**Product Requirements Document: "SecureAWS" MS Teams Chatbot**

**Version:** 1.0

**Date:** June 19, 2025

**Author:** Gemini

**Status:** Draft

**1. Introduction & Purpose**

"SecureAWS" is a conversational AI assistant integrated into Microsoft Teams. Its purpose is to provide employees with immediate and accurate answers to questions about AWS security best practices. The chatbot will serve as a centralized knowledge hub, drawing information from the official AWS Well-Architected Framework, our internal Confluence documentation, and the general knowledge of Amazon Q. This will reduce the time engineers spend searching for information, ensure consistent application of security principles, and improve the overall security posture of our AWS environments.

**2. User Personas**

* **DevOps Engineer (Priya):** Needs to quickly find the correct security configuration for a new S3 bucket. Wants to get a direct answer with a code example without leaving MS Teams.
* **Software Developer (David):** Is building a new feature and needs to understand the security implications of using a specific AWS service. Wants to ask high-level questions and get links to relevant internal policies.
* **Security Analyst (Sarah):** Is reviewing a new application and wants to verify that the team is following the latest internal security guidelines documented in Confluence.

**3. Features & User Stories**

**3.1. Core Chat Functionality**

* **As a user, I want to ask security-related questions in natural language so that I can get immediate answers.**
* **As a user, I want the chatbot to maintain the context of my conversation so that I can ask follow-up questions.**
* **As a user, I want to receive answers that clearly cite the source (e.g., "From AWS Well-Architected Framework:", "From Confluence:") so that I can trust the information.**

**3.2. Knowledge Sources**

* **As a user, I want the chatbot to provide answers based on the latest AWS Well-Architected Framework so that I am following industry best practices.**
* **As a user, I want the chatbot to search our internal Confluence space for security policies and guidelines so that I am compliant with our company's standards.**
* **As a user, I want the chatbot to leverage the general knowledge of Amazon Q to answer a broad range of AWS security questions.**

**3.3. User Experience & Integration**

* **As a user, I want to interact with the chatbot directly within a dedicated channel or as a personal app in MS Teams.**
* **As a user, I want to receive responses formatted with rich text (bold, italics, code blocks, links) for better readability.**
* **As a user, I want to provide feedback (e.g., thumbs up/down) on the answers so that the chatbot's performance can be improved.**

**4. Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Priority** |
| FR-01 | The chatbot MUST be accessible as an application within Microsoft Teams. | Must Have |
| FR-02 | The chatbot MUST integrate with Amazon Q as its core AI engine. | Must Have |
| FR-03 | Amazon Q MUST be configured to use the AWS Well-Architected Framework as a knowledge source. | Must Have |
| FR-04 | The chatbot MUST securely connect to and index a specified Confluence space. | Must Have |
| FR-05 | The system MUST support natural language queries from users. | Must Have |
| FR-06 | Responses MUST be rendered in real-time. | Must Have |
| FR-07 | The chatbot SHOULD be able to handle conversational context (follow-up questions). | Should Have |
| FR-08 | Responses SHOULD include source attribution (AWS WAF, Confluence, etc.). | Should Have |
| FR-09 | The system MUST support rich-text formatting in responses. | Must Have |
| FR-10 | The chatbot SHOULD include a mechanism for users to provide feedback on answer quality. | Should Have |
| FR-11 | The system MUST log all queries and responses for analytics and auditing. | Must Have |

**5. Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Details** |
| NFR-01 | **Performance:** | Responses should be delivered in under 3 seconds for 95% of queries. |
| NFR-02 | **Security:** | All data in transit and at rest must be encrypted. Confluence integration must use secure, read-only credentials. The principle of least privilege must be applied to all IAM roles. |
| NFR-03 | **Scalability:** | The backend infrastructure must autoscale to handle concurrent users during peak business hours. |
| NFR-04 | **Availability:** | The service should have an uptime of 99.9%. |
| NFR-05 | **Data Privacy:** | The chatbot must not store any Personally Identifiable Information (PII) from user queries. |

**6. Technology Stack (Proposed)**

* **Chat Platform:** Microsoft Teams
* **AI Assistant:** Amazon Q
* **Backend Logic:** AWS Lambda (Node.js or Python)
* **API Gateway:** Amazon API Gateway
* **Deployment:** AWS Serverless Application Model (SAM) or Terraform
* **Authentication:** Azure Active Directory for Teams App, AWS IAM for services.
* **CI/CD:** GitHub Actions