DNS Server Report

Configuration:

This DNS server is built using Python, leveraging the dnslib library. The server is designed to handle DNS queries and provide appropriate responses based on a predefined list of domains and addresses. It runs on IP 0.0.0.0 and the default DNS port (53). Incoming DNS queries are parsed, and only A records are responded to. PTR queries are not processed. Additionally, the server implements caching to reduce load and speed up queries.

Security Measures:

- **Rate Limiting:** The server restricts the number of requests from a single IP address within a specific time window. This DNS server allows only 10 requests per 10 seconds.
- **IP Blocking:** IP addresses that exceed the specified rate limit are temporarily blocked for 30 seconds. Given that the local IP address 127.0.0.1 is used for testing, a permanent blacklist was not implemented to avoid testing complications.
- **Request Validation:** The server verifies that incoming DNS queries do not exceed 512 bytes, as the DNS server only supports UDP. Additionally, queries must have the RD flag set; otherwise, they are dropped.

Tests/Attacks:

- **Invalid Data:** A malformed request is sent to the DNS server via a Python script to ensure the server can handle it without breaking.
- **Flood:** A Python script sends hundreds of queries in a short period to test the rate limiting feature and ensure correct IP addresses are blocked for the specified duration.
- **Cache Poisoning:** A Python script attempts to insert a fake DNS response into the server's cache.
- **Oversized Packet**: A Python script sends a request to the server that exceeds 512 bytes to ensure it is handled properly.
- **Recursion Desired:** A Python script sends a packet with the RD flag set to false to test if the server drops such requests.

Results: The tests performed as intended, with the server behaving correctly. More detailed information about the tests will be presented in the demonstration video.

Project Overview:

This project was an excellent introduction to networks and security. It was my first college course dealing with these topics. Throughout the project, I became familiar with programming with sockets and network concepts. I also gained a deeper understanding of DNS functionality and importance. Implementing the tests provided practical knowledge of common security practices when programming servers like this DNS server.