8: Vaultwarden - Password Management Solution

Vaultwarden is a lightweight, self-hosted password management solution that provides full Bitwarden API compatibility while using significantly fewer resources than the official Bitwarden server. It supports all Bitwarden clients and offers enterprise-grade security features. For comprehensive configuration options, API documentation, and advanced features, please refer to the official Vaultwarden documentation.

Architecture Overview

Vaultwarden provides the following core functionality:

- **Bitwarden API Compatibility** - Full compatibility with all Bitwarden clients
- **End-to-End Encryption** - AES-256 encryption for all data
- **Multi-User Support** - Organization and user management
- **WebSocket Support** - Real-time synchronization
- **Admin Panel** - Comprehensive management interface
- **Database Flexibility** - SQLite, PostgreSQL, MySQL support

Prerequisites

Before installing Vaultwarden, ensure the following infrastructure is in place:

- **Traefik installed** (from Chapter 4)
- **Docker running** (from Chapter 3)
- **Borgmatic installed** (from Chapter 5) - Automated backup protection
- **Domain configured** (from Chapter 4.5)
- **SSL certificates** (Let's Encrypt via Traefik)

Installation Methods

Via Infinity Tools Menu

Navigate to the Infinity Tools menu and select:

```
□□APPLICATIONS → Vaultwarden → Install
```

Command Line Installation

```
# Direct script execution
sudo bash /opt/InfinityTools/Solutions/setup-vaultwarden.sh --install

# With environment variables
export VW_DOMAIN="vault.domain.com"
export VW_USE_TRAEFIK="true"
export VW_SIGNUPS="false"
export PROXY_NETWORK="proxy"
sudo -E bash /opt/InfinityTools/Solutions/setup-vaultwarden.sh --install
```

Configuration Parameters

Required Configuration

During installation, you'll configure:

- **SSL Mode:** Traefik integration or standalone
- **Domain:** FQDN for web vault access
- **Signup Policy:** Open registration or admin-only
- **Admin Token:** Generated automatically for admin access

Environment Variables

```
# SSL and Domain Configuration
export VW_USE_TRAEFIK="true"              # Use Traefik for SSL termination
export VW_DOMAIN="vault.domain.com"       # FQDN for web vault
export VW_PORT="8443"                     # Port for standalone mode

# User Management
export VW_SIGNUPS="false"                 # Disable open registration
export VW_SIGNUPS_VERIFY="true"           # Require email verification

# Network Configuration
```

```
export PROXY_NETWORK="proxy"             # Docker network name
```

Generated Configuration

Docker Compose Configuration (Traefik Mode)

Location: `/opt/speedbits/vaultwarden/docker-compose.yml`

```
version: '3.8'

services:
  vaultwarden:
    image: vaultwarden/server:1.34.3
    container_name: vaultwarden
    restart: unless-stopped
    environment:
      DOMAIN: https://vault.domain.com
      ADMIN_TOKEN_FILE: /run/secrets/admin_token.txt
      SIGNUPS_ALLOWED: "false"
      SIGNUPS_VERIFY: "true"
      DATABASE_URL: /data/db.sqlite3
      WEBSOCKET_ENABLED: "true"
      WEBSOCKET_ADDRESS: 0.0.0.0
      WEBSOCKET_PORT: 3012
    volumes:
      - /opt/speedbits/vaultwarden/data:/data
      - /opt/speedbits/vaultwarden/admin_token.txt:/run/secrets/admin_token.txt:ro
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.vaultwarden.rule=Host(`vault.domain.com`)"
      - "traefik.http.routers.vaultwarden.entrypoints=websecure"
      - "traefik.http.routers.vaultwarden.tls.certresolver=myresolver"
      - "traefik.http.services.vaultwarden.loadbalancer.server.port=80"
      - "traefik.http.routers.vaultwarden-websocket.rule=Host(`vault.domain.com`) &&
Path(`/notifications/hub`)"
      - "traefik.http.routers.vaultwarden-websocket.entrypoints=websecure"
      - "traefik.http.routers.vaultwarden-websocket.tls.certresolver=myresolver"
      - "traefik.http.services.vaultwarden-websocket.loadbalancer.server.port=3012"
```

```
    networks:
      - proxy

networks:
  proxy:
    external: true
```

Standalone Configuration

For environments without Traefik:

```
version: '3.8'

services:
  vaultwarden:
    image: vaultwarden/server:1.34.3
    container_name: vaultwarden
    restart: unless-stopped
    environment:
      DOMAIN: https://localhost:8443
      ADMIN_TOKEN_FILE: /run/secrets/admin_token.txt
      SIGNUPS_ALLOWED: "false"
      DATABASE_URL: /data/db.sqlite3
      WEBSOCKET_ENABLED: "true"
      WEBSOCKET_ADDRESS: 0.0.0.0
      WEBSOCKET_PORT: 3012
      ROCKET_TLS: '{certs="/ssl/vaultwarden.crt",key="/ssl/vaultwarden.key"}'
      ROCKET_PORT: 443
    volumes:
      - /opt/speedbits/vaultwarden/data:/data
      - /opt/speedbits/vaultwarden/admin_token.txt:/run/secrets/admin_token.txt:ro
      - /opt/speedbits/vaultwarden/ssl:/ssl:ro
    ports:
      - "8443:443"
    networks:
      - proxy
```

Security Configuration

Admin Token Management

Admin tokens are stored securely and provide access to the admin panel:

```
# Generate new admin token
openssl rand -base64 48

# Store in secure location
echo "generated_token" > /opt/speedbits/vaultwarden/admin_token.txt
chmod 600 /opt/speedbits/vaultwarden/admin_token.txt
```

Security Headers

Traefik middleware provides comprehensive security headers:

```
labels:
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.X-Content-Type-Options=nosniff"
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.X-Frame-Options=SAMEORIGIN"
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.X-XSS-Protection=1; mode=block"
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.Strict-Transport-Security=max-age=31536000; includeSubDomains"
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.Referrer-Policy=strict-origin-when-cross-origin"
  - "traefik.http.middlewares.vaultwarden-security.headers.customResponseHeaders.Content-Security-Policy=default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'; img-src 'self' data: https:; connect-src 'self' wss://vault.domain.com https://vault.domain.com; font-src 'self' data:; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none';"
```

Database Configuration

SQLite (Default)

Vaultwarden uses SQLite by default for simplicity:

```
environment:
  DATABASE_URL: /data/db.sqlite3
```

PostgreSQL Configuration

For production environments, PostgreSQL is recommended:

```
environment:
  DATABASE_URL: postgresql://vaultwarden:password@postgres:5432/vaultwarden

# Add PostgreSQL service
services:
  postgres:
    image: postgres:15-alpine
    container_name: vaultwarden-postgres
    restart: unless-stopped
    environment:
      POSTGRES_DB: vaultwarden
      POSTGRES_USER: vaultwarden
      POSTGRES_PASSWORD: secure_password
    volumes:
      - /opt/speedbits/vaultwarden/postgres:/var/lib/postgresql/data
    networks:
      - proxy
```

Advanced Configuration

Environment Variables

Vaultwarden supports extensive configuration via environment variables:

```
environment:
  # Domain and SSL
  DOMAIN: https://vault.domain.com
  ROCKET_TLS: '{certs="/ssl/vaultwarden.crt",key="/ssl/vaultwarden.key"}'

  # Database
```

```
  DATABASE_URL: /data/db.sqlite3

  # User Management
  SIGNUPS_ALLOWED: "false"
  SIGNUPS_VERIFY: "true"
  SIGNUPS_VERIFY_RESEND_TIME: "3600"
  SIGNUPS_VERIFY_RESEND_LIMIT: "6"

  # Security
  ADMIN_TOKEN_FILE: /run/secrets/admin_token.txt
  INVITATIONS_ALLOWED: "true"
  INVITATION_ORG_NAME: "Organization Name"

  # WebSocket
  WEBSOCKET_ENABLED: "true"
  WEBSOCKET_ADDRESS: 0.0.0.0
  WEBSOCKET_PORT: 3012

  # SMTP (for email verification)
  SMTP_HOST: smtp.example.com
  SMTP_FROM: vaultwarden@example.com
  SMTP_PORT: 587
  SMTP_SECURITY: starttls
  SMTP_USERNAME: smtp_user
  SMTP_PASSWORD: smtp_password
```

Organization Management

Configure organization settings for team password sharing:

```
environment:
  ORG_CREATION_USERS: "admin@domain.com"
  ORG_NAME: "Company Name"
  ORG_OWNER_EMAIL: "admin@domain.com"
```

Monitoring and Logging

Health Checks

```
# Add health check to docker-compose.yml
healthcheck:
  test: ["CMD", "curl", "-f", "http://localhost:80/alive"]
  interval: 30s
  timeout: 10s
  retries: 3
  start_period: 30s
```

Logging Configuration

```
environment:
  LOG_LEVEL: info
  LOG_FILE: /data/vaultwarden.log
  EXTENDED_LOGGING: "true"
  LOG_TIMESTAMP: "true"
```

Backup and Recovery

Data Backup

Vaultwarden data is stored in the mounted volume:

```
# Backup Vaultwarden data
tar -czf vaultwarden-backup-$(date +%Y%m%d).tar.gz -C /opt/speedbits/vaultwarden/data .

# Backup configuration
cp /opt/speedbits/vaultwarden/docker-compose.yml /backup/vaultwarden-compose.yml
cp /opt/speedbits/vaultwarden/admin_token.txt /backup/vaultwarden-admin-token.txt
```

Disaster Recovery

```
# Restore from backup
tar -xzf vaultwarden-backup-20241201.tar.gz -C /opt/speedbits/vaultwarden/data/

# Restart service
cd /opt/speedbits/vaultwarden
```

```
docker compose down
docker compose up -d
```

Performance Optimization

Resource Limits

```
services:
  vaultwarden:
    deploy:
      resources:
        limits:
          memory: 512M
          cpus: '0.5'
        reservations:
          memory: 256M
          cpus: '0.25'
```

Database Optimization

For PostgreSQL, configure connection pooling:

```
environment:
  DATABASE_URL:
postgresql://vaultwarden:password@postgres:5432/vaultwarden?sslmode=require&max_connections=20
```

Troubleshooting

Common Issues

WebSocket Connection Failures:

```
# Check WebSocket configuration
docker logs vaultwarden | grep -i websocket

# Verify Traefik routing
```

```
curl -H "Host: vault.domain.com" http://localhost/notifications/hub
```

Database Connection Issues:

```
# Check database file permissions
ls -la /opt/speedbits/vaultwarden/data/db.sqlite3

# Verify database integrity
sqlite3 /opt/speedbits/vaultwarden/data/db.sqlite3 "PRAGMA integrity_check;"
```

SSL Certificate Problems:

```
# Check Traefik certificate status
docker logs traefik | grep -i acme

# Verify domain resolution
dig vault.domain.com
nslookup vault.domain.com
```

Debug Commands

```
# Container status
docker ps | grep vaultwarden

# Container logs
docker logs vaultwarden

# Container exec
docker exec -it vaultwarden /bin/sh

# Network connectivity
docker network inspect proxy

# Port binding
ss -tulnp | grep :443
```

Integration with Other Services

Borgmatic Backup Integration

Include Vaultwarden in automated backups:

```
# Add to borgmatic configuration
locations:
  directories:
    - /opt/speedbits/vaultwarden/data

# Exclude temporary files
exclude_patterns:
  - "*.tmp"
  - "*.log"
```

Monitoring Integration

Add Vaultwarden to monitoring systems:

```
# Health check endpoint
curl -f https://vault.domain.com/alive

# Metrics endpoint (if enabled)
curl https://vault.domain.com/metrics
```

Security Best Practices

Access Control

- Disable open registration in production
- Use strong admin tokens
- Implement IP whitelisting for admin access
- Enable two-factor authentication for all users

Network Security

- Use Traefik for SSL termination
- Implement rate limiting
- Configure fail2ban for brute force protection

- Regular security updates

Next Steps

With Vaultwarden installed and configured, you can now:

- Configure user accounts and organizations
- Set up SMTP for email notifications
- Implement backup strategies
- Integrate with existing identity providers

Verification Checklist

- [ ] Vaultwarden container running and healthy
- [ ] Web vault accessible via HTTPS
- [ ] Admin panel accessible with admin token
- [ ] WebSocket connections working
- [ ] Database accessible and writable
- [ ] SSL certificates valid

Next: Application Deployment and Management (Coming Soon)

Revision #3
Created 31 October 2025 13:22:13 by bjoern
Updated 17 November 2025 16:36:29 by bjoern