Cybersecurity Threat Detection Dashboard

Configuration & Installation Guide

Table of Contents

System Requirements

Minimum Hardware Requirements

- **CPU**: 2-core processor (Intel i3 or AMD equivalent)
- **RAM**: 4GB minimum, 8GB recommended
- **Storage**: 20GB free disk space
- **Network**: Stable internet connection for threat intelligence feeds

Software Requirements

- **Operating System**:
  - Linux (Ubuntu 20.04+ / CentOS 8+ / RHEL 8+)
  - Windows Server 2019+
  - macOS 10.15+
- **Web Server**: Apache 2.4+ or Nginx 1.18+
- **PHP**: Version 7.4+ (PHP 8.1+ recommended)
- **Database**: MySQL 8.0+ or MariaDB 10.5+
- **Web Browser**: Chrome 90+, Firefox 88+, Safari 14+

PHP Extensions Required

```bash
# Ubuntu/Debian
sudo apt-get install php-mysql php-curl php-json php-mbstring php-xml php-zip php-gd

# CentOS/RHEL
sudo yum install php-mysql php-curl php-json php-mbstring php-xml php-zip php-gd
```

Pre-Installation Setup

1. Update System Packages

```bash
# Ubuntu/Debian
sudo apt update && sudo apt upgrade -y

# CentOS/RHEL
sudo yum update -y
```

2. Install LAMP/LEMP Stack

Apache Installation (Ubuntu/Debian)

```bash
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
```

Nginx Installation (Alternative)

```bash
sudo apt install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

MySQL Installation

bash

```
sudo apt install mysql-server -y
sudo systemctl start mysql
sudo systemctl enable mysql
sudo mysql_secure_installation
```

PHP Installation

bash

```
sudo apt install php libapache2-mod-php php-mysql -y
sudo systemctl restart apache2
```

Database Configuration

1. Create MySQL Database and User

sql

```
-- Login to MySQL as root
mysql -u root -p

-- Create database
CREATE DATABASE cybersecurity_dashboard CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

-- Create user
CREATE USER 'cyber_user'@'localhost' IDENTIFIED BY 'SecurePassword123!';

-- Grant privileges
GRANT ALL PRIVILEGES ON cybersecurity_dashboard.* TO 'cyber_user'@'localhost';
FLUSH PRIVILEGES;

-- Exit MySQL
EXIT;
```

2. Database Schema Setup

sql

```sql
-- Use the database
USE cybersecurity_dashboard;

-- Create threats table
CREATE TABLE threats (
    id INT AUTO_INCREMENT PRIMARY KEY,
    threat_type VARCHAR(100) NOT NULL,
    severity ENUM('Low', 'Medium', 'High', 'Critical') NOT NULL,
    source_ip VARCHAR(45),
    destination_ip VARCHAR(45),
    description TEXT,
    status ENUM('Active', 'Resolved', 'Under Investigation') DEFAULT 'Active',
    detected_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    resolved_at TIMESTAMP NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    INDEX idx_severity (severity),
    INDEX idx_status (status),
    INDEX idx_detected_at (detected_at)
);

-- Create security_events table
CREATE TABLE security_events (
    id INT AUTO_INCREMENT PRIMARY KEY,
    event_type VARCHAR(100) NOT NULL,
    severity VARCHAR(20) NOT NULL,
    source VARCHAR(255),
    message TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_event_type (event_type),
    INDEX idx_timestamp (timestamp)
);

-- Create users table for dashboard access
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role ENUM('admin', 'analyst', 'viewer') DEFAULT 'viewer',
    last_login TIMESTAMP NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
```

```
);

-- Create system_logs table
CREATE TABLE system_logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    log_level ENUM('DEBUG', 'INFO', 'WARNING', 'ERROR', 'CRITICAL') NOT NULL,
    message TEXT NOT NULL,
    source VARCHAR(100),
    user_id INT,
    ip_address VARCHAR(45),
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE SET NULL,
    INDEX idx_log_level (log_level),
    INDEX idx_timestamp (timestamp)
);

-- Insert sample data
INSERT INTO threats (threat_type, severity, source_ip, destination_ip, description, status) VALUES
('Malware Detection', 'High', '192.168.1.100', '10.0.0.5', 'Suspicious executable detected on endpoint', 'Active'),
('DDoS Attack', 'Critical', '203.0.113.1', '192.168.1.1', 'High volume traffic detected from external source', 'Under Investiga
('Unauthorized Access', 'Medium', '192.168.1.150', '192.168.1.10', 'Failed login attempts detected', 'Active'),
('Data Exfiltration', 'High', '192.168.1.75', '8.8.8.8', 'Large data transfer to external server', 'Resolved');

INSERT INTO security_events (event_type, severity, source, message) VALUES
('Login Failure', 'Medium', 'Web Portal', 'Multiple failed login attempts detected'),
('Firewall Block', 'Low', 'Network Firewall', 'Blocked connection attempt from suspicious IP'),
('Antivirus Alert', 'High', 'Endpoint Protection', 'Malware quarantined on workstation WS-001'),
('Network Intrusion', 'Critical', 'IDS System', 'Potential intrusion attempt detected on network segment');
```

Web Server Configuration

Apache Configuration

Create virtual host configuration file:

```
bash

sudo nano /etc/apache2/sites-available/cybersecurity-dashboard.conf
```

Add the following configuration:

apache

```
<VirtualHost *:80>
    ServerName cybersecurity-dashboard.local
    DocumentRoot /var/www/cybersecurity-dashboard

    <Directory /var/www/cybersecurity-dashboard>
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/cybersecurity-dashboard_error.log
    CustomLog ${APACHE_LOG_DIR}/cybersecurity-dashboard_access.log combined
</VirtualHost>
```

Enable the site and required modules:

bash

```
sudo a2ensite cybersecurity-dashboard.conf
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Nginx Configuration (Alternative)

```nginx
server {
    listen 80;
    server_name cybersecurity-dashboard.local;
    root /var/www/cybersecurity-dashboard;
    index index.php index.html;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Application Installation

1. Create Project Directory

```bash
sudo mkdir -p /var/www/cybersecurity-dashboard
sudo chown -R www-data:www-data /var/www/cybersecurity-dashboard
sudo chmod -R 755 /var/www/cybersecurity-dashboard
```

2. Create Configuration File

```bash
sudo nano /var/www/cybersecurity-dashboard/config/database.php
```

Add database configuration:

php

```php
<?php
return [
    'host' => 'localhost',
    'database' => 'cybersecurity_dashboard',
    'username' => 'cyber_user',
    'password' => 'SecurePassword123!',
    'charset' => 'utf8mb4',
    'options' => [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
        PDO::ATTR_EMULATE_PREPARES => false,
    ],
];
?>
```

3. Create Environment Configuration

bash

```bash
sudo nano /var/www/cybersecurity-dashboard/.env
```

Add environment variables:

env

```
# Database Configuration
DB_HOST=localhost
DB_NAME=cybersecurity_dashboard
DB_USER=cyber_user
DB_PASSWORD=SecurePassword123!

# Application Configuration
APP_NAME="Cybersecurity Dashboard"
APP_ENV=production
APP_DEBUG=false
APP_URL=http://cybersecurity-dashboard.local

# Security Configuration
SESSION_LIFETIME=7200
CSRF_TOKEN_EXPIRY=3600
MAX_LOGIN_ATTEMPTS=5
LOCKOUT_DURATION=1800

# API Configuration
API_RATE_LIMIT=100
API_RATE_WINDOW=3600

# Email Configuration (Optional)
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=your-email@gmail.com
MAIL_PASSWORD=your-app-password
```

4. Set File Permissions

bash

```
sudo chown -R www-data:www-data /var/www/cybersecurity-dashboard
sudo find /var/www/cybersecurity-dashboard -type d -exec chmod 755 {} \;
sudo find /var/www/cybersecurity-dashboard -type f -exec chmod 644 {} \;
sudo chmod 600 /var/www/cybersecurity-dashboard/.env
```

Security Configuration

1. SSL Certificate Setup (Recommended)

Using Let's Encrypt:

```bash
sudo apt install certbot python3-certbot-apache -y
sudo certbot --apache -d cybersecurity-dashboard.local
```

Self-Signed Certificate (Development):

```bash
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/private/cybersecurity-dashboard.key \
    -out /etc/ssl/certs/cybersecurity-dashboard.crt
```

2. Firewall Configuration

```bash
# UFW (Ubuntu)
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 22/tcp
sudo ufw enable

# iptables (Alternative)
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

3. PHP Security Settings

Edit PHP configuration:

```bash
sudo nano /etc/php/8.1/apache2/php.ini
```

Update security settings:

ini

```ini
expose_php = Off
display_errors = Off
log_errors = On
error_log = /var/log/php_errors.log
max_execution_time = 30
max_input_time = 60
memory_limit = 256M
post_max_size = 50M
upload_max_filesize = 50M
session.cookie_httponly = 1
session.cookie_secure = 1
session.use_strict_mode = 1
```

4. Create Admin User

bash

```bash
php /var/www/cybersecurity-dashboard/scripts/create_admin.php
```

Testing & Verification

1. System Health Check

bash

```bash
# Check web server status
sudo systemctl status apache2  # or nginx

# Check MySQL status
sudo systemctl status mysql

# Check PHP configuration
php -v
php -m | grep mysql
```

2. Database Connection Test

bash

```bash
mysql -u cyber_user -p cybersecurity_dashboard -e "SELECT COUNT(*) FROM threats;"
```

3. Web Application Test

1. Open browser and navigate to: `http://cybersecurity-dashboard.local`
2. Verify login page loads correctly
3. Test admin login credentials
4. Check dashboard functionality
5. Verify threat detection data displays

4. Log File Verification

```bash
# Check Apache logs
sudo tail -f /var/log/apache2/cybersecurity-dashboard_access.log
sudo tail -f /var/log/apache2/cybersecurity-dashboard_error.log

# Check PHP logs
sudo tail -f /var/log/php_errors.log

# Check MySQL logs
sudo tail -f /var/log/mysql/error.log
```

Troubleshooting

Common Issues and Solutions

1. Database Connection Failed

Problem: Cannot connect to MySQL database **Solution**:

```bash
# Check MySQL service
sudo systemctl status mysql
sudo systemctl restart mysql

# Verify user privileges
mysql -u root -p -e "SHOW GRANTS FOR 'cyber_user'@'localhost';"
```

2. Permission Denied Errors

Problem: Web server cannot access files **Solution**:

bash

```
sudo chown -R www-data:www-data /var/www/cybersecurity-dashboard
sudo chmod -R 755 /var/www/cybersecurity-dashboard
```

3. PHP Modules Missing

Problem: Required PHP extensions not installed **Solution**:

bash

```
sudo apt install php-mysql php-curl php-json php-mbstring php-xml
sudo systemctl restart apache2
```

4. Session Issues

Problem: User sessions not working **Solution**:

bash

```
# Check session directory permissions
sudo chmod 1733 /var/lib/php/sessions
sudo chown root:root /var/lib/php/sessions
```

Maintenance

Daily Tasks

- Monitor system logs for errors
- Check threat detection alerts
- Verify database backup completion
- Review security event logs

Weekly Tasks

- Update threat intelligence feeds
- Review user access logs
- Check system resource usage
- Update security signatures

Monthly Tasks

- Apply security patches
- Review and rotate API keys
- Audit user accounts and permissions
- Performance optimization review

Backup Strategy

bash

```bash
#!/bin/bash
# Daily backup script
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/var/backups/cybersecurity-dashboard"

# Create backup directory
mkdir -p $BACKUP_DIR

# Database backup
mysqldump -u cyber_user -p cybersecurity_dashboard > $BACKUP_DIR/db_backup_$DATE.sql

# Application files backup
tar -czf $BACKUP_DIR/app_backup_$DATE.tar.gz /var/www/cybersecurity-dashboard

# Keep only last 7 days of backups
find $BACKUP_DIR -name "*.sql" -mtime +7 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +7 -delete
```

Monitoring Setup

Create monitoring script:

bash

```bash
#!/bin/bash
# System monitoring script
LOG_FILE="/var/log/cybersecurity-dashboard-monitor.log"

# Check web server
if ! systemctl is-active --quiet apache2; then
    echo "$(date): Apache is not running" >> $LOG_FILE
    systemctl restart apache2
fi

# Check database
if ! systemctl is-active --quiet mysql; then
    echo "$(date): MySQL is not running" >> $LOG_FILE
    systemctl restart mysql
fi

# Check disk space
DISK_USAGE=$(df /var/www/cybersecurity-dashboard | awk 'NR==2 {print $5}' | sed 's/%//')
if [ $DISK_USAGE -gt 80 ]; then
    echo "$(date): Disk usage is at ${DISK_USAGE}%" >> $LOG_FILE
fi
```

Additional Configuration

Email Alerts Setup

Configure SMTP settings in `/var/www/cybersecurity-dashboard/config/mail.php`:

php

```php
<?php
return [
    'host' => 'smtp.gmail.com',
    'port' => 587,
    'encryption' => 'tls',
    'username' => 'your-email@gmail.com',
    'password' => 'your-app-password',
    'from_address' => 'noreply@cybersecurity-dashboard.local',
    'from_name' => 'Cybersecurity Dashboard',
];
?>
```

API Rate Limiting

Configure in `/var/www/cybersecurity-dashboard/config/api.php`:

```php
<?php
return [
    'rate_limit' => 100,
    'rate_window' => 3600,
    'max_requests_per_minute' => 60,
    'blocked_duration' => 1800,
];
?>
```

This completes the comprehensive Configuration and Installation Guide for the Cybersecurity Threat Detection Dashboard.