20: Warpgate - SSH Bastion Host

Warpgate is a modern SSH/RDP bastion host providing centralized access control, session recording, and web-based management. It acts as a gateway for all SSH connections, reducing the attack surface by eliminating direct server access.

For protocol specifications, advanced configuration, and technical documentation, see the official Warpgate documentation.

Prerequisites

- **Docker installed** (Chapter 3)
- **Docker Compose** (Chapter 3)
- **Optional: Traefik installed** (Chapter 4) for HTTPS with Let's Encrypt
- **Optional: Domain configured** (Chapter 4.5), e.g., `warpgate.example.com`
- **Firewall access** - Ability to open port 2222 (SSH bastion)

Installation via Infinity Tools

Menu Installation

```
APPLICATIONS → Warpgate → Install
```

CLI Installation

```
sudo bash /opt/InfinityTools/Solutions/setup-warpgate.sh --install

# With domain (Traefik mode)
export WARPGATE_DOMAIN="warpgate.example.com"
export WG_USE_TRAEFIK="true"
sudo -E bash /opt/InfinityTools/Solutions/setup-warpgate.sh --install
```

```
# Standalone mode
export WG_USE_TRAEFIK="false"
export WG_PORT="8888"
sudo -E bash /opt/InfinityTools/Solutions/setup-warpgate.sh --install

# With domain argument
sudo bash /opt/InfinityTools/Solutions/setup-warpgate.sh --install warpgate.example.com
```

Architecture

Container

- **warpgate** - Warpgate container (ghcr.io/warp-tech/warpgate:latest)

Ports

- **2222** - SSH bastion port (exposed directly, TCP)
- **8888** - Web interface port (via Traefik or standalone, HTTPS)

Data Persistence

- **Data:** `/opt/speedbits/warpgate/data/` (configuration, database)
- **Config:** `/opt/speedbits/warpgate/data/warpgate.yaml` (main configuration)
- **Database:** `/opt/speedbits/warpgate/data/db/` (SQLite database)
- **SSL:** `/opt/speedbits/warpgate/ssl/` (standalone mode certificates)

Deployment Modes

Traefik Mode (Default)

Uses Traefik for SSL termination and domain routing:

- Automatic Let's Encrypt certificate provisioning
- Domain-based access: `https://warpgate.example.com`
- SSH bastion: `ssh -p 2222 user@warpgate.example.com`
- Requires: Traefik running, DNS A record configured

Standalone Mode

Direct access with HTTPS (self-signed):

- HTTPS: `https://SERVER_IP:8888` (self-signed cert)
- SSH bastion: `ssh -p 2222 user@SERVER_IP`
- Default web UI port: 8888 (configurable)
- No domain required

Installation Process

Configuration Steps

1. **SSL Mode Selection:** Choose Traefik or Standalone
2. **Domain Configuration:** If Traefik, specify domain (e.g., warpgate.example.com)
3. **Port Configuration:** If Standalone, specify web UI port (default: 8888)
4. **Container Creation:** Warpgate container created and started
5. **Interactive Setup:** Admin account creation via `warpgate setup` command

What Gets Created

- **Directory:** `/opt/speedbits/warpgate`
- **Container:** `warpgate`
- **Docker Compose:** `/opt/speedbits/warpgate/docker-compose.yml`
- **Configuration:** `/opt/speedbits/warpgate/data/warpgate.yaml`
- **Database:** SQLite database in `/opt/speedbits/warpgate/data/db/`

Access Methods

Traefik Mode

```
# Web interface
https://warpgate.example.com


# SSH bastion
ssh -p 2222 user@warpgate.example.com
```

Direct web access after DNS propagation and SSL certificate generation (30-60 seconds).

Standalone Mode

```
# Web interface
https://SERVER_IP:8888

# SSH bastion
ssh -p 2222 user@SERVER_IP
```

Accept self-signed certificate warning (Advanced → Proceed).

Initial Setup

Admin Account Creation

After container creation, Warpgate runs interactive setup:

```
docker run --rm -it \
  -v /opt/speedbits/warpgate/data:/data \
  ghcr.io/warp-tech/warpgate:latest \
  setup
```

Prompts:

- **Admin username:** Username for admin account
- **Admin password:** Password for admin account
- **Confirm password:** Password confirmation

Configuration File

After setup, configuration is stored in:

```
/opt/speedbits/warpgate/data/warpgate.yaml
```

File permissions: `600` (owner: uid 1000)

Authentication

Web Interface Authentication

- Username/password authentication
- Admin account created during setup
- Additional users created via web interface

SSH Bastion Authentication

- Warpgate username/password authentication
- After authentication, user selects target
- Warpgate connects to target using configured credentials

Target Configuration

Adding Targets

Targets are servers that users can connect to through Warpgate:

- **Name:** Friendly name for the target
- **Host:** IP address or hostname (use `localhost` for same server)
- **Port:** SSH port (usually 22)
- **Username:** SSH username for the target
- **Key-based auth:** Optional SSH key configuration

Same-Server Target

For accessing the server where Warpgate runs:

- **Host:** `localhost` or `127.0.0.1`
- **Port:** `22` (or custom SSH port)
- **Username:** Server username

User Management

Web Interface

- Create users via web interface
- Assign access to specific targets
- Manage user permissions

- View user sessions

User Access Control

- Users can only access targets they're granted access to
- Access can be granted/revoked per user per target
- Session recording available per user/target

SSH Connection Flow

Connection Process

1. Client connects to Warpgate on port 2222
2. Warpgate authenticates user (username/password)
3. Warpgate presents available targets
4. User selects target
5. Warpgate connects to target using configured credentials
6. Session is established and optionally recorded

SSH Command

```
# Traefik mode
ssh -p 2222 warpgate-user@warpgate.example.com


# Standalone mode
ssh -p 2222 warpgate-user@SERVER_IP
```

Security Configuration

Access Security

- Traefik mode uses Let's Encrypt SSL (production-ready)
- Standalone HTTPS uses self-signed certificates (acceptable for internal use)
- SSH bastion port (2222) exposed directly
- Direct SSH port (22) can be closed after Warpgate setup

Firewall Best Practices

```
# Open Warpgate SSH bastion port
sudo ufw allow 2222/tcp

# Close direct SSH access (after testing Warpgate)
sudo ufw delete allow 22/tcp

# Open web interface port (if standalone)
sudo ufw allow 8888/tcp
```

Container Security

- Runs as uid 1000 (non-root)
- Data directory mounted with proper permissions
- Configuration file secured (600 permissions)

Environment Variables

Standalone Mode

- `WARPGATE_HTTP_LISTEN` - HTTP listen address (default: 0.0.0.0:8888)
- `WARPGATE_SSH_LISTEN` - SSH listen address (default: 0.0.0.0:2222)

Troubleshooting

Web Interface Issues

- Check container status: `docker ps | grep warpgate`
- View logs: `docker logs warpgate`
- Verify configuration: `cat /opt/speedbits/warpgate/data/warpgate.yaml`
- Check file permissions: `ls -la /opt/speedbits/warpgate/data/`

SSH Connection Issues

- Verify firewall: `sudo ufw status | grep 2222`
- Test connectivity: `nc -v SERVER_IP 2222`
- Check user credentials in web interface
- Verify target configuration

- Check user access permissions

Target Connection Failures

- Verify target host/IP is correct
- Check target SSH port
- Verify target username
- Test direct connection to target
- Check SSH key configuration (if using key-based auth)

Production Considerations

- **Access Method:** Use Traefik mode for production (trusted SSL)
- **Firewall:** Close direct SSH port (22) after Warpgate verification
- **User Management:** Regularly review and remove unused users
- **Session Recording:** Enable for security auditing
- **Monitoring:** Monitor SSH sessions and access patterns
- **Backup:** Backup configuration and database regularly
- **Updates:** Re-run install script periodically for updates

Integration with Infinity Tools

Warpgate complements Infinity Tools by providing:

- Centralized SSH access management
- Secure gateway for all server access
- Session recording and auditing
- User access control

Recommended Setup:

- Open only Warpgate SSH port (2222) publicly
- Close direct SSH port (22) after testing
- Use Traefik for web interface HTTPS
- Enable session recording for security
- Regularly audit user access

Advanced Configuration

Custom Ports

```
# Custom web UI port (standalone)
export WG_PORT="9999"
sudo -E bash setup-warpgate.sh --install

# SSH port is always 2222 (exposed directly)
```

Configuration File

Edit configuration directly:

```
# Backup first
cp /opt/speedbits/warpgate/data/warpgate.yaml
/opt/speedbits/warpgate/data/warpgate.yaml.backup

# Edit configuration
nano /opt/speedbits/warpgate/data/warpgate.yaml

# Restart container
docker restart warpgate
```

Session Recording

Enabling Recording

Session recording can be enabled per user or per target in the web interface. Recorded sessions are stored in the database and can be reviewed for security auditing.

User Management Script

Infinity Tools provides a helper script for creating system users:

```
sudo bash /opt/InfinityTools/Infrastructure/add-warpgate-user.sh
```

This script creates a `warpgate` system user with SSH key access. Edit the script to add your SSH public key before running.

Next Steps

Warpgate is now operational. Use it to:

- Add targets (servers) users can connect to
- Create users and assign access
- Connect via SSH through Warpgate
- Monitor sessions and access
- Close direct SSH access for better security

For advanced features, API documentation, and development guides, refer to the official Warpgate documentation.

Revision #2
Created 17 November 2025 17:18:18 by bjoern
Updated 17 November 2025 17:22:04 by bjoern