title: How to Include/Exclude Log Sources With the File Input Plugin description: Blacklist/Whitelist log sources when using the File Input Plugin. This is useful when you want to exclude certain files from a directory you want to ship logs to Sematext. By using Glob patterns you can filter which files to tail.

To reduce the noise from logs you do not need or want to track, you can blacklist the log sources entirely. This guide refers to using the File Input Plugin and will explain an advanced approach to using Glob patterns.

This will contain a few examples that show how you can configure the File Input Plugin to filter log sources.

The most efficient way to exclude log sources is to stop collecting logs from noisy data sources. For a full reference on Glob patterns, check this out.

Running Logagent natively (bare-metal/VM)

Limit the data sources to dedicated log files or directories, instead of using `/var/log/**/*.log` which is the default setting in Logagent.

```
input:
  files:
    - /var/log/system.log
    - /var/log/kernel.log
    - /var/log/audit.log
    - /var/log/nginx/access_log
    - /var/log/myapp/*.log
    - /var/log/containers/myapp*.log
```

The file input takes Glob patterns. You can create custom Glob patterns that exclude log sources with `!(pattern)`

```
input:
  files:
    - /var/log/!(auth*.log)
    - /var/log/myapp/*.log
    - /var/log/containers/*.log
```

This pattern will collect all log files from the `/var/log` directory except for files that match `auth*.log`. This `*` means it matches 0 or more characters in a single path portion. It will also collect logs from the `myapp` directory, and `containers` directory.

You can also combine multiple patterns to exclude.

```
input:
  files:
    - /var/log/!(auth*.log|system*.log|kernel*.log)
    - /var/log/containers/!(kube*.log|storage*.log|etcd*.log|coredns*.log)
```

This pattern will collect all log files from the `/var/log` directory except for files that match `auth*.log`, `system*.log`, and `kernel*.log`. It will also collect all container logs except for the ones that are generated by containers in the Kube System namespace.

By using this approach you can exclude certain log sources. This may be easier than including a long list of log sources if you have many of them.

Running Logagent as a container

LOG_GLOB is an environment variable configured on the Logagent container. It's a semicolon-separated list of Glob patterns. The same way you would add Glob patterns in the File Input Plugin, you add Glob patterns to the `LOG_GLOB` environment variable, but instead in a semicolon-separated list.

This will collect all logs from both the `/mylogs` and `/var/log` directories.

```
LOG_GLOB=/mylogs/**/*.log;/var/log/**/*.log
```

Here's an example of excluding log sources, just as in the section above, but with the LOG_GLOB environment variable.

```
LOG_GLOB=/mylogs/containers/!(kube*.log|storage*.log|etcd*.log|coredns*.log)
```

Finally, don't forget to mount your server log files into the container using a Docker volume.

```
-v /var/log:/mylogs
```

You start Logagent as a container with the volume mount and LOG_GLOB environment variable.

```
docker pull sematext/logagent
docker run -d --restart=always --name logagent \
  -e LOGS_TOKEN=YOUR_LOGS_TOKEN \
  -e LOG_GLOB=/mylogs/**/*.log;/var/log/**/*.log  \
  -v /var/log:/mylogs
```

Using LOG_GLOB is needed when you want to ship logs from your host to Sematext with a containerized instance of Logagent. It's also needed when you're using Kubernetes with the Containerd container runtime. It stores container logs on the host instead of using a socket like Docker.