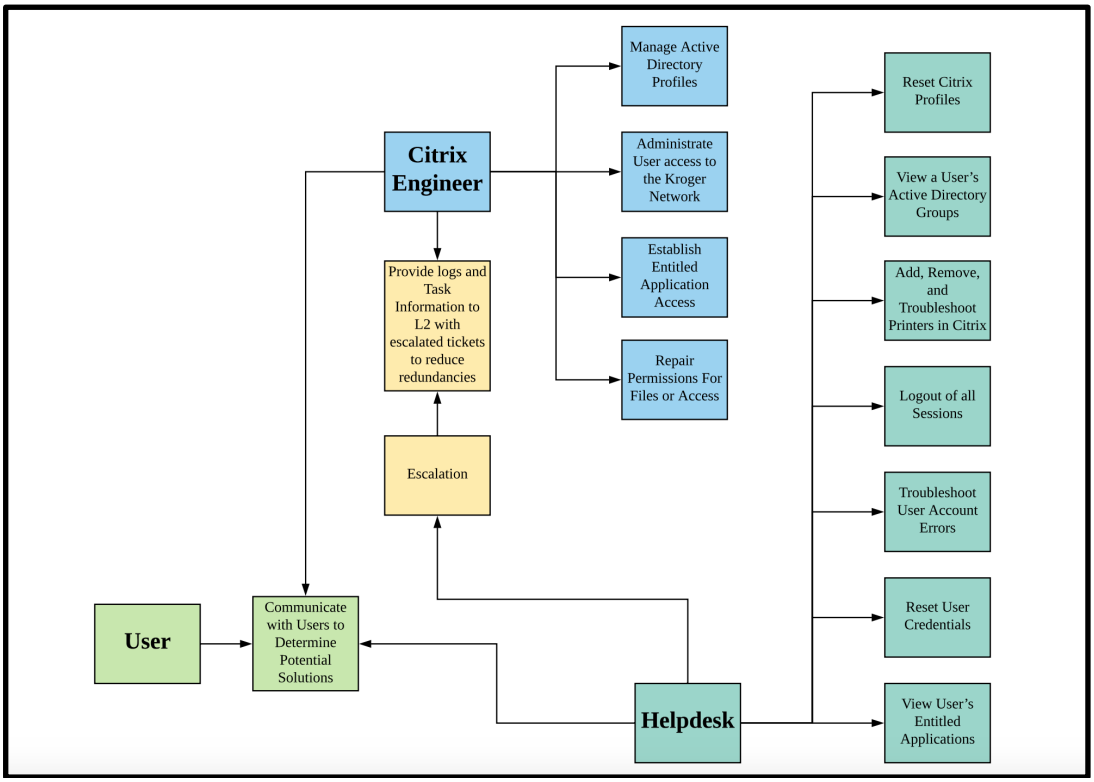

Figure 1. Use Case Diagram

2. PROJECT MANAGEMENT

2.1 Budget

CTTB's cost includes the labor and infrastructure required to build, test, and maintain the solution. The Citrix environment the tool will assist management of is outside of the scope of this figure. Initially, two out of three of the CTTB team interned at Kroger at ~$16.00 per hour, but during the development of the CTTB the company asked we develop the tool during off-hours as a personal project. More recently Kroger has shown additional interest in retaking ownership of the application, after the proof of concept performed better than expected.

Table 1: The budget presents costs for the development of the Citrix Tactical Tool Belt, including testing, hosting, and development.

Item	Unit Multiplier (Hours/Amount)	Unit Price	Line Item Total
Hardware			
Server	1	$2400	$2400.00
Project Planning/Management/Admin			
Labor	60	$16	$960.00
Development/Testing			
Labor	150	$16	$2400.00
Total			**$5,750.00**

Table 1: Budget

2.2 Objectives/Deliverables

The project deliverables and deadlines are presented in ***Table 2: Project Objectives/Deliverables Due Dates***.

Major Project Milestones	
Fall 2018	
Stakeholder Requirements Meeting	09/05/2018
Paper GUI Framework Testing	10/02/2018
Web UX Prototyping	10/17/2018
Client UI Prototyping	10/26/2018
Spring 2019	
Develop Backend	01/10/2019
Develop Frontend	01/21/2019
Alpha Testing	01/30/2019
Final Security & Pen Test Audit	03/22/2019
Live Deployment Testing	03/26/2019

Table 2: Project Objectives/Deliverables Due Dates

2.3 Project Schedule

Below are the projects schedule of deliverables and major milestones. This includes the Gantt chart and the WBS.

2.3.1 Gantt Chart

Figure 2: Gantt Chart presents our project schedule with major milestones listed. This chart list milestones for fall 2018 and spring 2019.

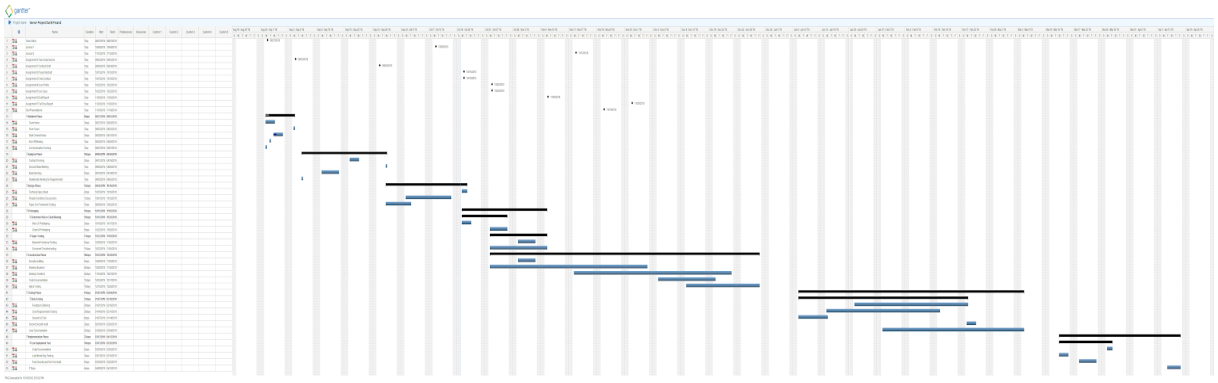

Figure 2. Gantt Chart

2.3.2 WBS

1. Initiation Phase
    1.1. Senior Design Team
        1.1.1. Team Intros
        1.1.2. Communication Forming
        1.1.3. Kick Off Meeting
        1.1.4. Start Contract Ideas
        1.1.5. Form Team
2. Analysis Phase
    2.1. Stakeholder Meeting for Requirements
    2.2. Brainstorming
    2.3. Contract Development
    2.4. Second Stake Meeting
3. Design Phase
    3.1. Paper GUI Framework Testing
    3.2. Finalize Functions Discussions
    3.3. Technical Spec Sheet
    3.4. Prototyping
        3.4.1. Determine Web or Client Meeting
            3.4.1.1. Web UX Prototyping
            3.4.1.2. Client UI Prototyping
        3.4.2. Super Testing
            3.4.2.1. Document Troubleshooting
            3.4.2.2. Backend Functional
4. Construction Phase
    4.1. Develop Backend
    4.2. Security Auditing

- 4.3. Develop Frontend
- 4.4. Code Documentation
- 4.5. Alpha Testing

5. Testing Phase
   - 5.1. Beta Testing
     - 5.1.1. Second UX Test
     - 5.1.2. User Requirements Testing
     - 5.1.3. Feedback Gathering
   - 5.2. User Documentation
   - 5.3. Second Security Audit
6. Implementation Phase
   - 6.1. Live Deployment Test
     - 6.1.1. Last Minute Bug Testing
     - 6.1.2. Final Security and Pen Test Audit
     - 6.1.3. Code Documentation
   - 6.2. IT Expo

3. TECHNICAL ELEMENTS

3.1 Network

Kroger's network consists of several tiered components spread over two redundant data centers, the HDC and CDC. Each data center houses several thousand hosts and infrastructure components, ranging from application servers to multitiered switches. Citrix is hosted mirrored hosts inside these data centers.

3.2 Application

Citrix delivers client solutions that offer secure application hosting and virtualization services to large corporate enterprises to reduce costs and enable effective management at scale. This tool will alleviate a significant amount of manual work for Level 1 Helpdesk necessary to maintain high productivity for Kroger's in-store and remote workforce, by automating simple or repetitive tasks to reduce ticket lead time and improve efficiency.

3.3 Technical Discussion

CTTB at its core will have a login page that verifies credentials and grants access to the main application. The main menu will show a textbox to input the desired Enterprise User Identification (EUID) for resolution. Once entered, a PHP console will run several PowerShell scripts using the given EUID. These scripts will accomplish two primary functions:

The first PowerShell script will query Active Directory for the user's EUID AD information and memberships, then store that data into an array which will then be passed back to the PHP console where the AD membership information will be displayed in the left column, with user information presented in the center column.

The second PowerShell script queries the Application Delivery Controller for all enabled applications the user has permissions for. The values are stored in a second array and passed back to the PHP console to be displayed on the main page in the right column.

Figure 3: Search Function shows how the search function is processed.

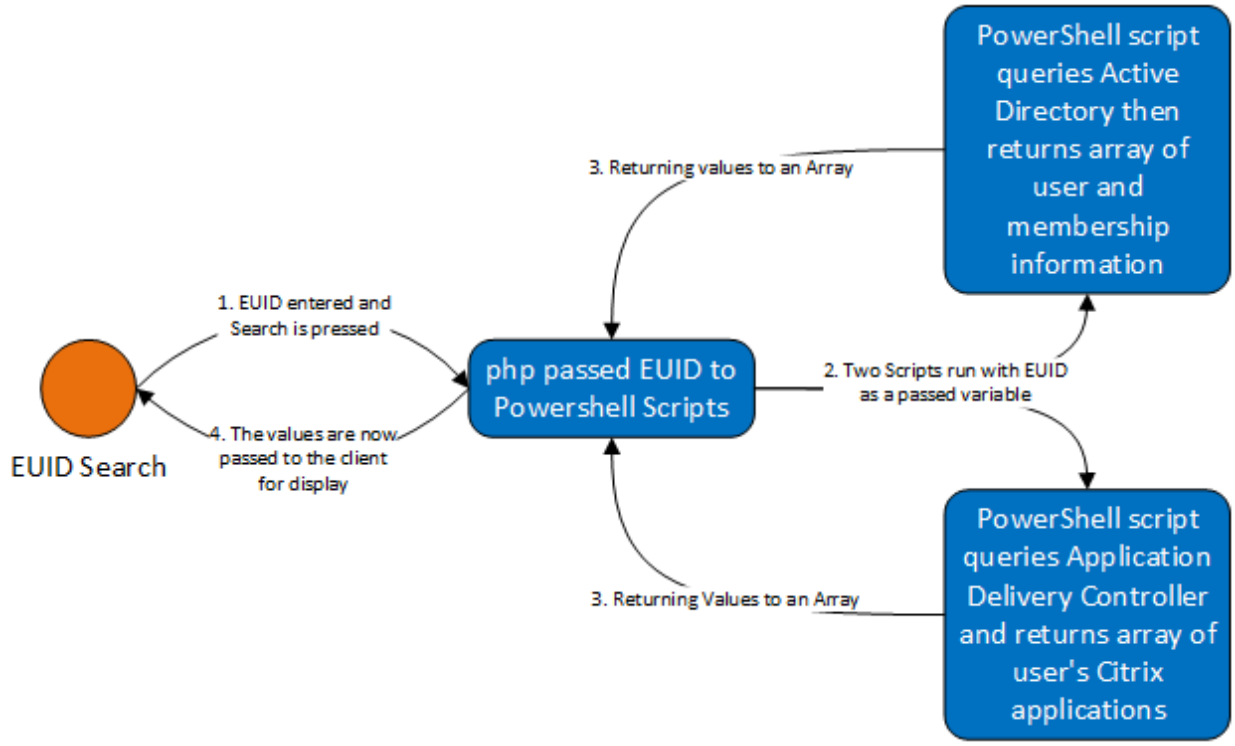

Figure 3: Search Function

Figure 4: Reset Profile Function: shows how the reset profile function is processed.

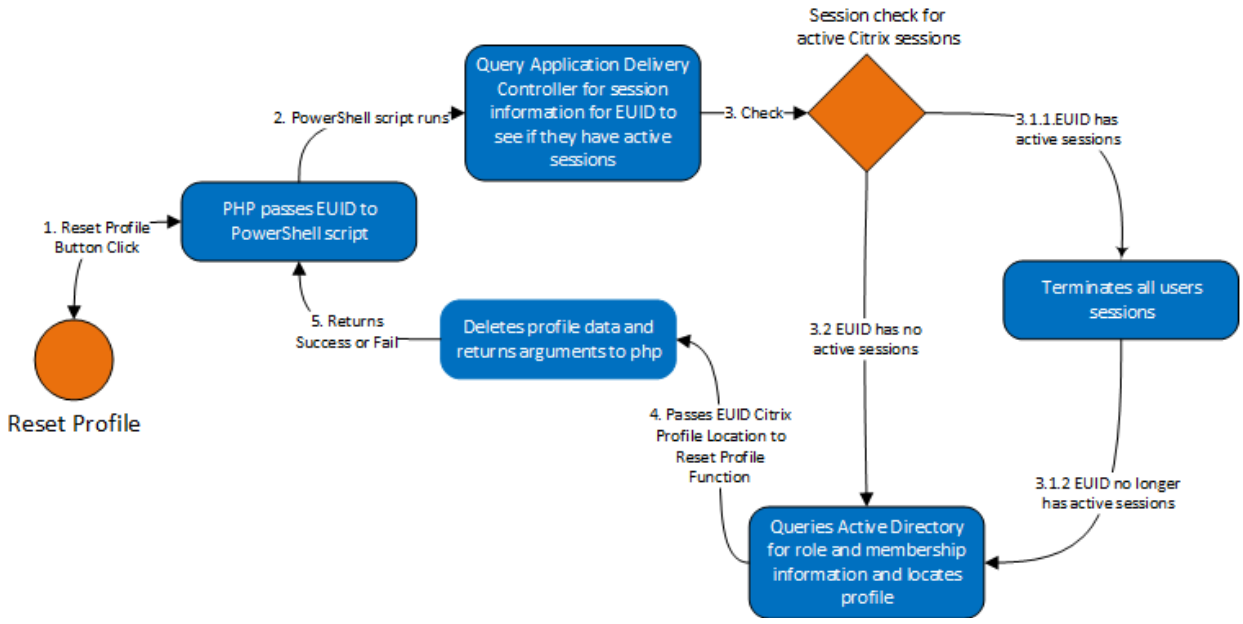

Figure 4: Reset Profile Function

3.4 Technical Architecture

Figure 5: Technical Architecture presents our technical architecture.

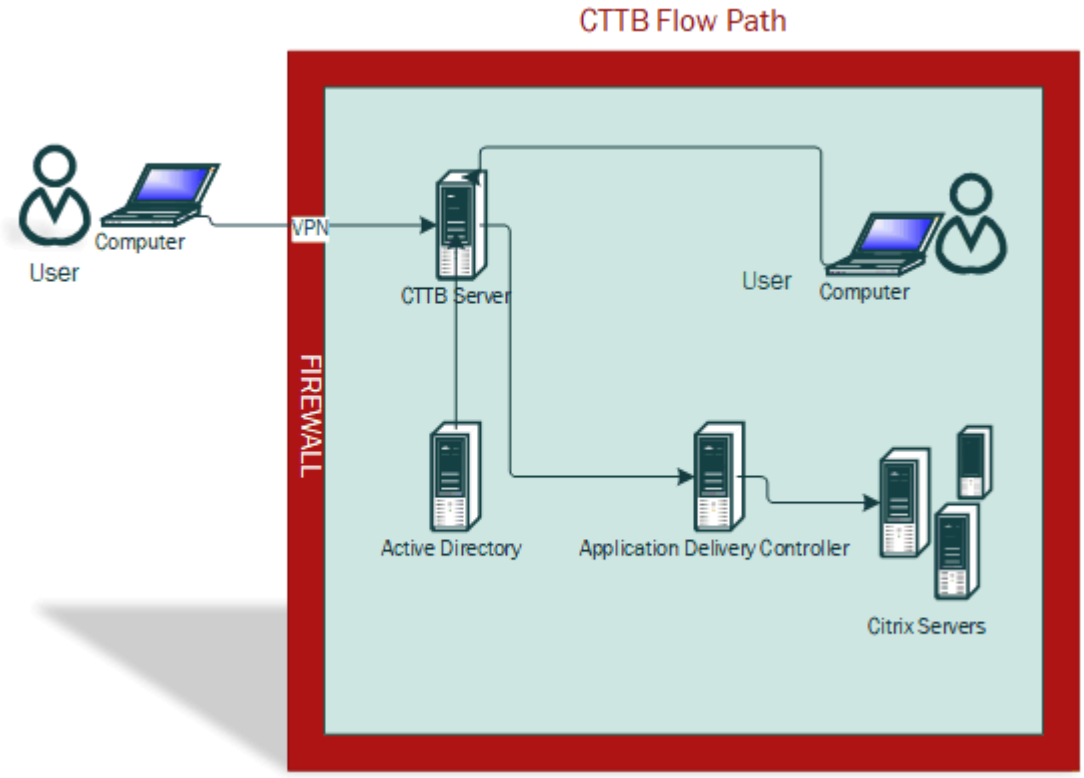

Figure 5: Technical Architecture

3.5 Testing

3.5.1 Testing Methodology and Approach

Testing consisted of several rounds of security and input testing to verify the program's functions were working as intended and provided adequate security measures to prevent malicious access. User testing is still ongoing, and the general methodology of the user interface is "simple and functional." The UI is minimalist and provides feedback at various stages to indicate what actions have been taken and what actions are available. Security testing included different methods of code and input attacks for various stages of development, including SQL injection, PowerShell scripts, malformed data, cross-site scripting, and html code-injection.

Security Testing In Depth

Kroger has a very robust internal network that is impervious to most general attacks like DDOS and compromised permissions. Additionally, some of the testing we intended to perform was limited by our access to the Kroger environment: for example, port scanning on the Kroger network was strictly forbidden. There are some known attributes of the datacenter and the network to consider; generally, Kroger's physical hosts are hardened with a custom ESXi image, separate VLANs exist for every function, and the application will be in a cloud environment on a separate segment isolated from different applications.

Concerning the security of a remote session on a webapp, when controlling for versioning and user access our initial design with an on-system solution was far less robust. A dedicated app requires significantly more upkeep; updating every system the application is installed on requires more time and resources and may not be 100% accurate. Additionally, coding any dependencies or hostnames into an application to "call home" is an added risk vector. With a hosted app, ensuring the user's webapp is always up to date is as simple as pushing an update to a web server. The code that is exposed to the user is very small - only a small front-end webpage and any user data that is requested from Active Directory. Users can't download the application and the scripts that enable its functions to a USB and get critical infrastructure data like DNS names or IP addresses because they are never exposed to the user.

3.5.2 Test Plan

Our plan includes user testing for interface design, requested features, the quality of the services provided, and any additional requests and functions that may be desired.

The Kroger Support Center will be the focus of the test, as the primary users for the application. Considering the weight of feature requests and time investment, discussions with management on process flows and on-the-ground use will have the most impact. Three separate tests will be conducted:

The first will be with Level 3 engineers to confirm the functionality of the tools and verifying that the desired output is received. Any obvious coding mistakes will be apparent here.

The second round will consist of a small group of KSC members to test for user interface experience and expectations, as well as requests for additional features. This test will focus on how the user interacts with the tool, how quickly they grasp the features of the tool, and if there are any missing features they need.

The final test will consist of a predetermined list of security testing, approved by the Kroger infrastructure team.

3.5.3 Test Results

The test consisted of two Citrix engineers, one senior Level 1 analyst, and two junior support analysts. The UI and some other features were adjusted to accommodate the requests of our testers. The most requested feature by the analysts was to add a one-click option to clear the EUID box. Our solution was to add a button and an “X” icon on the right side of the text box; this was positively received by the testers, and they ultimately opted for the “X” icon inside the textbox.

During the two rounds of user testing, we found several vulnerabilities in our error-checking code when submitting an EUID for a disabled Active Directory user. To

correct this, we added an integrity check to verify the “enabled” value was set to “true” when searching for an EUID.

3.6 Future Recommendations

The need for a tool like the CTTB at Kroger was very clear from the beginning. What was not clear initially was how users would receive the tool, interact with it, or how the team would create and execute its functions. If the team was presented with a chance to do it all over again, we would spend more time researching other tools and ask for feedback from other developers. The team worked with the tools we knew at the time, and while these tools could technically complete the tasks, there were other more efficient or effective options readily available. This change is apparent with our move from strictly a PowerShell program to a web-based application.

If we had more time, we would continue to add features to the app that were out of our original scope. For example, we are still working on the printer function to allow KSC to add them for a user through the web interface. We would also consider using ReactJS; while we started to convert our application to it, we ran out of time and didn’t feel confident presenting the changes we wanted to make yet.

Initially the application was going to be entirely built with Microsoft tools, like .NET, that would open through Citrix. The Citrix manager indicated there would be the potential for failure if the troubleshooting tool for Citrix was hosted within Citrix. This realization lead us down the webpage path, which had different challenges. We later reached out to other web developers and using their feedback opted to pursue Materialize and other more modern tools.

Our final plans for this project are to keep developing it, finalizing the stretch goal features, and working to automate more Level 1 troubleshooting procedures to recover additional

time. Finally, we are researching methods for adding and removing environments from the tool to increase the portability of the tool.

4. USER INTERFACE

4.1 User Interface

Below are screenshots of the login page and landing page for the CTTB's user interface.

Figure 6: User Login Page is our login page for the Citrix Tactical Tool Belt.

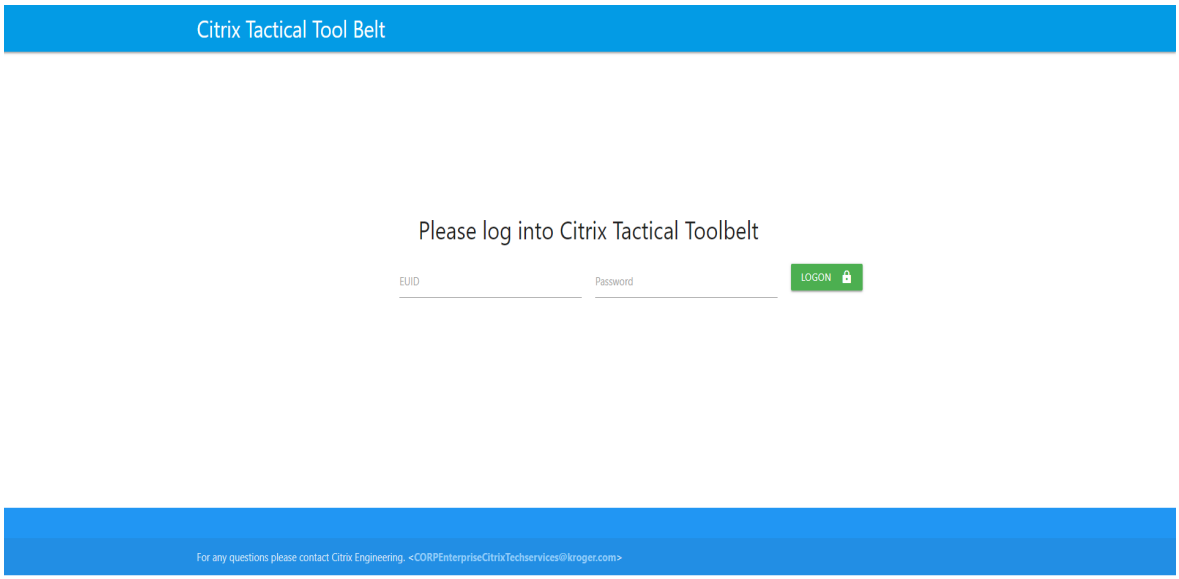

Figure 6: User Login Page

Figure 7: User Interface is our landing page after login.

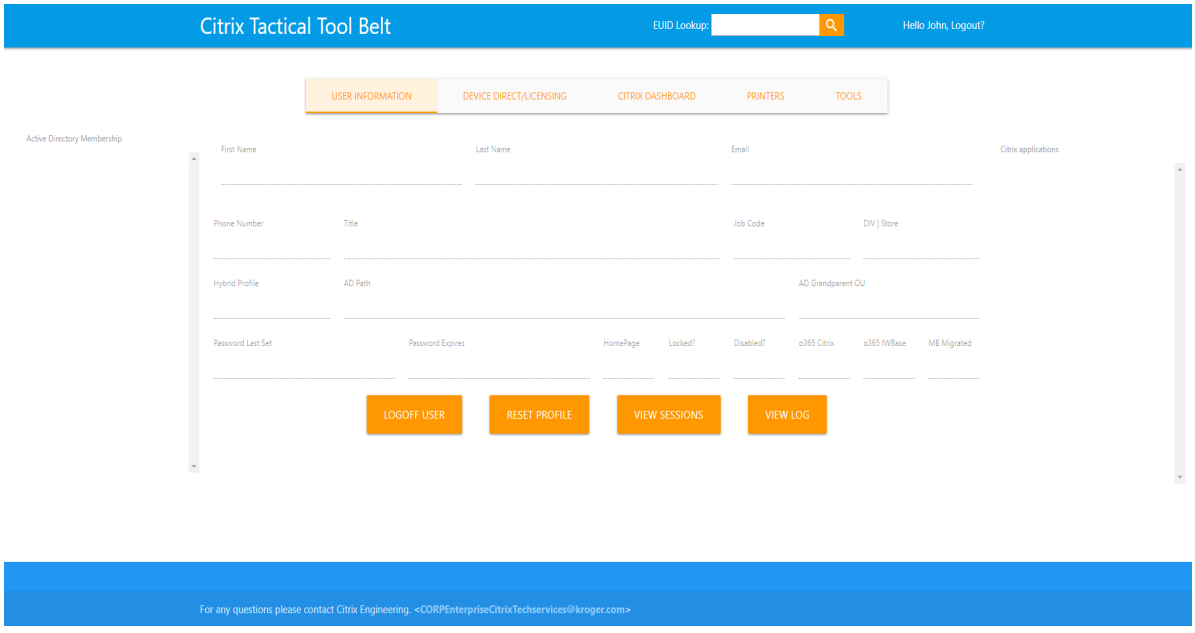

Figure 7: User Interface

5. CONCLUSION

5.1 Fall Semester 2018

During the Fall Semester we had to dust off our books and projects from our UX, Agile, and Project Management classes. To plan out our path for the Citrix Tactical Tool Belt, we needed to understand what it is and why it is needed. There were some setbacks throughout the planning stage. Setting time aside for exploring other options also consumed some time. We were able to backtrack without issues and adjusted accordingly.

We had to pick up on PowerShell to automate the tasks in the background, including modifying files in a profile directory, accessing Active Directory and returning information from it, and opening other programs to send them commands for certain functions. PowerShell was new to the whole team and wasn't covered in any of our classes, so the process was challenging.

Our largest roadblock was a change in frontend access. Initially we had planned to do the whole app in PowerShell, including the interface, hosting it from Citrix. This was later shot down due this becoming a liability in the event the analyst's Citrix access was also compromised, leading to no access to the Citrix Tactical Tool Belt to proceed with fixing the issue. So, our team went back to the drawing board and decided to make a web front-end.

5.2 Spring Semester 2019

During the Spring Semester, the team was tested for skill and endurance. We spent significant time researching alternative tools and asking for feedback from other developers, discussing the tool between ourselves and our advisors. Initially the application was going to be entirely built with Microsoft tools, like .NET, that would open through Citrix. After deciding to not host the tool directly on Citrix, we focused on the webapp development path, which had different challenges.

Our final plans for this project are to keep developing it, finalize our stretch goals, and automate more Level 1 troubleshooting procedures. The team also plans to continue to add features to provide more tools to the Kroger Support Center.

After our research, we pursued Materialize and PHP over our initial design. As a team, we have grown in our ability to communicate and understand user needs. Pushing the boundaries of the tools available to us showed how technological limitations can lead to frustration. Even with the three of us separated by different schedules, work, and school, we collaborated on a great Senior Project. We met all our goals and even managed additional stretch goals.

Our last stand was at the IT Expo. The team agrees we successfully delivered our product, and many commented that it would be a great tool at their company. We can be proud of the effort we put in.

References:

Wang, A., Chang, A., Mark, A., & Louie, K. (n.d.). Materializecss. Retrieved January 22, 2019, from https://materializecss.com/

JS Foundation. (n.d.). JQuery.ajax(). Retrieved January 22, 2019, from https://api.jquery.com/jQuery.ajax/

Gołębiowski, A. (2016, December 18). XenApp 6.5 powershell commands cheat sheet. Retrieved November 10, 2018, from https://blog.citrix24.com/xenapp-6-5-powershell-commands-cheat-sheet/

Fisher, J. D. (2019, January 08). PowerShell Cheatsheet - Essential Commands You Can't Live Without. Retrieved November 10, 2018, from https://www.comparitech.com/net-admin/powershell-cheat-sheet/

SCHÄFERHOFF, N. (2018, November 13). PHP Cheat Sheet (.PDF Version Included). Retrieved January 22, 2019, from https://websitesetup.org/php-cheat-sheet/

Malik, R. (2016, December 13). Executing PowerShell using PHP and IIS. Retrieved January 22, 2019, from http://theboywonder.co.uk/2012/07/29/executing-powershell-using-php-and-iis/

Schwalbe, K (2015). *Information Technology Project Management.* Boston, MA: Course Technology

Appendix A: Technology Utilized

Appendix B: Poster