Acc Module

Jiri Kuthan
iptel.org

Acc Module

by Jiri Kuthan

Revision History

Revision $Revision: 1.2 $ $Date: 2003/07/14 21:19:21 $

Table of Contents

List of Examples

Chapter 1. User’s Guide

1.1. Overview

acc module is used to report on transactions to syslog, SQL and RADIUS.

To report on a transaction using syslog, use “setflag” to mark a transaction you are interested in with a flag, load accounting module and set its “log_flag” to the same flag number. The acc module will then report on completed transaction to syslog. A typical usage of the module takes no acc-specific script command -- the functionality binds invisibly through transaction processing. Script writers just need to mark the transaction for accounting with proper setflag.

What is printed depends on module’s “log_fmt” parameter. It’s a string with characters specifying which parts of request should be printed:

- c = Call-Id
- d = To tag (Dst)
- f = From
- i = Inbound Request-URI
- m = Method
- o = Outbound Request-URI
- r = fRom
- s = Status
- t = To
- u = digest Username
- p = username Part of inbound Request-URI

If a value is not present in request, "n/a" is accounted instead.

Note that:

- A single INVITE may produce multiple accounting reports -- that’s due to SIP’s forking feature
- Subsequent ACKs and other requests do not hit the server and can’t be accounted unless record-routing is enforced. The ACKs assert very little useful information anyway and reporting on INVITE’s 200 makes most accounting scenarios happy.
- There is no session accounting -- ser maintains no sessions. If one needs to correlate INVITEs with BYEs for example for purpose of billing, then it is better done in the entity which collects accounting infromation. Otherwise, SIP server would have to become sessions-stateful, which would very badly impact its scalability.
- If a UA fails in middle of conversation, a proxy will never learn it. In general, a better practice is to account from an end-device (such as PSTN gateway), which best knows about call status (including media status and PSTN status in case of the gateway).

Support for SQL and RADIUS works analogously. You need to enable it by recompiling the module with properly set defines. Uncomment the SQL_ACC and RAD_ACC lines in modules/acc/Makefile. To compile SQL support, you need to have mysqlclient package on your system. To compile RADIUS support, you need to have radiusclient installed on your system (http://www.mcs-cityline.net/~lf/radius/). The radius client needs to be configured properly. To do so, use the template at etc/radiusclient.conf and make sure that module's radius_config parameter points to its location. In particular, accounting secret must match that one configured in server. Uses along with FreeRadius (http://www.freeradius.org/) and Radiator (http://www.open.com.au/radiator/) servers have been reported to us.

Both mysql and radius libraries must be dynamically linkable. You need to configure your OS so that SER, when started, will find them. Typically, you do so by manipulating LD_LIBRARY_PATH environment variable or condiguring ld.so.

1.1.1. General Example

```
loadmodule "modules/acc/acc.so"
modparam("acc", "log_level", 1)
modparam("acc", "log_flag", 1)

if (uri=~"sip:+49") /* calls to Germany */ {
    if (!proxy_authorize("iptel.org" /* realm */,
                         "subscriber" /* table name */))  {
        proxy_challenge("iptel.org" /* realm */, "0" /* no qop */ );
        break;
    }

    if (method=="INVITE" & !check_from()) {
        log("from!=digest\n");
        sl_send_reply("403","Forbidden");
        break;
    }

    setflag(1); /* set for accounting (the same value as in log_flag!)
    t_relay();  /* enter stateful mode now */
};
```

1.2. Dependencies

The module depends on the following modules (in the other words the listed modules must be loaded before this module):

- *tm* -- Transaction Manager
- *a database module* -- If compiled with database support.

1.3. Exported Parameters

1.3.1. `secret` (string)

Default value is randomly generated string.

Example 1-1. Setting secret module parameter

```
modparam("auth", "secret", "johndoessecretphrase")
```

1.3.2. `log_level` (integer)

Log level at which accounting messages are issued to syslog.

Default value is L_NOTICE.

Example 1-2. log_level example

```
modparam("acc", "log_level", 2)   # Set log_level to 2
```

1.3.3. `log_fmt` (string)

Defines what parts of header fields will be printed to syslog, see "overview" for list of accepted values.

Default value is "miocfs".

Example 1-3. log_fmt example

```
modparam("acc", "log_fmt", "mfs")
```

1.3.4. `early_media` (integer)

Should be early media (183) accounted too ?

Default value is 0 (no).

Example 1-4. early_media example

```
modparam("acc", "early_media", 1)
```

1.3.5. `failed_transactions` (integer)

Should be failed transactions (status>=300) accounted too ?

Default value is 0 (no).

Example 1-5. failed_transactions example

```
modparam("acc", "failed_transactions", 1)
```

1.3.6. `log_flag` (integer)

Request flag which needs to be set to account a transaction.

Default value is 1.

Example 1-6. log_flag example

```
modparam("acc", "log_flag", 2)
```

1.3.7. `log_missed_flag` (integer)

Request flag which needs to be set to account missed calls.

Default value is 2.

Example 1-7. log_missed_flag example

```
modparam("acc", "log_missed_flag", 3)
```

1.3.8. `report_ack` (integer)

Shall acc attempt to account e2e ACKs too ? Note that this is really only an attempt, as e2e ACKs may take a different path (unless RR enabled) and mismatch original INVITE (e2e ACKs are a separate transaction).

Default value is 1 (yes).

Example 1-8. report_ack example

```
modparam("acc", "report_ack", 0)
```

1.3.9. `report_cancels` (integer)

By default, CANCEL reporting is disabled -- most accounting applications are happy to see INVITE's cancellation status. Turn on if you explicitely want to account CANCEL transactions.

Default value is 0 (no).

Example 1-9. report_cancels example

```
modparam("acc", "report_cancels", 1)
```

1.3.10. `radius_config` (string)

This parameter is radius specific. Path to radius client configuration file, set the referred config file correctly and specify there address of server, shared secret (should equal that in /usr/local/etc/raddb/clients for freeRadius servers) and dictionary, see etc for an example of config file and dictionary.

Default value is "/usr/local/etc/radiusclient/radiusclient.conf".

Example 1-10. radius_config example

```
modparam("acc", "radius_config", "/etc/radiusclient/radiusclient.conf")
```

1.3.11. `service_type` (integer)

Radius service type used for accounting.

Default value is 15 (SIP).

Example 1-11. service_type example

```
modparam("acc", "service_type", 16)
```

1.3.12. `radius_flag` (integer)

Request flag which needs to be set to account a transaction -- RADIUS specific.

Default value is 1.

Example 1-12. radius_flag example

```
modparam("acc", "radius_flag", 2)
```

1.3.13. `radius_missed_flag` (integer)

Request flag which needs to be set to account missed calls -- RADIUS specific.

Default value is 2.

Example 1-13. radius_missed_flag example

```
modparam("acc", "radius_missed_flag", 3)
```

1.3.14. `db_url` (string)

SQL address -- database specific.

Default value is "mysql://ser:heslo@localhost/ser"

Example 1-14. db_url example

```
modparam("acc", "db_url", "mysql://user:password@localhost/ser")
```

1.3.15. `db_flag` (integer)

Request flag which needs to be set to account a transaction -- database specific.

Default value is 1.

Example 1-15. db_flag example

```
modparam("acc", "db_flag", 2)
```

1.3.16. `db_missed_flag` (integer)

Request flag which needs to be set to account missed calls -- database specific.

Default value is 2.

Example 1-16. db_missed_flag example

```
modparam("acc", "db_missed_flag", 3)
```

1.4. Exported Functions

1.4.1. `acc_log_request(comment)`

`acc_request` reports on a request, for example, it can be used to report on missed calls to off-line users who are replied 404. To avoid multiple reports on UDP request retranmission, you would need to embed the action in stateful processing.

Meaning of the parameters is as follows:

- *comment* - Comment to be appended.

Example 1-17. acc_log_request usage

```
...
acc_log_request("Some comment");
...
```

1.4.2. `acc_db_request(comment, table)`

Like `acc_log_request`, `acc_db_request` reports on a request. The report is sent to database at "db_url", in the table referred to in the second action parameter

Meaning of the parameters is as follows:

- *comment* - Comment to be appended.
- *table* - Database table to be used.

Example 1-18. acc_db_request usage

```
...
acc_log_request("Some comment", "Some table");
...
```

1.4.3. `acc_rad_request(comment)`

Like `acc_log_request`, `acc_rad_requst` reports on a request. It reports to radius server as configured in "radius_config".

Meaning of the parameters is as follows:

- *comment* - Comment to be appended.

Example 1-19. acc_rad_request usage

```
...
acc_rad_request("Some comment");
...
```

Chapter 2. Developer's Guide

To be done.

Chapter 3. Frequently Asked Questions

1. What is the meaning of life ?

42