InkBridge Networks

We Authenticate The Internet

unlang - The FreeRADIUS Policy Language

Document	Policy
Prepared by	InkBridge Networks
Date	2024-12

unlang Overview

What is unlang?

unlang is the name given to the policy language used by FreeRADIUS.

It is a simple language primarily based around flow control with keywords such as `if` and `switch`.

Anyone familiar with languages such as Perl will find uvery simple to understand.

Where is the unlang documentation?

For the current stable release of FreeRADIUS (3.2.x), the documentation is available online as part of the FreeRADIUS documentation.

The upcoming FreeRADIUS version 4 has taken the opportunity to improve some parts of unlang syntax, e.g.:

- `update` is no longer required for amending attribute values.
- The syntax for expansions has been amended to be more consistent with other languages - rather than `%{func: arg1 arg2}` the syntax is now `%func(arg1, arg2)`.

In addition, FreeRADIUS v4 has added new functionality to unlang including:

- `try catch` blocks for better error handling.
- `timeout` blocks for managing response times from external resources.
- Many new expansions for manipulating data.

Documentation for FreeRADIUS v4 is also available online.

How stable is unlang?

Aside from the syntax changes introduced in FreeRADIUS v4, unlang is very stable.

The only changes done within the stable release are bug fixes.

Once FreeRADIUS v4 is released, the only changes will be adding additional functionality.

How does FreeRADIUS use unlang?

During server start up, FreeRADIUS parses its configuration, including all the unlang in processing sections.

The unlang is compiled during this startup process so that at run time the policies are run efficiently.

When a packet is received by FreeRADIUS, it is passed to one or more processing section depending on the type of packet, causing the unlang within that section to be run.

FreeRADIUS supports Python and Perl - why not use those?

FreeRADIUS does provide the ability to call Python or Perl scripts as part of packet processing using the `rlm_python` or `rlm_perl` modules.

These modules are intended for use where a third-party library needs to be used which is only available in one of those languages.

When calling one of these modules, there is a significant overhead.

The data which FreeRADIUS is using to process the packet has to be re-structured into the format which that language expects, then the script has to be launched.

When the module has finished processing, the data then has to be read back and restructured back into FreeRADIUS's internal format.

In addition, debugging policy written in Python or Perl is much harder. When FreeRADIUS is run in debug mode, its output clearly shows every line of unlang being run and what the result is. No such output is available for policies written in Python or Perl; they have to have debug logging added to the scripts at key points to get any feedback.

Contact

InkBridge Networks

26 rue Colonel Dumont
38000 Grenoble
France

T +33 4 85 88 22 67
F +33 4 56 80 95 75
W https://inkbridgenetworks.com
E sales@inkbridge.io

InkBridge Networks (Canada)

100 Centrepointe Drive, Suite 200
Ottawa, ON, K2G 6B1
Canada

T +1 613 454 5037
F +1 613 280 1542

InkBridge Networks

We authenticate the Internet