nftables switchdev support

Pablo Neira Ayuso
<pablo@netfilter.org>

Netdev 1.1
February 2016
Sevilla, Spain

nftables switchdev support

- Steps:
  - Check if switchdev is available
  - If so, transparently insertion into hardware (offload flag is set)
  - Front-end normalization to intermediate representation (IR)
  - IR: expressions & statements
  - Helper functions to generate hardware representation

Intermediate Representation (IR)

- Similar to the model in userspace nft.
- Normalize front-end input to intermediate representation.

```
        Relational                     Relational
           /   \                        /       \
    payload   value                 binop      value
                                     /  \
                                 meta  value
```

Intermediate Representation (IR)

```
enum nft_ast_expr_type {
        NFT_AST_EXPR_UNSPEC     = 0,
        NFT_AST_EXPR_RELATIONAL,
        NFT_AST_EXPR_VALUE,
        NFT_AST_EXPR_META,
        NFT_AST_EXPR_PAYLOAD,
        NFT_AST_EXPR_BINOP,
};
```

Intermediate Representation (IR)

```
struct nft_ast_expr {
        enum nft_ast_expr_type          type;
        enum nft_ast_expr_ops           op;
        u32                               len;
        union {
                struct {
                        struct nft_data         data;
                } value;
                struct {
                        enum nft_meta_keys      key;
                } meta;
                struct {
                        enum nft_payload_bases  base;
                        u32                     offset;
                } payload;
                struct {
                        struct nft_ast_expr     *left;
                        struct nft_ast_expr     *right;
                } relational;
                struct {
                        struct nft_ast_expr     *left;
                        struct nft_ast_expr     *right;
                } binop;
        };
};
```

Intermediate Representation (IR)

```
enum nft_ast_stmt_type {
    NFT_AST_STMT_EXPR          = 0,
    NFT_AST_STMT_PAYLOAD,
    NFT_AST_STMT_META,
    NFT_AST_STMT_COUNTER,
    NFT_AST_STMT_VERDICT,
};

struct nft_ast_stmt {
    struct list_head               list;


    enum nft_ast_stmt_type             type;
    union {
        struct nft_ast_expr        *expr;
        /* Other statement definitions here */
    };
};
```

Intermediate Representation (IR)

- struct nft_ast_expr *nft_ast_expr_alloc(enum nft_ast_expr_type type)
- void nft_ast_expr_destroy(struct nft_ast_expr *expr)
- struct nft_ast_stmt *nft_ast_stmt_alloc(enum nft_ast_stmt_type type);
- void nft_ast_stmt_list_release(struct list_head *ast_stmt_list)
- int nft_delinearize(struct list_head *ast_stmt_list, struct nft_rule *rule)

Nftables delinearization

```
@@ -333,6 +360,7 @@ static const struct nft_expr_ops nft_meta_get_ops = {
        .eval           = nft_meta_get_eval,
        .init           = nft_meta_get_init,
        .dump           = nft_meta_get_dump,
+       .delinearize    = nft_meta_get_delinearize,
 };

 static const struct nft_expr_ops nft_meta_set_ops = {

@@ -114,6 +189,7 @@ static const struct nft_expr_ops nft_cmp_ops = {
       .eval           = nft_cmp_eval,
       .init           = nft_cmp_init,
       .dump           = nft_cmp_dump,
+      .delinearize    = nft_cmp_delinearize,
 };

 static int nft_cmp_fast_init(const struct nft_ctx *ctx,
```

Backend parser call graph

-
  ```
  struct nft_ast_xfrm_desc {
              const struct nft_ast_proto_desc *proto_desc;
              const struct nft_ast_meta_desc  *meta_desc;

  };
  ```

-
  ```
  struct nft_ast_proto_desc {
         enum nft_payload_bases base;
         u32 protonum;

         int (*xfrm)(const struct nft_ast_expr *dlexpr, struct nft_ast_xfrm_state *state, void *data);
        const struct nft_ast_proto_desc *protocols[ ];

  };
  ```

-
  ```
  struct nft_ast_meta_desc {
         int (*xfrm)(const struct nft_ast_expr *dlexpr, struct nft_ast_xfrm_state *state, void *data);

  };
  ```

Backend parser call graph

-
  ```
  struct nft_ast_xfrm_state {
          const struct nft_ast_xfrm_desc *xfrm_desc;
          const struct nft_ast_proto_desc
                          *pctx[NFT_PAYLOAD_TRANSPORT_HEADER + 1];
          void *data;
  };
  int nft_ast_xfrm(const struct list_head *ast_stmt_list,
                   const struct nft_ast_xfrm_desc *xfrm_desc, void *data)
  ```

-
  ```
  int nft_ast_xfrm_update_pctx(u32 base, u32 proto,
                               struct nft_ast_xfrm_state *state)
  ```

Backend parser call graph

- static const struct nft_ast_proto_desc rocker_eth_proto_desc = {

  ```
      .base         = NFT_PAYLOAD_LL_HEADER,
      .xfrm         = rocket_eth_proto_xfrm,
      .protocols    = {
          &rocker_proto_ipv4,
          &rocker_proto_ipv6,
          NULL
      },
  };
  ```

- static const struct nft_ast_proto_desc rocker_proto_ipv4 = {

  ```
      .base         = NFT_PAYLOAD_NETWORK_HEADER,
      .protonum     = htons(ETH_P_IP),
      .xfrm         = rocker_ipv4_proto_xfrm,
      .protocols    = {
          &rocker_proto_tcp,
          &rocker_proto_udp,
          NULL
      },
  };
  ```

nftables switchdev integration

```
--- a/include/net/netfilter/nf_tables.h
+++ b/include/net/netfilter/nf_tables.h
@@ -788,6 +788,7 @@ struct nft_stats {

 #define NFT_HOOK_OPS_MAX               2
 #define NFT_BASECHAIN_DISABLED         (1 << 0)
+#define NFT_BASECHAIN_SWITCHDEV                (1 << 1)
```

- @@ -48,6 +48,7 @@ enum switchdev_obj_id {

  ```
          SWITCHDEV_OBJ_PORT_VLAN,
          SWITCHDEV_OBJ_IPV4_FIB,
          SWITCHDEV_OBJ_PORT_FDB,
  +       SWITCHDEV_OBJ_NFT,
   };
  ```

nftables switchdev integration

```
@@ -73,6 +74,10 @@ struct switchdev_obj {
                         const unsigned char *addr;
                         u16 vid;
                 } fdb;
+                 struct switchdev_obj_nft {
+                          struct list_head *stmt_list;
+                          u64 handle;
+                 } nft;
        } u;
 };
```

nftables switchdev integration

- From nf_tables_api.c commit path:
  - Check if switchdev is available
  - Call nf_tables_commit_switchdev() before software commit.
  - Normalize nftables software representation into AST.
  - Pass AST as switchdev object
  - Walk AST and generate hardware internal representation.

nftables switchdev support

Pablo Neira Ayuso
<pablo@netfilter.org>

Netdev 1.1
February 2016
Sevilla, Spain