Browser Launch Troubleshooting Guide for Hardened RHEL 9.6 Environments

Quick Start - Run Diagnostics First

Before contacting support, please run our diagnostic tool:

```bash
# Inside the Docker container
python3 /app/diagnostics.py > /tmp/diagnostics.log 2>&1

# Share the output with your support team
cat /tmp/diagnostics.log
```

Common Issues and Solutions

Issue 1: SELinux Blocking Browser Launch

Symptoms:

- Error: `spawn EACCES`
- Browser works when run manually but fails via application
- `getenforce` shows "Enforcing"

Solution A: Disable SELinux for Container (Recommended)

```bash
docker run --security-opt label=disable your-image
```

Or in `docker-compose.yml`:

```yaml
services:
  your-app:
    security_opt:
      - label=disable
```

Solution B: Create Custom SELinux Policy

```bash
# 1. Temporarily set permissive mode
sudo setenforce 0

# 2. Run your application to generate denials
docker run your-image

# 3. Collect denials
sudo ausearch -m avc -ts recent > /tmp/denials.txt

# 4. Generate policy module
sudo audit2allow -M browser_policy -i /tmp/denials.txt

# 5. Install policy
sudo semodule -i browser_policy.pp

# 6. Re-enable enforcing
sudo setenforce 1
```

Issue 2: Shared Memory (/dev/shm) Issues

Symptoms:

- Error mentions `/dev/shm`
- Crashes during browser startup

Solution:

```bash
# Increase shared memory size
docker run --shm-size=2g your-image

# Or mount host /dev/shm
docker run -v /dev/shm:/dev/shm your-image
```

Issue 3: Process Limits / Zygote Errors

Symptoms:

- Process spawn failures
- "Cannot allocate memory" errors
- Zygote process errors

Solution:

Check and increase limits:

```bash
# Check current limits
ulimit -a

# Increase in docker-compose.yml
ulimits:
  nproc: 65535
  nofile:
    soft: 65535
    hard: 65535
```

Issue 4: Network Restrictions

Symptoms:

- Cannot reach external URLs
- CDP connection timeouts

Solution:

Ensure network access:

```bash
```

```
# Test connectivity
docker exec -it container_name curl https://example.com

# If blocked, configure proxy in environment variables
docker run \
  -e HTTP_PROXY=http://proxy:port \
  -e HTTPS_PROXY=http://proxy:port \
  your-image
```

Docker Run Configurations for Different Security Levels

Level 1: Standard (Preferred)

```
bash

docker run \
  --shm-size=2g \
  --security-opt label=disable \
  your-image
```

Level 2: Medium Security

```
bash

docker run \
  --shm-size=2g \
  --cap-add=SYS_ADMIN \
  your-image
```

Level 3: Maximum Compatibility (Last Resort)

```
bash

docker run \
  --privileged \
  --shm-size=2g \
  --security-opt seccomp=unconfined \
  --security-opt label=disable \
  your-image
```

⚠️ **Note:** Level 3 should only be used for testing, not production.

Verification Steps

After applying any solution:

1. **Check SELinux Status:**

```bash
getenforce
# Should show: Permissive or Disabled (for container)
```

2. **Verify Browser Binary:**

```bash
docker exec -it container_name \
  /root/.cache/ms-playwright/chromium*/chrome-linux/headless_shell --version
```

3. **Test Application:**

```bash
docker logs -f container_name
# Look for "Browser launched successfully"
```

Advanced Debugging

Enable Detailed Logging

```bash
# Set environment variables
docker run \
  -e DEBUG=pw:api \
  -e PLAYWRIGHT_CHROMIUM_DEBUG_PORT=9222 \
  your-image
```

Check for Security Module Denials

```bash
# SELinux denials
ausearch -m avc -ts recent | grep chrome

# AppArmor denials
dmesg | grep -i DENIED | grep chrome

# Seccomp violations
dmesg | grep -i seccomp
```

Kernel Security Features

```bash
# Check what's enabled
cat /proc/self/attr/current  # SELinux
cat /sys/kernel/security/apparmor/profiles  # AppArmor
```

Working with Security Team

If you need to involve your security team, provide them with:

1. **Diagnostic Log** (from step 1)
2. **Required Capabilities:**
   - Process spawning (fork, clone)
   - IPC operations
   - Access to /dev/shm
   - Network access to target URLs
3. **Specific Requests:**
   - Allow Docker container to spawn browser subprocesses
   - Allow Chrome's zygote process creation
   - Permit crashpad handler execution
   - Allow shared memory usage

Still Having Issues?

Contact support with:

- Complete diagnostic log (`diagnostics.log`)
- Docker run command or docker-compose.yml
- Output of `getenforce` and `sestatus`
- Any SELinux/AppArmor denial logs

Support Email: [your-support-email] **Documentation:** [your-docs-url]