openresty / encrypted-session-nginx-module

encrypt and decrypt nginx variable values http://openresty.org

74 commits **2** branches **6** releases **5** contributors

agentzh doc: updated copyright notice. Latest commit 37e8f24 on Dec 18, 2016

src	bugfix: fixed one potential memory leak in an error condition.	3 months ago
t	added (passing) tests for using "encrypted_session_expires" with "if".	4 years ago
util	util/build.sh: fixed build failures with nginx 1.11.2.	7 months ago
.gitattributes	added a .gitattributes file to correct GitHub's language tag.	a year ago
.gitignore	updated .gitignore a bit.	4 years ago
.travis.yml	travis-ci: reduced environment settings in scripts.	9 months ago
README.md	doc: updated copyright notice.	2 months ago
config	feature: this module can now be compiled as a dynamic module with NGI...	11 months ago
valgrind.suppress	removed the hard-coded luajit library path from the rpath of the deve...	2 years ago

README.md

Name

encrypted-session-nginx-module - encrypt and decrypt nginx variable values

This module is not distributed with the Nginx source. See the installation instructions.

Table of Contents

Status

This module is production ready.

Synopsis

```
# key must be of 32 bytes long
encrypted_session_key "abcdefghijklmnopqrstuvwxyz123456";

# iv must not be longer than 16 bytes
#   default: "deadbeefdeadbeef" (w/o quotes)
encrypted_session_iv "1234567812345678";

# default: 1d (1 day)
encrypted_session_expires 2; # in sec

location /encrypt {
    set $raw 'text to encrypted'; # from the ngx_rewrite module
    set_encrypt_session $session $raw;
    set_encode_base32 $session; # from the ngx_set_misc module

    add_header Set-Cookie 'my_login=$session';  # from the ngx_headers module
```

```
    # your content handler goes here...
}

location /decrypt {
    set_decode_base32 $session $cookie_my_login; # from the ngx_set_misc module
    set_decrypt_session $raw $session;

    if ($raw = '') {
        # bad session
    }

    # your content handler goes here...
}
```

Description

This module provides encryption and decryption support for nginx variables based on AES-256 with Mac.

This module is usually used with the ngx_set_misc module and the standard rewrite module's directives.

This module can be used to implement simple user login and ACL.

Usually, you just decrypt data in nginx level, and pass the unencrypted data to your FastCGI/HTTP backend, as in

```
location /blah {
    set_decrypt_session $raw_text $encrypted;

    # this directive is from the ngx_set_misc module
    set_escape_uri $escaped_raw_text $raw_text;
```

```
        fastcgi_param QUERY_STRING "uid=$uid";
        fastcgi_pass unix:/path/to/my/php/or/python/fastcgi.sock;
    }
```

Lua web applications running directly on ngx_lua can call this module's directives directly from within Lua code:

```
    local raw_text = ndk.set_var.set_decrypt_session(encrypted_text)
```

Back to TOC

Directives

Back to TOC

encrypted_session_key

syntax: *encrypted_session_key <key>*

default: *no*

context: *http, server, server if, location, location if*

Sets the key for the cipher (must be 32 bytes long). For example,

```
    encrypted_session_key "abcdefghijklmnopqrstuvwxyz123456";
```

Back to TOC

encrypted_session_iv

syntax: *encrypted_session_iv <iv>*

default: *encrypted_session_iv "deadbeefdeadbeef";*

context: *http, server, server if, location, location if*

Sets the initial vector used for the cipher (must be *no longer* than 16 bytes).

For example,

```
encrypted_session_iv "12345678";
```

Back to TOC

encrypted_session_expires

syntax: *encrypted_session_expires <time>*

default: *encrypted_session_expires 1d;*

context: *http, server, server if, location, location if*

Sets expiration time difference (in seconds by default).

For example, consider the following configuration:

```
encypted_session_expires 1d;
```

When your session is being generated, ngx_encrypted_session will plant an expiration time (1 day in the future in this example) into the encrypted session string, such that when the session is being decrypted later, the server can pull the expiration time out of the session and compare it with the server's current system time. No matter how you transfer and store your session, like using cookies, or URI query arguments, or whatever.

People may confuse this setting with the expiration date of HTTP cookies. This directive simply controls when the session gets expired; it knows nothing about HTTP cookies. Even if the end user intercepted this session from cookie by himself and uses it later manually, the server will still reject it when the expiration time gets passed.

Back to TOC

set_encrypt_session

syntax: *set_encrypt_session $target <value>*

default: *no*

context: *http, server, server if, location, location if*

Encrypts the string value specified by the `value` argument and saves the result into the variable specified by `$target`.

For example,

```
set_encrypt_session $res $value;
```

will encrypts the value in the variable $value into the target variable `$res`.

The `value` argument can also be an nginx string value, for example,

```
set_encrypt_session $res "my value = $value";
```

The resulting data can later be decrypted via the set_decrypt_session directive.

Back to TOC

set_decrypt_session

syntax: *set_decrypt_session $target <value>*

default: *no*

context: *http, server, server if, location, location if*

Similar to set_encrypt_session, but performs the inverse operation, that is, to decrypt things.

Back to TOC

Installation

You're recommended to install this module (as well as the Nginx core and many other goodies) via the ngx_openresty bundle. See the detailed instructions for downloading and installing ngx_openresty into your system. This is the easiest and most safe way to set things up.

Alternatively, you can install this module manually with the Nginx source:

Grab the nginx source code from nginx.org, for example, the version 1.11.2 (see nginx compatibility), and then build the source with this module:

```
wget 'http://nginx.org/download/nginx-1.11.2.tar.gz'
tar -xzvf nginx-1.11.2.tar.gz
cd nginx-1.11.2/

Here we assume you would install you nginx under /opt/nginx/.
./configure --prefix=/opt/nginx \
    --with-http_ssl_module \
    --add-module=/path/to/encrypted-session-nginx-module

make -j2
make install
```

Download the latest version of the release tarball of this module from encrypted-session-nginx-module file list.

Also, this module is included and enabled by default in the ngx_openresty bundle.

OpenSSL should not be disabled in your Nginx build.

Back to TOC

Building as a dynamic module

Starting from NGINX 1.9.11, you can also compile this module as a dynamic module, by using the `--add-dynamic-module=PATH` option instead of `--add-module=PATH` on the `./configure` command line above. And then you can explicitly load the module in your `nginx.conf` via the load_module directive, for example,

```
load_module /path/to/modules/ndk_http_module.so;  # assuming NDK is built as a dynamic module too
load_module /path/to/modules/ngx_http_encrypted_session_module.so;
```

Back to TOC

Compatibility

The following versions of Nginx should work with this module:

- **1.11.x** (last tested: 1.11.2)
- **1.10.x**
- **1.9.x** (last tested: 1.9.15)
- **1.8.x**
- **1.7.x** (last tested: 1.7.10)
- **1.6.x**
- **1.5.x** (last tested: 1.5.12)
- **1.4.x** (last tested: 1.4.4)
- **1.2.x** (last tested: 1.2.9)
- **1.1.x** (last tested: 1.1.5)
- **1.0.x** (last tested: 1.0.11)
- **0.9.x** (last tested: 0.9.4)
- **0.8.x** (last tested: 0.8.54)
- **0.7.x >= 0.7.46** (last tested: 0.7.68)

Earlier versions of Nginx like 0.6.x and 0.5.x will *not* work.

If you find that any particular version of Nginx above 0.7.44 does not work with this module, please consider reporting a bug.

Back to TOC

Report Bugs

Although a lot of effort has been put into testing and code tuning, there must be some serious bugs lurking somewhere in this module. So whenever you are bitten by any quirks, please don't hesitate to

1. send a bug report or even patches to agentzh@gmail.com,
2. or create a ticket on the issue tracking interface provided by GitHub.

Back to TOC

Source Repository

Available on github at openresty/encrypted-session-nginx-module.

Back to TOC

Getting involved

You'll be very welcomed to submit patches to the author or just ask for a commit bit to the source repository on GitHub.

Back to TOC

Author

Yichun "agentzh" Zhang (章亦春) <agentzh@gmail.com>

Back to TOC

Copyright & License

Back to TOC

See Also

- NDK

- ngx_set_misc module

Back to TOC