OPNsense

The open source firewall in practice

THOMAS KRENN®

Table of contents

OPNsense

The open source firewall in practice

Comprehensive IT security and flexibly expandable plugins.

IT managers regularly see the IT they manage exposed to new threats. Having the latest firewall is no longer sufficient. It is much more important to be able to react flexibly to security risks. The open source firewall OPNsense is a digital platform that offers many additional features such as intrusion detection & prevention, VPN, two-factor authentication and high availability. In this e-book, we present OPNsense as an alternative to commercial firewall solutions. Our focus is on practical use cases in the SME context and on how functionality can be expanded via plugins.

1. How companies can benefit from OPNsense

Since OPNsense is an open source solution, there are no license costs. This is a great advantage, particularly for SMEs, which often have small IT budgets and could invest the costs saved into the continuous support of their IT via an external service provider, for instance. Moreover, IT managers often do not have the time to regularly update their security solutions. This represents a considerable risk as data theft can quickly threaten the existence of an SME.

OPNsense has a very active developer community that regularly releases new versions. The OPNsense software is regularly improved with new plugins, which relieves IT managers and also ensures maximum security. Anti-spam and antivirus features are just a few examples. This is also the biggest advantage of OPNsense. The system is open source and can therefore be flexibly expanded and optimized. As a result, OPNsense offers features that are sometimes missing in commercial solutions. One example is the administration of Radius servers, which is provided via a plugin from m.a.x. Informationstechnologie AG.

The advantages of OPNsense at a glance:

- Completely open source (BSD 2-Clause license), unlimited access to the source code
- The system can be optimized at any time via your own plugins
- Professional support available
- Time and cost savings thanks to simple administration and the lack of license fees

---

For example, high availability solutions and intelligent networking can be implemented and several security solutions can be interconnected on a central platform using OPNsense. The software's well-structured management interface makes it very easy to administer the system and react flexibly to security risks. In addition, recurring processes are partially automated: A cron daemon enables automated execution of tasks ("jobs") at predefined times. These tasks include automatic updates, reloading IPS signatures or refreshing blacklists.

2. What makes OPNsense unique?

Many companies grapple with the question: Which is better: an affordable open source solution or an expensive commercial solution? Straight answer: It depends. More precisely: It depends on the specific application. Commercial solutions usually offer a greater depth of functionality such as multiple scenarios for high availability, application detection and live traffic blocking.
The biggest advantages of an open source solution like OPNsense remain cost and flexibility. There are no license costs, neither for the core system nor for extensions, in contrast to commercial solutions, which sometimes require additional licenses for individual functions. With OPNsense, all functions are included and can be flexibly adapted and expanded at any time.
Moreover, the OPNsense license also allows the development of commercial plugins, for instance to enable better integration into existing infrastructures, which individual companies can commission from suitable service providers without having to ask the core developers for permission or requiring special extended licenses.

2.1 The OPNsense Business Edition

In addition to the freely available version, Deciso offers a Business Edition of OPNsense for a moderate annual license fee. The functional scope of the software itself does not differ from the base version. However, it is based on a somewhat less current and thereby usually more stable source code. This provides companies with a more conservative upgrade path. There are discounts on the time-based support packages as well as access to a GeoIP database that makes it easier to block IPs from specific countries or regions. Additionally, purchasers of OPNsense Business Edition have access to OPNcentral, a commercial plugin currently being developed by Deciso for the centralized management of multiple OPNsense instances. Last but not least, the Business Edition also offers the possibility to support the developers as a commercial user with a regular amount and thereby contribute to the further development of the software.

3. A closer look at OPNsense

The software has its origin in the year 2015 and is a fork of the well-known open source firewall software pfSense, which itself started in 2004 as a project fork of m0n0wall. The basic idea of all these projects was to combine the functionalities and management of firewall rules under one graphical interface. However, there are important technical differences, which are explained in the technology blog produced by m.a.x. Informationstechnologie AG1.

The name "OPNsense" is derived from the combination of "pfSense" and "open source". The founders of OPNsense cite five central arguments for the fork2:

Technology – High code quality and well-structured development methods as well as achievable goals in a roadmap with regular releases

Security – No tasks executed in the GUI require root access and potential security risks are dealt with at an early stage

Quality – All new functions are created using a solid framework (Phalcon) with a Model–View–Controller approach

Community – A productive community of developers and users with barrier-free access to codes and systems

Transparency – Possible changes are communicated transparently and OPNsense is based on the proven, open source 2-clause BSD license

---

The OPNsense project is financially and technically supported by the Dutch company Deciso B.V., which was already active as a sponsor and co-developer for the two OPNsense predecessors pfSense and m0n0wall and also supports other open source projects. OPNsense is based on the HardenedBSD operating system, a derivative of the Unix operating system FreeBSD, which has been developed by a highly active community for decades. The core functionality of OPNsense focuses on the configuration of interfaces (LAN, WAN, DMZ) as well as firewall rules, Network Address Translation (NAT) and local services such as DNS and DHCP. Under the hood, the packet filter pf is used as the actual firewall and NAT software. As a user, however, you will not come into contact with pf's complex firewall rules, as the configuration is done entirely via the intuitive web interface.

---

1 Differences between pfSense and OPNsense: https://techcorner.max-it.de/wiki/OPNsense_vs._pfSense_-_Im_Vergleich [German]
2 Reasons for the OPNsense fork: https://docs.opnsense.org/history/thefork.html

3.1 Intrusion and malware detection

Suricata, a powerful IDS/IPS system, is integrated into OPNsense to detect and defend against intrusion attempts. This analyzes network traffic at different protocol levels according to rule-based patterns (deep packet inspection) and thereby detects unusual events. Rule sets from different providers can be integrated automatically and, of course, it is also possible to create your own rules. Suricata is considered extremely fast and consumes few hardware resources compared to other IDS systems.

Server-side malware detection provides an additional layer of protection to safeguard users. To detect infected websites and downloads, OPNsense relies on ClamAV, an antivirus software widely used in the open source sector, and the ICAP format, which various manufacturers use to offer malware signatures. Virus protection can be supplemented with additional security measures, such as category-based web filters, to further enhance security.

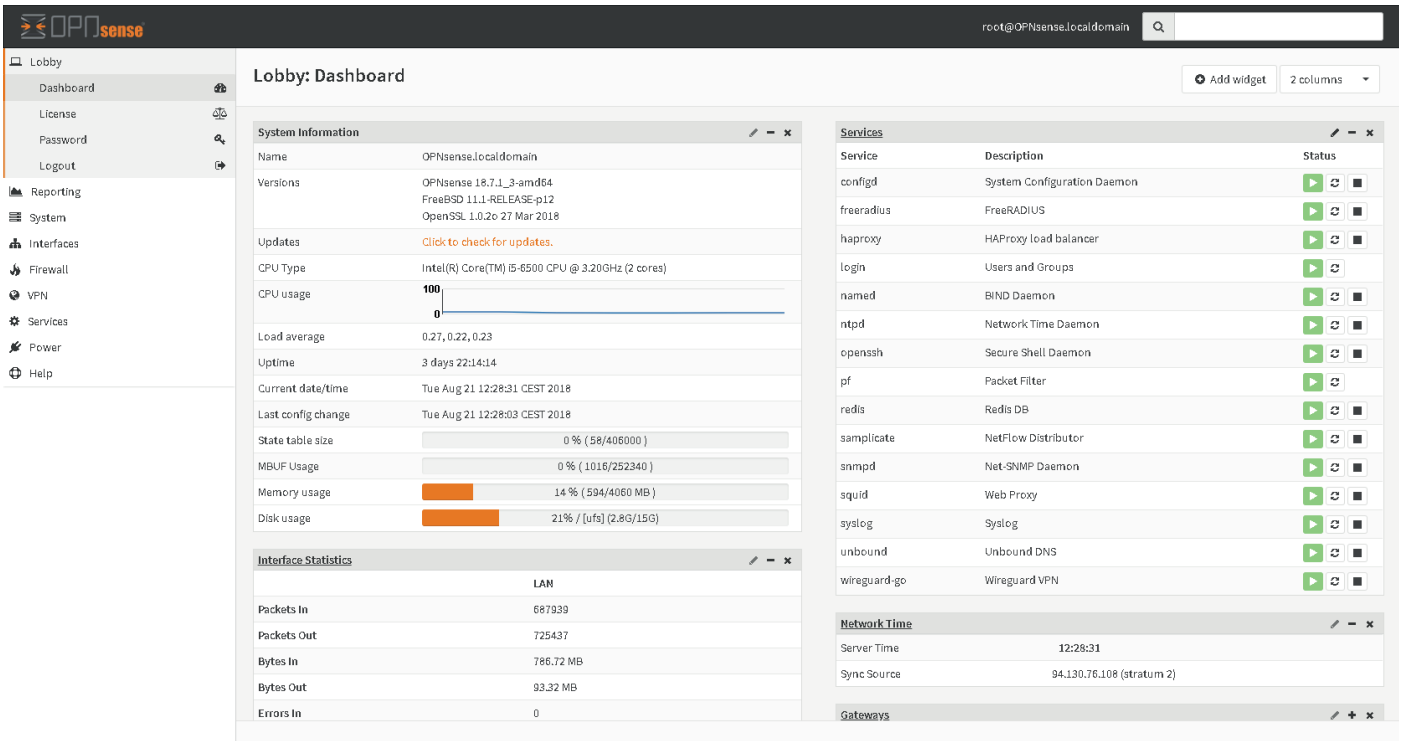

Figure 1: The OPNsense GUI. The dashboard shows the current system status and statistics of the interfaces.

3.2 Virtual Private Networking (VPN)

Virtual Private Networks (VPN) are indispensable, especially for companies that offer home offices, operate several locations or work with external freelancers. OPNsense offers several options here in the core system, such as OpenVPN and IPsec, as well as older protocols such as PPTP via plugins.

Here, too, the setup is carried out completely via the management interface and is well explained in the documentation on both the server and client side – a valuable resource as VPN installations are notoriously error-prone. Additional VPN technology can be added via plugins.

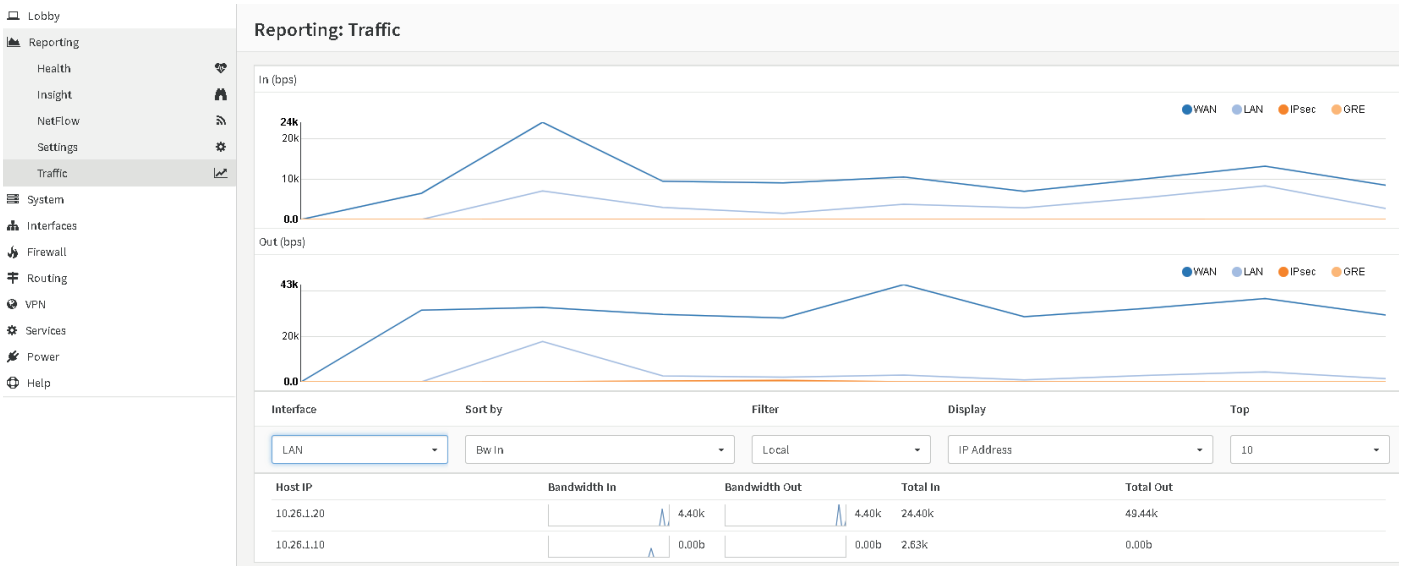

Figure 2: Traffic data is displayed graphically in the reporting area.

3.3 High availability

Fail-safe reliability is an important issue for firewall servers, as a hardware defect can affect the entire organization. As a result, such services should operate on a redundant basis whenever possible. OPNsense uses the CARP (Common Address Redundancy Protocol) standard for high availability. This, like the packet filter pf, comes from the OpenBSD environment and is particularly well-suited for high-availability firewalls. Thanks to its straightforward web interface and online how-to documentation, setting up a failover device takes mere minutes.

3.4 Traffic analysis and traffic shaping

Detailed monitoring of network traffic, broken down by source and destination IP, is also part of the core system. It is based on Netflow, a monitoring system originally developed by Cisco. The netflow analyzer is directly integrated into the system's web interface. OPNsense is the only open source solution with an integrated netflow analyzer.
In many cases, traffic monitoring is not enough, and a certain portion of bandwidth must be allocated to individual services, clients or networks. This traffic shaping is especially important for the quality of service (QoS) of real-time services such as VoIP, but also when bandwidth is to be distributed to users according to certain rules. Traffic shaping is configured via a two- or three-tier concept using pipes, queues and rules in the web GUI and can thereby be adapted to any application case.

3.5 Further features in the core system

The core system offers many additional features, including the so-called captive portal, which can be used to configure temporary WLAN access, including a voucher system. Hotels and public institutions can save considerably on costs here compared to commercial voucher systems. OPNsense is also equipped with a transparent proxy function.
These can be used to restrict access through various

models, such as blacklists, whitelists, subnets or user agents. The cache can be combined with the traffic shaper to implement bandwidth throttling or category based filters. SSL inspection in the cache is also possible.

3.6 Plugins

Alongside the many features in the core system, there is also the possibility of optimizing the system for specific applications by means of individually programmed plugins. Such plugins can be requested from the OPNsense project managers if they are of use to other users, or from specialized IT service providers such as m.a.x. Informationstechnologie AG in Munich. The second part of this e-book is dedicated to the possibilities offered by OPNsense plugins.

4. Hardware for OPNsense

When using BSD-based systems such as OPNsense, it should be noted that hardware is often not supported to the same extent as Linux, for example. Therefore the use of tested systems is recommended, such as those available from Thomas-Krenn. The variety of application purposes sometimes makes it difficult to assess performance requirements correctly. Pure firewall operation for SMEs may require very few resources and can be well covered by an inexpensive and low-maintenance and energy-efficient LES system.
Other components such as large VPNs, the captive portal or a web proxy require more powerful CPUs and more RAM. It is therefore important to have the widest possible range of OPNsense-optimized systems to choose from with a high degree of flexibility in terms of CPUs, RAM and network interfaces. The OPNsense website provides some general information on hardware sizing3, which is also available in summary form in the Thomas-Krenn wiki4. All available OPNsense-optimized systems are shown on the overview page in the Thomas-Krenn online shop5. The consultants at Thomas-Krenn are also more than happy to help for cases requiring more individualized solutions.

Three examples of OPNsense-optimized servers from Thomas-Krenn: The inexpensive and power-saving LES v3, a mini-server equipped with four 1 Gbit/s LAN ports, the LES compact 4L and the infrastructure server RI 1102D-F (pictured) with front IO, an Xeon CPU and up to 256 GB of RAM.

5. Using OPNsense – two examples

m.a.x. Informationstechnologie AG is one of the official partners of the OPNsense project and has developed a number of useful plugins. The most important plugins are described in detail in this e-book.

---

3 Hinweise zum Hardware Sizing: https://wiki.opnsense.org/manual/hardware.html
4 Hardware-Anforderungen von OPNsense: https://www.thomas-krenn.com/de/wiki/Hardwareanforderungen_OPNsense
5 OPNsense-optimierte Server: https://www.thomas-krenn.com/de/produkte/einsatzzweck/hardware-it-security/opnsense-firewalls.html

The Munich-based IT service provider has already realized a large number of customer projects with OPNsense. The following provides an overview of two specific examples.

5.1 Example customer A: An SME specializing in the production of technical ropes

The existing Linux-based firewall system was replaced by an OPNsense appliance. Both the firewall for central administration as well as the production facilities were equipped with OPNsense. The online data traffic from production passes through the head office without local breakout and is scanned for viruses while the integrated IPS (Intrusion Prevention System) protects against malware and other threats. An intelligent VPN implementation allows for a fully automated failover across lines as both the headquarters and production facilities have primary and secondary internet access lines. Thanks to the networking concept, one WAN can fail in each of the two locations without users losing network access to the headquarters.

5.2 Example customer B: An SME consulting firm in the social sector

The sites were networked on the basis of an OPNsense solution, which also provided VPN access for the employees. By networking with a central OPNsense platform using IKEv2, the routing and firewall rules are centrally controlled. This simplifies administration and ensures maximum security in the network. It does not matter what hardware is used at the sites as long as the IKEv2 standard is met. An automated failover can also be integrated with a few adjustments. Every employee can easily connect to the locations via a VPN client from the road or at home using the OPNsense platform. Authentication takes place via the local Windows Active Directory of the respective location. This allows the administrator at each location to decide which user is allowed to log into the VPN.

6. Plugins for OPNsense

The plugin system is one of the great strengths of OPNsense. It is also where the advantages of a consistent open source approach are fully exploited. Most plugins are developed by the community, for the community. This means that every plugin is designed for a useful purpose arising from practical IT situations. A complete list of all the available plugins of the stable branch can be obtained via the system's plugin page. Any plugin that appears there can be considered stable and mature for use in productive systems. Plugins are installed directly within the OPNsense interface, making the process extremely simple. It is always worth taking a look at the official documentation beforehand, as this provides information on the requirements and resource usage (Example ClamAV). Plugins are also available from the related project pfSense as well as the predecessor software m0n0wall. The difference: The OPNsense plugin system is based on a modern framework with an MVC concept (Model, View, Controller). Furthermore, all code is freely available. An API functionality already integrated in the core system allows interaction with the core system and between plugins. Plugins are already listed in

the GUI and only need to be activated. Secure code execution is guaranteed by "configd", a proprietary service wrapper from OPNsense.

7. Transparent development model

In addition to the stable plugins, other plugins that are in development and may appear later can also be collected as a list on GitHub6. The plugin source code and further information is also available there, sorted by topic7. The transparent commit history and the issues and questions listed under "Issues" give a good impression of the state of the plugin and the activities of its developers in advance. Most plugins can be found in the network infrastructure directories and often provide interfaces for various protocols and applications. Maximum diversity is in high demand here. Companies often use "unusual" solutions which, in the absence of a viable interface, may prevent OPNsense from being used at all. This again shows the fundamental difference to commercial solutions. Commercial providers have to spend their own resources to provide as many interfaces as possible, which consequently impacts the price for users. Customers therefore indirectly pay for features and interfaces that they may not even need.

8. Plugins in an example scenario

Using a fictitious but realistic example, we want to illustrate how OPNsense plugins can meet demanding requirements for network security, email protection, monitoring and failover reliability. The example is based on real-world experience with OPNsense. The company "StartUP AG" operates a small network with 50 users and a Proxmox virtualization environment with a local mail and file server, which is protected against power failures by a UPS. The company also depends on personalized WLAN logins for its employees. Recently, the management decided that IT needs to secure the network as a whole and ensure monitoring. More specifically, this means securing the WLAN via user authentication, connecting a mail relay to the local mail server, connecting to the existing Grafana installation and integrating the UPS. AS we will see, OPNsense plugins allow these requirements to be optimally covered.

8.1 Centralized WLAN protection with the FreeRADIUS plugin

Over the years, WLAN has become an integral part of life. WLAN also plays an important role for businesses – be it for the printer in the corridor, where no patch box is installed, or for warehouse workers who use an app to record the goods. But WLAN shifts the boundaries of the local network – allowing for the possibility of external access. Currently, the best protection is provided by WPA2 Enterprise8, which does not use a common pre-shared key (PSK) as with WPA2 Personal, but allows

6 Plugin list: https://github.com/opnsense/plugins#a-list-of-currently-available-plugins
7 Details on the plugins: https://github.com/opnsense/plugins
8 Information about WPA2: https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access#WPA2

greater control of authentication via a RADIUS service. In this way, user accounts can be created for some or all employees and when a colleague leaves, it is not necessary to change the PSK at all access points. This functionality is best set up using the FreeRADIUS plugin. As with all plugins, the installation is conveniently performed via System – Firmware – Plugins (Figure 4).

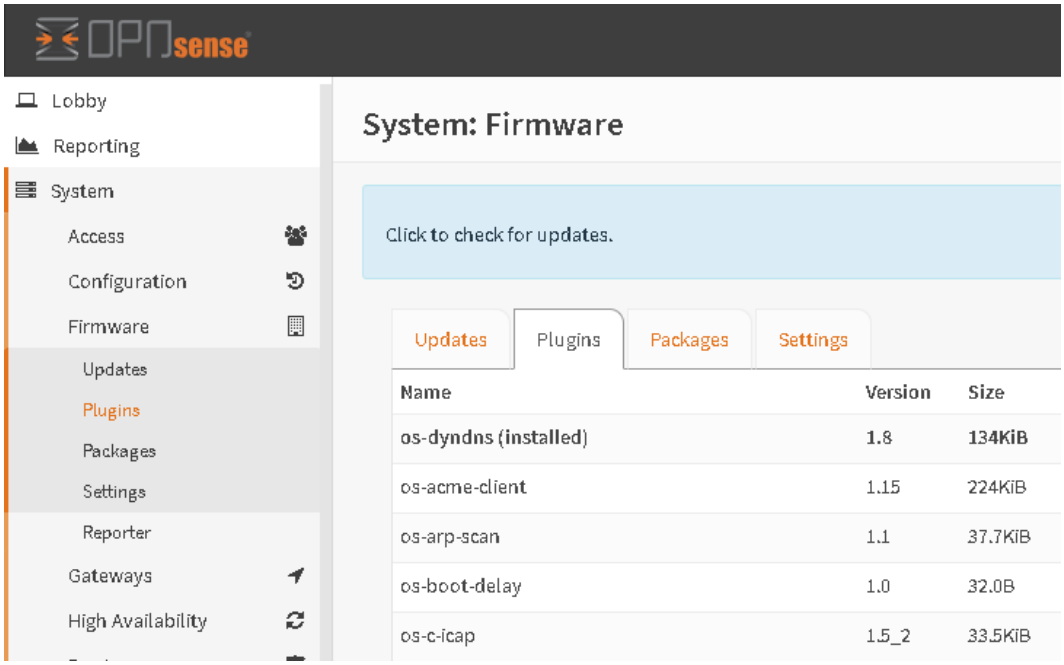

Figure 4: Plugin activation in the "Firmware" submenu of the web interface

The service can then be activated under Services – FreeRADIUS – General (Figure 5). The access points (AP) must then be stored in the "Clients" submenu (Figure 6). Only a name, the IP address and a shared secret are required. This must be stored on all APs to enable communication between the APs and the RADIUS service. (Figure 6)

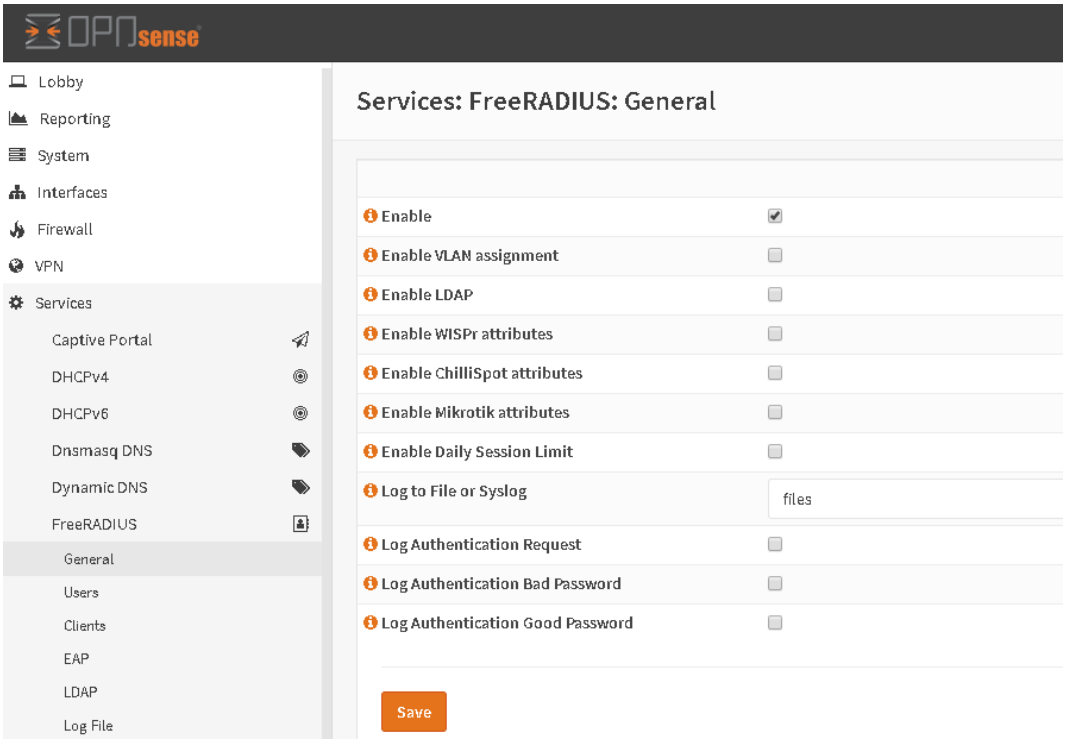

Figure 5: Activating the FreeRADIUS plugin

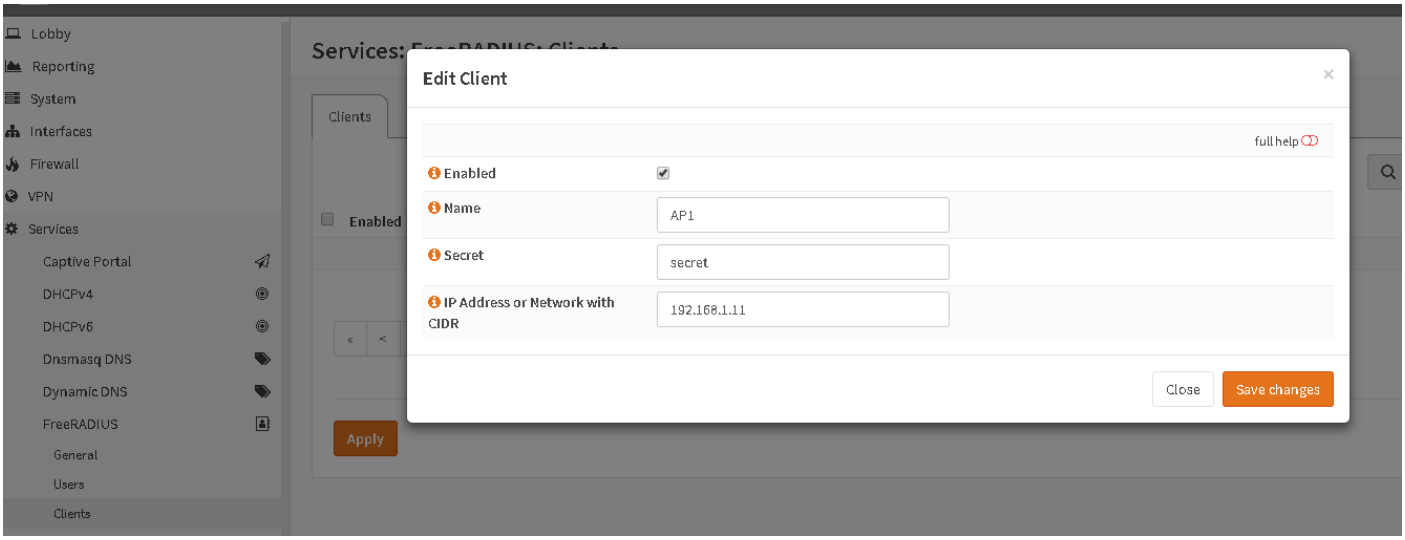

Figure 6: Adding access points as clients for RADIUS

Now users can be created in the "Users" submenu. During setup, some additional options are offered, but only username and password are necessary. If any unsupported fields are filled, the AP will ignore them (Figure 7).

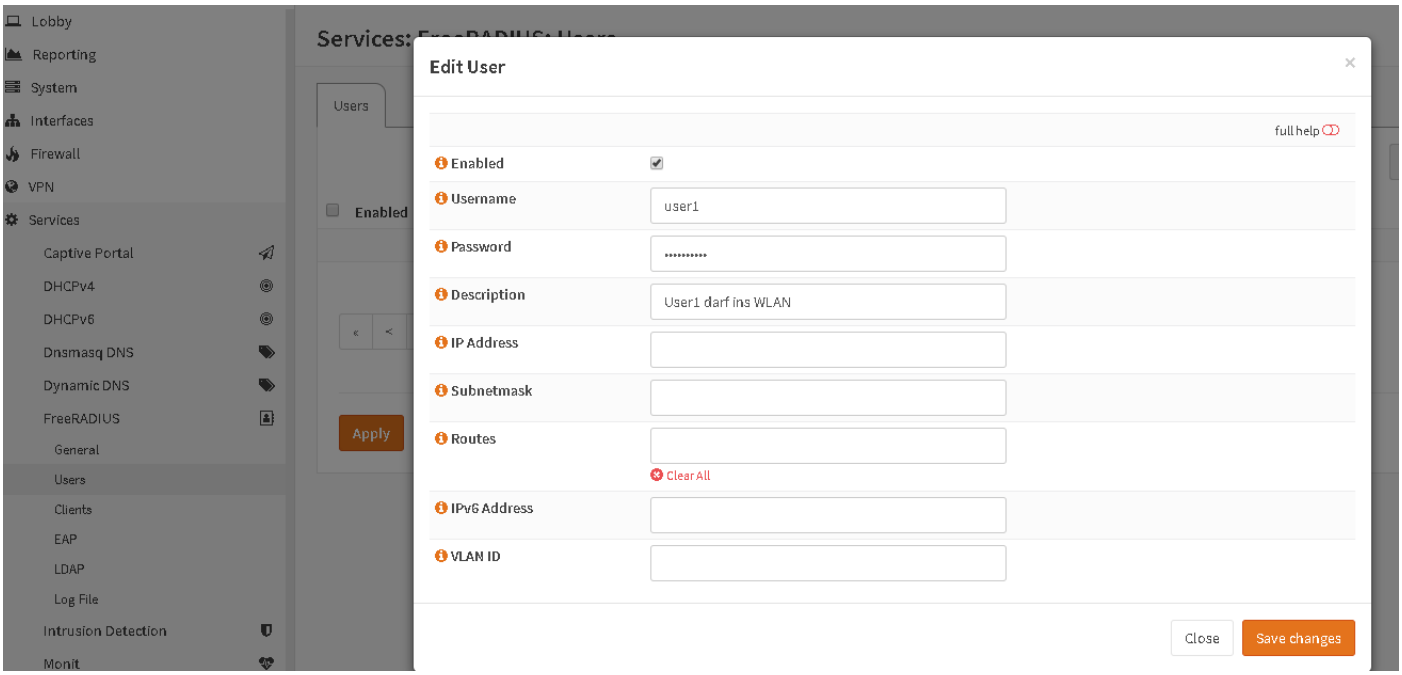

Figure 7: Manual user management with the FreeRADIUS plugin

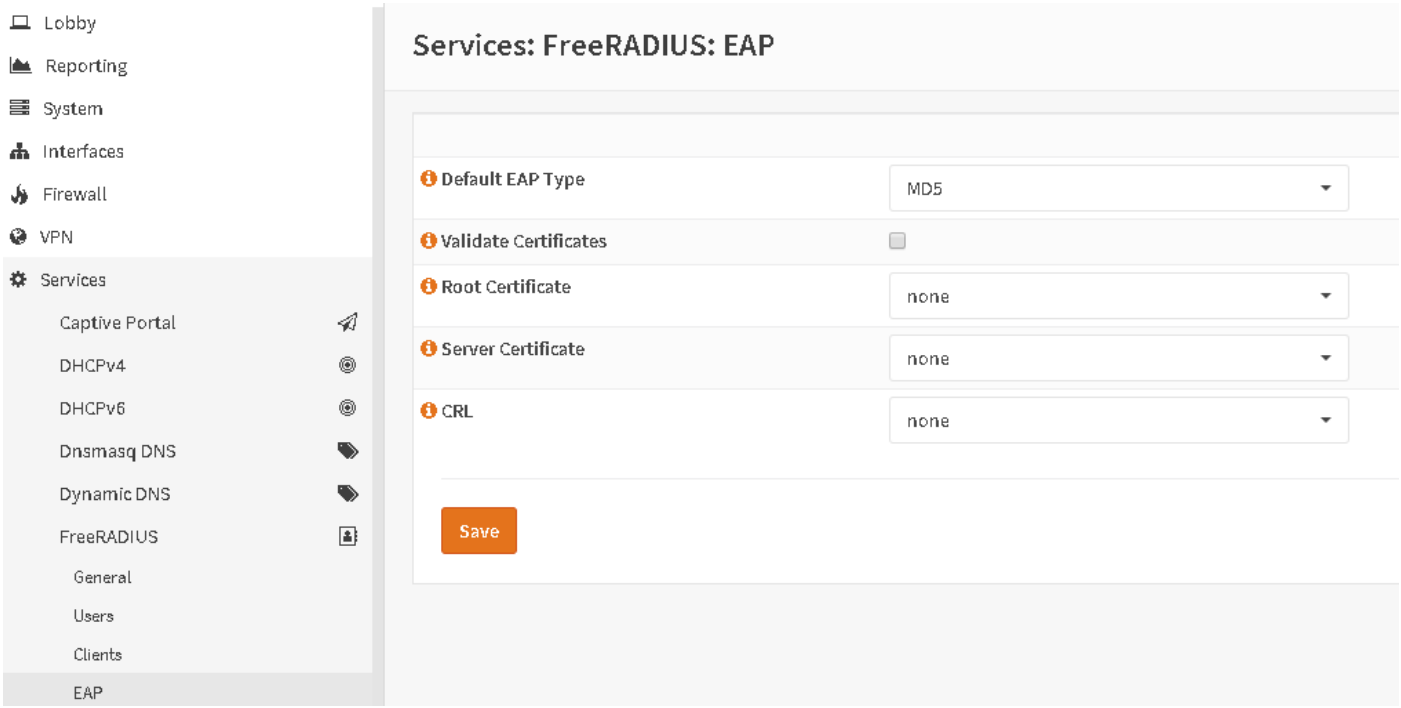

Figure 8: Authentication options for RADIUS

In the EAP submenu, the authentication method and certificate can be selected, but the default values are designed for compatibility so that the installation of a certificate is not required (Figure 8). For technically experienced users, the plugin also offers the possibility of coupling with the local Windows domain. But the possibility of using certificate-based authentication via EAP-TLS and the PKI management integrated into OPNsense give users great flexibility in terms of compatibility and security. The plugin is now configured and we can continue with the configuration of the APs. In our example, we are using a Cisco WAP150, but the configuration should be similar for most models from mainstream manufacturers. First, the OPNsense Radius server must be stored in the system. To do this, we log on to the web interface of the WAP150 and navigate to System Security – Radius Server. There we define the IP address, shared key and ports (Figure 9).

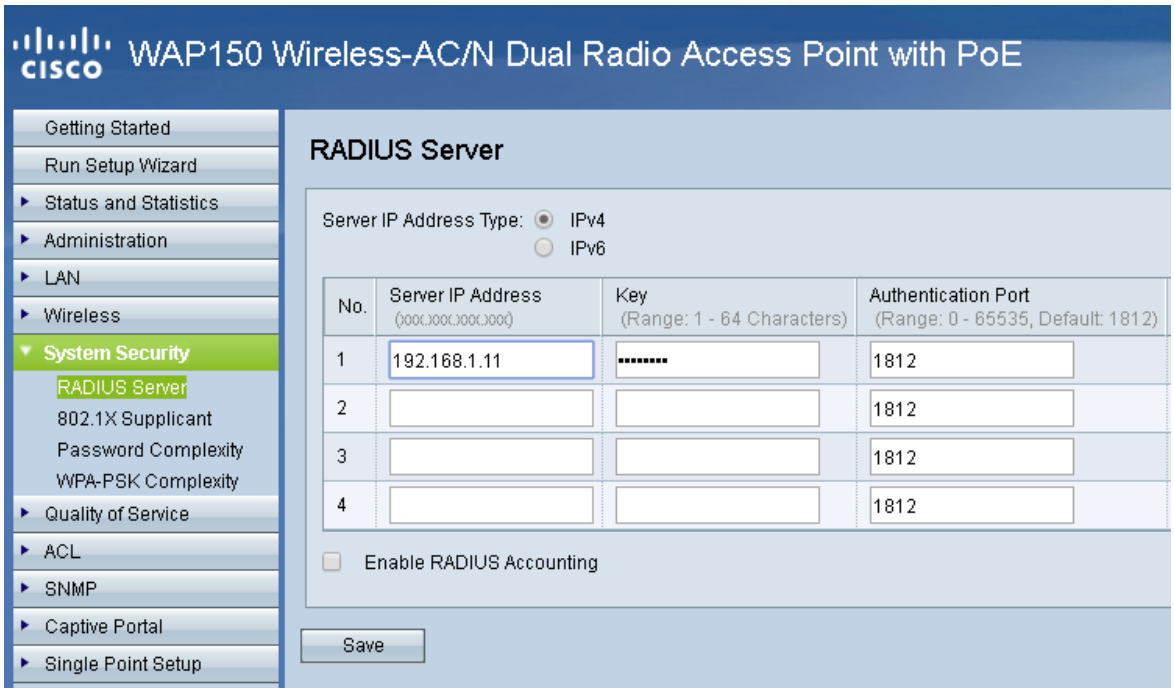

Figure 9: The Radius server data is stored at the access point

We then switch to Wireless – Networks and change the security for each frequency range (2.4 and 5 GHz) to WPA2 Enterprise and activate "Use global RADIUS server settings" (Figure 10). Users can now log on to the WLAN with their login.

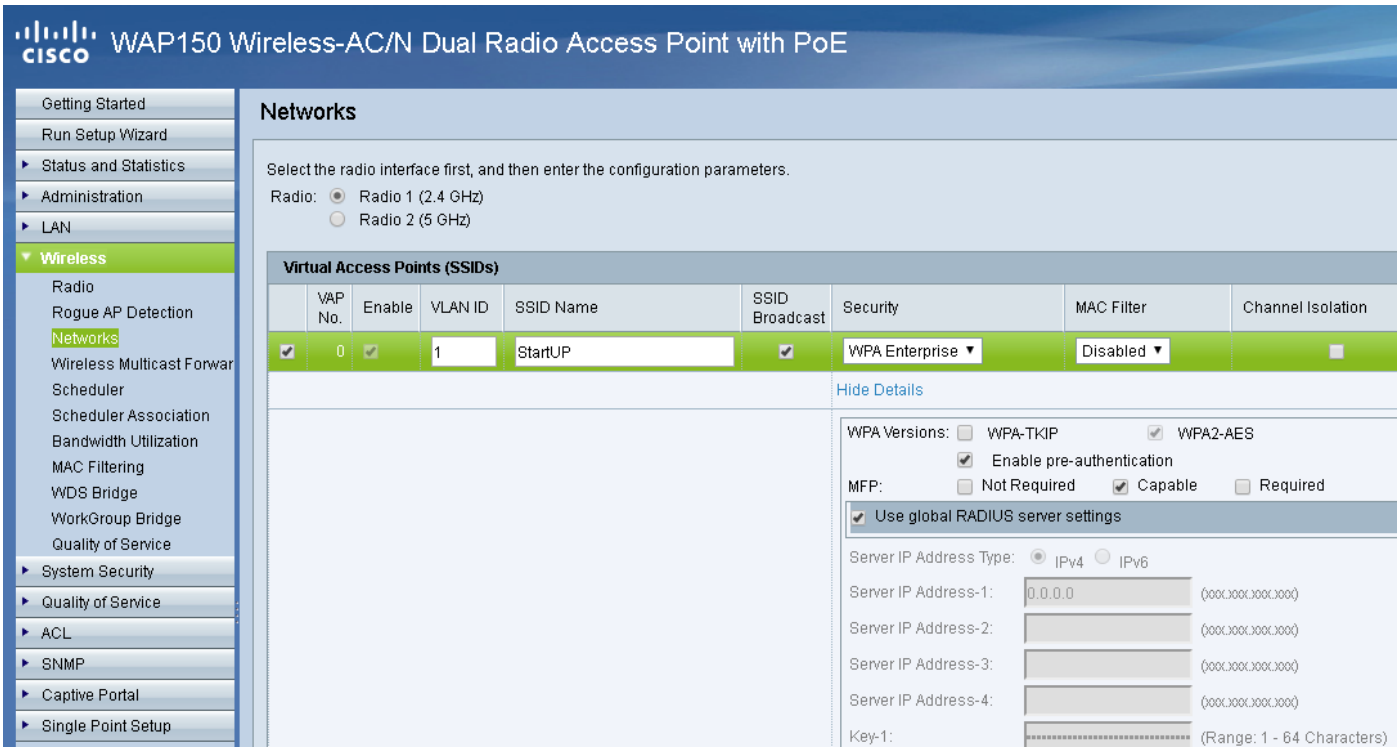

Figure 10: The security on the AP is set to WPA2 Enterprise for each frequency range.

8.2 Securing an Exchange Server with the Postfix plugin

Postfix has been a popular mail server in the open source area for many years. Continuous further development and a focus on security are the hallmarks of this project. In our example, it is used to secure an internal Microsoft Exchange server, which should not be publicly accessible if at all possible. The Postfix plugin therefore activates an additional SMTP instance. This prevents brute force attacks on local users or the exploitation of bugs that have not yet been fixed. The plugin also enables a pre-scan of known mailboxes in addition to anti-spam features. For our case study with the company StartUP, we allow the internal LAN to send e-mails, route company e-mails to the local Exchange server and block some unwanted advertising senders. Moreover, in our use case, the anti-spam plugin "Rspamd" should be evaluated. As usual, the installation of the plugins is performed via System – Firmware. To cover the whole range – mail server, spam and virus protection – we install the plugins "Postfix", "ClamAV" and "Rspamd". First, we navigate to Services – Postfix and the submenu "General" to activate the service, set the system hostname and domain and add our local LAN to the Trusted Networks. Optionally, a banner can be set, the maximum mail size can be edited and local certificates can be configured for encrypting mail traffic (Figure 11).

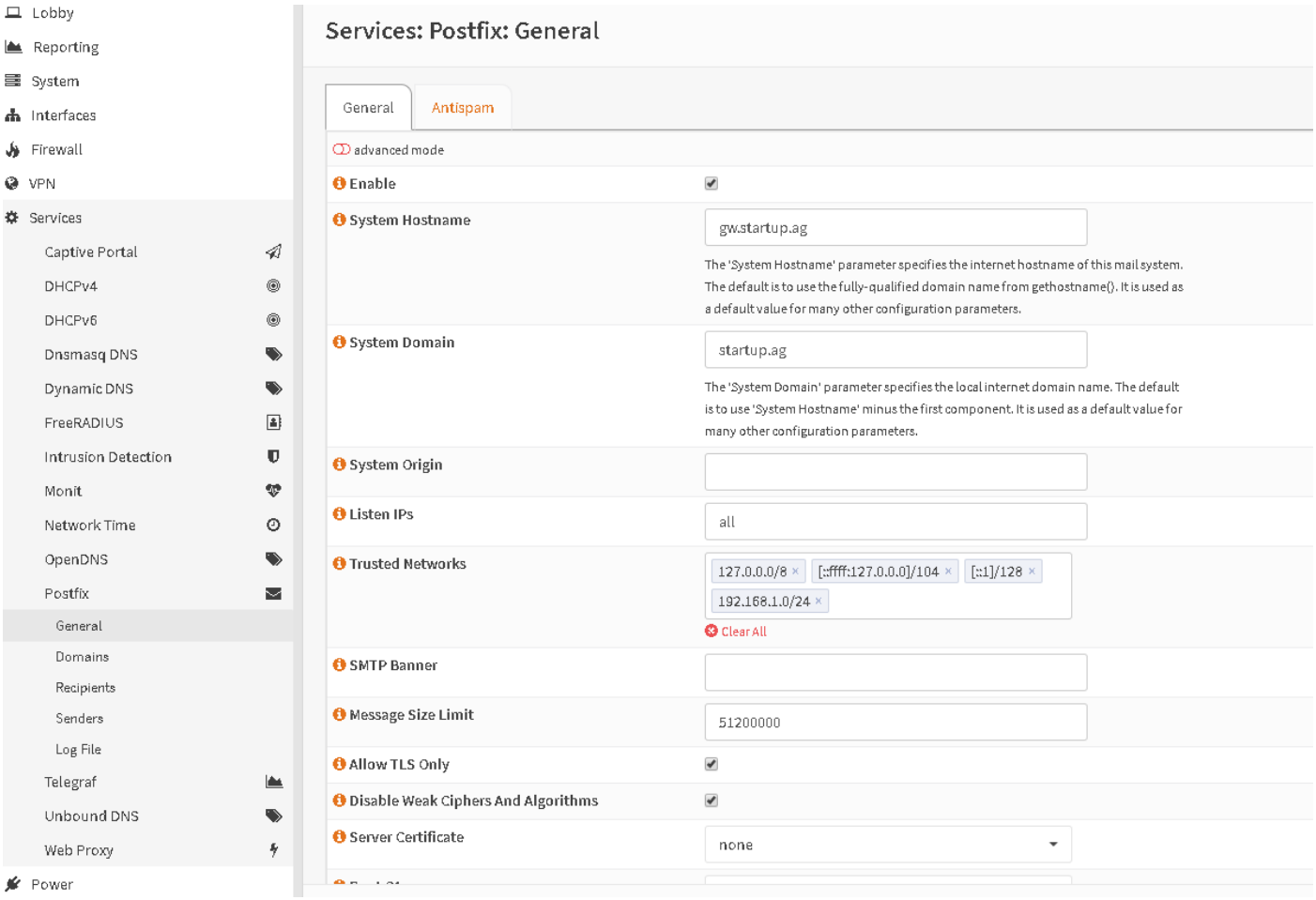

Figure 11: General settings for Postfix

We then set the two company domains used for e-mail traffic in the "Domains" submenu and send them to the Exchange server (Figure 12). Using the submenus "Senders" and "Recipients", rules can be defined based on the sender and recipient. As an example, we could block a particularly intrusive distributor from Asia, who sends us their current offers every day.

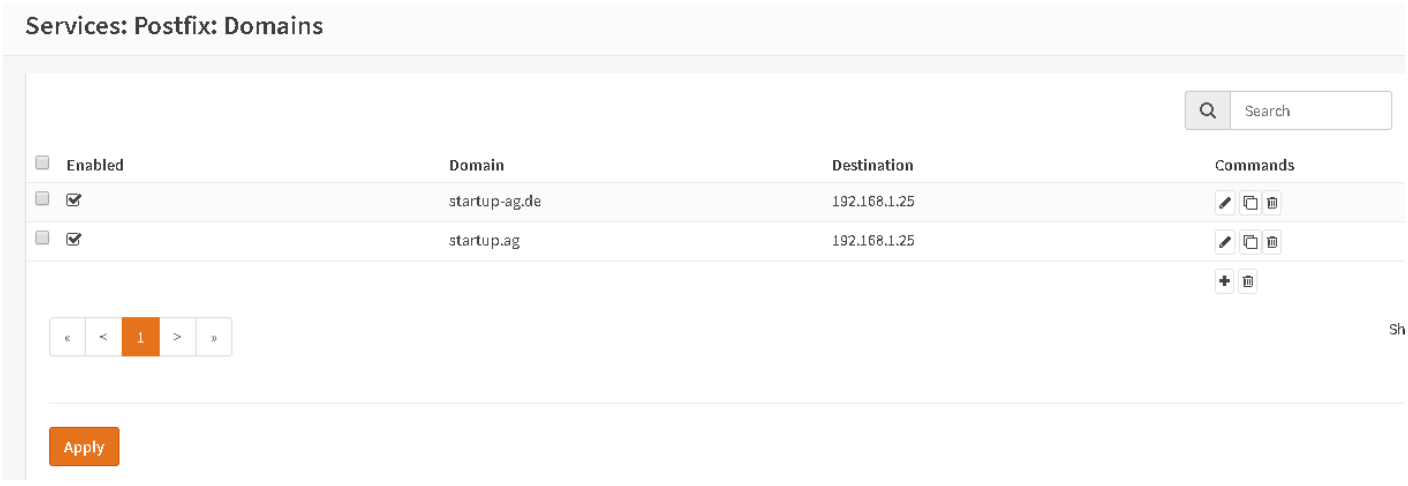

Figure 12: Domain settings for mail forwarding to Exchange

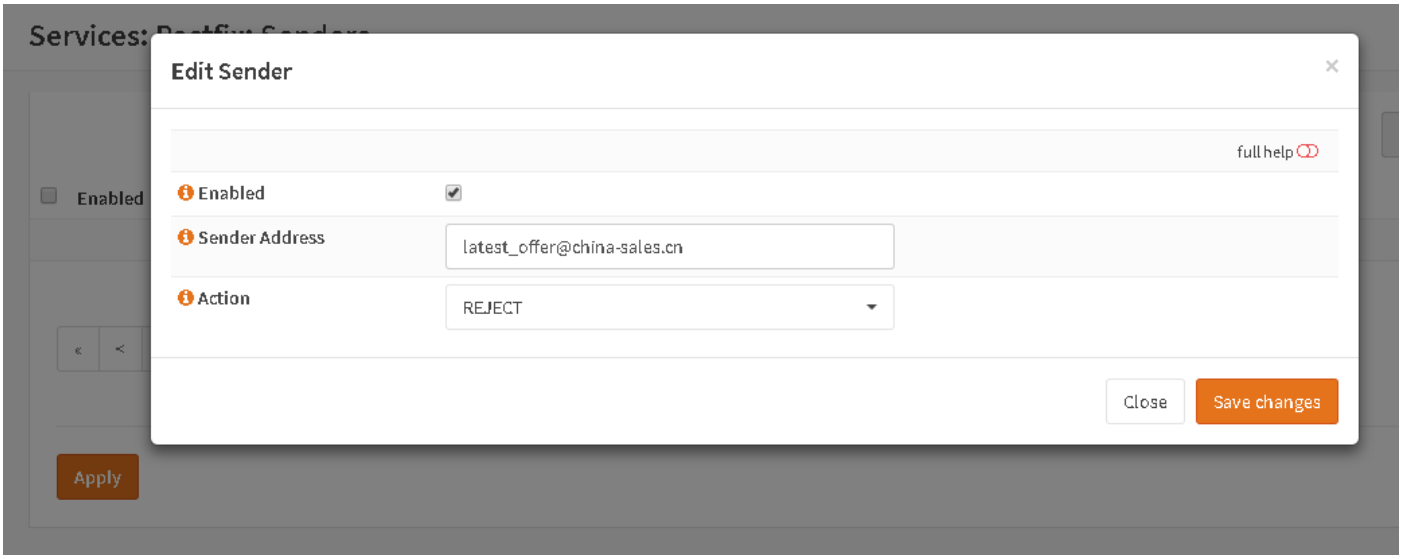

Figure 13: Sender and recipient-based rules

The basic configuration is now complete and we can take a first look at anti-spam and anti-malware. To do this, we switch to Services – ClamAV and are initially asked to load the antivirus database (Figure 14).

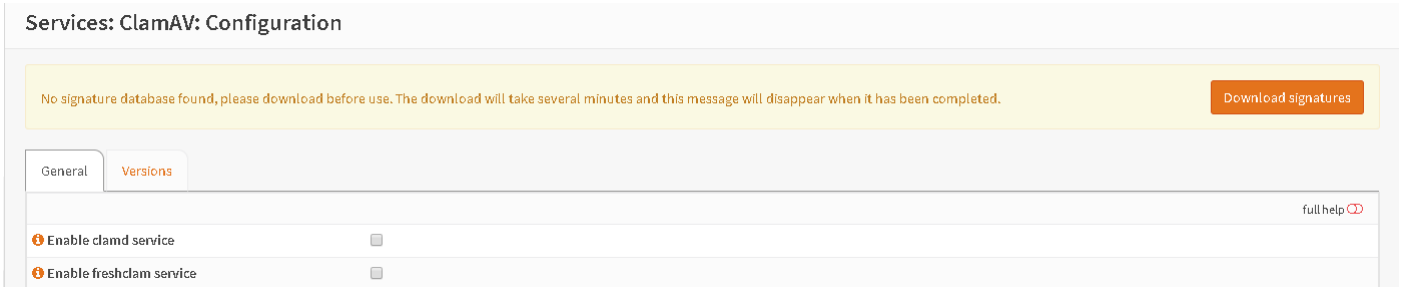

Figure 14: Loading the antivirus database in ClamAV

After loading we activate the services "Clamd" and "Freshclam", leave the default values unchanged and switch to the plugin Rspamd. The setting options in this plugin are very extensive. As a result, the help setting should be turned on to provide brief explanations of the various fields. The default values are sometimes very restrictive, e.g. .exe attachments are prohibited by default. However, it is possible to adjust the file types under General Settings. Via "Milter Headers", the mail headers can be adjusted similar to SpamAssassin – e.g. the "Spamd Bar" (Figure 15). The plugins that Rspamd offers under the "Spam Protection" menu are so extensive that we will not cover them in detail here. For now, we would simply recommend using the default settings as far as possible, but to allow .exe attachments and mark spam in the header (Figure 16).

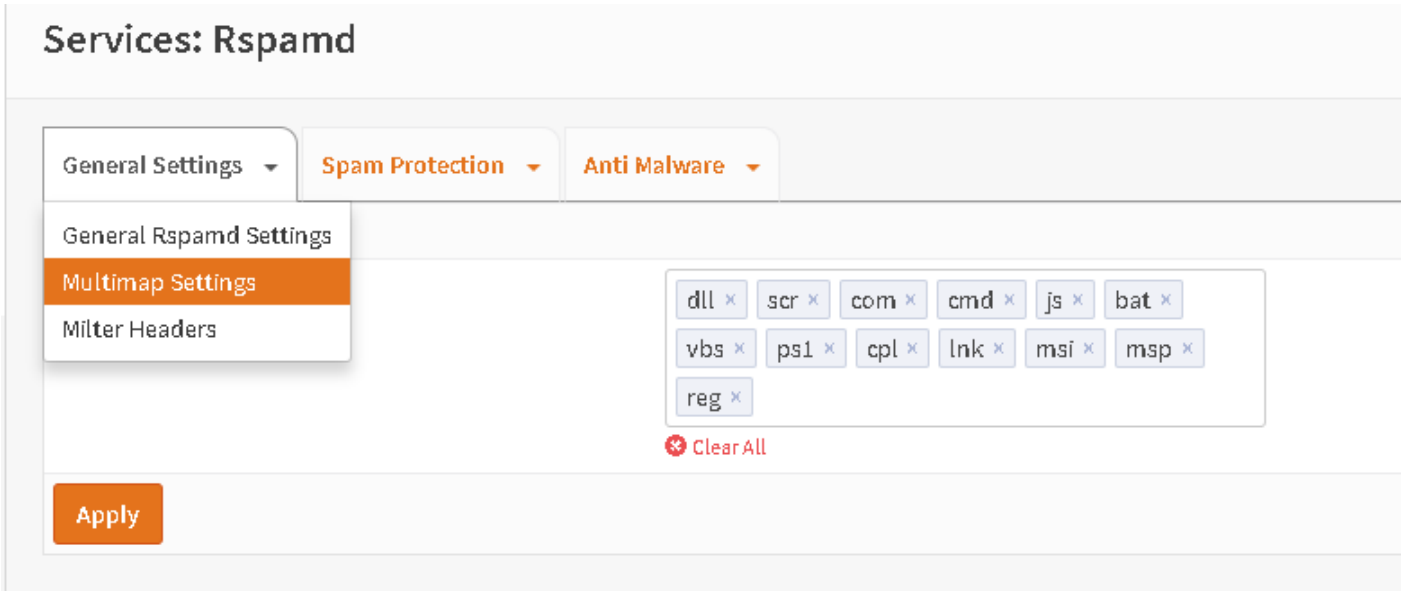

Figure 15: Adjusting the Milter headers in Rspamd

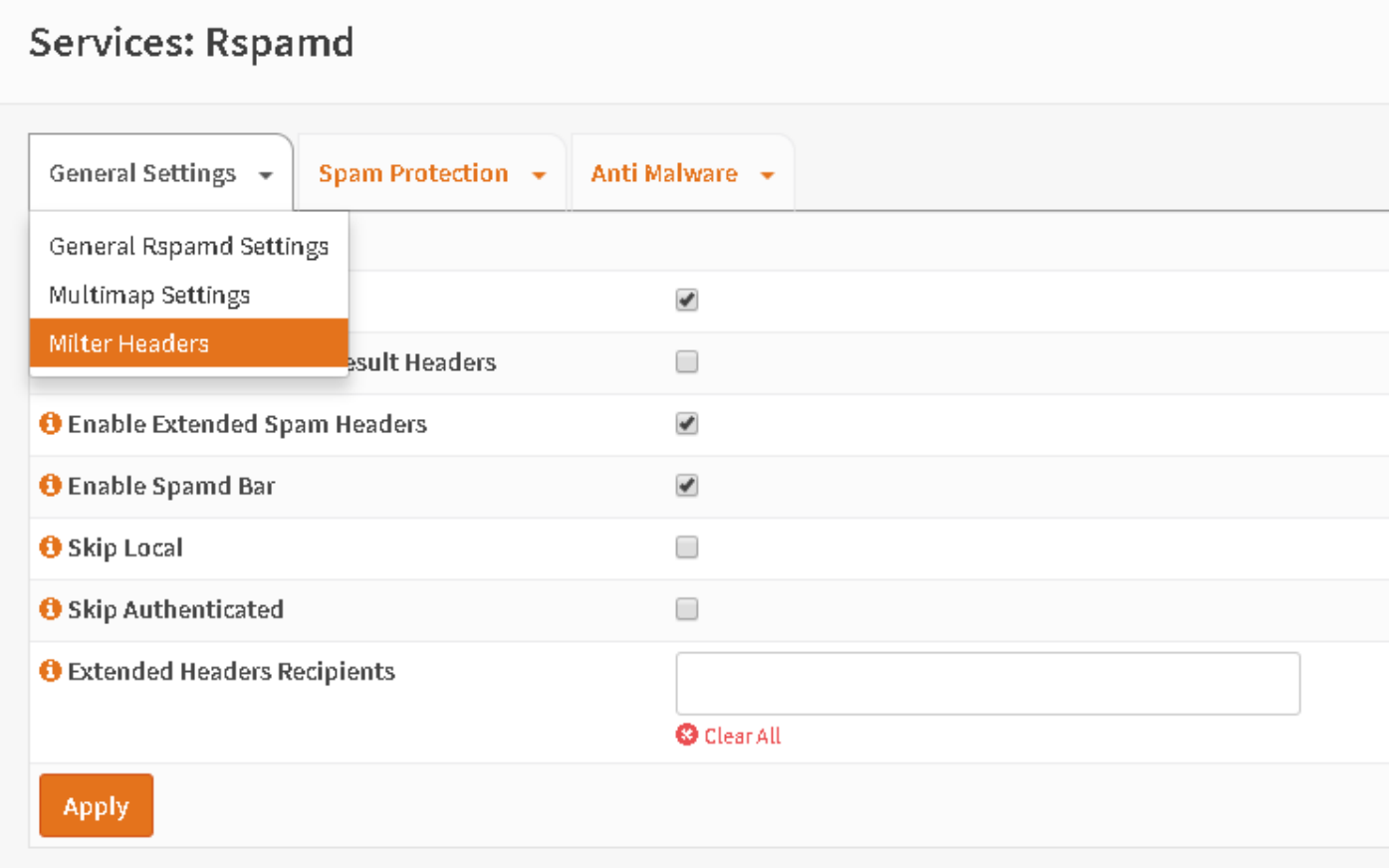

Figure 16: Setting the spam protection rules in Rspamd

To link Postfix with Rspamd, we now go to the Antispam tab under Services – Postfix – General. Here, we can add and remove services for testing purposes.

8.3 Monitoring with the Telegraf plugin

Telegraf is a monitoring agent that was developed in the Go language and can therefore be used on nearly every common platform. The agent collects performance data from the local system, so-called inputs, and delivers them to any target (output). Best suited for this is InfluxDB, which is also made by the main developer of Telegraf. InfluxDB is a Time Series Database (TSDB) and stores metrics more efficiently than MySQL, for example. With Grafana, the data can be conveniently visualized and alerting options are also available. The plugin is installed as usual via System – Firmware – Plugins. It can then be found under Services. In the "General" submenu, the default values are well chosen so that only the "Enable" option needs to be chosen and the hostname entered (Figure 17).

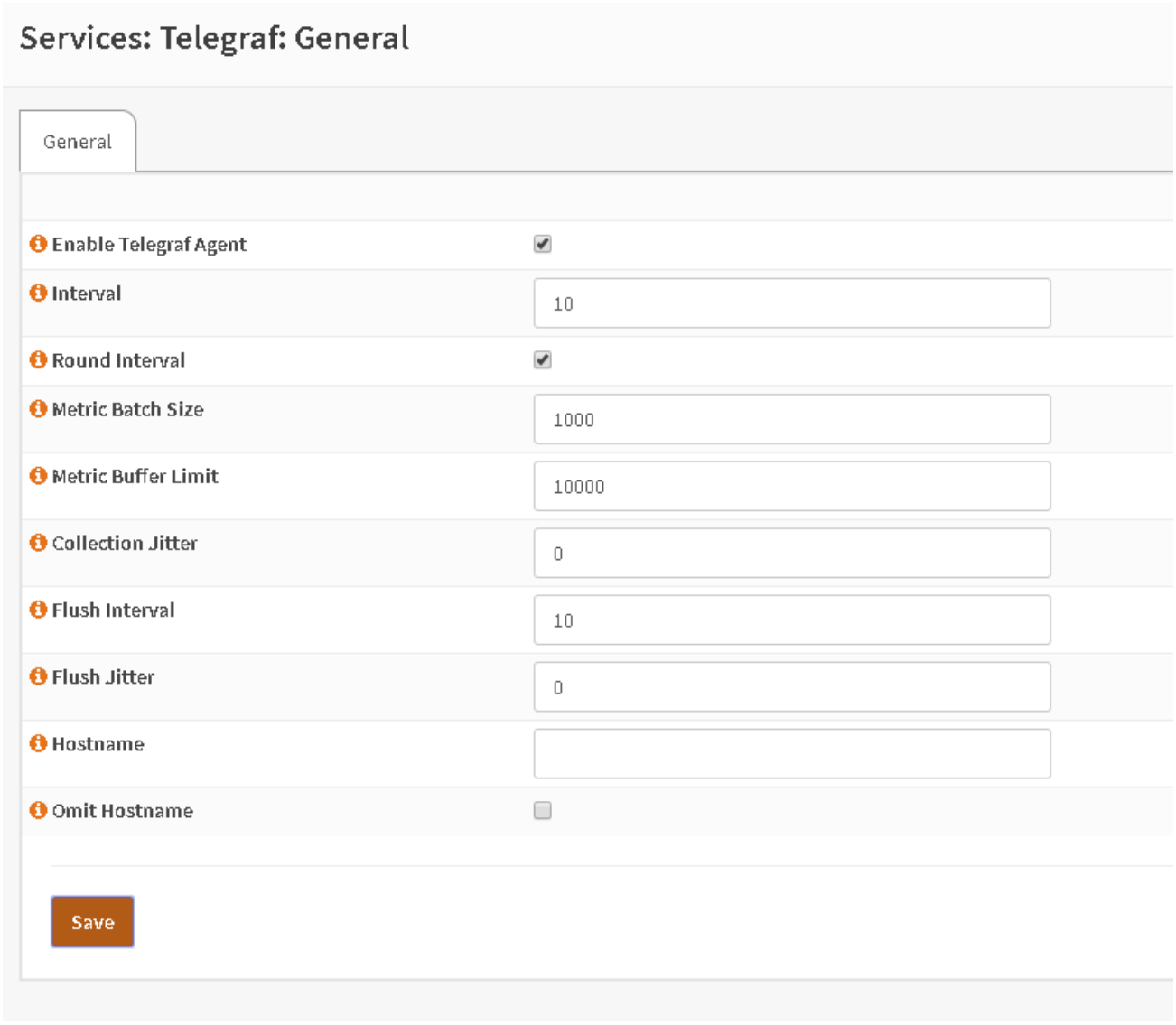

Figure 17: General settings of the monitoring agent Telegraf

The Inputs menu (Figure 18) lists all available monitoring points, of which all the most critical are already activated. Optionally, external systems can be queried with the ping input.
In the Outputs section, the configuration is completely missing, because they can be sent to Influx, Graphite or Graylog. In our case, we activate the Enable checkbox, set the hostname of the database as well as the username and password and we are ready to go (Figure 19).

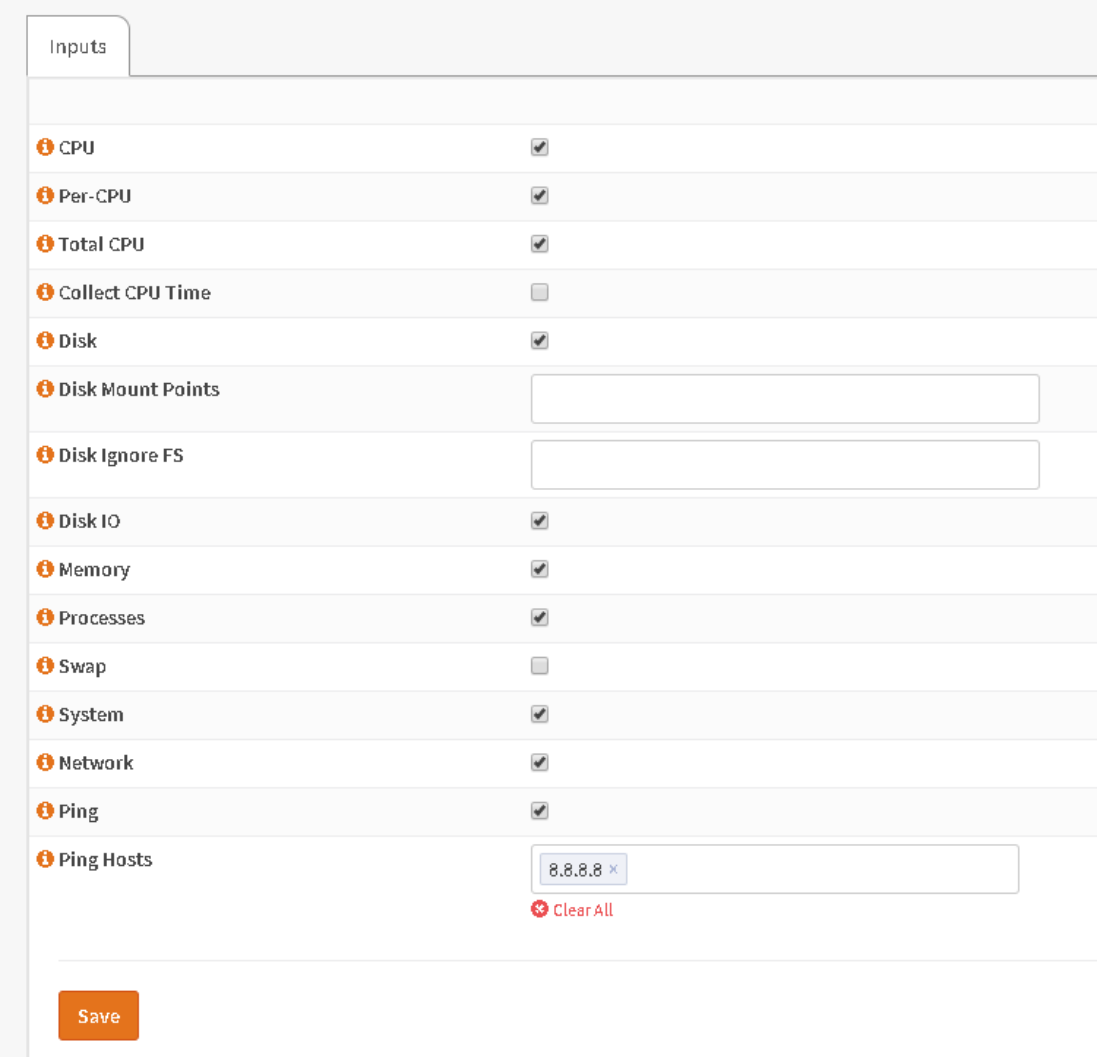

Figure 18: Inputs for Telegraf

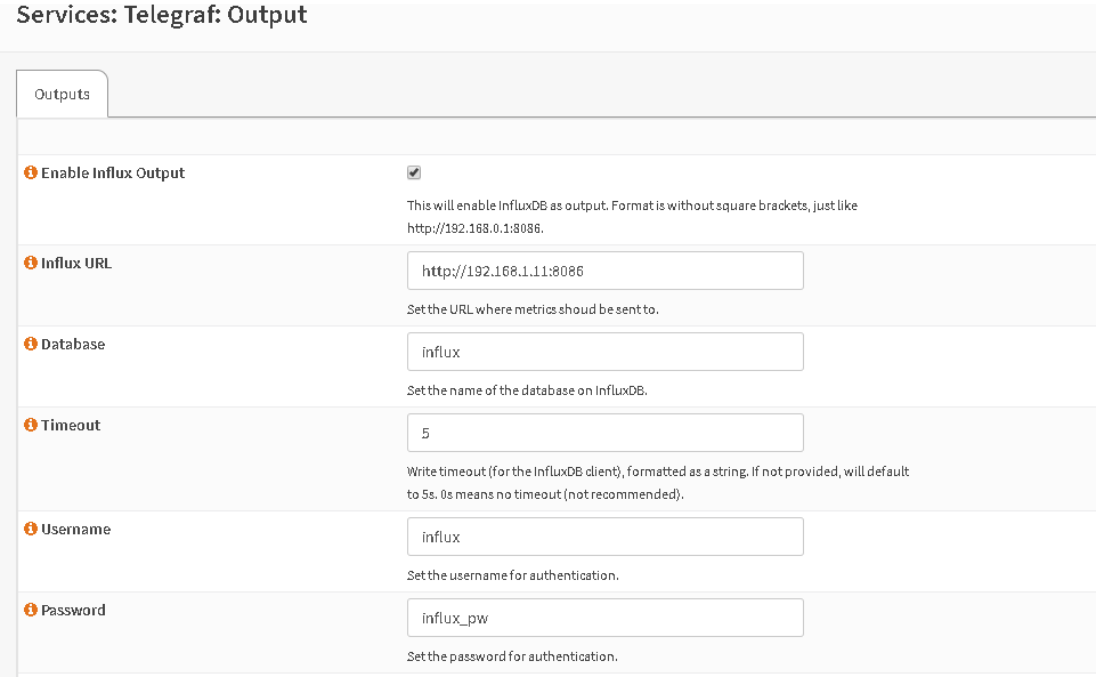

Figure 19: Configuration of the InfluxDB for Telegraf

Grafana is used for visualization and alerting. Detailed instructions on Telegraf, InfluxDB and Grafana would go beyond the scope of this e-book but are available online from various websites and blogs.

A quick guide to the setup of InfluxDB and Grafana can be found on m.a.x. it's TechCorner wiki9. Figure 20 shows how a Grafana dashboard visualizes the metrics. Alerting is also easy to set up (Figure 21).

Figure 20: Visualized metrics in the Grafana dashboard

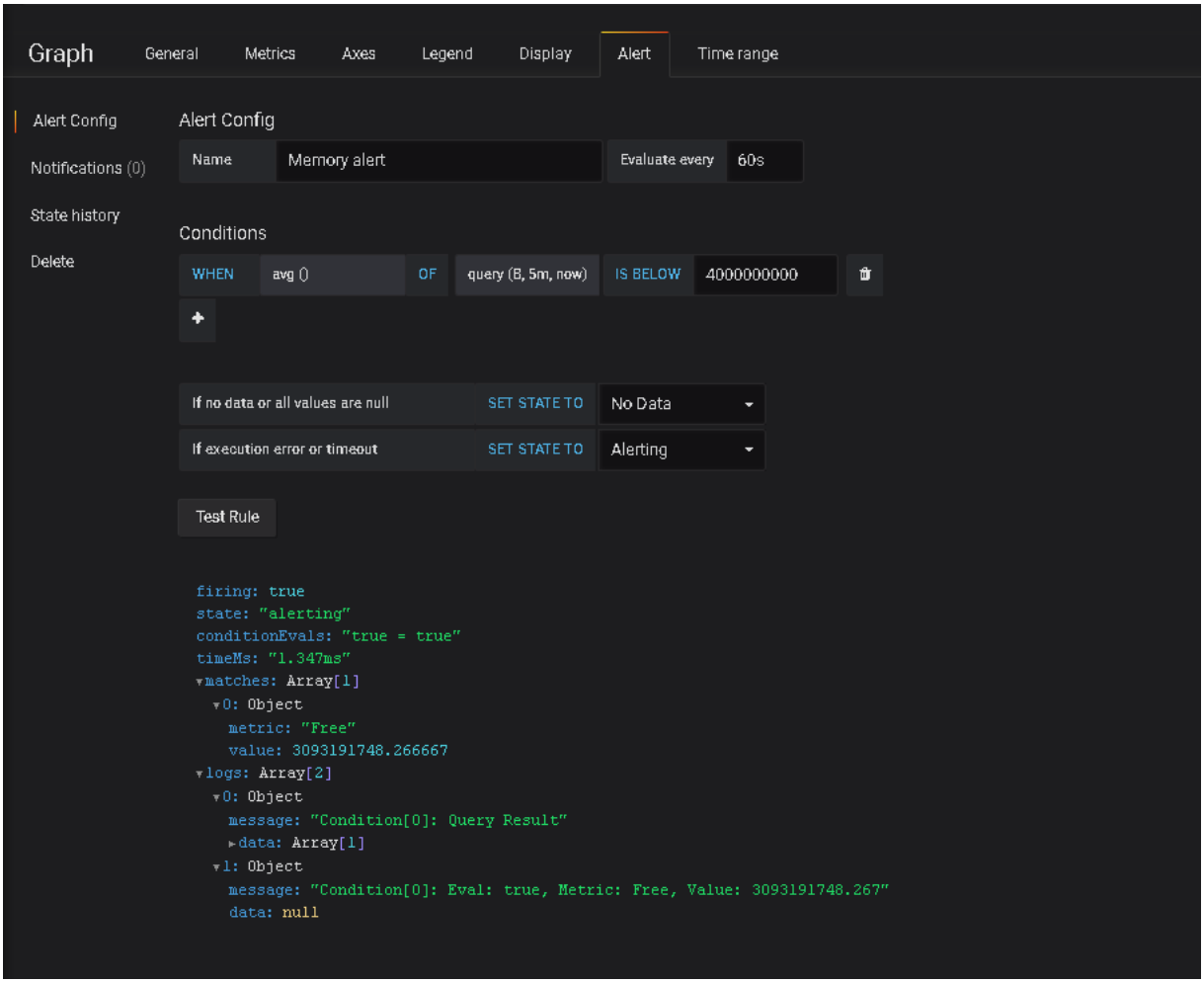

Figure 21: Setting up alerts in Grafana

9 Installation of InfluxDB: https://techcorner.max-it.de/wiki/Installation_von_InfluxDB_und_Grafana_auf_Debian_9 [German]

8.4 NUT plugin for UPS integration

NUT stands for Network UPS Tools and is a collection of scripts and UPS drivers to connect with a wide range of UPS devices from common manufacturers. The basic idea is that a system is connected to the UPS via a USB cable and retrieves the current status. This makes it possible to determine, for instance, whether it is receiving power and how much charge is left in the battery. What makes this plugin unique is that the firewall can shut itself down in a controlled manner when it sees that the battery has only ten percent remaining – thereby preventing a possible corruption of its file system. The program offers two different modes: Standalone and Netclient. In Standalone mode, the firewall itself is attached to the UPS and can thereby grant UPS access to other devices that have Network UPS Tools installed.

As usual, we install the plugin via System – Firmware – Plugins and find it in the Services section. There, we set the mode to Standalone and give the UPS a name. Important: The plugin should not be activated yet! This should only be done after the other settings have been set, otherwise the service will fall into a timeout loop and will no longer respond. This bug has not yet been fixed, although it has been known to the NUT developers for a long time. We switch to the tab UPS Type and select the driver USBHID-Driver from the drop-down menu. The driver supports many UPSs from APC and other manufacturers. The Extra Arguments field should already be set correctly, so we just need to enable the service with the checkbox (Figure 22).

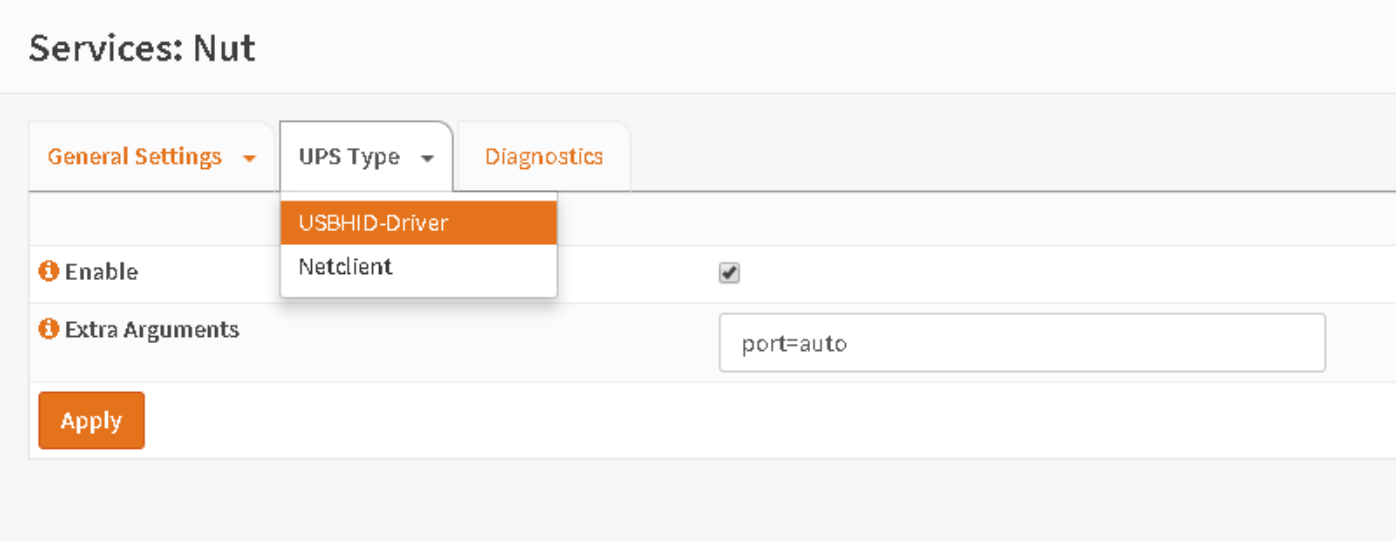

Figure 22: Selecting the USBHID driver

You should now check whether the USB cable is connected to the UPS. Next, return to General Settings and activate the plugin. UPS performance data should now appear in the Diagnostics tab and update every ten seconds. If another device running the NUT tools is already connected to the UPS, the firewall can still connect to the service there in "Netclient" mode. To do this, select Netclient as the driver, enter the IP address, username and password and then select and activate the mode in General Settings.

Local access data can also be entered via General Settings – Nut Account Settings. Having done this, we only need one port forward, which forwards external requests to localhost. This will allow other devices to communicate with the UPS via our OPNsense firewall. To do this, simply add a new entry in Firewall – NAT – Port Forward with the interface set as LAN, the source as LAN net, destination as LAN address, the port range as 3493 and the redirect target as 127.0.0.1 (Figure 23).

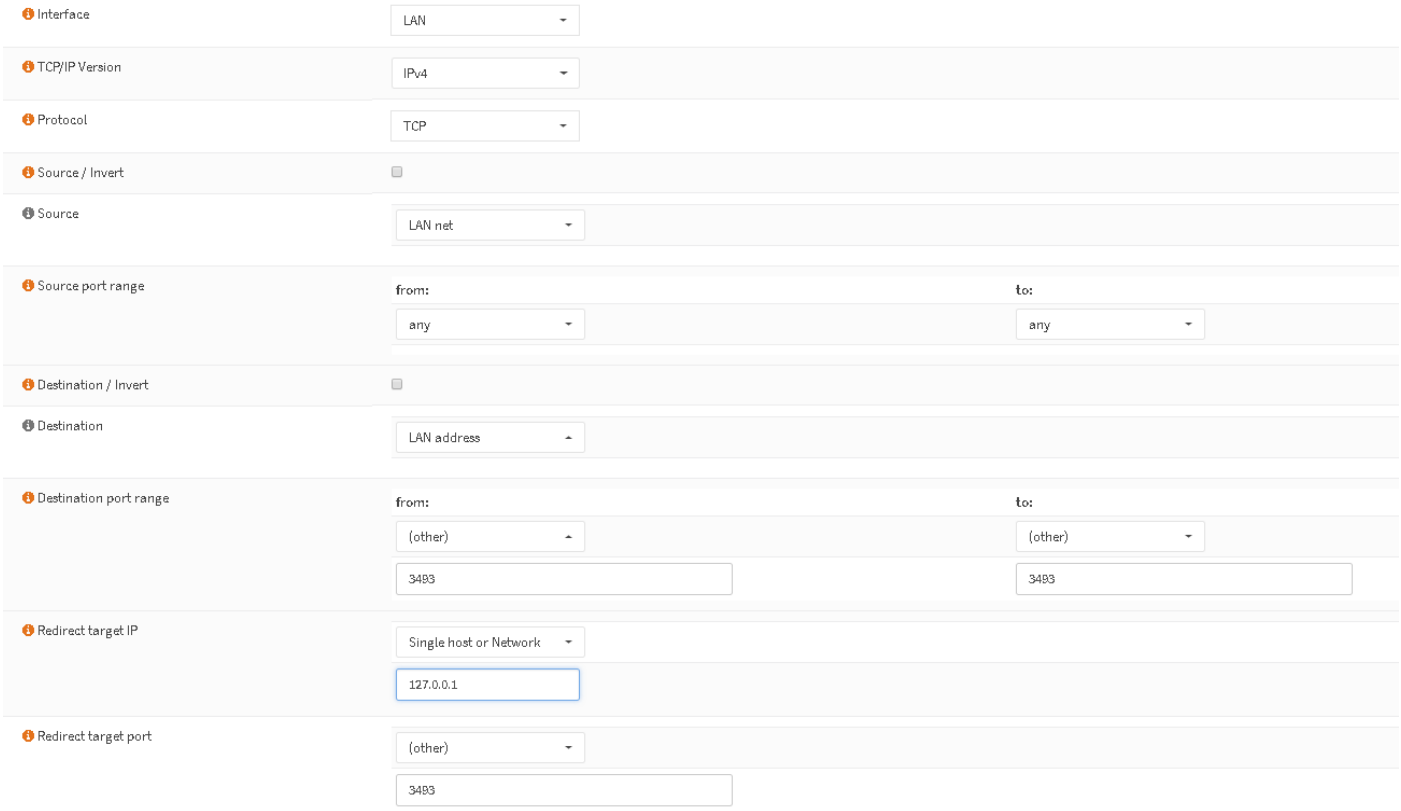

Figure 23: Port forwarding for NUT

Summary

We were able to successfully meet the requirements of our example company "StartUP AG" with only a few adjustments. StartUP AG is now well-prepared for future challenges, such as anti-spam and antivirus protection for web-based applications. This example highlights how OPNsense's plugin system covers a variety of different areas – from WLAN security to mail server protection and hardware monitoring. The installation and configuration of the plugins always follow the same pattern. The default settings are usually reasonable. Both of these facts save administrators valuable time.

In conjunction with OPNsense's core functions, such as NAT, interesting deployment scenarios can be realized without having to install additional services outside of OPNsense, which increases the complexity of the infrastructure. Small bugs in individual plugins occur occasionally but can be fixed with the necessary know-how. Above all, its ability to connect with central IT components, such as the mail server or modern monitoring systems, shows that OPNsense is not merely a pure firewall solution, but an easy to administer infrastructure hub for the entire corporate network. Due to the pure open source philosophy of the project, future security is also guaranteed and vendor lock-in is avoided. External service providers can be commissioned at any time to adapt existing plugins or develop new ones.

Thomas-Krenn.AG
Speltenbach-Steinäcker 1
D-94078 Freyung
thomas-krenn.com

THOMAS KRENN®