Installation Guide

BigLan Network Monitoring System

Contents

Introduction

This document guides you through the steps of installing the BigLan Network Monitoring System.

NOTE: Successful installation and configuration requires experience and knowledge of Linux server and web server components (such as Apache).
The application also includes a Laravel-based web interface, so Laravel experience is also highly recommended for more efficient troubleshooting and customization.

IMPORTANT! For optimal and secure operation of the application, it is recommended to use it within a local area network (LAN) or intranet. This environment minimizes security risks, as sensitive network data does not leave the internal network. Although the application may be able to communicate over the Internet, this option should only be used over an encrypted channel (e.g. VPN) and with appropriate firewall rules in place. Use over the open Internet poses increased security risks, including the possibility of unauthorized access and data interception.

WARNING! Although the system can operate with a self-signed certificate, which can be used for testing purposes or on an internal network, this solution is not secure. Browsers may display a warning and the communication is not properly encrypted. It is strongly recommended to create your own internal certificate authority (Local CA). This method allows the web server operating on the internal network to have a trusted certificate. Certificates issued by your own CA must be installed by all devices on the network (for example, client computers) to be reliable.

WARNING! The user is responsible for the security settings and risks encountered when using the product on an external network.

IMPORTANT! For the application to function properly, the server must have access to the VLANs in which the monitored workstations and servers are located.

IMPORTANT! For the application to function properly, the server must have at least one IP address in each IP range where the monitored workstations and servers are located. This ensures that the application can access different network segments.

Hardware requirements:

The basic hardware requirements for the application are as follows, depending on the number of monitored devices:

- Processor: Quad-core CPU
- Memory: 4GB RAM
- Storage: 40 GB free space

These values are rough estimates and apply to a typical network environment where the load consists of:

- Monitoring of 500 workstations/servers
- Monitor 100 network printers
- Monitor 100 monitoring status changes per minute

IMPORTANT! For larger networks and higher loads, hardware requirements may also increase.

Installation

1. Installing Ubuntu Server 24.04 LTS

https://ubuntu.com/tutorials/install-ubuntu-server

2. Installing Apache 2

sudo apt install apache2

3. Installing MySQL

sudo apt install mysql-server

After installation, it is worth running the MySQL security script .

sudo mysql_secure_installation

4. Create a database and set up a user

mysql -u [username] –p

CREATE DATABASE [database_name e.g. : biglan] CHARACTER SET utf8mb4 COLLATE utf8mb4_ unicode _ ci ;

*CREATE USER ' admin '@' localhost ' IDENTIFIED BY ' password '; GRANT ALL PRIVILEGES ON *.* TO ' admin '@' localhost ' WITH GRANT OPTION; FLUSH PRIVILEGES;*

exit ;

5. Installing SNMP

sudo apt update
sudo apt install snmp snmpd

6. Installing and configuring PHP

sudo apt install php libapache2-mod-php php-mysql php-snmp php-xml php-zip php-curl

7. Firewall settings

sudo ufw allow 8080/tcp

8. Git (if needed)

sudo apt install git

9. Create a folder for your files

sudo mkdir /var/www/biglan

Permission for the web server to write to the folder

sudo chown -R $USER:www-data /var/www/biglan

10. Download project from GitHub to folder

cd /var/www/biglan
git clone https://github.com/atlantisguru/biglan-server.git

11. Installing Composer

sudo apt install composer

12. public folder setup

sudo nano /etc/apache2/sites-available/000-default.conf

```
<VirtualHost *:80>
   ServerAdmin webmaster@localhost
   DocumentRoot /var/www/biglan/public
<Directory /var/www/biglan/public>
      AllowOverride All
      Require stand granted
</Directory>
   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

13. Enable SSL

sudo a2enmod ssl
sudo a2ensite default-ssl
sudo nano /etc/apache2/sites-available/default-ssl.conf

```
<VirtualHost *:443>
```

```
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/biglan/public
<Directory /var/www/biglan/public>
        AllowOverride All
        Require stand granted
</Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

sudo a2enmod rewrite

14. Laravel APP key generation

cd /var/www/biglan
php artisan key : generate

15. Laravel .env file settings

cd /var/www/biglan

sudo cp .env.example .env

sudo nano .env

```
APP_NAME=BigLan
APP_ENV=local
APP_TIMEZONE=[Your time zone]
APP_URL=https://[Server IP Address]

MASTER_KEY=[A 32-character token generated from lowercase and uppercase letters and numbers]

APP_LOCALE=hu
APP_FALLBACK_LOCALE=en

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=biglan
DB_USERNAME=[Username]
DB_PASSWORD=[Password]
```

16. Apache restart web server

sudo systemctl restart apache2

17. Database migration

php artisan cache:table
php artisan migrate

18. Laravel clear cache

php artisan cache:clear
php artisan config:cache

19. Populate database with default data

php artisan db:seed

20. Schedule a CronJob for Laravel

```
sudo crontab -e
```

Choose option 1 (edit with nano).

Add this line to the document:

```
* * * * * cd /var/www/biglan && php artisan schedule:run >>
/dev/null 2>&1
```

21. Load the website in your browser and register a user

After installation, the first registered user will be automatically confirmed and given full administrative privileges.

Find the BigLan User Guide in the *Documents* menu.

Troubleshooting

Restart the web server after every operation that affects the web server.

sudo systemctl restart apache2

Run a config cache flush after every modification to the Laravel's .env file. Always initiate this from the project folder (/var/www /biglan).

php artisan cache: clear
php artisan config :cache

To detect database errors, it is recommended to install phpMyAdmin .

sudo apt install phpmyadmin

sudo nano /etc/apache2/conf-available/phpmyadmin.conf

```
#
# Apache config file for phpMyAdmin
#

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
   Options Indexes FollowSymLinks
   DirectoryIndex index.php

<IfModule mod_authz_core.c>
# Apache 2.4
      Require stand granted
</IfModule>
<IfModule !mod_authz_core.c>
# Apache 2.2
      Order Deny,Allow
      Deny from All
      Allow from 127.0.0.1
      Allow from : :1
</IfModule>
</Directory>
```

sudo a2enconf phpmyadmin
sudo systemctl restart apache2