+++ title = “Security” description = “Security Docs” keywords = [“grafana”, “security”, “documentation”] aliases = [“/docs/grafana/latest/installation/security/”] weight = 500 +++

Security

If you run non-Grafana web services on your Grafana server or within its local network, then they might be vulnerable to exploitation through the Grafana data source proxy or other methods.

To prevent this type of exploitation from happening, we recommend that you apply one or more of the precautions listed below.

Limit IP addresses/hostnames for data source URL

You can configure Grafana to only allow certain IP addresses or hostnames to be used as data source URLs and proxied through the Grafana data source proxy. Refer to [data_source_proxy_whitelist]({{< relref “../administration/configuration/#data-source-proxy-whitelist” >}}) for usage instructions.

Request security

The request security configuration option allows users to limit requests from the Grafana server. It targets requests that are generated by users. For more information, refer to Request security({{< relref “../enterprise/request-security.md” >}}) in [Grafana Enterprise]({{< relref “../enterprise” >}}).

> **Note:** Request security is available in Grafana Enterprise v7.4 and later versions.

Firewall rules

Configure a firewall to restrict Grafana from making network requests to sensitive internal web services.

There are many firewall tools available, refer to the documentation for your specific security tool. For example, Linux users can use iptables.

Proxy server

Require all network requests being made by Grafana to go through a proxy server.

Limit Viewer query permissions

Users with the Viewer role can enter *any possible query* in *any* of the data sources available in the **organization**, not just the queries that are defined on the dashboards for which the user has Viewer permissions.

For example: In a Grafana instance with one data source, one dashboard, and one panel that has one query defined, you might assume that a Viewer can only see the result of the query defined in that panel. Actually, the Viewer has access to send any query to the data source. With a command-line tool like curl (there are lots of tools for this), the Viewer can make their own query to the data source and potentially access sensitive data.

To address this vulnerability, you can restrict data source query access in the following ways:

- Create multiple data sources with some restrictions added in data source configuration that restrict access (like database name or credentials). Then use the [Data Source Permissions]({{< relref "../enterprise/datasource_permissions.md" >}}) Enterprise feature to restrict user access to the data source in Grafana.
- Create a separate Grafana organization, and in that organization, create a separate data source. Make sure the data source has some option/user/credentials setting that limits access to a subset of the data. Not all data sources have an option to limit access.

Implications of enabling anonymous access to dashboards

When you enable anonymous access to a dashboard, it is publicly available. This section lists the security implications of enabling Anonymous access.

- Anyone with the URL can access the dashboard.
- Anyone can make view calls to the API and list all folders, dashboards, and data sources.
- Anyone can make arbitrary queries to any data source that the Grafana instance is configured with.