libnftables

É. Leblond

Stamus Networks

June 28, 2016

Work in progress

Work for me

- Running a libnftables version of nft
- Test program is working too

Diff statistics

```
32 files changed, 1003 insertions(+), 682 deletions(-)
```

STAMVS NETWORKS

Lib usage: basic usage

```
#include <nftables/nftables.h>
const char CMD[] = "list ruleset";
int main(int argc, char * const *argv)
{
        nft_context_t *context = NULL;
        /* Global init to be call once */
        nft_global_init();
        /* Open a thread-safe context */
        context = nft_open();
        /* Run a command */
        nft_run_command(context, CMD, strlen(CMD));
        /* Free ressouce for context */
        nft_close(context);
        /* Global deinit function */
        nft_global_deinit();
        return 0;
}
```

STAMVS NETWORKS

transaction support

```
#include <nftables/nftables.h>
const char ADD1[] = "add rule inet filter input counter";
const char ADD2[] = "add rule inet filter output position 40 counter";
int main(int argc, char * const *argv) {
        nft_context_t *context = NULL;
        nft_global_init();
        context = nft_open();
        /* Start a transaction */
        if (nft_transaction_start(context) != 0) {
                nft_print_error(context); return -1;
        }
        /* Add a command to the transaction */
        if (nft_transaction_add(context, ADD1, strlen(ADD1)) !=0) {
                nft_print_error(context); return -1;
        }
        nft_transaction_add(context, ADD2, strlen(ADD2));
        /* Commit transaction to kernel */
        if (nft_transaction_commit(context) != 0) {
                nft_print_error(context); return -1;
        }
        nft_close(context);
        nft_global_deinit();
        return 0;
}
```

Main structure

```
typedef struct _nft_context {
        struct netlink_ctx *nl_ctx;
        struct mnl_socket *nf_sock;
        struct mnl_socket *mon_sock;
        struct nftnl_batch *batch;

        struct list_head cmds;
        struct list_head msgs;
        int seq;
        unsigned int batch_seqnum;
        bool cache_initialized;
        bool batch_supported;
        const struct input_descriptor *indesc;
} nft_context_t;
```

STAMVS NETWORKS

Current API

```
void nft_global_init(void);
void nft_global_deinit(void);

nft_context_t * nft_open(void);
int nft_close(nft_context_t *ctx);

int nft_run_command(nft_context_t *ctx, const char * buf,
                    size_t buflen);

int nft_transaction_start(nft_context_t *ctx);
int nft_transaction_add(nft_context_t *ctx, const char * buf,
                        size_t buflen);
int nft_transaction_commit(nft_context_t *ctx);

int nft_print_error(nft_context_t *ctx);
```

STAMVS NETWORKS

Short term TODO list

Output

- Redirect output
- Json output
- Error message

API update

- Use identifier
- Documentation

STAMVS NETWORKS

Long term TODO list

Monitor event

- Get event
- Filter them
- Select output format

Set handling

- Typed set handling

Bindings

- Python bindings

STAMVS NETWORKS