System Design Document for BharatDNS

Sarthak Gupta, Dhanush Hebbar, Arjun Nair, Rohit Reddy

***Abstract*—This document presents a detailed system design for BharatDNS, a DNS resolver system that detects and mitigates malicious activities such as DNS tunneling. The design includes system architecture, component design, database design, and security considerations.**

I. Introduction

The BharatDNS system is designed to serve as a DNS resolver capable of detecting and mitigating malicious activities such as DNS tunneling. This document outlines the system design for BharatDNS, covering various aspects including system architecture, component design, database design, and security considerations.

A. Purpose of the Document

The purpose of this document is to describe the system design of BharatDNS, providing a comprehensive overview of its architecture, components, and functionality.

B. Scope

The scope of this document is to provide detailed insights into the design of BharatDNS, including its system architecture, component design, database schema, and security measures. It is intended for use by developers, testers, and project managers involved in the development and deployment of BharatDNS.

C. Definitions, Acronyms, and Abbreviations

- **BharatDNS**: DNS resolver system designed to detect and mitigate malicious activities.
- **SDD**: System Design Document.
- **UI**: User Interface.
- **DFD**: Data Flow Diagram.
- **ERD**: Entity Relationship Diagram.

II. System Overview

A. System Context

BharatDNS is a DNS resolver system designed to provide secure and efficient resolution of domain names to IP addresses. It operates as a web-based application and interacts with external systems such as threat intelligence services and TAXII servers to enhance its threat detection capabilities.

Indian Institute of Information Technology, Kottayam

B. System Functions

BharatDNS provides the following key functions:

- DNS resolution: Resolves domain names to IP addresses.
- Malicious activity detection: Detects and mitigates malicious activities such as DNS tunneling.
- Integration with threat intelligence services: Obtains real-time threat data to enhance detection capabilities.
- Machine learning-based detection: Utilizes machine learning algorithms to improve detection accuracy.
- User interface: Provides an intuitive interface for system administrators to monitor and manage the system.

C. User Characteristics

BharatDNS is designed for two types of users:

- System administrators: Users responsible for managing the BharatDNS system, including configuring settings, monitoring threats, and performing administrative tasks.
- End users: Users who access BharatDNS for DNS resolution services. They may include employees within an organization or external clients.

System administrators should have knowledge of managing web-based systems and familiarity with DNS protocols and security practices. End users may vary in technical expertise but should be able to use DNS resolution services without extensive training.

D. Constraints

BharatDNS operates under the following constraints:

- Scalability: The system must be capable of handling a large volume of DNS resolution requests and scaling to accommodate increasing demand.
- Security: BharatDNS must ensure the confidentiality, integrity, and availability of DNS resolution services, protecting against unauthorized access and malicious activities.
- Availability: The system should be available 24/7, with minimal downtime for maintenance and upgrades.

E. Assumptions and Dependencies

BharatDNS relies on the following assumptions and dependencies:

- Reliable threat intelligence services: External threat intelligence services provide accurate and timely threat data to enhance the detection capabilities of BharatDNS.
- Secure network infrastructure: The underlying network infrastructure supporting BharatDNS, including servers and communication channels, is secure and resilient to attacks.

- Stable internet connection: End users accessing BharatDNS have stable internet connectivity to ensure uninterrupted DNS resolution services.

III. Overview

The BharatDNS system serves as a DNS resolver capable of detecting and mitigating malicious activities such as DNS tunneling. It consists of various components that work together to ensure efficient and secure DNS resolution.

A. Hardware Architecture

BharatDNS operates on server machines located in data centers. These servers are equipped with sufficient hardware resources such as CPU, memory, and storage to handle DNS resolution requests efficiently. Additionally, network admin dashboards are provided for system administrators to monitor network calls.

B. Data Architecture

BharatDNS utilizes a relational database management system (RDBMS) to manage system data. The database stores information related to DNS resolution requests, threat intelligence data fetched from TAXII servers in STIX protocol, and network calls displayed to system administrators.

C. Interface Architecture

BharatDNS provides two interfaces: one for users and one for system administrators. The user interface is a network-based CLI application that allows users to perform DNS resolution, view resolution history, and manage their accounts. The system administrator interface is a network admin dashboard that provides administrators with tools to monitor network calls and manage system settings.

D. Data Flow Diagrams

Data flow diagrams illustrate the flow of data and interactions between various components of BharatDNS, including the servers, database, and user interfaces.

E. User Interface Design

The user interface design of BharatDNS is based on principles of usability and user experience. It is intuitive, easy to use, and visually appealing, providing users with all the necessary information they need to perform DNS resolution and manage their accounts.

IV. Data Design

This section describes the data design of BharatDNS, including a description of the data used by the system, a data dictionary, and information about data storage, access, and manipulation.

A. Data Description

BharatDNS utilizes the following data:

- Customer information, including name, contact details, and payment information
- Bike information, including bike type, availability, and location
- Rental information, including rental start and end dates, rental cost, and rental status

B. Data Dictionary

The data dictionary for BharatDNS is as follows:

1) DNS_Resolution_Data:

- **Request_ID** (Primary Key): Unique identifier for each DNS resolution request.
- **Site_URL**: URL of the site requested for resolution.
- **IP_Address**: Corresponding IP address resolved for the site URL.
- **Resolution_Time**: Timestamp of the resolution request.

2) Threat_Intelligence_Data:

- **Threat_ID** (Primary Key): Unique identifier for each threat intelligence entry.
- **Site_URL**: URL of the site identified as malicious.
- **Threat_Type**: Type of threat associated with the site (e.g., malware, phishing).
- **Detection_Time**: Timestamp of threat detection.

C. Data Storage

The data for BharatDNS will be stored in a relational database management system (RDBMS). The RDBMS will provide data consistency, data integrity, and data security. The database will be hosted on a server with appropriate backup and recovery mechanisms.

D. Data Access

Data access to BharatDNS will be provided through a web-based interface. The interface will allow authorized users to perform CRD (Create, Read, Delete) operations on the data. Access to the data will be controlled through authentication and authorization mechanisms.

E. Data Manipulation

Data manipulation in BharatDNS will be performed using SQL (Structured Query Language). The system will use stored procedures to ensure data consistency and data integrity. Additionally, appropriate error-handling mechanisms will be implemented to ensure that invalid data is not entered into the system.

V. Component Design

The component design of BharatDNS is detailed below:

A. DNS Resolver Component

The DNS Resolver Component of BharatDNS is responsible for handling DNS resolution requests, analyzing DNS traffic patterns, and detecting anomalous behavior indicative of DNS tunneling. It acts as the core functionality of the system, ensuring that DNS queries are resolved efficiently and securely.

BharatDNS utilizes open-source DNS resolvers such as PowerDNS for DNS resolution purposes. PowerDNS is a robust and flexible DNS server that is widely used in various network environments due to its reliability and performance.

By leveraging PowerDNS as the DNS resolver component, BharatDNS benefits from its advanced features and capabilities, including:

1) High Performance:: PowerDNS is designed to handle large volumes of DNS queries efficiently, ensuring fast response times and low latency for DNS resolution requests.

2) Scalability:: PowerDNS is highly scalable and can be deployed in distributed architectures to handle increasing DNS traffic loads as network requirements grow.

3) DNSSEC Support:: PowerDNS supports DNS Security Extensions (DNSSEC), allowing BharatDNS to validate DNS responses and provide enhanced security against DNS spoofing and cache poisoning attacks.

4) Flexible Configuration:: PowerDNS offers extensive configuration options, enabling BharatDNS to tailor its behavior according to specific network policies and requirements.

5) Active Community Support:: PowerDNS has an active community of developers and users who contribute to its ongoing development and provide support through forums, mailing lists, and documentation.

Overall, by utilizing PowerDNS as the DNS resolver component, BharatDNS ensures reliable and efficient DNS resolution services, contributing to the overall security and performance of the network infrastructure.

B. Threat Intelligence Services Integration Component

BharatDNS utilizes a TAXII (Trusted Automated Exchange of Indicator Information) server to fetch malicious DNS data. The system is configured to retrieve this data periodically, with a frequency set to once every 7 days. This ensures that BharatDNS remains updated with the latest threat intelligence information regarding malicious DNS activity.

By fetching data from the TAXII server at regular intervals, BharatDNS enhances its ability to detect and mitigate DNS-related security threats effectively. The periodic updates allow the system to adapt to evolving threat landscapes and incorporate new indicators of compromise into its detection mechanisms.

Additionally, the use of a TAXII server facilitates standardized and automated information sharing, enabling BharatDNS to access threat intelligence data from a wide range of trusted sources. This helps in maintaining the system's accuracy and reliability in identifying malicious DNS activity, thereby strengthening the overall security posture of the network infrastructure.

C. Machine Learning Model Component

The Machine Learning Model Component implements the machine learning model for DNS tunneling detection in BharatDNS. It includes processes for feature extraction from DNS traffic data and model inference to identify patterns indicative of DNS tunneling. This component continuously learns and adapts to new threats, improving the system's effectiveness in detecting malicious activities.

1) Sequential Model:: The neural network model is defined using the Sequential class from the Keras library. This means that the layers of the network are stacked sequentially on top of each other.

2) Input Layer:: The input layer of the neural network consists of 96 neurons. This layer is responsible for receiving the input data, which in this case is the tokenized DNS information. The input shape=(96) parameter specifies that each input sample has 96 features.

3) Dense Layers:: There are two hidden layers in the neural network, each containing 256 neurons. These layers are referred to as "dense" layers because each neuron is connected to every neuron in the previous layer. The activation function used in these layers is the Rectified Linear Unit (ReLU), which introduces non-linearity into the network and allows it to learn complex patterns from the input data.

4) Dropout Layers:: Dropout is a regularization technique used to prevent overfitting in neural networks. It works by randomly setting a fraction of input units to zero during training, which helps to prevent the network from relying too heavily on any individual neuron or feature. In this configuration, dropout layers with a dropout rate of 0.5 (50

5) Output Layer:: The output layer of the neural network contains a single neuron with a sigmoid activation function. This is a binary classification problem, where the network predicts whether the input DNS data is positive (indicating malicious activity) or negative (indicating benign activity). The sigmoid activation function squashes the output value between 0 and 1, representing the probability of the input being classified as positive.

6) Compilation:: Finally, the model is compiled using the specified optimizer, which is denoted by optim, the binary cross-entropy loss function, and accuracy as the evaluation metric. The binary cross-entropy loss function is commonly used for binary classification problems, and it measures the difference between the predicted probabilities and the actual labels.

Overall, this configuration defines a feedforward neural network with two hidden layers, dropout regularization, and a sigmoid output layer, which is trained to classify tokenized DNS data into positive or negative categories based on their maliciousness.

D. User Interface Component

The User Interface Component provides an interface for system administrators to interact with BharatDNS. It offers a user-friendly dashboard where administrators can view detected threats, monitor system health, manage system settings, and perform administrative tasks. The interface is designed to be intuitive and easy to use, facilitating efficient system management and threat response.

VI. Software Infrastructure

The software infrastructure for BharatDNS will consist of the following components:

- **Operating System:** BharatDNS will run on Linux-based operating systems, such as Ubuntu or CentOS, chosen for their stability, security, and compatibility with open-source software.
- **Web Server:** Apache HTTP Server
- **Threat Intelligence Services Integration:** AlienVault TAXII API
- **Machine Learning Framework:** TensorFlow and Keras
- **Database Management System:** MySQL
- **Programming Languages:** Python, SQL, XML along with web technologies such as HTML, CSS, and JavaScript, along with frameworks like React or Angular.

VII. Network Infrastructure

The network infrastructure for BharatDNS will be designed to ensure secure and reliable operation. Specific requirements include:

- **Local Area Network (LAN) Connectivity:** BharatDNS will be connected to the organization's local area network (LAN) to facilitate communication with other network components and systems.
- **Wide Area Network (WAN) Connectivity:** BharatDNS will have access to the internet via the organization's wide area network (WAN), enabling it to fetch threat intelligence data from external sources and communicate with remote systems.
- **Networking Components:**
  - **Switches:** Switches will be used to connect BharatDNS servers, workstations, and other networking components to the LAN, providing efficient data transfer within the local network.
  - **Router:** A router will be used to connect the LAN to the WAN, enabling internet access for BharatDNS and facilitating communication with external systems.
  - **Firewall:** A firewall will be deployed to secure BharatDNS and prevent unauthorized access to the system. The firewall will implement access control policies, intrusion detection/prevention, and other security measures to protect against network-based threats.

VIII. Security Considerations

To ensure the security of BharatDNS and its data, the following measures are implemented:

A. User Authentication

All users accessing BharatDNS must authenticate themselves using a username and password.

B. Data Encryption

Sensitive data such as user information and threat intelligence entries are encrypted using industry-standard encryption algorithms.

C. Regular Backups

Regular backups of the database are performed to prevent data loss in the event of system failure or other disasters.

IX. Conclusion

In conclusion, the System Design Document (SDD) for BharatDNS provides a comprehensive overview of the architecture, components, database design, and security considerations of the system. BharatDNS, a DNS resolver system, is meticulously designed to detect and mitigate malicious activities, particularly DNS tunneling, thereby ensuring the security and integrity of network infrastructure.

The SDD outlines the system's purpose, scope, and user characteristics, establishing a clear understanding of its intended functionality and target audience. It elucidates the system's constraints, assumptions, and dependencies, laying the groundwork for its development, deployment, and operation.

Through detailed descriptions of system components such as the DNS Resolver, Threat Intelligence Services Integration, Machine Learning Model, and User Interface, the document elucidates the inner workings of BharatDNS. Each component is meticulously designed to fulfill specific functions critical to the system's overall operation, from efficiently resolving DNS queries to leveraging machine learning algorithms for threat detection.

Moreover, the document delves into the software and network infrastructure supporting BharatDNS, outlining the technologies and protocols employed to ensure its reliable and secure operation. From the choice of operating systems and web servers to the implementation of encryption and backup mechanisms, every aspect of the system's infrastructure is carefully considered to mitigate risks and enhance resilience.

Security considerations are paramount throughout the SDD, with measures such as user authentication, data encryption, regular backups, and network security protocols outlined to safeguard BharatDNS against potential threats and vulnerabilities. These measures are integral to maintaining the confidentiality, integrity, and availability of the system and its data, ensuring its reliability and trustworthiness in real-world deployments.

In summary, the BharatDNS System Design Document provides a solid foundation for the development, deployment, and operation of a robust and secure DNS resolver system. By adhering to the principles and specifications outlined in this document, stakeholders can confidently proceed with the implementation of BharatDNS, confident in its ability to effectively detect and mitigate malicious activities while providing reliable and efficient DNS resolution services.

X. Glossary

- **BharatDNS**: A DNS resolver system designed to detect and mitigate malicious activities such as DNS tunneling.
- **SDD**: System Design Document.
- **UI**: User Interface.
- **DFD**: Data Flow Diagram.
- **ERD**: Entity Relationship Diagram.
- **TAXII**: Trusted Automated Exchange of Indicator Information, a protocol used for exchanging cyber threat intelligence.
- **DNS Tunneling**: A technique used by cyber attackers to bypass security controls by encapsulating non-DNS traffic within DNS packets.
- **PowerDNS**: An open-source DNS server software that provides high performance and extensive features for DNS resolution.
- **Threat Intelligence**: Information about potential or current threats to network security, obtained from various sources such as threat feeds, security vendors, and research organizations.
- **Machine Learning Model**: A computational model trained on data to identify patterns and make predictions or decisions without being explicitly programmed.
- **Neural Network**: A type of machine learning model inspired by the structure and function of the human brain, consisting of interconnected nodes (neurons) organized in layers.
- **TensorFlow**: An open-source machine learning framework developed by Google for building and training neural network models.
- **Keras**: A high-level neural networks API written in Python, capable of running on top of TensorFlow and other machine learning frameworks.
- **SQL**: Structured Query Language, a domain-specific language used for managing and querying relational databases.
- **RDBMS**: Relational Database Management System, a software system used for managing relational databases.
- **LAN**: Local Area Network, a network that connects devices within a limited geographical area such as a home, office, or campus.
- **WAN**: Wide Area Network, a network that connects devices over a wide geographical area, often spanning multiple cities or countries.
- **Firewall**: A network security device that monitors and controls incoming and outgoing network traffic based on predetermined security rules.
- **Encryption**: The process of encoding data in such a way that only authorized parties can access it, typically using cryptographic algorithms.
- **Backup**: A copy of data stored separately from the original source to protect against data loss due to system failure, corruption, or other disasters.