Govroam Site Monitoring

Summary

We have little or no visibility of the connected sites and this impacts our ability to ensure that they're properly configured and offering a good service. Implementing a monitoring system that provides both Jisc staff and site administrators with suitably detailed information would speed up initial site configuration, quickly highlight site issues and greatly improve the quality of the service for all.

Problem

Govroam offers no site monitoring beyond 'keeping an eye' on syslog information from the Govroam NRPS. Whilst it's possible to spot some issues and imply the underlying problem, it is by no means a deterministic or reliable approach.

Even something as simple as 'Is RRPS X configured to accept requests from the NRPS?' or even 'Is RRPS X alive?' would be very manual tasks and require configuration changes to the NRPS.

The recent change of adding a new NRPS was easy to perform at the Jisc end but there's been no particuarly reliable way of determining whether an RRPS or ORPS has actually made that the change at their end, or done the change correctly.

As well as not having any visibility of remote site issues within the Govroam team, the remote site administrators have no visibility of how their systems are interacting with the outside world. They might be making RADIUS or network changes which internally appear successful but break some aspect of the service between them and Jisc.

Solution

Provide a monitoring system which performs regular checks of all the sites in various ways and reports back the status.

What systems to monitor?

In the eduroam world it's possible to directly monitor every ORPS because eduroam only has a single tier of RADIUS servers. This provides a complete picture of the service status. The multi-tiered approach of Govroam means that we can't expect to monitor more than one level down directly. Many RFOs have sites with RFC1918 addresses on their ORPS and/or highly restrictive firewalls and security policies.

However, it should be possible to monitor authentications all the way down to the IdP level

using RADIUS requests. Thus we can monitor connected hosts directly and realms indirectly.

How are these monitored?

- Monitoring hosts generally starts with ICMP echo/response (Pings). A successful ICMP response indicates that there is a network connection and a working kernel capable of responding. It doesn't give any indication of how capable RADIUS software of authenticating or proxying a request. Monitoring beyond ICMP seems unlikely as it would require running agent software on the hosts (difficult if it belongs to someone else, impossible where the hosts are appliances).
- RADIUS requests address the application layer monitoring and can be done in a number of ways:

    1. An authentication request of something@realm, even if 'something' is an non-existant account, should generate a Reject response from the RADIUS infrastructure. This indicates that there's a RADIUS infrastructure that behaves somewhat normally. However, this infrastructure might be no more than a single RADIUS server configured to always Reject no matter what.
    2. An authentication request for valid@realm, where 'valid' is an existing account should generate an Accept. This demonstrates a fully working RADIUS infrastructure capable of successfully handling requests for a realm.
    3. A non-EAP (MSCHAPv2 or PAP) authentication requests between NRPS and RRPS which can confirm basic connectivity. Requires an account creating on each RRPS.
    4. Status-Server requests. By default RADIUS will mark a remote server as down if it fails to receive a response in a set time period. However, within a multi-tiered environment it's entirely possible for a NRPS to mark an RRPS down when it's actually an IdP failing to respond. Status-Server checks the availablity of the next hop thus preventing false conclusions being drawn. It can also be used as a RADIUS 'ping' and to gather performance data from RADIUS servers. It is the 'Ping' approach that we could use to monitor the state of RRPSes.

Complications

Currently, load balancing across RADIUS servers means it's not possible to submit RADIUS requests which follow a specific path so some effort will be needed to find a way to test each RRPS from each NRPS in turn. Essentially this would require running a series of tests from each NRPS for each RRPS.

Some Federations have configured their firewalls not to allow ICMP to their RRPS.

Only some RADIUS software support Status-Server and the RRPS/ORPS directly connect to Jisc mostly use MS NPS, which doesn't.

Federations whose RRPS are run by third parties don't necessarily have IdP functionality at the

RFO level, making it difficult/impossible to test authentication an account in the RFO's own realm. e.g. BT don't provide an IdP for PSBA so psba.gov.wales has no valid accounts.

What action should be taken for failures?

Since there's no option to directly remediate configuration problems with remote sites there are but two approaches:

1. Web-based status display which can be checked by both Jisc and sites alike, akin to the information provided by the eduroam support portal.
2. Notifications by email (or some other method) directly to the registered adminstrators. The 'other methods' might include an RSS feed or social media.

Implementation

Which software?

There are plenty of status monitoring systems available from monit to Solarwinds. They vary in functionality and complexity. Cacti is very powerful but takes a lot of effort to configure, Observium can monitor a wide range of system, easy to configure but only support the checks it supports. Solarwinds is very expensive.

The nature of Govroam means that a tool does not need to support the monitoring of thousands of hosts, hundred of metrics, be distributed or high-scalable. It only has to support two checking methods, ping and RADIUS, but it has to support RADIUS **very** well, with the capability of adding new, arbitary RADIUS tests at will and displaying the results suitably.

Since most of the tests would have be run directly on the four NRPS and software would need to be able to either run tests on remote servers or interface with custom software that runs on remote servers.

It's unlikely that, out of the box, any software would support the sort of display that we'd want to offer the site administrators (which might be authenticated and provide different levels of detail for different roles i.e. public v. administrator) so has to be capable of exporting data to a custom tool.

Some monitoring systems offer a history along with the current status and whilst this is desirable, it's not essential.

Icinga meets many of these criteria – it is free, established, respected, runs on Linux, easy to configure, can distribute checks, has a variety of checks, is extendable, can be resilient and support a large number of end points. It's interface is unlikely to be customisable enough to limit access with the control we might require, or alter the format into a way that's palatable by administrators but it has an API and a number of third party interfaces.

Configuration

Unfortunately all the site configuration information exists in files and is not organised in a sensible way. To be useful the data needs to be structured something like:

RFO → Sites → Realms

→ Administrators

→ RRPS details

→ Administrators

so that a monitoring system could reflect the structure and display the appropriate data accordingly. This information is spread across files on five systems, a wiki and a spreadsheet, all manually updated. Inevitably there will be transcription errors and inconsistencies between them due to human error.

This data would need to be pulled together into a relational database from which the monitoring system could extract the right data in the right format in a sensible way. With sensibly structured data it should be possible to allow remote site administrators to have access to just their results and no one elses.

As a consequence of storing the data in a properly structured way, kept current and accurate it would be possible generate the other 'views', such as the contacts list on the wiki, the technical mailing list and the membership spreadsheet.

Enhancements

Once there is a monitoring infrastructure able to run arbitary checks against servers and realms then we could consider adding further checks:

- Existence of:
  - Operator-Name
  - CUID
  - CSI
- Correct information on websites (Logo, link)
- Visitor access tests
- Security checks
- Anything else which sites MUST or SHOULD be implementing

It's conceivable that such an infrastructure could also act as the basis for client side monitoring e.g. as a recipient of test results supplied by probes installed on the client sites.