Remote Deployment of RADIUS and Syslog Services

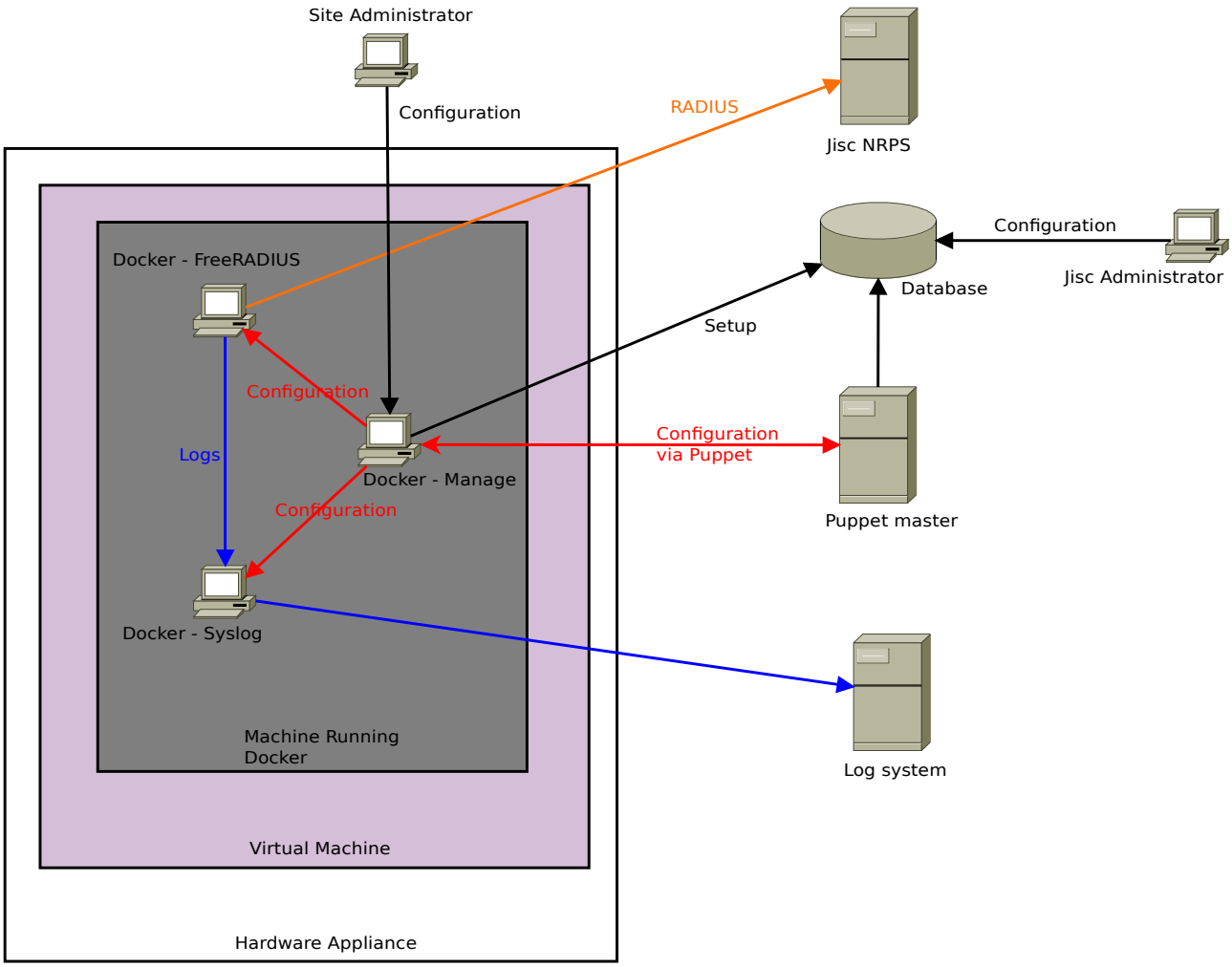
Summary

There are several challenges facing Govroam which can be solved by remotely deploying RADIUS and Syslog services on client sites. The main problem is that most sites are running MS NPS, which is impossible to configure in a way that fulfils many of the Govroam requirements. Lack of attribute additions, sensible logging, remote syslog all mean that there is not enough audit and logging information being passed to Jisc. The secondary problem is that the RADIUS services, NPS or otherwise, are proving a difficult challenge for administrators to configure properly. Many aren't putting in sufficient checks on realms and much of the initial set up is slowed by them having to learn how EAP works and mistakes with the shared secrets. There appears to be very few administrators in the public sector who use Unix/Linux systems, which makes it much less likely for them to run the more capable RADIUS servers.

Anything that removes the burdens from the administrators, removes MS NPS from the RRPS level and increases the logging will make Govroam appear more attractive and improve the marketability of the service.

Solution

Provide a way for system administrators to deploy a RRPS easily, quickly, accurately, with minimal knowledge required. If this system is, to an extent, managed by Jisc then most of these criteria can be fulfilled. However, Jisc don't want the overhead of managing the day-to-day configuration changes associated with adding new sites, just with the initial deployment and communication with the NRPS and central logging facilities.



Benefits

Very simple deployment and management for site administrators

- No specific Unix/Linux knowledge required.
- No specific RADIUS knowledge required.

Flexible options designed to accommodate most sites

Easy to manage infrastructure for Jisc

Flexible configuration for site administrators

Fully configured RADIUS with all required and desired options

Full logging for F-TICKS and any other information Jisc wants

Can be marketed as a quick option for going live – RRPS could be up in hours.

Proposition

A three layer approach that should cover 99% of the use cases. These can be developed in turn as and when needed and resources are available:

1. Docker containers for FreeRADIUS and Syslog,
2. A virtual machine image running the aforementioned Docker containers,
3. A hardware appliance, which is essentially a commodity server running the aforementioned VM.

This approach reduces to just managing the Docker containers, a minimal OS image for the VM and the maintenance on the hardware. There are not three separate services that need development, maintenance and patching.

Sites which have docker capability already can use the docker containers, sites which don't have docker but do have a VM infrastructure can use the virtual machine image. Sites which have neither can install the appliance.

The critical parts are the docker containers and will need the most development.

FreeRADIUS

FreeRADIUS is the most fully functional RADIUS server around and has the ability to add Operator-Name, send F-TICKS via a syslog server, load balance between RADIUS proxies, modify/add/filter attributes and sensibly handle badly formed realms. The general configuration would include up-to-date bad realm lists to ignore and customised logging to suit Jisc's needs.

Syslog

The syslog server provides a path way for the F-TICKS from FreeRADIUS to the Govroam log handler software, ELK. In the future, if other Jisc managed proxies are installed further inside the organisation then the top level, external facing Syslog server could proxy F-TICKS from these to the NRPS.

Docker

Docker runs natively on Linux systems but is commonly installed and run on Windows and Mac OS X.

Docker containers are operating system subsets that generally run just one command or service. In this case the key containers would be for FreeRADIUS and Syslog which are stored in a central Docker repository, most likely hosted by Jisc. By providing the site administrators with a *docker compose* file they can quickly and easily download, install and run these images. The 'cloud' based storage and access is one of the key Docker features.

Each will need both a general and a site specific configuration. For instance, FreeRADIUS will be configured with a general Govroam proxy configuration and will also need the specific shared secrets need for communication between its site and the NRPS. The general configuration would be configured into the container by default and the specific configuration downloaded.

Configuration management

A popular tool for managing configuration is *puppet*. Clients with an installed puppet agent talk to a

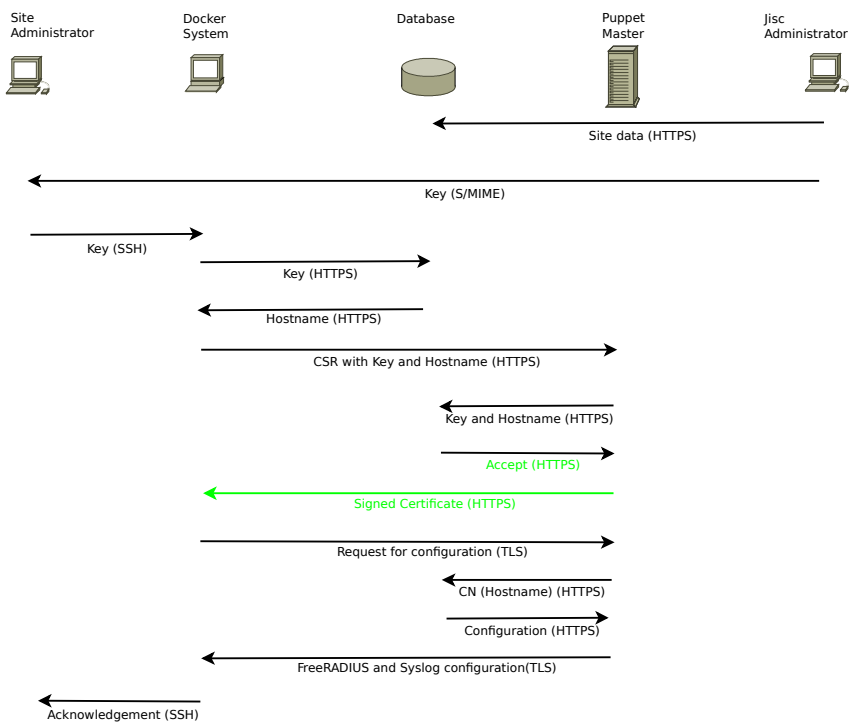
central server, a Puppet Master, where the configuration for each client is stored. The configuration is pulled down and installed on the client. This happens periodically and is driven by the client, so doesn't need the clients to be accessible directly from the internet (i.e. they can be behind a NATed gateway).

Thus if there is a third docker container running a puppet client then it can be used to configure the FreeRADIUS and Syslog servers with the correct site-specific information without any intervention from the local site administrators. At the Jisc end all that is needed is to have those details (shared secrets) available in the puppet server configuration.

Security

Automatic transfer of a site's shared secrets has to be secure and possible only by an approved person for that site. The way to do this is:

1. At the point where a site joins the site administrator is sent a key, securely.
2. They run a docker command on their infrastructure which takes this key and does the following:
    1. Requests a host name via HTTPS from a central Jisc server using the key as the index,
    2. Configures up a CSR with the key and the host name,
    3. Sends the CSR as part of the puppet sign up process,
    4. The server checks the key and host name against the central Jisc server,
    5. if matching, the CSR is signed and the puppet client is allowed access,



3. The puppet client now has a secure communication path to the puppet server and the server can uniquely identify the client from its host name,
4. Puppet requests its site specific configuration which is then propagated to FreeRADIUS and Syslog.

Whilst there is no way of securing the docker images from inspection from site administrators, there is nothing in them which could be used to compromise the system. They can only create a

certificate that matches the data on the Jisc server, they can only use the shared secrets with the IP addresses of the server, they can only use the key to request a host name. They can't use the key twice because the puppet server would baulk at two identical host names. They can't create a CSR which would work unless they had another key and matching host name.

Local Site v. Jisc Administration

Jisc would be responsible for keeping current the docker images, secrets shared with the NRPS, bad realm lists and anything else which is 'internal' to Jisc.

The local sites would be responsible for maintaining the configuration between the RRPS and the ORPS/IdPs. They would be provided with a method of adding/editing/removing ORPS/IdP shared secret information. This could be in the form of a simple command, a basic text interface accessed by sshing into another docker image or a web interface on another docker instance.

Virtual Machines

The VMs are just a wrapper for the docker containers. Minimal resources are required (1 CPU, <1GB memory, 1GB storage). The operating system would be the smallest capable of running docker. It would run its own puppet instance and would be sent out with minimal software installed. A similar approach for the above would be used where a key is given to the administrators. They'd give the VM an IP address, ssh into it, which runs a script where the key is entered, the containers are downloaded, configured as above. Then some form of management interface is provided (ssh, web page etc.) for managing the local site configuration.

Hardware Appliance

This is, say, a Dell server running a hypervisor with the above VM pre-installed. Configuration is just as above.

Full On-boarding Process

Docker

1. Site joins Govroam by filling in a boarding form with RADIUS IP addresses, secure contact email address and request for a docker-based system
2. Jisc sends a key for each IP address to the secure email address with the 'docker-compose' script and instructions.
    - Jisc configures the puppet database with the key, IP, host name and shared secrets for each remote system.
3. Site runs the docker setup command on the system with the first IP address.
4. They enter the key for that IP addresses
5. The RADIUS server and Syslog server configuration are downloaded and site informed

6. They run other docker commands to start up the RADIUS and Syslog servers. Communicated between RRPS and NRPS is established. Logging is established.
7. They add the configuration for their local ORPS (IP address/host name, shared secret, clustering etc.)
8. Repeat 3 – 7 for all their servers.

Virtual Machine

1. Site joins Govroam by filling in a boarding form with RADIUS IP addresses, secure contact email address and request for a docker-based system
2. Jisc sends a key for each IP address to the secure email address
    - Jisc configures the puppet database with the key, IP, host name and shared secrets for each remote system.
3. Site starts up the VM with the first IP address.
4. Site connects to the VM and enters the key for that IP addresses
5. The RADIUS server and Syslog server configuration are downloaded and site informed
6. They add the configuration for their local ORPS (IP address/host name, shared secret, clustering etc.)
7. Repeat 3 – 6 for all their servers.

Development

Whilst the key tools such as docker, puppet and virtual machines exist and are mature there is work required to connect these together and turn them into a service.

Docker containers – at least three are needed for the setup/management, RADIUS and syslog servers. The syslog configuration is trivial, the RADIUS would be based almost entirely on our existing configurations but the setup/management image would need purpose written software for the initial setup and the local site RADIUS configuration management.

Puppetmaster – the basic configuration is straightforward but purpose written software would be needed for the *autosigning* and *External Node Configuration* which interacts with the database of keys. Provision should be made to handle atypical site configurations.

Database – the underlying schema is fairly simple but it requires a management interface to add, remove, edit, revoke etc. the site configurations.