Domain Generation Algorithm (DGA)

Design Document

C-DAC, Bangalore

Version <version No.>

10-07-2024

Prepared By:

Aditya Londhe

Shefali Jakale

Pranjal Chavan

Arti Kumari

Yadnik Borse

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed design for the "DGA Detection Using Deep Learning" project. It explains the architecture, component design, data design, and UI design. This document serves as a guide for developers to implement the functionality requirements, including development and deployment architectures, database design, API development, and UI development.

1.2 Scope

This design document pertains to the implementation of the "DGA Detection Using Deep Learning" project. While scalability and extensibility at the component level are considered, practical implementation testing has not been performed.

1.3 Intended Audience

This document is intended for design, development, and testing teams and should be referenced at all stages of the project.

1.4 References

- Functional Requirement Specifications (FRS)

2.0 Acronyms, Terms, and Definitions

- FRS: Functional Requirements Specification
- User: Refers to department/employee/worker

3.0 Assumptions and Constraints

- The application follows API-based development, requiring user subscription for functionality access.
- Assumptions and constraints specific to the project are documented.

4.0 Basic Design Approach

- The "DGA Detection Using Deep Learning" project follows a Structured Design approach.
- Elements are organized based on FRS requirements, with APIs developed to handle related functions.
- APIs are designed with minimal interdependence for independent functionality.

5.0 Risks

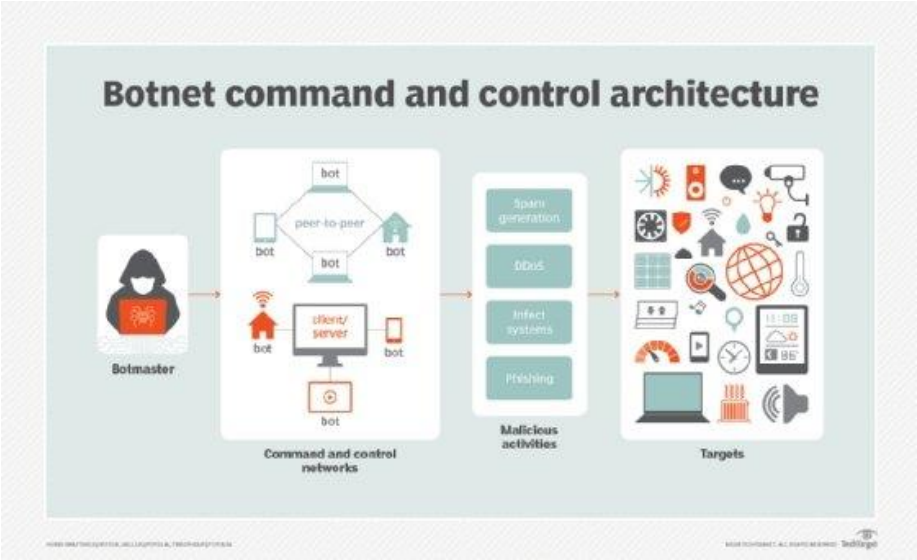
- Risks include handling personal data securely and compliance with data protection regulations.

6.0 System Overview

Describe the overall system architecture and its components.

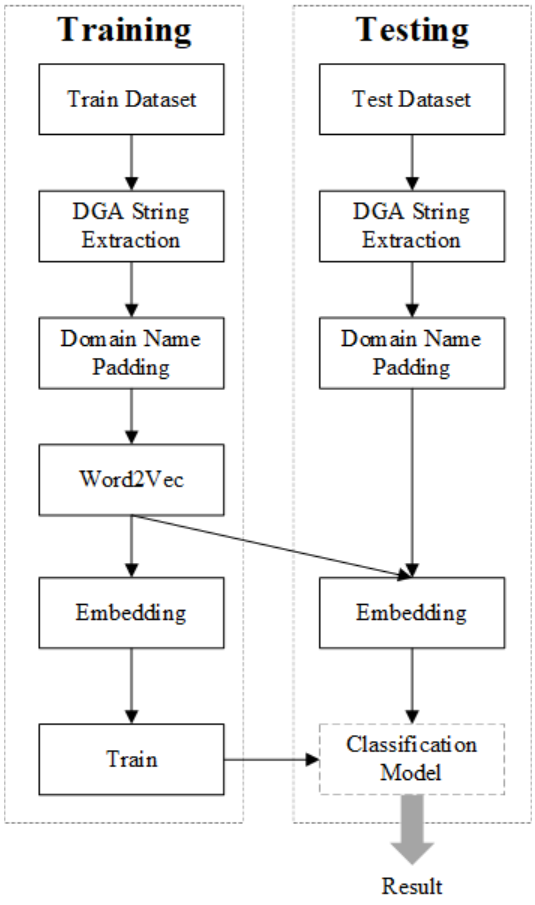
7.0 Architecture Design

Figure 1: Architecture Diagram



8.0 Data Design

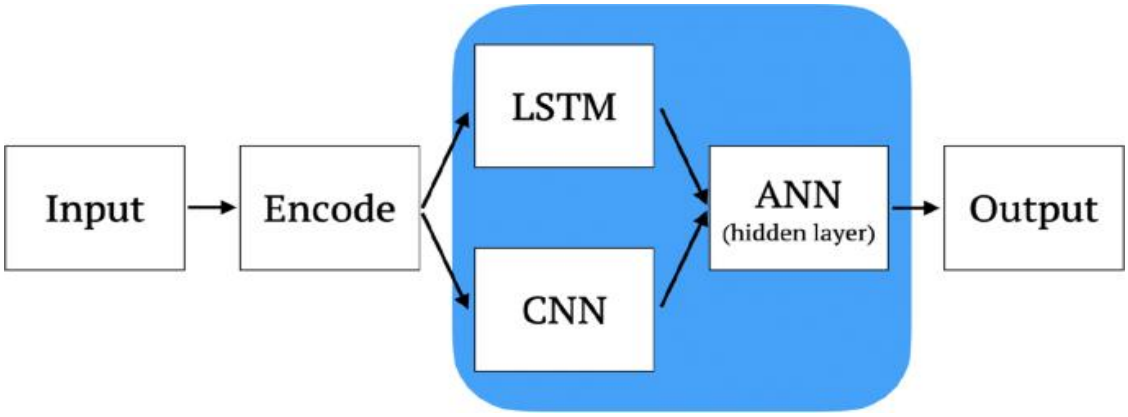
Database ER Diagram



- Explain database tables and structures.
- Example: Master tables (prefixed by 'm'), logs (prefixed by 'l'), and transaction tables.

9.0 Component Design

Figure 2: Component Diagram

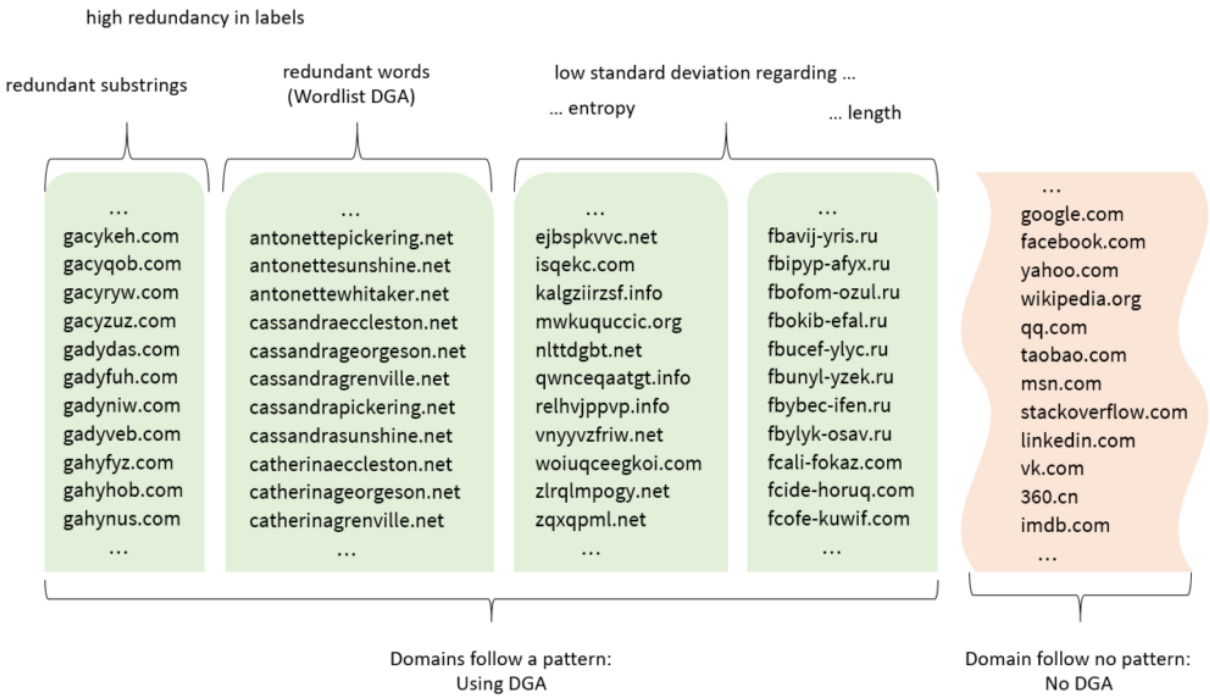


9.1 Component 1

9.1.1 Component Overview

- Explain the purpose and function of Component 1.
- Stepwise process followed in component implementation.

Activity Diagram/Use Case Diagram



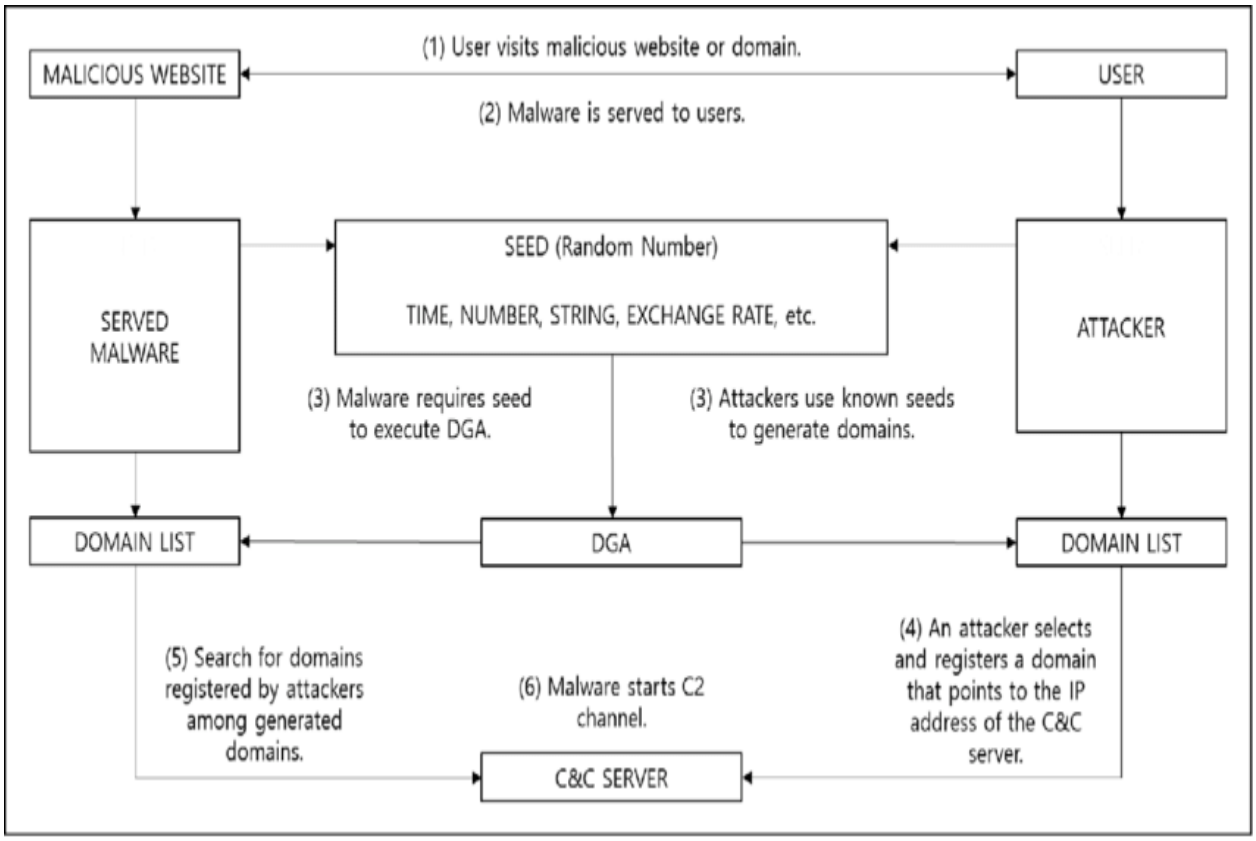
9.2 Component 2

9.2.1 Component Overview

- Explain the purpose and function of Component 2.

- Stepwise process followed in component implementation.

Activity Diagram/Use Case Diagram



10.0 Interface Design

GUI Layouts

- Provide GUI layouts for each interface in the application.

11.0 Any Specific Design Considerations

- The sandbox is limited to a specified number of simultaneous processing for API services.
- APIs are functionally tested with required security controls via API gateway.
- API services are accessible through a web-based interface, including API Explorer and Administrative Dashboard.
- APIs are designed as generic domain-specific implementations for integration into larger systems.

12.0 Design Test

- [Testing Team to update this section]

User Registration

- **Description**: Manages user sign-up, login, and profile management. Ensures secure and authorized access to the DGA detection system.
- **Key Features**:

Identified Components List for the DGA Detection System

1. **User Registration**
    - **Description**: Manages user sign-up, login, and profile management. Ensures secure and authorized access to the DGA detection system.
    - **Key Features**:
        - User sign-up with email verification
        - Secure login and token-based authentication
        - Profile management for updating user details

13.0 Cross Reference with System Requirement Specification

System Requirement Specification Features

COMPONENT NAME	API Name	Requirements	Adherence/Limitations
Component 1	/register	FR1.1	Meets the FR completely
Component 2	/search	FR2.1	Partially meets the FR

Appendix A – Glossary

Architecture Diagram

ERD(Entity Relationship Diagram)

Use Case Diagram

Component Diagram

Appendix B - References

1) Crotti M, Dusi M, Gringoli F, et al (2007). Detecting HTTP Tunnels with Statistical Mechanisms[C]. Communications, 2007. ICC '07. IEEE International Conference on. IEEE.
2) Nussbaum L, Neyron P, Richard O (2009). On Robust Covert Channels Inside DNS[C]. Emerging Challenges for Security, Privacy & Trust, Ifip Tc
3) International Information Security Conference. Aiello M, Merlo A, Papaleo G (2013). Performance assessment and analysis of DNS tunneling tools[J]. Logic Journal of IGPL, 21(4), 592-602.

Appendix C - Additional Diagrams

Include additional diagrams as needed.