Mismanaged Credential Finder

by

Kevin Kadakia, Michael Tan, and Richard McDonald

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

© Copyright 2021 Kevin Kadakia, Michael Tan, and Richard McDonald

The authors grant permission to the School of Information Technology
to reproduce and distribute copies of this document in whole or in part.

Kevin Kadakia	04/26/2021 Date
Michael Tan	04/26/2021 Date
Richard McDonald	04/26/2021 Date
Yahya Gilany, Faculty Advisor	04/26/2021 Date

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

April 2021

TABLE OF CONTENTS

LIST OF ILLUTRATIONS

ABSTRACT

Knowledge bases are a growing industry due to the information required in the background to support multiple applications and daily tasks for employees. These knowledge bases are filled with an abundance of information. This information could include passwords, keys and other credentials that could be critical for an organization. According to the Verizon Data Breach Investigations Report, 78% of attacks on web applications involved the use of stolen credentials with credentials and personally identifiable information as a target. With the Mismanaged Credential Finder, this application is looking to find and help remedy credentials that are in these knowledge bases. By doing so, this application is looking to help IT professionals manage their knowledge base and enforce cybersecurity policies. In doing so, our goal is to minimize risk due to a knowledge base breach for an organization.

1. INTRODUCTION

1.1 Problem

In cyber security, data breaches are one thing that is feared by all professionals. According to Verizon's 2020 Data Breach Investigations Report (DBIR), 78% of attacks in North America involved using stolen credentials, with the target of such breaches primarily are personally identifiable information and credentials. With the rise of knowledge bases backing the technical support side of organization, mismanaged credentials start to become a concern. When a user creates knowledge bases, it is a standard to include relevant information about their systems and their management to help troubleshoot problems that arise. However, due to this ease of access, people may decide to add critical information such as passwords, keys, and other credentials to their systems within this knowledge base. If this knowledge base were to be breached, this information could be used to further damage an organization.

1.2 Solution

Mismanaged Credential Finder or MCF for short will be an application with the ability to scan and search through webpages associated with a specific knowledge base. When running the scan of a knowledge base, this application will analyze the webpages and find texts that could be considered a credential. Credentials include, but are not limited to passwords, server access keys and credit card information. After the scan runs it will then generate a readable report for the user to help point out credentials, location, and possible remediations. This application will be targeted towards the security and audit professionals to help enforce security standards.

1.3 Project Goals

Our Project goal for the Mismanaged Credential Finder was to help promote the security policies in knowledge bases. This included building a web application and a one-of-a-kind scanner that will allow users to scan for credentials in any knowledge base. Other aspects of this included a reporting feature that will notify the users of any credentials that are placed into a knowledge base, along with a scoring system that will help organizations determine their security readiness.

1.4 Overview

Throughout this final report, there will be information on how the project was completed. The report includes in-depth processes and includes the following sections: design objectives, methodology, budget, timeline, problems encountered, and future recommendations.

2. DISCUSSION

2.1 Project Concept

The concept of the Mismanaged Credential Finder was created on the idea that security should not be sacrificed for convenience. Because knowledge bases could contain credentials for logins and sensitive and limited access items, this becomes a security issue for any organization that relies on these knowledge bases. For that, the Mismanaged Credential Finder is responsible for finding these credentials so that security professionals in these organizations can have them removed and stored in more secure and controlled locations.

2.2 Design Objectives

Our team's design objectives started out by looking into current solutions, which none seemed to exist other than a few GitHub repositories that pertain to scraping keys and passwords, however none of them tackled the idea that we were attempting to make. We had decided that we would design our own application with inspiration from these repositories. We opted to use coding languages like Python with its vast web scraping library of Selenium and Beautiful Soup that will allow us to read web pages. We later added HTML, CSS, and Bootstrap to help build our frontend which will serve our users.

2.3 Methodology and Technical Approach

Due to the nature of this project, there was no underlying code to be built upon, which meant it had to be all built from scratch. In our case, we broke our project down into separate

components that we would need to make. The components are the frontend which includes the web application that will be facing the user, and the backend which will be running the scan and finding all the sensitive information. The frontend will consist of a greeting page, scanner page, and a dashboard.

The greeting page is a basic introduction to what this application is and what it does. The scanner page is where all the information is gathered from the user to run a credential scan. This information includes the knowledge base URL, type of knowledge base and whether authentication is required. Finally, the dashboard displays the results in a simplified and easy-to-read way with charts and scan highlights, while also allowing the user to generate a report containing detailed information about the scan.

The backend is further broken up into different parts to develop this application from the ground up. The first part of the backend is a web scraper that will connect to a knowledge base gathering all information such as a list of all pages belonging to the knowledge base and their content. The second part is an analyzer that will determine if a string is a credential. The last part is an authentication portion that will login into the knowledge base as required.

2.4 User Profiles

There will be two types of users interacting with our web-application, Mismanaged Credentials. The first set of users described in Figure 1 are the Cyber Security Professionals who will mostly be running the application. The second set of users in Figure 2 are the Web developers who will be maintaining the website after its creation.

Figure 1 User Profile 1: Cyber Security Professionals User Profile

User Profile Form 1
Application: Mismanaged Credential Finder Scanner
Potential Users: • Cyber Security Professionals
Software and Interface: • The security professionals should be used and familiar with the credential finder scanner application. They also should be familiar with coding used in the creation of the application.
Experience with Similar Applications: • Cyber security professionals should be familiar with the programming languages used in the creation of the application, also have the familiarity of GitHub and cloud storage services
Task Experience: • The user should be able to interact with the application above using keyboard and mouse. This application should also be easy to navigate to run a report on credentials and access other functions.
Frequency of Use: • The frequency of use of this application depends on the company of whoever is running this application. Or they can even have it set up to auto run the application and have it set so that the application will run every day at the same time and just

have an email sent automatically to the person in charge.

Key Interface Design Requirements that the Profile Suggests:

- This user will need to be able to work with the programming languages and interfaces to create and maintain this application and keep up with the new languages.

Figure 2 User Profile 2: Web Developers and Administrator User Profile

User Profile Form 2

Application:

Amazon Web Services (AWS), Python, HTML/CSS, Linux/Windows Server.

Potential Users:

- Web Developers and Administrators

Software and Interface Experience: Users should be experienced in the use of the software development environments, database management software, and multiple services within Amazon Web Services.

Experience with Similar Applications: Users should have a familiarity to similar cloud storage services, OSes, and programming languages.

Task Experience: Users should be experienced in the applications listed above for the creation and continued maintenance of the application.

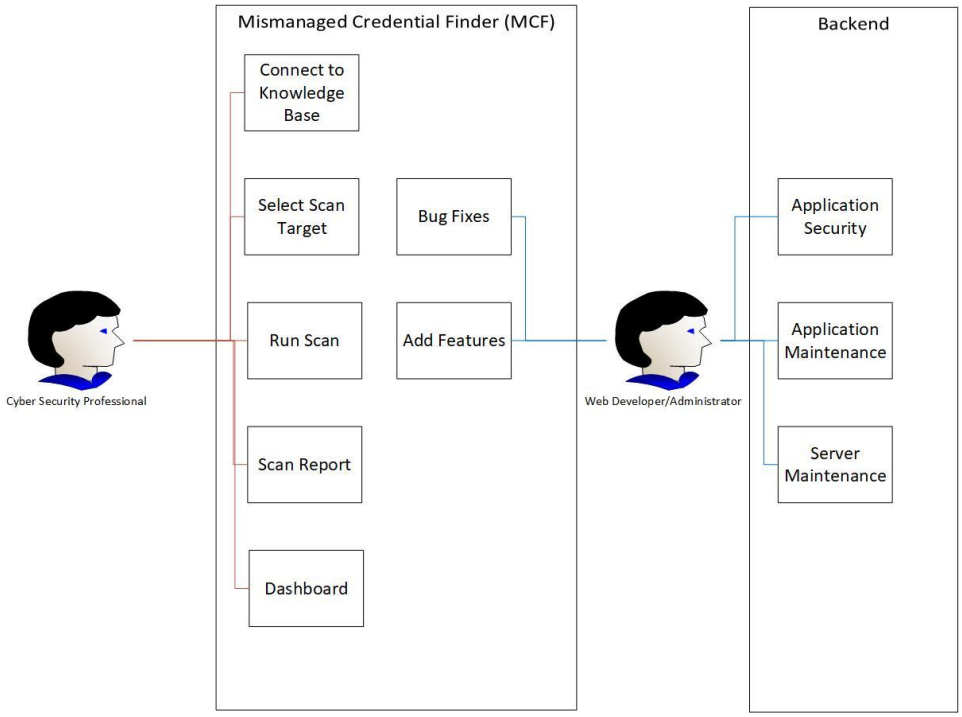
Frequency of Use: Use will be frequent during the duration of the development of the application and then will only become periodical and as needed to resolve both security and general bug issues.

Key Interface Design Requirements that the Profile Suggests: Users should be able to use the listed programming languages, databases, cloud services, and operating systems to create a functioning and well-maintained application.

2.5 Use Case Diagrams

The use case diagram, depicted in Figure 3, demonstrates the use case for Mismanaged Credential Finder with two users, the Cyber Security Professional and Web Developer/Administrator with their corresponding tasks.

Figure 3 Use Case Diagram



2.6 **Technical Architecture**

Mismanaged Credential Finder or MCF was built through three distinct parts, the frontend webpage, backend scanner and analysis, and infrastructure.

To build the frontend for the application, HTML, CSS, JavaScript, and a Bootstrap template were used to create an interactive home webpage along an UI that allows a user to navigate and interact with the application.

The base code for the application was written in Python along with the usage of Flask. With the use of Python and Flask, it allows the application to be hosted as a web service and interact between the multiple parts of the frontend and backend. For the scanner and analysis, web scrapers such as BeautifulSoup and Selenium were used. Using Selenium allows MCF to interact with a given knowledge base as it would a normal computer. With the addition of BeautifulSoup, it gives additional functionality to extract all the text data on a given webpage to feed into the analyzer. The analyzer uses pattern recognition all coded within Python. Regex was used to match text and requirements to match credentials. For example, “Password=”, “Begin RSA Key”, and the length of numeric characters for financial information. In addition to simple pattern recognition, a technique known as Entropy Analysis was also implemented. This uses math to assign BASE64 and HEX Characters values that will then be inserted into a math equation to determine and entropic value. If this value exceeds a certain threshold, it will be marked as a credential.

The infrastructure for MCF utilizes services provided by Amazon Web Services or AWS. The core component of that was used is the Elastic Compute Cloud or EC2 service. This service allows for the creation of cloud hosted virtual machines that can be used for cloud computing needs. With this service we created Amazon Linux servers to host our application and running

off Nginx, we were able to push it to the web so that the application can be accessed and used.

To further support the application, an Application Load Balancer or ALB was also created so that multiple servers can be used to host the application and provide redundancy. Finally, an Auto Scaling Group or ASG was created and associated with the ALB so that when servers are created, they are automatically added to the load balancer. The ASG creates servers until it reaches the set minimum amount of server for that group and will build them with a set template.

The template that was used was the Amazon Linux OS and when it launches it runs what is called a launch script that automatically configures the server and will also install and start the application on the server. The ASG also has a set maximum and when it goes over, it automatically terminates the oldest server, allowing for deployment of newer servers with newer versions of the application without any down time. Conversely if the ASG goes below the minimum a new server will also be created. This ensures that there is little to no downtime for the application.

2.7 Testing

The testing section will discuss the methodology, scope, objectives, and results in the user acceptance testing process. Through the testing, friends, family members and coworkers were used to test features that the primary users will be utilizing.

The approach to testing will include friends, family members and coworkers in and out of the professional field of IT. This approach was decided as it would be allowing the team to get the most information and testing done within a short amount of time. By using this test group, our team can review the testing information as soon as testing has been completed, along with

feedback and comments during the testing on items that can improve upon.

Through user acceptance testing, the tests will revolve around the functionality for the two use cases revolving around the Security Professional and Web Developer/Administrator. For the use-case around based around the Security Professional. The testing will include the scanning and reporting functions. This is a core aspect of the application and will be used on the daily basis. While not as commonly used, testing was also conducted around the use-case of the Web Administrator. This part of the testing ensures that the service runs on the server and is scalable for an organization's needs.

Our team is looking to ensure that the major features of the application are accounted for and to ensure that all functions are working as intended for all users. Any bugs that exist and have been found during the testing process will be resolved before the IT Expo.

The following figure, Figure 4, shows the test logs of the user assurance testing. It marks all the procedures and results encountered during the testing.

Figure 4 Testing Results

Record #	Test Case #	Input	Expected	Actual Output	Result (Pass/Fail)	Reason For Result	Date
1	1A	Login to MCF using provided credentials	User Login Successful, directs to homepage	Error in login. User does not pass login screen	Fail	User was not able to login due to login not communicating with the backend database	2/1/2021
2	2A	Initiate Scan with Type "Misc."	Scan starts running and notifies user when completed	Scan starts running and redirects users after Scan is complete	Pass	Scan was able to run and complete successfully on a non-credential knowledge base.	2/7/2021
3	2B	Initiate scan with type Confluence and with credentials	Scan starts running and notifies user when completed	Scan starts running and redirects users after Scan is complete	Partial Pass/Fail	No Visual Indicator such as timer of scan time. "Scan is Running" would show even when it was not running.	2/7/2021
4	2C	Initiate Scan with type Confluence and with incorrect Credentials	Notification of Failure	Scan completes and does not notify user of incorrect credentials	Fail	Scan does not notify user of incorrect credentials nor does it check for correct credentials.	2/7/2021
5	3A	Generate Report	Report generates and credentials are shown	Report is Generated and found credentials are shown	Pass	Report is generated successfully	2/8/2021
6	4A	Create new server with application ready	New server with application already set up and running	New server with application already set up and running	Pass	Correct set up of Auto Scaling Groups in AWS with auto launch configuration script that runs.	2/24/2021
7	4B	Stopping Server	Stop/Delete an old server so a new one can take its place.	Sever was stopped and deleted, with a new server automatically taking its place	Pass	Auto Scaling Groups allow one to set a limit for the number of servers in it, going over deletes the oldest one.	2/24/2021

2.8 Budget

The budget (Figure 5) illustrates the financial requirements of Mismanaged Credential Finder. When calculating the total cost of the project we factored in the software and labor costs.

Figure 5 Budget

Mismanaged Credential Finder Budget				
NO.	ITEM	UNIT, HOURS	UNIT PRICE	TOTAL (Yearly)
SOFTWARE				
1	Amazon Web Services	1	$20 (Monthly)	$240
	Subtotal			$240
LABOR				
2	Website Development and Design	256	$20 (Hourly)	$5,120
3	Software and Backend Development and Design	256	$20 (Hourly)	$5,120
4	Infrastructure Development and Design	256	$20 (Hourly)	$5,120
5	Maintenance and Support	2,080	$20 (Hourly)	$41,600

	Subtotal			$56,960
	Total			$57,200

2.9 Project Timeline

The project timeline (Figure 6) is a visual representation of the team's schedule over the duration of two semesters.

Figure 6 Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Project Scope and Planning	1 Week	09/28/2020	10/05/2020
2	Project Design	1 Week	10/05/2020	10/12/2020
3	Development	9 Weeks	10/12/2020	12/14/2020
4	Semester Presentation (Fall 2021)	1 Day	11/30/2020	11/30/2020
5	Catchup & Break	4 Weeks	12/15/2020	01/10/2021
6	Maintenance & Revisions	10 Weeks	01/11/2021	03/21/2021
7	User Acceptance Testing	4 Weeks	02/01/2021	02/28/2021
8	Semester Presentation (Spring 2021)	1 Day	03/22/2021	03/22/2021
9	Final Preparations for IT Expo	1 Week	04/05/2021	04/12/2021
10	IT Expo	1 Day	04/13/2021	04/13/2021

2.10 Problems Encountered and Analysis of Problems Solved

One of the first problems our group had run into was simply how to build the analyzer and how to make it efficient. One of these solutions was the use of entropy analysis, but because

of the limited knowledge of our group, we had initial problems in trying to figure out how to build our own entropy analyzer. Luckily, we were able to find a few GitHub Repositories such as TruffleHog and YAR that allowed us to investigate how others are running their scans.

Another problem we had run into was the authentication piece of the project. It was problematic because each knowledge base and their system is different from each other. We were able to investigate the pages through the effort of Google searches and view the source code to develop a way to authenticate with a knowledge base.

Lastly, we had issues with using and getting the application deployed on AWS. The deployment issue mostly revolved around getting the application on a web server, having the application be pushed to the web and finally having it working while on the web server. To get it hosted we needed to use Nginx for Flask to work and run the code, and then for the Selenium to work we needed to get Google Chrome installed on the server along with the Google Chrome Driver. Finally, after all that we had to figure out how to create a launch script that could configure the server, pull the application off GitHub, and then automatically start the application. This took several tries and version before it all was working and was ready for the ASG template.

2.11 Future Recommendations

If our group were to recreate our project from the start, our group would start out with getting a server set-up first and ensure that we had a testing environment before going forward. Along with this, we would like to increase the amount of time spent developing and testing our product. Such as the development and testing of a log-in database that would be able to authenticate users. Some additional features that could be added to our project could include

storing previous results and allowing users to customize their experience through rules. Such as excluding certain results or allowing the user to print off a PDF or EXCEL document of the results.

3. CONCLUSION

3.1 Lessons Learned

Throughout this project, we have learned a lot about the development and the implementation of an application. This included choosing the correct technologies and code to bring together to make our application work. There were also lessons learned on the side of project management and the idea of how time management, group work allocation and other team/group related topics were heavily used during our project.

3.2 Abilities and Skills Developed Throughout Project

We developed skills such as problem solving and time management which helped us in terms of developing our application. While working on this project, we came across new technologies which gave us the ability to learn more and obtain a deeper understanding of them. We received exposure to code, such as Python and Python related add-ons including Flask and Beautiful Soup. We also were exposed to servers and cloud services, such as Amazon Web Services.

BACK MATTER

References

1. Version (2020). *2020 Data Breach Investigations Report.* Verizon. https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf