Security Assessment Report

Backend Exposure and Reverse Proxy Evaluation

Conducted by: Syed Md Sameer Shah, eJPT (eLearnSecurity Junior Penetration Tester)

Table of Contents

1. Introduction

This document provides an in-depth analysis of a security assessment conducted on a target web infrastructure deployed behind a Cloudflare reverse proxy. The objective was to determine the exposure of backend services, test for performance vulnerabilities, and evaluate the effectiveness of obfuscation and protective layers. This assessment demonstrates critical vulnerabilities that can occur when origin servers are improperly configured, even when protected by enterprise-grade security services like Cloudflare.

All assessments were conducted ethically under explicit authorization and serve as an educational resource for understanding the importance of comprehensive security implementation.

2. Objective

The primary goals of this assessment were:

- Identify and document methods to discover the real IP address behind Cloudflare's reverse proxy
- Evaluate the backend server's response under simulated load conditions
- Assess the potential exposure of critical services such as MySQL
- Demonstrate common Cloudflare bypass techniques and their implications
- Provide comprehensive mitigation strategies to prevent origin server exposure
- Document the methodology and tools used for ethical penetration testing

3. Overview of Testing Environment

The target infrastructure included a web server with the hostname **sathyabama.ac.in** utilizing Cloudflare as a reverse proxy service. The server stack configuration was identified as:

Component	Details
Web Server	nginx/1.18.0 (Ubuntu)
Database	MySQL port 3306 (critically exposed)
Reverse Proxy	Cloudflare (CDN IPs in 103.x.x.x range)
Origin Server	Backend IP in 65.x.x.x range (obfuscated)

Infrastructure Assessment

The target demonstrated a common enterprise setup where Cloudflare acts as the primary protective layer, handling incoming requests and filtering malicious traffic before forwarding legitimate requests to the origin server.

4. Understanding Reverse Proxies and Cloudflare Architecture

4.1 What is a Reverse Proxy?

A **reverse proxy** is a server application that sits between client devices and backend servers, acting as an intermediary for requests from clients seeking resources from servers. Unlike a forward proxy that acts on behalf of clients, a reverse proxy acts on behalf of the server.

Key Functions of Reverse Proxies:

- **Load Distribution**: Distributing incoming requests across multiple backend servers
- **SSL Termination**: Handling SSL/TLS encryption and decryption
- **Caching**: Storing frequently requested content to reduce backend load
- **Security Filtering**: Blocking malicious requests before they reach origin servers
- **Compression**: Reducing bandwidth usage through content compression
- **Rate Limiting**: Controlling request frequency from individual clients

4.2 Cloudflare's Reverse Proxy Architecture

Cloudflare operates one of the world's largest reverse proxy networks, with data centers in over 320 cities globally. When a domain is configured with Cloudflare:

1. **DNS Resolution**: Domain DNS records point to Cloudflare's IP addresses
2. **Request Interception**: All incoming requests hit Cloudflare's edge servers first
3. **Security Processing**: Requests undergo security screening, bot detection, and DDoS filtering
4. **Origin Communication**: Legitimate requests are forwarded to the origin server
5. **Response Optimization**: Responses are cached, compressed, and optimized before delivery

Cloudflare's Network Flow:

```
Client Request → Cloudflare Edge Server → Security Filters → Origin Server
Origin Response → Cloudflare Processing → Cached/Optimized Response → Client
```

5. IP Discovery Methodology and Cloudflare Bypass Techniques

5.1 Understanding Origin IP Exposure

When properly configured, Cloudflare should completely mask the origin server's IP address. However, several common misconfigurations and information leakage vectors can expose the real backend IP, effectively bypassing Cloudflare's protective measures.

5.2 Shodan-Based IP Discovery Method

Primary Discovery Technique: Shodan Internet Scanning Database

Shodan is a search engine that continuously scans the entire internet, indexing devices, servers, and services based on their response banners, SSL certificates, and other identifying information. Unlike traditional search engines that index web pages, Shodan indexes the actual services running on internet-connected devices.

> **How Shodan Works:**
>
> 1. **Continuous Scanning**: Shodan's crawlers scan all IPv4 addresses across all ports
> 2. **Banner Grabbing**: Captures service banners, SSL certificates, and HTTP headers
> 3. **Indexing**: Stores this information in a searchable database
> 4. **Search Interface**: Allows users to search for specific services, software versions, or configurations

Discovery Process in This Assessment:

Step 1: Initial Shodan Search

The assessment utilized Shodan's API to search for the target domain and identify associated IP addresses.

Step 2: Filter Results and Identify False Positives

Initial search results showed multiple IPs in the 103.x.x.x range, which were identified as Cloudflare proxy servers based on:

- Uniform response headers indicating Cloudflare infrastructure
- Geographic distribution matching Cloudflare's edge network
- Similar SSL certificate configurations

Step 3: Identify True Origin Server

The real origin IP (65.x.x.x) was distinguished by:

- Different server response headers (nginx/1.18.0 Ubuntu)
- Unique SSL certificate not matching Cloudflare's edge certificates
- Direct response without Cloudflare's characteristic headers
- Exposed additional services (like the MySQL port 3306)

6. Tools Used

6.1 Network Reconnaissance Tools

a. nmap - Network Mapper

Purpose: Port scanning and service enumeration

Usage in Assessment: Comprehensive port scanning revealed open services and their versions, including the critical MySQL exposure on port 3306.

> **Critical Discovery:**
>
> ```
> PORT     STATE SERVICE VERSION
> 22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu
> 80/tcp   open  http    nginx 1.18.0
> 443/tcp  open  ssl/http nginx 1.18.0
> 3306/tcp open  mysql   MySQL 5.7.33
> ```

b. dig - DNS Lookup Tool

Purpose: DNS record enumeration and analysis

Used for analyzing DNS records and performing reverse DNS lookups on discovered IP addresses.

6.2 Load Testing and Performance Tools

a. hping3 - Network Packet Generator

Purpose: Simulate high-volume network traffic for stress testing

Used to generate SYN flood attacks to test server resilience and response under load.

b. wrk - HTTP Benchmarking Tool

Purpose: Generate high-concurrency HTTP requests

Configured to test server performance under various load conditions with different connection counts and durations.

7. Attack Vectors and Observations

7.1 SYN Flood Attack Simulation

Attack Methodology

The SYN flood attack exploits the TCP three-way handshake process by sending a large volume of SYN packets without completing the connection establishment. This consumes server resources and can lead to denial of service.

Observed Impact

Performance Degradation Metrics:

- **Latency Increase**: From baseline ~50ms to peak ~12,000ms (12 seconds)
- **Packet Loss Rate**: Up to 70% during peak attack periods
- **Connection Timeout**: New connections failing after 30-second timeout
- **Recovery Time**: 5-10 minutes for normal service restoration

7.2 Application Layer Load Testing

Concurrent Connection Testing

Progressive load testing was conducted with increasing connection counts to identify performance thresholds.

Load Level	Connections	Requests/sec	Avg Latency	99th Percentile	Error Rate
Level 1	200	2,340	85ms	156ms	0%
Level 2	500	4,120	121ms	287ms	0%
Level 3	1000	6,890	145ms	445ms	0%
Level 4	1900	8,234	231ms	1,234ms	12%

8. Service Port Exposure Analysis

8.1 Exposed Services Discovery

The assessment revealed that while the web application was protected by Cloudflare, several backend services remained directly accessible from the internet, completely bypassing Cloudflare's protective measures.

Port Scan Results

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu
80/tcp   open  http    nginx 1.18.0
443/tcp  open  ssl/http nginx 1.18.0
3306/tcp open  mysql   MySQL 5.7.33-0ubuntu0.20.04.3
```

8.2 Security Implications of Direct Service Exposure

Critical Risk Factors

Bypass of Web Application Security:

- Database and SSH services accessible without passing through Cloudflare filters
- No geographic restrictions or IP whitelisting applied to backend services
- Vulnerable to automated attack tools and botnets targeting these specific ports
- Complete bypass of all web application firewall protections

9. Cloudflare Bypass Prevention Strategies

9.1 Preventing Shodan-Based Discovery

Strategy 1: Complete Origin Server Isolation

Firewall-Based IP Restriction Procedure

The most effective mitigation is to configure the origin server's firewall to only accept connections from Cloudflare's IP ranges:

Implementation Steps:

1. **Obtain Current Cloudflare IP Ranges**: Download the latest IP ranges from Cloudflare's official sources
2. **Configure Firewall Rules**: Set up rules to only allow HTTP/HTTPS traffic from Cloudflare IPs
3. **Block Direct Access**: Deny all other incoming connections to web services
4. **Restrict Administrative Access**: Limit SSH access to specific management IP addresses
5. **Database Security**: Block external access to database ports completely
6. **Regular Updates**: Implement automated updates for Cloudflare IP ranges

Strategy 2: Change Origin Server IP Address

Complete IP Address Rotation Procedure:

Migration Steps:

1. Provision new server with different IP address
2. Install and configure all required services
3. Migrate website files and databases
4. Update Cloudflare DNS configuration
5. Configure security rules on new server
6. Test functionality through Cloudflare
7. Monitor for 24-48 hours
8. Decommission old server
9. Verify old IP no longer responds

10. Recommendations and Mitigations

10.1 Immediate Actions Required

Critical Priority (Implement within 24 hours):

1. Block Direct IP Access:

- Configure firewall to only allow Cloudflare IP ranges
- Block all direct access to ports 80 and 443
- Implement default deny policy for incoming connections

2. Secure Database Access:

- Completely block external access to MySQL port 3306
- Allow database connections only from localhost
- Review and strengthen database authentication

3. Restrict Administrative Access:

- Limit SSH access to specific management IP addresses
- Implement key-based authentication only
- Disable password authentication for SSH

High Priority (Implement within 1 week):

4. Enable Authenticated Origin Pulls:

- Configure Cloudflare Authenticated Origin Pulls
- Update web server to validate client certificates
- Test and verify proper functionality

5. Implement Monitoring:

- Set up automated Shodan monitoring
- Configure alerts for new IP exposures
- Establish incident response procedures

11. Conclusion

This security assessment has revealed significant vulnerabilities in the current infrastructure configuration, primarily stemming from the exposure of the origin server's IP address and critical services bypassing Cloudflare's protective measures. While Cloudflare provides excellent protection for web traffic, the assessment demonstrates that incomplete implementation can leave significant security gaps.

Key Findings Summary:

1. **Origin IP Exposure**: The real server IP was discoverable through Shodan, completely bypassing Cloudflare protection
2. **Critical Service Exposure**: MySQL database and SSH services were directly accessible from the internet
3. **Performance Vulnerabilities**: The server showed significant performance degradation under load testing
4. **Configuration Gaps**: Firewall rules were insufficient to prevent direct access to backend services

Final Recommendations:

1. **Prioritize Critical Fixes**: Implement firewall restrictions and service isolation immediately
2. **Plan Infrastructure Changes**: Schedule origin IP changes and architectural improvements
3. **Establish Monitoring**: Set up continuous security monitoring and alerting
4. **Regular Assessment**: Conduct periodic security assessments to identify new vulnerabilities
5. **Documentation**: Maintain comprehensive security documentation and procedures

---

Assessment Completed by: Syed Md Sameer Shah, eJPT

Date: 30 May, 2025

Classification: I honestly think no one cares....