Securing OLAMA API using NGINX Reverse Proxy (Windows Setup)

Overview

In the previous lab, we demonstrated how an exposed OLAMA API can be exploited remotely without any authentication. In this lab, we'll secure that exposure using NGINX as a reverse proxy with basic authentication. This will act as a gateway, enforcing access control before requests reach the OLAMA API.

This document provides a step-by-step guide to mitigate security risks associated with exposing OLAMA APIs directly over the internet by configuring NGINX as a reverse proxy on a Windows Azure VM.

Pre-requisites

- Azure-hosted Windows VM with OLAMA already installed and running on default port 11434.

- Internet access and RDP access to the VM.

- Administrative privileges on the Windows VM.

Step-by-Step Mitigation Guide

1. Download and Install NGINX (Windows)

1. Visit https://nginx.org/en/download.html and download the stable Windows version.
2. Extract the downloaded ZIP file to a preferred location (e.g., C:\nginx).

2. Configure Basic Authentication

- Go to https://www.web2generators.com/apache-tools/htpasswd-generator.
- Enter a username and password of your choice, e.g., `lama` and `123456`.
- Click 'Generate .htpasswd file'.
- Copy the generated string and paste it into a new Notepad file.
- Save the file as `.htpasswd` (remove the `.txt` extension).
- Place this file inside the NGINX config directory (e.g., C:\nginx\).

Note:

- The password is hashed, and it's not stored in plain text.
- Enable file extensions in Windows Explorer to ensure .txt is removed.

3. Update NGINX Configuration File

1. Open `nginx.conf` in a text editor from the NGINX config folder.
2. Replace or add the following configuration block inside the `http` section:

```
server {
  listen 80;
  server_name localhost;

  location / {
    proxy_pass http://localhost:11434/;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    auth_basic "Ollama Secure API";
    auth_basic_user_file C:/nginx/.htpasswd;
  }
}
```

4. Modify Azure NSG Rules

1. Go to the Azure Portal > Networking > Inbound Port Rules.
2. Remove public rule for port 11434.
3. Add a rule to allow port 80 (HTTP) from required IP ranges.

5. Start NGINX

1. Open Command Prompt as Administrator.
2. Navigate to the NGINX folder.
3. Run: `start nginx.exe`

6. Test Secure Access with cURL (Recommended)

1. Use the following command:

`curl -u yourusername:yourpassword http://<public-ip>/api/tags`

Important Note: Without username and password, it should result with unauthorized message

2. The expected failure when no credentials are passed:

curl http://<public-ip>/api/tags

This should return a 401 Unauthorized message, confirming that unauthenticated access is blocked.

Conclusion

By using NGINX as a reverse proxy and enabling authentication, you significantly reduce the attack surface of your exposed OLAMA APIs. This setup should be a baseline for any production or public-facing environments.

As an advanced step, you can configure NGINX to serve traffic over HTTPS using certificates from Let's Encrypt. This will ensure credentials are encrypted during transmission — a must-have for production-grade deployments.