System Requirements Specification

for

CyberTool

Version 4.0 approved

Prepared by Nicolas Rodriguez, Jeremiah Webb, Olivia Meholic, Sarah Gleixner, Joseph Alesandrini, Troy Neubauer

Embry-Riddle Aeronautical University

February 5th, 2024

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Group	09/29/23	Creation	V1.0
Group	10/18/23	Revision	V2.0
Group	11/16/23	Revision	V3.0
Group	02/05/24	Revision	V4.0

1. Introduction

1.1. Purpose

This is version 1.0 of our CyberTool, which is a public website that will enable users to filter through a variety of cybersecurity tools depending on the needs of the user (commercial, personal, etc.). This website will be hosted on an AWS server including all necessary architecture for the product. Among this architecture, there will be a database containing the cybersecurity products and the key values that were determined during research of said products. Recommendations will be generated from the database based on user preferences/requirements. Upon user request, a comparative report will be generated containing prospective tools and/or tools they already have.

1.2. Document Conventions

This SRS will be broken into numerical sections with each title bolded. The requirements will be separated into three categories of Front-End, Back-End, and Research with each of these containing the subcategories: "Must have", "Should have", "Could have", and "Would like to have". The glossary will contain all of the names and abbreviations of the third-party software and AWS services that are utilized.

1.3. Intended Audience and Reading Suggestions

The CyberTool is designed for all readers. To effectively build the tool, developers should look into the "Front-End" and "Back-End" sections for technical insights. Project Managers overseeing the tool's progression should review the entire document to ensure everything is on track. Marketing Staff will benefit from the "Purpose" and "Product Scope" sections, which highlight the tool's main features. Users can learn about what the tool offers by reading the "Purpose" section. Testers should focus on the technical requirements to ensure the tool works as planned. Lastly, Documentation Writers creating guides or other materials should read the entire SRS, with special attention to the "Glossary". For a complete understanding, all readers are advised to start with the "Purpose" section and then move to sections most relevant to their role.

1.4. Product Scope

CyberTool is software designed to help users find the right cybersecurity tools. Its goal is to make the search for security tools easy and quick. By using this tool, users, whether individuals or big companies, can find the best security solutions for their needs. This aligns with our corporate goal of making cybersecurity accessible and understandable for everyone.

1.5. References

2. Overall Description

2.1. Product Perspective

CyberTool is software developed to assist users in selecting appropriate cybersecurity tools. It operates by interfacing with a database hosted on AWS, which contains detailed information about various cybersecurity products. The tool then provides recommendations based on user-specific requirements.

2.2. Product Functions

User Preferences/Requirements: Users input their specific needs.

- Database on AWS: The tool interfaces with a database hosted on AWS containing information about various cybersecurity products.
- Recommendations: Based on the user's preferences and the data from the database, the tool provides recommendations.
- Comparative Report: Upon user request, a report comparing prospective tools and/or tools they already have is generated.

Filter Cybersecurity Tools: Users can filter through cybersecurity tools.

- Commercial: Tools for commercial use.
- Personal: Tools for personal use

Figure for 2.1 Product Perspective

2.3. User Classes and Characteristics

2.4. Operating Environment

The entire framework will operate in the AWS environment including the hosting of the website. As a result, users with a web browser (see versions in 2.7) will be able to interact with the product, CyberTool.

2.5. Design and Implementation Constraints

2.5.1. Design Constraints

- System can only be hosted by Amazon Web Services
- System can only be accessible by web browsers (See versions in 2.7).

2.5.2 Implementation Constraints

- System Speed is dependent on the quality of service paid to AWS.

2.6. User Documentation

2.6.1. The website shall be hosted via a GitHub repository.

2.6.2. The complete code of the AWS backend shall be hosted via a GitHub repository.

2.6.3. A user manual of the AWS backend will be provided on GitHub.

2.6.4. A system architecture diagram will be provided on GitHub.

2.6.5. The website shall provide links to documentation to GitHub Website.

2.6.6. The website shall provide a web page to explain physical website usage of the website.

2.7. Assumptions and Dependencies

2.7.1. The developer must use Version 118.0.5993.72 (Official Build) (64-bit) Google Chrome Web Browser.

2.7.2. The developer may use Version 118.0.2 (64-bit) Firefox Web Browser.

2.7.3. The developer must have at least version 7.0.0 of npm.

2.7.4. The developer must have at least version 21.0.0 of nodejs.

2.7.5. The developer must have at least version 2.10.0 of aws-cli.

2.7.6. The developer must have internet access.

2.7.7. The developer must be on a computer running Windows 10.

2.7.8. The developer must have Administrative Access to AWS Dashboard.

3. External Interface Requirements

3.1. User Interfaces

The user shall interface with CyberTool through our AWS-hosted web application. The following requirements detail how the user interface shall be displayed to offer application functionality to the user.

The user shall navigate CyberTool starting from the Login Page and then to the appropriate following pages according to their input.

Login Page: A simple interface asking for a username and password, accompanied by a 'Forgot Password' link and a 'Sign up' button for new users.

3.1.1 Login Page: The system shall provide input fields for the username.

3.1.2 Login Page: The system shall provide input fields for the password.

3.1.3 Login Page: The system shall accept a string between 6 and 20 characters for the username field.

3.1.4 Login Page: The system shall accept a string between 6 and 20 characters for the password field.

3.1.5 Login Page: The system shall provide a 'forgot password' link beneath the password field.

3.1.6 Login Page: The system shall provide a 'Login' button.

3.1.7 Login Page: The systems shall provide a 'Sign Up' button underneath the 'Login' button.

3.1.8 Login Page: The systems shall provide a 'Continue as guest' button underneath the Signup button.

3.1.9 Login Page: The system shall allow authenticated users to take the survey.

3.1.10 Login Page: The system shall allow unauthenticated users to view CyberTool page.

3.1.11 Login Page: The system shall allow authenticated users to be directed to survey page.

3.1.12 Login Page: The system shall allow authenticated users to view the main page.

3.1.13 Login Page: The system shall detect if an input field has not been filled out.

3.1.14 Login Page: The system shall detect if a username is incorrect.

3.1.15 Login Page: The system shall detect if a password is incorrect.

Signup Page: A simple interface asking for a name, email, username, and password.

3.1.16 Signup Page: The system shall provide a signup page

3.1.17 Signup Page: The system shall provide input fields for the username.

3.1.18 Signup Page: The system shall provide input fields for the password.

3.1.19 Signup Page: The system shall provide input fields for the 'retype password' field.

3.1.20 Signup Page: The system shall provide input fields for the email

3.1.21 Signup Page: The system shall accept a string between 6 and 20 characters for the username field.

3.1.22 Signup Page: The system shall accept a string between 6 and 20 characters for the password field.

3.1.23 Signup Page: The system shall accept a valid email for the 'email' input

3.1.24 Signup Page: The system shall ensure the 'password' and the 're-type password' are the same

3.1.25 Signup Page: The system shall provide a ‘Sign Up’ button

3.1.26 Signup Page: The system shall provide a check box for the user agreement

3.1.27 Signup Page: The system shall detect if an input field has not been filled out

3.1.28 Signup Page: The system shall detect if check box has been checked

3.1.29 Signup Page: The system shall detect if an email is already in use

3.1.30 Signup Page: The system shall detect if password is UTF8 characters

3.1.31 Signup Page: The system shall allow authenticated user to view main page

3.1.32 Signup Page: The system shall allow authenticated users to be directed to survey page

Survey Page: Multiple-choice questions designed to gather user needs regarding cybersecurity tools. Questions might revolve around user expertise, organization size, nature of the threat they face, and the technical stack they use.

3.1.33 Survey Page: The system shall provide check boxes.

3.1.34 Survey Page: The system shall provide questions for the user.

3.1.35 Survey Page: The system shall allow users to check off box.

3.1.36 Survey Page: The system shall detect if a box is not checked off.

3.1.37 Survey Page: The system shall allow authenticated users to be directed to the main page.

3.1.38 Survey Page: The system shall provide a ‘Continue’ button.

Tools Page: Displays a list of recommended cybersecurity tools based on the user’s answers, with brief descriptions.

3.1.39 Tools Page: The system shall provide a list of cybersecurity tools

3.1.40 Tools Page: The system shall provide the name of the cybersecurity tools

3.1.41 Tools Page: The system shall provide the version of the cybersecurity tools

3.1.42 Tools Page: The system shall provide the status of the cybersecurity tools

3.1.43 Tools Page: The system shall provide the launch date of the cybersecurity tools

3.1.44 Tools Page: The system shall provide a search bar on top of the cybersecurity tools

3.1.45 Tools Page: The system shall accept a string between 1 to 20 characters in search bar

3.1.46 Tools Page: The system shall show numbers of cybersecurity tools shown on page.

3.1.47 Tools Page: The system shall have pagination for cybersecurity tool display

3.1.48 Tools Page: The system shall provide left arrow button

3.1.49 Tools Page: The system shall provide right arrow button

3.1.50 Tools Page: The system shall move to next page when right arrow is pressed

3.1.51 Tools Page: The system shall move to previous page when left arrow is pressed

3.1.52 Tools Page: The system shall provide a filter checkbox menu

3.1.53 Tools Page: The system shall provide an aviation specific checkbox filter option

3.1.54 Tools Page: The system shall provide a toolbox checkbox filter option

3.1.55 Tools Page: The system shall provide a maturity level ranging from levels 1-4 as a checkbox filter option

3.1.56 Tools Page: The system shall provide an AI/ML use checkbox filter option

3.1.57 Tools Page: The system shall provide a tool function menu with checkboxes

3.1.58 Tools Page: The system shall provide a drop-down menu with different company options

3.1.59 Tools Page: The system shall filter cybersecurity tools based on filter user choice (see appendix item ---- for filter choices)

3.1.60 Tools Page: The system shall provide a 'Download as PDF' button

3.1.61 Tools Page: The system shall provide a link to retrieve old reports

3.1.62 Tools Page: The system shall allow the user to be directed to reports page

3.1.63 Tools Page: The system shall allow the user to download a comparative report

3.1.64 Tools Page: The system shall allow the user to click on tool

3.1.65 Tools Page: The system shall display tool information

Document Page: Displays a list of previously created reports with the option to view each report with an interactive link to the report

3.1.66 Document Page: The system shall display all documents previously made

3.1.67 Document Page: The system shall provide a 'View Report' link

3.1.68 Document Page: The system shall allow user to view a single report.

3.1.69 Document Page: The system shall allow a user to download a report via a presigned URL.

Forgot Password Page: Allows user to reset password if user has an account

3.1.70 Forgot Password Page: The system shall provide input fields for the email

3.1.71 Forgot Password Page: The system shall accept a string between 6 and 20 characters for the email field

3.1.72: Forgot Password Page: The system shall provide a "Send Code" button

Logical Characteristics:

Standards/Guidelines:

The GUI should ensure it is user-friendly.

Consistent font and color schemes throughout all its pages

Standard buttons should be used for login, signup, survey, and result pages.

Error messages should be clear, concise, and displayed in red font near the point of error.

3.2 Hardware Interfaces

- **Supported Devices:** Desktops, Laptops, and Tablets

3.3 Software Interfaces

- **Operation System:** Platform independent but can be used with Windows 11, macOS, and Linux
- **Tools & Libraries**
  - AWS Amplify for frontend development
  - AWS Cognito for storing user profiles
  - DynamoDB Database for tool data and storage of URLs
  - Amazon API Gateway for communications between different AWS tools

Data Interaction:

- Incoming Data: User login credentials, user responses to the survey
- Outgoing Data: Recommended tool list, error or success messages

3.4 Communication Interfaces

Communication Protocol

HTTPS SSL/TLS Connection
All client-server communication within the CyberTool system will exclusively use HTTPS (Hypertext Transfer Protocol Secure) over SSL/TLS (Secure Sockets Layer/Transport Layer Security) for enhanced security and data integrity. This ensures that all data transmitted between clients and servers is encrypted, making it extremely difficult for unauthorized parties to intercept or tamper with the communication.

Message Formatting

JSON (JavaScript Object Notation)
Message formatting between the client and server components of the CyberTool system will be accomplished using JSON (JavaScript Object Notation). JSON is a lightweight and widely supported data interchange format that facilitates efficient data transmission and parsing.

Data Transfer
Data transfer within the CyberTool system is facilitated through the following AWS (Amazon Web Services) services:

- **Amazon Cognito**
  - Amazon Cognito will be used for user authentication and management. It provides secure and scalable user identity and access control.
- **Amazon CloudFront**
  - Amazon CloudFront will serve as the content delivery network (CDN) to optimize the delivery of content, including static assets, to users. This will enhance the overall performance and reliability of the system.
- **Amazon API Gateway**
  - Amazon API Gateway will serve as the interface for external clients to interact with the system's APIs. It manages API requests, enforces security, and facilitates the integration of various backend services.

User Authentication Requirements
Users must provide at least an email address for sign-up and authentication purposes.

4. System Features

4.1. Comparative Report

4.1.1. Description and Priority

4.1.2. Stimulus/Response Sequences

4.1.3. Functional Requirements

4.1.3.1 The system shall develop a report in a PDF format.

4.1.3.2 The system shall develop a report in an HTML format.

4.1.3.3 The system shall store the report in an S3 Bucket. The system shall create a presigned URL for the report that is valid for 7 days.

4.1.3.4 The system shall store the source URL of the report in DynamoDB. The system shall store the fonts needed for reports in an S3 Bucket.

4.1.3.5 The system shall use AWS Lambda to run its code.

4.1.3.6 The system shall use the Rust Bootstrap for Lambda Runtime.

4.1.3.7 The system shall create error logs via CloudWatch.

4.1.3.8 The system shall create user usage logs via CloudWatch.

4.1.3.9 The system shall notify the user if an error occurs during report creation error via AWS SNS email.

4.1.3.10 The system shall notify website administrators of any report creation errors via AWS SNS email.

4.1.3.11 The report shall query from DynamoDB for tool data via Amazon API Gateway. The system shall deserialize JSON from DynamoDB Tool Database.

4.1.3.12 The system shall derive JSON output from recommendation algorithm.

4.1.3.13 The system shall use the programming language Rust for creation of the report.

4.1.3.14 The report shall display the tools selected as a table.

4.1.3.15 The report table shall display the technology’s name.

4.1.3.16 The report shall contain the user’s username.

4.1.3.17 The report table shall contain tools already used by the user.

4.1.3.18 The report table shall contain links to each technology’s homepage.

4.1.3.19 The report table shall contain what type of technology each row contains.

4.2. Recommendation Algorithm

4.3. Cognito Authentication

4.3.1. Description and Priority

The system will integrate Amazon Cognito to handle user authentication and management. This feature is high priority because of user security, access control, and personalization features within application.

Benefit:
Using Amazon Cognito improves user trust and makes sign-in easy. It also helps personalize the user's experience.
Penalty:
Without Cognito, sign-in might be less safe and harder for users.

Cost: Fees for using Cognito and costs to add it to our system.
Risk:
Possible issues during setup. Costs might go up if many users join. We rely on Cognito's performance.

4.3.2. Stimulus/Response Sequences

Stimulus: User attempts to register or sign into the application.

- Response: The system initiates the authentication flow with Amazon Cognito, presenting the user with the necessary UI for the process.

Stimulus: User completes authentication details.

- Response: Amazon Cognito verifies credentials. Upon verification, the user will be granted access depending on the user pool.

Stimulus: User forgets password or login details

- The system triggers Amazon Cognito's password recovery flow, sending the user a reset link or code to their registered email

4.3.3. Functional Requirements

4.3.3.1 The system shall enforce Cognito data protection standards.

4.3.3.2 The system shall authenticate users through Cognito.

4.3.3.3 The system shall use user groups for user management.

4.3.3.4 The system shall sync front-end validations with Cognito.

4.3.3.5 The system shall monitor login attempts via Cognito.

4.3.3.6 The system shall direct authenticated users with Cognito.

4.3.3.7 The system shall confirm user email through Cognito.

4.3.3.8 The system shall use user group assignments in Cognito.

4.3.3.9 The system shall permit role-based access with Cognito.

4.3.3.10 The system shall support temporary guest credentials.

4.3.3.11 The system shall suspend user accounts with Cognito.

4.3.3.12 The system shall support password policies with Cognito.

4.3.3.13 The system shall scale based on user pool.

4.3.3.14 The system shall suspend user pools with Cognito.

4.3.3.15 The system shall integrate with AWS via Cognito.

4.3.3.16 The system shall allow for customizable user attributes in Cognito.

4.3.3.17 The system shall manage tokens via Cognito.

4.3.3.18 The system shall notify users via Cognito.

4.4. Database Access

4.4.1. Description and Priority

The system will utilize Amazon DynamoDB, a NoSQL database service provided by AWS with features including predictable performance and scalability, to store and query information related to each tool covered by the project. Partition keys will be set based on the primary purpose of each tool (i.e., its function), and a sort key with a unique identifier. Each item (or tool) in the table will contain attributes or additional information such as the name of the tool, its features, accessibility and requirements, etc. A full list of attributes is available in Appendix E.

4.4.2. Stimulus and Response Sequences

Stimulus: User makes a search in a search field on the website associated with a particular attribute

- Response: The system shall display each tool where that tool's particular attribute matches the search term

Stimulus: Developer updates a database item through AWS

- Response: DynamoDB table updates and is reflected on the website

4.4.3. Functional Requirements

4.4.3.1 The system shall store tables that each contain a unique string-type partition key that is associated with the function the user wants the tool to serve. (See Appendix D for partition keys)

4.4.3.2 The system shall store tables that contain string-type global secondary indexes (GSI) that will contain a list of attributes. (See Appendix E for GSIs)

4.4.3.3 The system shall perform queries based on desired partition keys and GSIs designated by the user.

4.4.3.4 The system shall allow the user to form complex queries equivalent to AND, OR, or NOT operations.

4.4.3.5 The system shall provide the user with information about a tool based on queries to the database.

4.4.3.6 If the user enters a query and no tool matches the input given, the system shall display an error message that there are no tools available with the given criteria.

4.4.3.7 The system shall generate tables with columns created by the GSIs the user has made selections for.

4.4.3.8 The system shall generate an additional column with a personal recommendation rating for each tool based on analysis from the developer team.

4.4.3.9 Information provided by the database shall be accessible on the website.

4.4.3.10 Information provided by the database shall be able to be integrated into the recommendation reports.

4.4.3.11 The system shall allow the developer team to view and modify the database through AWS.

4.5. Application Programming Interface (API)

4.5.1. Description and Priority

The system will utilize Amazon API Gateway, a service used to facilitate communication between different AWS services including AWS Lambda, AWS Amplify, and DynamoDB. Amazon API Gateway allows for the development of a RESTful API and uses URL endpoints to invoke different HTTP methods to access or modify resources.

This feature is high priority, as it allows for developers to continue developing the website with integration between the front-end and back-end and enables end-user features including searching the tool database and storing user account information or reports.

4.5.2. Stimulus and Response Sequences

Stimulus: Developer invokes an API method to update a database item

- Response: The API sends a command to DynamoDB to update the item in the database

Stimulus: User queries a search on the available tools

- Response: The API sends a command to DynamoDB to retrieve information on the available tools

4.5.3. Functional Requirements

4.5.3.1 The system shall utilize Amazon API Gateway

4.5.3.2 The system shall seamlessly integrate with other AWS services, including AWS Lambda, AWS Amplify, and DynamoDB.

4.5.3.3 The system shall provide logging and monitoring capabilities for API usage via AWS CloudWatch

4.5.3.4 The system shall expose API endpoints with a clear and consistent URL structure

4.5.3.5 The system shall define and expose endpoints for main resources, such as /tools, /users, etc.

4.5.3.6 The system shall implement API authentication using Cognito User Pools

4.5.3.7 The system shall support JSON as the standard format for request and response bodies

4.5.3.8 Each API request shall require appropriate headers, including Content-Type, Authorization, and Accept.

4.5.3.9 The system shall use JSON for data serialization in API communication

4.5.3.10 The system shall support the HTTP methods GET, POST, PUT, and DELETE

4.5.3.11 The system shall follow RESTful principles for a consistent and predictable API

4.5.3.12 HTTP status codes shall be used appropriately to indicate the result of API requests

4.5.3.13 The system shall provide clear error messages in the API response, including error codes and descriptions.

4.5.3.14 Additional error details and a trace ID shall be included for debugging purposes.

4.5.3.15 The system shall maintain comprehensive API documentation that includes details on endpoints, request/response formats, authentication, and examples.

4.5.3.16 Documentation shall be easily accessible to developers.

4.5.3.17 The system shall implement thorough testing.

4.5.3.18 The system shall utilize AWS Lambda with a Python 3.10 runtime to run code through API Gateway

5. Other Nonfunctional Requirements

5.1. Performance Requirements

To achieve an optimal user experience, the CyberTool web application shall incorporate a Content Delivery Network (CDN) as a critical component of its infrastructure. The selected CDN for this purpose is Amazon CloudFront, renowned for its capabilities in improving file delivery performance.

Requirements

- CDN Implementation: The CyberTool web application shall integrate Amazon CloudFront as its designated CDN to enhance the delivery of essential files.
- Latency Reduction: Amazon CloudFront shall be configured to minimize latency, ensuring that content is delivered to end-users with minimal delay, irrespective of their geographical location.
- Scalability: The CDN solution shall possess inherent scalability features. It should seamlessly adapt to accommodate increased user traffic and growing content delivery demands without compromising application performance.

5.2. Safety & Security Requirements

Data Encryption in Transit
To ensure the confidentiality and integrity of data during transmission, our product will exclusively employ the HTTPS (Hypertext Transfer Protocol Secure) protocol. HTTPS employs strong encryption to secure data as it travels between our server and the client's device. SSL/TLS certificate will be provided by AWS Certificate Manager.

Data Encryption at Rest
Sensitive personal information, including emails, passwords, and saved reports, will be subject to encryption through Server-Side Encryption (SSE) using the AES-256 (Advanced Encryption Standard with a 256-bit key) encryption standard. AES-256 is a robust encryption algorithm that offers a high level of security for data stored on AWS servers. This ensures that even if unauthorized access occurs, the data remains protected and unreadable.

Logging and Monitoring
To maintain transparency and track all data transactions originating from databases and object storage, we will employ CloudWatch. CloudWatch will meticulously log all relevant data transactions, helping us monitor and analyze the system's performance and security. All log data will be securely stored in Amazon S3 (Simple Storage Service) with SSE (Server-Side Encryption) enabled. This additional layer of encryption ensures the confidentiality and integrity of log data, guarding against potential breaches.

Denial-of-Service Protection
To proactively safeguard our website against denial-of-service attacks, which can disrupt services and lead to data loss, we will implement the utilization of AWS Web Application Firewall (WAF) and AWS Shield. These security measures are designed to identify and mitigate various forms of attacks, including cross-site scripting (XSS) attacks, which can compromise the integrity of user data.

- **AWS Web Application Firewall (WAF)**: This service provides real-time protection against common web exploits and vulnerabilities, including XSS attacks. It filters and monitors web traffic, allowing only legitimate requests to reach our servers.
- **AWS Shield:** AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards against large-scale and sophisticated DDoS attacks. It helps ensure the availability of our services and protects against potential data loss due to service disruption.

5.3. Software Quality Attributes

High Availability
High availability is a critical aspect of the CyberTool product, as it ensures that our services are accessible and reliable, even in the face of hardware failures or other disruptions. AWS’s multi-AZ and serverless aspects of services can guarantee high availability of data and service to customers.

Multi-Availability Zone Deployment

The product will be deployed across multiple Availability Zones (AZs) provided by AWS. This approach ensures redundancy and fault tolerance. In the event of an issue affecting one AZ, traffic is automatically redirected to a healthy AZ, minimizing downtime and service interruptions.

5.4. Business Rules

User Accounts for Website Access
For users who intend to interact with the CyberTool application via the website's front-end interface, the following requirements apply:

Account Creation: Website users must create an individual account using a valid email address. This account creation process serves to uniquely identify users and facilitate personalized interactions with the application.

Once registered, website users will gain access to the following functionalities within the CyberTool application:

Query Storage: Users can save and access queries for future reference.

Questionnaire for Recommendations: The application will provide users with the capability to complete a questionnaire to receive tailored recommendations for cyber tools.

Report Generation: Users will have the ability to generate reports that document the recommendations provided by the application.

6. Other Requirements

The database used will be NoSQL DynamoDB.

Appendix A: Glossary

AWS – Amazon Web Services
WAF – Web Application Firewall
SSE – Server-Side Encryption
IAM – Identity and Access Management
OU - Organization Unit
SCP - Service Control Policy
RBAC – Role-based Access Control
CIAM – Customer Identity and Access Management

Appendix B: Analysis Models

Appendix C: To Be Determined List

Continue to develop filter for cybersecurity tools
Add additional cybersecurity tools to DynamoDB database
Finalize report format
Develop report html page for user to use

Appendix D: Partition Keys

Log Analysis
Industrial Control Systems
Operational Technology
Identity and Access Management
Indicators of Compromise

Appendix E: Attributes

Name
Company Name
Company Website
Company Phone Number
Device
Launch Year
Active
Requirements
Features
Drawbacks
Accuracy
Pricing
Compliance