SL1 PowerFlow: System Security Plan for Docker Enterprise

SL1 PowerFlow Version 3.0.0

SL1 PowerFlow: System Security Plan for Docker Enterprise

The following System Security Plan for Docker Enterprise Edition describes how PowerFlow is configured for Military Unique Deployment (MUD).

> **NOTE:** Where relevant, this document lists the **SV-<*number*>_rule** ID related to the STIG requirement check for that feature.

This document covers the following topics:

PowerFlow Services Configuration: the docker-compose File

This section covers the configuration of the **docker-compose** file used in a PowerFlow system.

Ports Used by PowerFlow

SV-235804r627539_rule, SV-235837r627638_rule, SV-235819r627584_rule

PowerFlow Docker images expose the following ports, but only some of them are externally published in the GUI service:

- steprunners: 5555/tcp
- dexserver: 5556/tcp
- pypiserver: 3141/tcp
- redis: 6379/tcp
- flower: 5555/tcp
- rabbit: 4369/tcp, 5671-5672/tcp, 25672/tcp
- couchbase: 8091-8097/tcp, 9123/tcp, 11207/tcp, 11210/tcp, 11280/tcp, 18091-18097/tcp
- api: 5000/tcp, 8080/tcp, 8443/tcp
- gui: 3141/tcp, 5556/tcp, 8091/tcp, 8102/tcp, 15672/tcp, 8443/tcp, 6443/tcp, 8080/tcp

NOTE: The steprunner image is shared between the scheduler, flower, steprunner, and syncpacks_ steprunner services.

Ports for the GUI Service

The only externally published ports by a PowerFlow system are published by the GUI service, which is the ingress proxy running nginx. These externally published ports include the following:

```
ports:
- published: 80
  target: 8080
  mode: host
- published: 443
  target: 6443
  mode: host
- published: 5556
  target: 5556
  mode: host
```

TIP: For more port information, see the *System Requirements*.

Ports for Couchbase

In MUD deployment, Couchbase communicates internally through encrypted overlay network only, using the following ports:

- 8091: Couchbase Web console, REST/HTTP interface
- 8092: Views, queries, Cross Data Center Replication (XDCR)
- 8093: Query services (Couchbase 4.0 and later)
- 8094: Full-text search (Couchbase 4.5 and later)
- 8095: Analytics (Couchbase 5.5 and later)
- 8096: Eventing (Couchbase 5.5 and later)
- 11207: Smart client library data node access (SSL)
- 11210: Smart client library/moxi data node access
- 11211: Legacy non-smart client library data node access
- 18091: Couchbase Web console, REST/HTTP interface (SSL)
- 18092: Views, query, XDCR (SSL)
- 18093: Query services (SSL) (Couchbase 4.0 and later)
- 18094: Full-text search (SSL) (Couchbase 4.5 and later)
- 18095: Analytics (SSL) (Couchbase 5.5 and later)
- 18096: Eventing (SSL) (Couchbase 5.5 and later)

For more information about the ports, refer to the Couchbase documentation at https://docs.couchbase.com/server/current/install/install-ports.html#detailed-port-description.

Docker cgroup Usage

SV-235815r627572_rule: PowerFlow services do not set another cgroup while running, so the default Docker cgroup is used

Docker cpu_shares Configurations by Service

SV-235807r627548_rule: The **cpu_shares** cannot be configured by service because Docker Swarm does not honor that configuration. For more information, see the Docker documentation: https://docs.docker.com/compose/compose-file/compose-file-v3/#resources.

PowerFlow Memory Limits and CPU Shares

SV-235806r627545_rule: As memory settings are based on specific host specifications, PowerFlow provides only examples about configuring these settings for PowerFlow services, based on the following requirements:

Minimum at 1,000 Synced Objects			Minimum at 10,000 Synced Objects			Minimum at 50,000 Synced Objects		
CPU Cores	RAM (GB)	Disk (GB)	CPU Cores	RAM (GB)	Disk (GB)	CPU Cores	RAM (GB)	Disk (GB)
8	24	100	8	36	100	8	48	200

Note that these system requirements ultimately depend on the amount of workload you plan on running on your PowerFlow service. This table offers a conservative starting point for sizing based on a typical environment (any object being processed by PowerFlow is considered a synced object).

PowerFlow needs its own dedicated memory. Thin provisioning is not supported.

All workloads are different. Storage requirements will vary based upon monitoring depth, frequency of integrations, and length of retention. Sizing recommendations may differ based on multi-SL1 stack support.

Use the Docker official documentation to configure memory limits on the **docker-compose** file based on system needs: https://docs.docker.com/compose/compose-file/compose-file-v3/#resources.

The ***Failure Scenarios*** topic in *Appendix B* of the ***SL1 PowerFlow Platform*** manual contains details about the default settings when deploying a PowerFlow system where the only services set with hard limits are api, steprunners, sp_steprunners and redis.

Sample docker-compose Files

docker-compose 24 GB

```
services:
  contentapi:
    deploy:
      resources:
        limits:
          memory: 2G
  couchbase:
    container_name: couchbase.isnet
    deploy:
      restart_policy:
        condition: any
      resources:
        limits:
          memory: 8G
  dexserver:
    deploy:
      restart_policy:
        condition: on-failure
        max_attemps: 5
      resources:
```

```
      limits:
        memory: 500M
flower:
  deploy:
    restart_policy:
      condition: on-failure
      max_attemps: 5
    resources:
      limits:
        memory: 1G
gui:
  deploy:
    resources:
      limits:
        memory: 2G
pypiserver:
  deploy:
    resources:
      limits:
        memory: 500M
rabbitmq:
  deploy:
    resources:
      limits:
        memory: 8G
redis:
  deploy:
    resources:
      limits:
        memory: 8G
scheduler:
  deploy:
    resources:
      limits:
        memory: 2G
steprunner:
  deploy:
    replicas: 5
    resources:
      limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
      placement:
        max_replicas_per_node: 5
  syncpacks_steprunner:
    deploy:
      mode: global
      resources:
        limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
version: '3.8'
```

docker-compose 36 GB

```
services:
  contentapi:
    deploy:
      resources:
        limits:
          memory: 2G
  couchbase:
    container_name: couchbase.isnet
    deploy:
      restart_policy:
        condition: any
      resources:
        limits:
          memory: 10G
  dexserver:
    configs:
    - source: isconfig
      target: /etc/iservices/isconfig.yml
    deploy:
      restart_policy:
        condition: on-failure
    resources:
      limits:
        memory: 1G
flower:
  deploy:
    restart_policy:
      condition: on-failure
    resources:
      limits:
        memory: 1G
gui:
  deploy:
    resources:
      limits:
        memory: 2G
pypiserver:
  deploy:
    resources:
      limits:
        memory: 1G
rabbitmq:
  deploy:
    resources:
      limits:
        memory: 8G
redis:
  deploy:
    resources:
      limits:
        memory: 8G
scheduler:
  deploy:
    resources:
      limits:
        memory: 2G
steprunner:
  deploy:
    replicas: 5
    resources:
      limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
      placement:
        max_replicas_per_node: 5
  syncpacks_steprunner:
    deploy:
      mode: global
      resources:
        limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
version: '3.8'
```

docker-compose 48 GB

```
services:
  contentapi:
    deploy:
      resources:
        limits:
          memory: 2G
  couchbase:
    container_name: couchbase.isnet
    deploy:
      restart_policy:
        condition: any
      resources:
        limits:
          memory: 12G
  dexserver:
    configs:
    - source: isconfig
      target: /etc/iservices/isconfig.yml
    deploy:
      restart_policy:
        condition: on-failure
    resources:
      limits:
        memory: 2G
flower:
  deploy:
    restart_policy:
      condition: on-failure
    resources:
      limits:
        memory: 2G
gui:
  deploy:
    resources:
      limits:
        memory: 2G
pypiserver:
  deploy:
    resources:
      limits:
        memory: 1G
rabbitmq:
  deploy:
    resources:
      limits:
        memory: 10G
redis:
  deploy:
    resources:
      limits:
        memory: 8G
scheduler:
  deploy:
    resources:
      limits:
        memory: 2G
steprunner:
  deploy:
    replicas: 5
    resources:
      limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
      placement:
        max_replicas_per_node: 5
  syncpacks_steprunner:
    deploy:
      mode: global
      resources:
        limits:
          memory: 2G
      restart_policy:
        condition: any
        delay: 10s
version: '3.8'
```

Setting Custom Interfaces

SV-235820r627587_rule: All ports that PowerFlow publishes are in the GUI service, which can be updated in the **docker-compose** file to set a specific interface by port. Below is an example of how that can be achieved:

```
gui:
  ports:
    - published: 10.2.3.4:80
      target: 8080
    - published: 10.2.3.4:443
      target: 6443
    - published: 10.2.3.4:3141
      target: 3141
    - published: 10.2.3.4:5556
      target: 5556
```

For more information, refer to the Docker documentation for exposing ports: https://docs.docker.com/compose/compose-file/compose-file-v3/#ports.

Docker Secret Management

Docker EE STIG: **SV-235826r627605_rule, SV-235824r627599_rule**

AppSec SRG: **SRG-APP-000176, SRG-APP-00023**

PowerFlow services currently use the following Docker secrets:

- **encryption_key**. The encryption key used to encrypt the information in the configuration files.
- **is_pass**. The common password that all the services share.

For more information, refer to the Docker documentation: https://docs.docker.com/engine/swarm/secrets/.

Adding or Dropping Capabilities

SV-235801r627530_rule: PowerFlow requires no additional capabilities beyond Docker defaults. Also, adding or removing capabilities is not supported by Swarm deployments.

Setting the on-failure Container Restart Policy

SV-235843r627656_rule: PowerFlow services have the restart_policy condition set to "any" by default because it is essential that they restart automatically so PowerFlow can be available and accessible.

The following can be set if this requirement needs to be met over PowerFlow services availability.

```
deploy:
  restart_policy:
    condition: on-failure
    delay: 30s # time based on the service
    max_attempts: 5
```

For more information, refer to the Docker documentation: https://docs.docker.com/compose/compose-file/compose-file-v3/#restart_policy.

Read-only Containers

SV-104789r1_rule: All PowerFlow containers run in **read_only** by default. To help facilitate this requirement, PowerFlow defined additional, temporary-named volumes rather than use tmpfs volumes due to this Docker bug. These named tmp volumes are used only for internal information that would otherwise be deleted on a container restart.

- iservices_tmp_rabbit_config
- iservices_tmp_api_logs
- iservices_tmp_syncpacks_envs

PowerFlow journald Log-driver Settings

SV-235832r695335_rule, SV-235833r627626_rule, SV-235831r627620_rule, SV-235787r627488_rule, SV-235786r627485_rule

The default Docker log driver in a PowerFlow system is set to **journald**. For more information, refer to the Docker documentation for configuring log-drivers: https://docs.docker.com/config/containers/logging/configure/.

PowerFlow Signed Images

Docker EE STIG: **SV-235839r627644_rule**, **SV-235846r627665_rule**

AppSec SRG: **SRG-APP-000386**, **SRG-APP-000475**

PowerFlow images are signed and uploaded to Docker Hub, and they are available externally to users with appropriate credentials, and for trust verification.

However, to satisfy deployments that are not Internet-connected, ScienceLogic also providea the same containers inside a signed, sha256/sha512 checksum-verified RPM package, which is also contained by our signed, checksum-validated ISO.

Users in non-Internet-facing environments can take these RPM-shipped containers and upload them to their own internally managed DTR, and sign and verify them there.

Linux CMD Verification Example

This section covers how to verify a signed or trusted container from ScienceLogic DockerHub acess or internal DTR.

Execute the following commands on the PowerFlow system as a user with access to the repository in DTR for which image signing is being enabled:

```
docker login [dtr_url]
docker trust signer add --key [ucp_client_bundle_cert].pem [ucp_user]
[dtr_url]/[namespace]/[imageName]
docker trust key load [ucp_client_bundle_key].pem
docker tag [source_image] [dtr_url]/[namespace]/[imageName]:[tag]
export DOCKER_CONTENT_TRUST=1
docker push [dtr_url]/[namespace]/[imageName]:[tag]
```

Checking Verification Requirements

To check for image signing and checksum verification requirements on a PowerFlow system, run the following command:

```
docker images --digests to check the checksum
```

PowerFlow Docker Daemon Default Settings

default-ulimits

SV-235844r627659_rule: The rule states that in a PowerFlow system, the Docker Enterprise default-ulimit values are not overwritten at run-time unless approved in the System Security Plan (SSP).

The following list includes the default ulimits declared at the daemon level when using Docker, and none of them are overwritten by run-time containers by default in a PowerFlow system:

```
"default-ulimits": {
    "core": {
      "Hard": 10000000,
      "Soft": 10000000,
      "Name": "core"
    },
    "nofile": {
      "Hard": 100000,
      "Soft": 100000,
      "Name": "nofile"
    },
    "nproc": {
      "Hard": 3000,
      "Soft": 1500,
      "Name": "nproc"
   }
}
```

PIDs cgroup Limits

SV-235828r627611_rule: To ensure that PIDs cgroup limits are used in Docker Enterprise, PowerFlow adds `ulimits nproc` at the daemon level, as `pid_limit` is not available in Swarm deployments.

WARNING: The Docker documentation states the following: "Be careful setting `nproc` with the `ulimit` flag as `nproc` is designed by Linux to set the maximum number of processes available to a user, not to a container."

```
"default-ulimits": {
  "nproc": {
      "Hard": 3000,
      "Soft": 1500,
      "Name": "nproc"
   }}
```

The userland Proxy Capability

SV-235791r627500_rule: In a PowerFlow system, the userland proxy capability in the Docker Engine Enterprise component of Docker Enterprise is disabled with the following command in the Docker daemon file:

```
{"userland-proxy": false}
```

Preventing Containers from Acquiring Additional Privileges

SV-235816r672380_rule: In a PowerFlow system, all Docker containers are restricted from acquiring additional privileges. Setting `{"no-new-privileges": true}` at the daemon level affects all the containers.

For more information, see the Docker documentation: https://docs.docker.com/engine/reference/commandline/dockerd/.

Full Docker daemon File

```
{
"storage-driver": "overlay2",
"selinux-enabled": true,
"default-ulimits": {
    "core": {
      "Hard": 10000000,
      "Soft": 10000000,
      "Name": "core"
    },
    "nofile": {
      "Hard": 100000,
      "Soft": 100000,
      "Name": "nofile"
    },
    "nproc": {  #used instead of pid_limits
      "Hard": 3000,
      "Soft": 1500,
      "Name": "nproc"
   }
},
"userland-proxy": false,
"no-new-privileges": true #capadd dissabled
}
```

PowerFlow Swarm init autolock and listen-addr configuration and examples

Docker Enterprise Swarm Manager autolock Enabled

Docker Swarm autolock commands

SV-235849r627674_rule, SV-235823r627596_rule, SSV-235849r627674_rule: PowerFlow enables Swarm autolock by default in Military Unique Deployment (MUD) systems when executing the following command when the **pull_start_iservices.sh** script is executed:

```
docker swarm init --autolock
```

When autolock is enabled, save the `unlock-key` in a safe place so you can unlock the Swarm when it is restarted. You can access the key by executing the following command while the Swarm is alive:

```
docker swarm unlock-key
```

Use the following command to unlock the Swarm:

```
docker swarm unlock
```

Use the following command to rotate the keys periodically:

```
docker swarm unlock-key --rotate
```

NOTE: ScienceLogic recommends that the PowerFlow system administrator creates a process for maintaining key rotation records and establishing a pre-defined frequency for key rotation.

For more information, see the Docker documentation: https://docs.docker.com/engine/swarm/swarm_manager_locking/#initialize-a-swarm-with-autolocking-enabled.

Restricting Access to the PowerFlow System

SV-235873r627746_rule, SV-235848r627671_rule: The PowerFlow system can be configured to restrict inbound connections from non-secure zones.

To bound the PowerFlow Swarm to a specific host interface, you can use the flag `--listen-addr` when initializing the Swarm, such as:

```
docker swarm init --listen-addr 10.2.3.4:2377
```

For more information, see the Docker documentation: https://docs.docker.com/engine/reference/commandline/swarm_init/#--listen-addr.

You can use the following **firewalld** rule to limit the number of connections by default:

```
firewall-offline-cmd --direct --add-rule ipv4 filter INPUT_direct 0 -p tcp
-m limit --limit 25/minute --limit-burst 100 -j INPUT_ZONES
```

NOTE: These settings should be sufficient for standard clusters, but you might need to increase the `limit-burst` value based on how many workers are added into the cluster. Users in a non-MUD environment can also use this command to further protect their PowerFlow system.

For additional information, see *Configuring the PowerFlow System for High Availability* in the ***SL1 PowerFlow Platform*** manual.

Restricting Access to the PowerFlow Docker Swarm Stack

SRG-APP-000315-WSR-000004: The web server restricts inbound connections from non-secure zones.

To block or allow traffic into the PowerFlow Docker Swarm stack, you can run commands with the **powerflowcontrol** (pfctl) command-line utility in interactive or non-interactive mode.

interactive mode:

```
pfctl --host <host> <username>:<password> node-action --action=secure_
zones --interactive
```

non-interactive mode:

```
pfctl --host <host> <username>:<password> node-action --action=secure-
zones [--apply/--unapply] --interfaces=eth0,wlan0...
```

You can either apply or remove firewall rules for the PowerFlow system. The default behavior is `--apply.` You will need to provide a list of network interfaces to which the firewall rules will be applied.

With the commands described above, you can enter custom firewall-cmd rules into the DOCKER-USER chain. To add the DOCKER-USER filter chain to the firewall-cmd "direct" interface, run the following command:

```
firewall-cmd --permanent --direct --add-chain ipv4 filter DOCKER-USER
```

To apply rules when an interface needs to be blocked from forwarding traffic to **docker_gwbridge** ("applying" or installing rules), run the following command:

```
firewall-cmd --permanent --direct --add-rule ipv4 filter DOCKER-USER 0 -o
docker_gwbridge -i net_interface_name -j DROP
```

To remove rules to allow the interface to forward traffic to **docker_gwbridge** ("unapplying" or removing rules), this command is being executed:

```
firewall-cmd --permanent --direct --remove-rule ipv4 filter DOCKER-USER 0
-o docker_gwbridge -i net_interface_name -j DROP
```

You can also choose to invoke a per-network command manually (as it is not supported in the command-line utility) instead of using the -o argument for both `--add-rule` and `--remove-rule`:

```
firewall-cmd --permanent --direct --add-rule ipv4 filter DOCKER-USER 0 -s
172.21.0.1/16 -i net_interface_name -j DROP
```

NOTE: For more information, see the official firewalld documentation:https://firewalld.org/documentation/man-pages/firewall-cmd.html and the docker documentation about using iptables policies https://docs.docker.com/network/iptables/.

After you make an update, restart both the firewalld and Docker services with the following commands:

```
sudo firewall-cmd --reload
```

```
sudo systemctl restart docker
```

NOTE: The update might take some time, because after restarting the Docker daemon, the Docker swarm stack services are updated, and then the ingress rules are updated.

PowerFlow Default Audit Policies

SV-235779r627464_rule: The following auditing policies are set by default in PowerFlow MUD systems.

```
auditctl -w /usr/bin/docker -k
```

```
auditctl -w /var/lib/docker -k docker
```

```
auditctl -w /etc/docker -k docker
```

```
auditctl -w /usr/lib/systemd/system/docker.service -k docker
```

```
auditctl -w /usr/lib/systemd/system/docker.socket -k docker
```

```
auditctl -w /etc/default/docker -k docker
```

```
auditctl -w /etc/docker/daemon.json
```

```
auditctl -w /usr/bin/docker-containerd -k docker
```

```
auditctl -w /usr/bin/docker-runc -k docker
```

Docker Enterprise Edition GPG Key

Docker EE STIG: **SV-235787r627488_rule**

AppSec SRG: **SRG-APP-000131**

PowerFlow includes Docker Enterprise Edition GNU Privacy Guard (GPG) keys.

Run the following command to verify that the docker-ee keys are added to the installation:

```
rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n'
```

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com. For more information, see https://sciencelogic.com/company/legal.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010