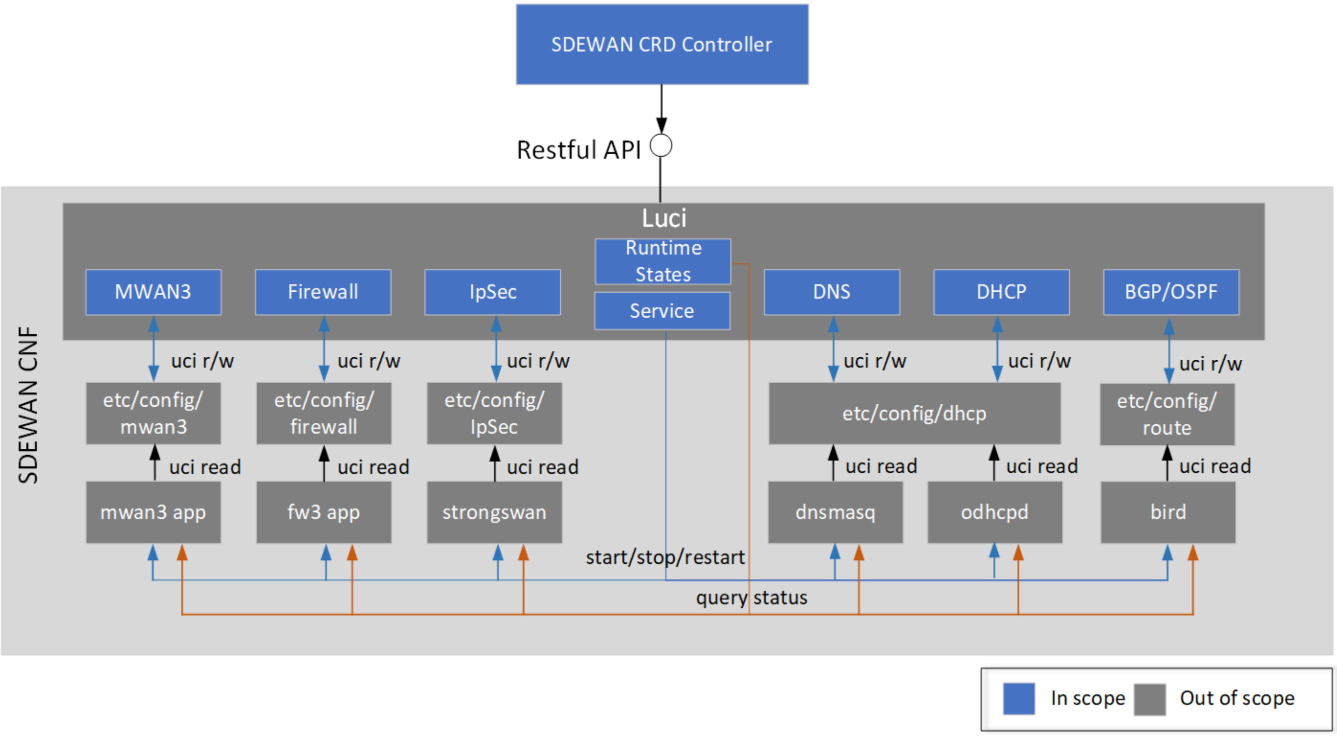
SDEWAN CNF

SDEWAN is implemented as CNF based on OpenWRT and it will support below functionalities:

- Export Restful API interface to support configuration of MWAN3, Firewall & NAT, IpSec.
- Site-to-Site tunnels across edges & edges & central orchestrators and application managers

Architecture

SDEWAN CNF enhances OpenWRT Luci web interface with SDEWAN controllers to provide Restful API for network functions' configuration and control.



CNF includes below modules:

- MWAN3: mwan3 configuration for multiple WAN links' management
- Firewall: fw3 configuration for firewall rule, NAT rule.
- IpSec: strongswan configuration to setup security tunnel between CNFs
- DNS/DHCP: dnsmasq configuration for DNS and DHCP (ip4) or odhcpd configuration for DHCP (ip6)
- BGP/OSPF: bird configuration for BGP/OSPF auto routing
- Service: manage (e.g. start, stop, restart etc.) lifecycle of network function applications (e.g. mwan3, fw3, strongswan etc.)
- Runtime States: exports system log for debugging

APIs

Common Error code:

Code	Description
400	Bad request

401	unauthorized -the security token is not provides or expired.
404	resource not found

Error Response:

Name	In	Type	Description
message	body	string	error message

SDEWAN Service

SDEWAN service restful API provides the capability to list available SDEWAN services, get service status and execute service operation.

PUT /cgi-bin/luci/sdewan/v1/services/{service-name}/

Execute a operation for a service

Request:

- Request Parameters

Name	In	Type	Description
service-name	path	string	service name, valid value are "mwan3", "firewall", "ipsec"
action	body	string	action to be executed. valid value are "start", "stop", "restart", "reload"

- Request Example

```
{
   "action": "start"
}
```

Response

- Normal response code: 200
- Error response code: 400 (e.g. invalid action)
- Response Parameters

Name	In	Type	Description
result	body	string	operation execution result

- Response Example

```
{
   "result": "success"
}
```

GET /cgi-bin/luci/sdewan/v1/services

Lists all available sdewan services supported by SDEWAN CNF

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
services	body	array	a list of supported service

- Response Example

```
{
   "services": ["mwan3", "firewall", "ipsec"]
}
```

SDEWAN Interface

SDEWAN interface API provide network interfaces information and control to up/down a network interface

PUT /cgi-bin/luci/sdewan/v1/interfaces/{interface}/

Execute a operation for a service

Request:

- Request Parameters

Name	In	Type	Description
interface	path	string	interface name, e.g. "eth0"
action	body	string	action to be executed. valid value are "up", "down"

- Request Example

```
{
    "action": "up"
}
```

Response

- Normal response code: 200
- Error response code: 400 (e.g. invalid action), 404 (e.g. interface not found)
- Response Parameters

Name	In	Type	Description
result	body	string	operation execution result

- Response Example

```
{
    "result": "success"
}
```

GET /cgi-bin/luci/sdewan/v1/interfaces

Lists all available network interfaces of the SDEWAN CNF

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
interfaces	body	array	a list of available network interfaces
ip_address	body	array	ip address of the interface
ip6_address	body	array	ipv6 address of the interface
mac_address	body	string	mac address of the interface
status	body	string	interface status, valid value are "UP", "DOWN"
received_packets	body	string	number of received packets
send_packets	body	string	number of send packaets

- Response Example

```
{"interfaces": [

   { "ip_address: ["10.0.0.1"],

      "name":"eth0",

      "status":"UP",

      "send_packets":"19148",

     "mac_address":"22:22:22:22:22:22",

    "receive_packets":"20923" }

]}
```

GET /cgi-bin/luci/sdewan/v1/interfaces/{interface-name}

get information of a network interfaces of the SDEWAN CNF

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
interface-name	path	string	interface name
ip_address	body	array	ip address of the interface
ip6_address	body	array	ipv6 address of the interface
mac_address	body	string	mac address of the interface
status	body	string	interface status, valid value are "UP", "DOWN"
received_packets	body	string	number of received packets
send_packets	body	string	number of send packaets

- Response Example

```
   { "ip_address: ["10.0.0.1"],

      "name":"eth0",

      "status":"UP",

      "send_packets":"19148",

     "mac_address":"22:22:22:22:22:22",

    "receive_packets":"20923" }
```

MWAN3

OpenWRT MWAN3 configuration includes below sections:

- Global: common configuration special used to configure routable loopback address (for OpenWRT 18.06)
- Interface: define how each WAN interface is tested for up/down status
- Member: represents an interface with a metric and a weight value
- Policy: defines how traffic is routed through the different WAN interface(s)
- Rule: describes what traffic to match and what policy to assign for that traffic.

SDEWAN CNF will be created with Global and Interface sections initialized based on CNF allocated interfaces.

SD-EWAN MWAN3 CNF API provides support to get/create/update/delete MWAN3 Rule, Policy (with Member).

MWAN3 Policy

POST /cgi-bin/luci/sdewan/mwan3/v1/policies

create a new policy

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/mwan3/v1/policies/{policy-name}

update a policy

Request:

- Request Parameters:

Name	In	Type	Description
policy-name	path	string	policy name
members	body	array	policy members
interface	body	string	member interface name
metric	body	int	(optional) default: 1, members within one policy with a lower metric have precedence over higher metric members
weight	body	int	(optional) default: 1, members with same metric will distribute load based on this weight value

- Request Example

  PUT /cgi-bin/luci/sdewan/mwan3/v1/policies/balanced

```
{
    "members": [
        {
            "interface": "net1",
            "metric" 1,
            "weight": 2
        }
        {
            "interface": "net2",
            "metric" 1,
             "weight": 1
        }
    ]
}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/mwan3/v1/policies

Lists all defined policies

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
policies	body	array	a list of defined policies

- Response Example

```
{
   "policies": [

      {

         "name":"balanced",

         "members": [

            {

                "interface": "net1",

                "metric" 1,

                "weight": 2

            }

            {

                "interface": "net2",

                "metric" 1,

                "weight": 1

            }

          ]

      }

   ]
}
```

GET /cgi-bin/luci/sdewan/mwan3/v1/policies/{policy-name}

Get a policy

Request: N/A

- Request Parameters

Name	In	Type	Description
policy-name	path	string	policy name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	policy name
members	body	array	policy members
interface	body	string	member interface name
metric	body	int	(optional) default: 1, members within one policy with a lower metric have precedence over higher metric members
weight	body	int	(optional) default: 1, members with same metric will distribute load based on this weight value

- Response Example

```
{
     "name": "balanced",
      "members": [
         {
            "interface": "net1",
            "metric" 1,
            "weight": 2
          }
          {
             "interface": "net2",
             "metric" 1,
              "weight": 1
          }
      ]
}
```

DELETE /cgi-bin/luci/sdewan/mwan3/v1/policies/{policy-name}

delete a policy

Request:

- Request Parameters

Name	In	Type	Description
policy-name	path	string	policy name

Response

- Normal response codes: 200
- Error response codes: 401, 404

MWAN3 Rule

POST /cgi-bin/luci/sdewan/mwan3/v1/rules

create a new rule

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/mwan3/v1/rules/{rule-name}

update a policy

Request:

- Request Parameters

Name	In	Type	Description

rule-name	path	string	rule name
policy	body	string	policy used for the rule
src_ip	body	string	(optional) source ip address
src_port	body	string	(optional) source port or port range
dest_ip	body	string	(optional) destination ip address
dest_port	body	string	(optional) destination port or port range
proto	body	string	(optional) protocol for the rule. Valid values: "tcp", "udp", "icmp", "all"
family	body	string	(optional) address family. Valid values: "ipv4", "ipv6", "all"
sticky	body	string	(optional) default: 0, allow traffic from the same source ip address within the timeout limit to use same wan interface as prior session
timeout	body	int	(optional) default: 600, Stickiness timeout value in seconds

- Request Example
  PUT /cgi-bin/luci/sdewan/mwan3/v1/rules/default_rule

```
{

   "dest_ip": "0.0.0.0/0"

    "policy": "balanced"

}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/mwan3/v1/rules

Lists all defined rules

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
rules	body	array	a list of defined rules

- Response Example

```
{
   "rules": [

      {

         "name":"default_rule",

         "dest_ip": "0.0.0.0/0"

         "policy": "balanced"

      }

    ]

}
```

GET /cgi-bin/luci/sdewan/mwan3/v1/rules/{rule-name}

Get a rule

Request: N/A

- Request Parameters

Name	In	Type	Description
rule-name	path	string	rule name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	rule name
policy	body	string	policy used for the rule
src_ip	body	string	(optional) source ip address
src_port	body	string	(optional) source port or port range
dest_ip	body	string	(optional) destination ip address
dest_port	body	string	(optional) destination port or port range
proto	body	string	(optional) protocol for the rule. Valid values: "tcp", "udp", "icmp", "all"
family	body	string	(optional) address family. Valid values: "ipv4", "ipv6", "all"
sticky	body	string	(optional) default: 0, allow traffic from the same source ip address within the timeout limit to use same wan interface as prior session
timeout	body	int	(optional) default: 600, Stickiness timeout value in seconds

- Response Example

```
#ipv4 example

{

  "name":"default_rule",

  "dest_ip": "0.0.0.0/0"

   "policy": "balanced"

}

#ipv6 example

{

  "name":"default_ipv6_rule",

  "dest_ip": "fdca:f00:ba3::/64"

   "policy": "balanced"

}
```

DELETE /cgi-bin/luci/sdewan/mwan3/v1/rules/{rule-name}

delete a rule

Request:

- Request Parameters

Name	In	Type	Description
rule-name	path	string	rule name

Response

- Normal response codes: 200
- Error response codes: 401, 404

Firewall

OpenWRT Firewall configuration includes below sections:

- Default: declares global firewall settings which do not belong to specific zones
- Include: used to enable customized firewall scripts
- Zone: groups one or more *interfaces* and serves as a *source* or *destination* for *forwardings*, *rules* and *redirects*.
- Forwarding: control the traffic between zones
- Redirect: defines port forwarding (NAT) rules
- Rule: defines basic accept, drop, or reject rules to allow or restrict access to specific ports or hosts.

SDEWAN CNF will be created with Default sections initialized. Include section will not be implemented in this release.

SD-EWAN Firewall API provides support to get/create/update/delete Firewall Zone, Redirect, Rule and Forwardings

Zone

POST /cgi-bin/luci/sdewan/firewall/v1/zones

create a new zone

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/firewall/v1/zones/{zone-name}

update a zone

Request:

- Request Parameters:

Name	In	Type	Description
zone-name	path	string	zone name
network	body	array	List of *interfaces* attached to this zone
masq	body	boolean	Specifies whether *outgoing* zone traffic should be masqueraded. "0" or "1"
masq_src	body	string	Limit masquerading to the given source subnets.
masq_dest	body	string	Limit masquerading to the given destination subnets
masq_allow_invalid	body	boolean	whether add `DROP INVALID` rules
mtu_fix	body	boolean	Enable MSS clamping for *outgoing* zone traffic
input	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *incoming* zone traffic.
forward	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *forwarded* zone traffic.
output	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *output* zone traffic.
family	body	string	The protocol family (`ipv4`, `ipv6` or `any`) these iptables rules are for.
subnet	body	string	List of IP subnets attached to this zone
extra_src	body	string	Extra arguments passed directly to iptables for source classification rules.
etra_dest	body	string	Extra arguments passed directly to iptables for destination classification rules.

- Request Example
  PUT /cgi-bin/luci/sdewan/mwan3/v1/zones/wan

```
{

      "network":"wan"

      "input": "REJECT"

      "output": "ACCEPT"

     "forward": "REJECT"

      "masq": "1"

      "mtu_fix": "1"

}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/firewall/v1/zones

Lists all defined zones

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
zones	body	array	a list of defined zones

- Response Example

```
{
   "zones": [

      {

         "name":"wan"

         "network":"wan"

         "input": "REJECT"

         "output": "ACCEPT"

         "forward": "REJECT"

         "masq": "1"

         "mtu_fix": "1"

       }

   ]

}
```

GET /cgi-bin/luci/sdewan/firewall/v1/zones/{zone-name}

Get a zone

Request: N/A

- Request Parameters

Name	In	Type	Description
zone-name	path	string	zone name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	(Required) zone name
network	body	array	List of *interfaces* attached to this zone
masq	body	boolean	Specifies whether *outgoing* zone traffic should be masqueraded. "0" or "1"
masq_src	body	string	Limit masquerading to the given source subnets.
masq_dest	body	string	Limit masquerading to the given destination subnets
masq_allow_invalid	body	boolean	whether add `DROP INVALID` rules
mtu_fix	body	boolean	Enable MSS clamping for *outgoing* zone traffic
input	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *incoming* zone traffic.
forward	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *forwarded* zone traffic.
output	body	string	Default policy (`ACCEPT`, `REJECT`, `DROP`) for *output* zone traffic.
family	body	string	The protocol family (`ipv4`, `ipv6` or `any`) these iptables rules are for.
subnet	body	string	List of IP subnets attached to this zone
extra_src	body	string	Extra arguments passed directly to iptables for source classification rules.
etra_dest	body	string	Extra arguments passed directly to iptables for destination classification rules.

- Response Example

```
{
    "name":"wan",
     "network":"wan"
     "input": "REJECT"
     "output": "ACCEPT"
    "forward": "REJECT"
     "masq": "1"
     "mtu_fix": "1"
}
```

DELETE /cgi-bin/luci/sdewan/firewall/v1/zones/{zone-name}

delete a zone

Request:

- Request Parameters

Name	In	Type	Description
zone-name	path	string	zone name

Response

- Normal response codes: 200
- Error response codes: 401, 404

Redirect

POST /cgi-bin/luci/sdewan/firewall/v1/redirects

create a new redirect

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/firewall/v1/redirects/{redirect-name}

update a redirect

Request:

- Request Parameters:

Name	In	Type	Description
redirect-name	path	string	redirect name
src	body	string	(Required for DNAT) traffic source zone
src_ip	body	string	Match incoming traffic from the specified *source ip address*.
src_dip	body	string	(Required for SNAT) For *DNAT*, match incoming traffic directed at the given *destination ip address*. For *SNAT* rewrite the *source address* to the given address.
src_mac	body	string	Match incoming traffic from the specified *mac address*.
src_port	body	port or range	Match incoming traffic originating from the given *source port or port range* on the client host.
src_dport	body	port or range	For *DNAT*, match incoming traffic directed at the given *destination port or port range* on this host. For *SNAT* rewrite the *source ports* to the given value.
proto	body	string	Match incoming traffic using the given *protocol*. Can be one of `tcp`, `udp`, `tcpudp`, `udplite`, `icmp`, `esp`, `ah`, `sctp`, or `all`
dest	body	string	Specifies the traffic *destination zone*. Must refer to one of the defined *zone names*.
dest_ip	body	string	For *DNAT*, redirect matches incoming traffic to the specified internal host. For *SNAT*, it matches traffic directed at the given address.
dest_port	body	port or range	For *DNAT*, redirect matched incoming traffic to the given port on the internal host. For *SNAT*, match traffic directed at the given ports.
mark	body	string	match traffic against the given firewall mark
target	body	string	(Required) NAT target: SNAT, DNAT
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for

- Request Example
  PUT /cgi-bin/luci/sdewan/mwan3/v1/redirects/dnat_lan

```
{

   "src":"wan",
   "src_dport":"19900",
   "dest":"lan",
   "dest_ip":"192.168.1.1",
   "dest_port":"22",
   "proto":"tcp",
   "target":"DNAT"

}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/firewall/v1/redirects

Lists all defined redirects

Request: N/A

Response

- Normal response codes: 200

- Response Parameters

Name	In	Type	Description
redirects	body	array	a list of defined redirects

- Response Example

```
{
  "redirects": [

    {

      "name":"dnat_lan",
      "src":"wan",
      "src_dport":"19900",
      "dest":"lan",
      "dest_ip":"192.168.1.1",
      "dest_port":"22",
      "proto":"tcp",
      "target":"DNAT"

   }

  ]

}
```

GET /cgi-bin/luci/sdewan/firewall/v1/redirects/{redirect-name}

Get a redirect

Request: N/A

- Request Parameters

Name	In	Type	Description
redirect-name	path	string	redirect name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	(Required) forwarding name
src	body	string	(Required for DNAT) traffic source zone
src_ip	body	string	Match incoming traffic from the specified *source ip address*.
src_dip	body	string	(Required for SNAT) For *DNAT*, match incoming traffic directed at the given *destination ip address*. For *SNAT* rewrite the *source address* to the given address.
src_mac	body	string	Match incoming traffic from the specified *mac address*.
src_port	body	port or range	Match incoming traffic originating from the given *source port or port range* on the client host.
src_dport	body	port or range	For *DNAT*, match incoming traffic directed at the given *destination port or port range* on this host. For *SNAT* rewrite the *source ports* to the given value.
proto	body	string	Match incoming traffic using the given *protocol*. Can be one of `tcp`, `udp`, `tcpudp`, `udplite`, `icmp`, `esp`, `ah`, `sctp`, or `all`
dest	body	string	Specifies the traffic *destination zone*. Must refer to one of the defined *zone names*.
dest_ip	body	string	For *DNAT*, redirect matches incoming traffic to the specified internal host. For *SNAT*, it matches traffic directed at the given address.
dest_port	body	port or range	For *DNAT*, redirect matched incoming traffic to the given port on the internal host. For *SNAT*, match traffic directed at the given ports.
mark	body	string	match traffic against the given firewall mark
target	body	string	(Required) NAT target: SNAT, DNAT
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for

- Response Example

```
{

    "name":"dnat_lan",
    "src":"wan",
    "src_dport":"19900",
    "dest":"lan",
    "dest_ip":"192.168.1.1",
    "dest_port":"22",
    "proto":"tcp",
    "target":"DNAT"

}
```

DELETE /cgi-bin/luci/sdewan/firewall/v1/redirects/{redirect-name}

delete a redirect rule

Request:

- Request Parameters

Name	In	Type	Description
redirect-name	path	string	redirect name

Response

- Normal response codes: 200
- Error response codes: 401, 404

Rule

POST /cgi-bin/luci/sdewan/firewall/v1/rules

create a new rule

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/firewall/v1/rules/{rule-name}

update a rule

Request:

- Request Parameters:

Name	In	Type	Description
rule-name	path	string	rule name
src	body	string	(Required) traffic source zone
src_ip	body	string	Match incoming traffic from the specified *source ip address*
src_mac	body	string	Match incoming traffic from the specified *mac address*
src_port	body	port or range	Match incoming traffic from the specified *source port* or *port range*
proto	body	string	Match incoming traffic using the given *protocol*. Can be one of `tcp`, `udp`, `tcpudp`, `udplite`, `icmp`, `esp`, `ah`, `sctp`, or `all`
icmp_type	body	string	For *protocol* `icmp` select specific icmp types to match.
dest	body	string	traffic *destination zone*. Must refer to one of the defined *zone names*, or * for any zone
dest_ip	body	string	Match incoming traffic directed to the specified *destination ip address*
dest_port	body	port or range	Match incoming traffic directed at the given *destination port or port range*

mark	body	string	If specified, match traffic against the given firewall mark
target	body	string	(Required) Firewall action (`ACCEPT`, `REJECT`, `DROP`, `MARK`, `NOTRACK`) for matched traffic
set_mark	body	string	Zeroes out the bits given by mask and ORs value into the packet mark.
set_xmark	body	string	Zeroes out the bits given by mask and XORs value into the packet mark
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for
extra	body	string	Extra arguments to pass to iptables. Useful mainly to specify additional match options, such as `-m policy --dir in` for IPsec.

- Request Example
  PUT /cgi-bin/luci/sdewan/mwan3/v1/rules/reject_lan_80

```
{
     "src":"lan"
     "src_ip": "192.168.1.2",
     "src_port": "80",
     "proto":"tcp",
     "target":"REJECT"
}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/firewall/v1/rules

Lists all defined rules

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
rules	body	array	a list of defined rules

- Response Example

```
{
   "rules": [
     {
        "name":"reject_lan_80"
        "src":"lan"
        "src_ip": "192.168.1.2",
        "src_port": "80",
        "proto":"tcp",
        "target":"REJECT"
    }
  ]
}
```

GET /cgi-bin/luci/sdewan/firewall/v1/rules/{rule-name}

Get a rule

Request: N/A

- Request Parameters

Name	In	Type	Description
rule-name	path	string	rule name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	(Required) rule name
src	body	string	(Required) traffic source zone
src_ip	body	string	Match incoming traffic from the specified *source ip address*
src_mac	body	string	Match incoming traffic from the specified *mac address*
src_port	body	port or range	Match incoming traffic from the specified *source port* or *port range*
proto	body	string	Match incoming traffic using the given *protocol*. Can be one of `tcp`, `udp`, `tcpudp`, `udplite`, `icmp`, `esp`, `ah`, `sctp`, or `all`
icmp_type	body	string	For *protocol* `icmp` select specific icmp types to match.
dest	body	string	traffic *destination zone*. Must refer to one of the defined *zone names*, or * for any zone
dest_ip	body	string	Match incoming traffic directed to the specified *destination ip address*
dest_port	body	port or range	Match incoming traffic directed at the given *destination port or port range*
mark	body	string	If specified, match traffic against the given firewall mark
target	body	string	(Required) Firewall action (`ACCEPT`, `REJECT`, `DROP`, `MARK`, `NOTRACK`) for matched traffic
set_mark	body	string	Zeroes out the bits given by mask and ORs value into the packet mark.
set_xmark	body	string	Zeroes out the bits given by mask and XORs value into the packet mark
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for
extra	body	string	Extra arguments to pass to iptables. Useful mainly to specify additional match options, such as `-m policy --dir in` for IPsec.

- Response Example

```
{
    "name":"reject_lan_80"
    "src":"lan"
    "src_ip": "192.168.1.2",
    "src_port": "80",
    "proto":"tcp",
    "target":"REJECT"
}
```

DELETE /cgi-bin/luci/sdewan/firewall/v1/rules/{rule-name}

delete a firewall rule

Request:

- Request Parameters

Name	In	Type	Description
rule-name	path	string	rule name

Response

- Normal response codes: 200
- Error response codes: 401, 404

Forwarding

POST /cgi-bin/luci/sdewan/firewall/v1/forwardings

create a new forwarding

Request:

- Request Parameters: same with PUT's request
- Request Example: same with PUT's example

Response

- Normal response codes: 201
- Error response codes: 400, 401

PUT /cgi-bin/luci/sdewan/firewall/v1/forwardings/{forwarding-name}

update a forwarding

Request:

- Request Parameters:

Name	In	Type	Description
forwarding-name	path	string	forwarding name
src	body	string	(Required) traffic source zone
dest	body	string	(Required) traffic destination zone
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for.

- Request Example
  PUT /cgi-bin/luci/sdewan/mwan3/v1/forwardings/lan_wan

```
{
    "src":"lan"
    "dest": "wan"
}
```

Response

- Normal response codes: 204
- Error response codes: 400, 401, 404

GET /cgi-bin/luci/sdewan/firewall/v1/forwardings

Lists all defined forwardings

Request: N/A

Response

- Normal response codes: 200
- Response Parameters

Name	In	Type	Description
forwardings	body	array	a list of defined forwardings

- Response Example

```
{
   "forwardings": [

      {

        "name":"lan_wan",

        "src":"lan"

        "dest": "wan"

     }

   ]

}
```

GET /cgi-bin/luci/sdewan/firewall/v1/forwardings/{forwarding-name}

Get a forwarding

Request: N/A

- Request Parameters

Name	In	Type	Description
forwarding-name	path	string	forwarding name

Response

- Normal response codes: 200
- Error response code: 404
- Response Parameters

Name	In	Type	Description
name	body	string	(Required) forwarding name
src	body	string	(Required) traffic source zone
dest	body	string	(Required) traffic destination zone
family	body	string	Protocol family (`ipv4`, `ipv6` or `any`) to generate iptables rules for.

- Response Example

```
{

     "name":"lan_wan",

     "src":"lan"

     "dest": "wan"

}
```

DELETE /cgi-bin/luci/sdewan/firewall/v1/forwardings/{forwarding-name}

delete a forwarding rule

Request:

- Request Parameters

Name	In	Type	Description
forwarding-name	path	string	forwarding name

Response

- Normal response codes: 200
- Error response codes: 401, 404