



Flowmon

Networks



Flowmon DDoS Defender 5.02.00

RESTful API Reference Guide

April 21, 2020

Contents

1	Introduction	6
2	Authorization	7
2.1	Obtaining authentication token	7
2.2	Getting Rest API information	8
2.2.1	Getting list of available methods	8
2.2.2	Getting current version of Flowmon	8
3	Attack	10
3.1	Data Types	10
3.2	Endpoints	13
3.2.1	Methods	13
3.2.2	Flags	14
3.2.3	Attacks	15
3.2.4	Attack info	19
3.2.5	Attacks search	23
3.2.6	Set Whitelists to attack	27
3.2.7	Unset Whitelists from attack	29
4	Alert	31
4.1	Data Types	31
4.2	Endpoints	31
4.2.1	Templates	31
4.2.2	Alerts	32
4.2.2.1	Get all Alerts	32
4.2.2.2	Get one specific Alert	34
4.2.2.3	Add new Alert	35
4.2.2.4	Update existing Alert	36
4.2.2.5	Delete specific Alert	38
5	Router	39
5.1	Data Types	39
5.2	Endpoints	40
5.2.1	Routers	40
5.2.1.1	Get all Routers	40
5.2.1.2	Get one specific Router	41
5.2.1.3	Add new Router	42
5.2.1.4	Update existing Router	44

5.2.1.5	Delete specific Router	46
6	Scrubbing Center	47
6.1	Data Types	47
6.2	Endpoints	47
6.2.1	Scrubbing Centers	47
6.2.1.1	Get all Scrubbing Centers	47
6.2.1.2	Get one specific Scrubbing Center	48
6.2.1.3	Add new Scrubbing Center	49
6.2.1.4	Update existing Scrubbing Center	50
6.2.1.5	Delete specific Scrubbing Center	50
7	Scrubbing Center Parameter	52
7.1	Data Types	52
7.2	Endpoints	52
7.2.1	Scrubbing Center Parameters	52
7.2.1.1	Get all defaults Scrubbing Center Parameters	52
7.2.1.2	Update existing default Scrubbing Center Parameters	53
8	Report Chapters	56
8.1	Data Types	56
8.2	Endpoints	57
8.2.1	Code List	57
8.2.2	Chapters	58
8.2.2.1	Get all Chapters	58
8.2.2.2	Get one specific Chapter	60
8.2.2.3	Add new Chapter	61
8.2.2.4	Update existing Chapter	62
8.2.2.5	Delete specific Chapter	64
9	Email Templates	65
9.1	Data Types	65
9.2	Endpoints	65
9.2.1	Languages	65
9.2.2	EmailTemplates	66
9.2.2.1	Get all Email Templates	66
9.2.2.2	Get one specific Email Template	67
9.2.2.3	Add new Email Template	68
9.2.2.4	Update existing Email Template	69
9.2.2.5	Delete specific Email Template	70

10 Rule	72
10.1 Data Types	72
10.2 Endpoints	73
10.2.1 DetectionMethod	73
10.2.2 TrafficType	74
10.2.3 Rules	77
10.2.3.1 Get all Rules	77
10.2.3.2 Get one specific Rule	79
10.2.3.3 Add new Rule	82
10.2.3.4 Update existing Rule	84
10.2.3.5 Delete specific Rule	86
11 Groups	87
11.1 Data types	87
11.2 Endpoints	87
11.2.1 Groups	87
11.2.1.1 Get all groups	87
11.2.1.2 Get one specific Group	88
11.2.1.3 Create new group	88
11.2.1.4 Update existing Group	89
11.2.1.5 Delete specific group	90
12 Segment	91
12.1 Data Types	91
12.2 Endpoints	93
12.2.1 Segments	93
12.2.1.1 Get all Segments	93
12.2.1.2 Get one specific Segment	96
12.2.1.3 Add new Segment	98
12.2.1.4 Update existing Segment	100
12.2.1.5 Delete specific Segment	103
13 Whitelists	104
13.1 Data types	104
13.2 Endpoints	105
13.2.1 Whitelists	105
13.2.1.1 Get all whitelists	105
13.2.1.2 Get one specific Whitelist	107
13.2.1.3 Create new whitelist	108
13.2.1.4 Update existing Whitelist	110

13.2.1.5	Delete specific whitelist	113
13.2.1.6	Get available TrafficTypes	113
13.2.1.7	Get available Week days	115
13.2.1.8	Get available Month days	116
14	REST API changes	118

1 Introduction

This document describes usage of REST API for Flowmon DDoS Defender plugin which is available in version 4.02.00 and newer.

The next chapter describes authorization process for REST API access. Following chapters contains description and usage of data types and methods.

In this documentation is used CodeList, which is more specific for certain methods. Its basic form is as follows:

CodeList

```
CodeList = {  
  id: Integer|String("Item id"),  
  name: String("Item name")  
}
```

The items with null values are also shown in this document (during element creating and editing). These values do not have to be sent, they are displayed only as a reference.

2 Authorization

The API for interception management uses the Oauth 2.0 authorization. An application, which want to use API methods, has to request for granting access token using following HTTPS request (change only the values of the username and password properties).

2.1 Obtaining authentication token

New access token request

```
POST /resources/oauth/token
FORM-DATA grant_type=password
           client_id=invea-tech
           username=<user_login>
           password=<user_password>
```

The server will verify username and password and returns following response:

```
{
  "access_token": "AnlSCIWYbchsCc5sdc5ac4caca8a2",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "DS6SA512ADCVa51adc54VDS51VD5"
}
```

Now the application has to add this access token to every call of API method. The token has to be inserted in the **Authorization** header field of HTTPS requests and value of this field has to be "**<token_type> <access_token>**" from **/resources/oauth/token** response. So for the example the **Authentication** header value will look like "**bearer AnlSCIWYbchsCc5sdc5ac4caca8a2**". The access token is valid for **<expires_in>** seconds. Then the application can request new access token using previous HTTPS request or it can use **<refresh_token>** field to acquire new token without supplying of username and password.

Example of refresh token request:

```
POST /resources/oauth/token
FORM-DATA grant_type=refresh_token
           refresh_token=DS6SA512ADCVa51adc54VDS51VD5
           client_id=invea-tech
```

2.2 Getting Rest API information

There is new possibility of getting REST API available methods since Invea OS version 7.2.0.

2.2.1 Getting list of available methods

- Returns list of available methods

Example Usage

```
GET /rest/methods

RETURNS:
[
  {
    "url": "/rest/fcc/device",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/info",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/cpu",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/ram",
    "method": "GET"
  },
  ...
]
```

2.2.2 Getting current version of Flowmon

- Returns Current version of FlowMon device.
- @returns 200 { "version": String("Version") }

Example Usage

```
GET /rest/version
```

```
RETURNS:
```

```
[  
  {  
    "version": "11.00.00"  
  }  
]
```

3 Attack

3.1 Data Types

Attacks Search

```
AttacksSearch = {
  ?start: {
    ?from: String("Timestamp"),
    ?to: String("Timestamp"),
  },

  ?end: {
    ?from: String("Timestamp"),
    ?to: String("Timestamp"),
  },

  ?segment: [Integer(ProtectedSegment.id)],
  ?status: [String(Status.id)],
  ?actions: [String("Action label - fulltext search")],
  ?activeMitigation: Boolean
  ?limit: Integer("Default 500")
  ?order: String("asc or desc")
}
```

Attack

```
Attack = {
  id: Integer,
  segment: {
    id: Integer,
    name: String
  },

  start: String(Timestamp),
  end: String(Timestamp),

  status: AttackStatus,
  activeMitigation: Boolean,
  whitelists: [Whitelist],

  actions: ActionStatus
}
```

Attack Status

```
AttackStatus = {
```

```
id: String,
name: String
} << [
  {
    id: "STARTING",
    name: "New about to be automatically mitigated attack"
  },
  {
    id: "ACTIVE",
    name: "Ongoing attack"
  },
  {
    id: "NOT ACTIVE",
    name: "Inactive attack"
  },
  {
    id: "ENDED",
    name: "Finished attack"
  }
]

DetailedAttack = {
  attackTraffic: AttackSummary("Summary of traffic during attack"),
  normalTraffic: TrafficSummary("Summary of normal traffic"),
  chartData: [ChartSerie]
} extends Attack

ActionStatus = {
  action: String,
  performed: String or Null,
  date: String(Y-m-d H:i:s),
  attackDetail: Integer or null,
  info: [DetectionMethodOutput]
}

DetectionMethodOutput = {
  method: Methods,
  flags: [Flag]
}

Flag = {
  name: Flags,
  isSuspected: Boolean
}

Methods extends CodeList << [
  {id: in_adaptive_baseline_rate, name: "Adaptive threshold"},
```

```
{id: in_out_rate, name: "Incoming / Outgoing Ratio"},  
{id: min_rate, name: "Minimal traffic"},  
{id: in_baseline_rate, name: "Manual threshold"}  
]
```

```
Flags extends CodeList << [  
  {id: in_adaptive_baseline_rate.tcp, name: "TCP"},  
  {id: in_adaptive_baseline_rate.tcp_r, name: "TCP Rst"},  
  {id: in_adaptive_baseline_rate.tcp_s, name: "TCP Syn"},  
  {id: in_adaptive_baseline_rate.tcp_sa, name: "TCP Syn Ack"},  
  {id: in_adaptive_baseline_rate.tcp_af, name: "TCP Ask Fin"},  
  {id: in_adaptive_baseline_rate.udp, name: "UDP"},  
  {id: in_adaptive_baseline_rate.icmp, name: "ICMP"},  
  {id: in_out_rate.in_out, name: "Incoming / Outgoing Ratio"},  
  {id: min_rate.in, name: "Packed per second"},  
  {id: min_rate.in_bps, name: "Bits per second"},  
  {id: in_baseline_rate.in, name: "Manual threshold"}  
]
```

Action Status

```
ActionStatus = {  
  action: String,  
  performed: String or Null,  
  date: String(Y-m-d H:i:s),  
  attackDetail: Integer or null,  
  info: [DetectionMethodOutput]  
}  
  
DetectionMethodOutput = {  
  method: Methods,  
  flags: [Flag]  
}  
  
Flag = {  
  name: Flags,  
  isSuspected: Boolean  
}  
  
Methods extends CodeList << [  
  {id: in_adaptive_baseline_rate, name: "Adaptive threshold"},  
  {id: in_out_rate, name: "Incoming / Outgoing Ratio"},  
  {id: min_rate, name: "Minimal traffic"},  
  {id: in_baseline_rate, name: "Manual threshold"}  
]  
  
Flags extends CodeList << [  
  {id: in_adaptive_baseline_rate.tcp, name: "TCP"},
```

```
{id: in_adaptive_baseline_rate.tcp_r, name: "TCP Rst"},
{id: in_adaptive_baseline_rate.tcp_s, name: "TCP Syn"},
{id: in_adaptive_baseline_rate.tcp_sa, name: "TCP Syn Ack"},
{id: in_adaptive_baseline_rate.tcp_af, name: "TCP Ask Fin"},
{id: in_adaptive_baseline_rate.udp, name: "UDP"},
{id: in_adaptive_baseline_rate.icmp, name: "ICMP"},
{id: in_out_rate.in_out, name: "Incoming / Outgoing Ratio"},
{id: min_rate.in, name: "Packed per second"},
{id: min_rate.in_bps, name: "Bits per second"},
{id: in_baseline_rate.in, name: "Manual threshold"}
]
```

3.2 Endpoints

3.2.1 Methods

Get all detection methods.

GET /rest/iad/attacks/methods

- Return CodeList Methods
- @return 200 CodeList Methods (3)

```
GET /rest/iad/attacks/methods
```

```
Return 200:
```

```
[
  {
    "id": "in_adaptive_baseline_rate",
    "name": "Adaptive threshold"
  },
  {
    "id": "in_out_rate",
    "name": "Incoming / Outgoing Ratio"
  },
  {
    "id": "min_rate",
    "name": "Minimal traffic"
  },
]
```

```
{
  "id": "in_baseline_rate",
  "name": "Manual threshold"
}
]
```

3.2.2 Flags

Get all detection flags.

GET /rest/iad/attacks/flags

- Return CodeList Flags
- @return 200 CodeList Flags (3)

```
GET /rest/iad/attacks/flags
```

Return 200:

```
[
  {
    "id": "in_adaptive_baseline_rate.tcp",
    "name": "TCP"
  },
  {
    "id": "in_adaptive_baseline_rate.tcp_r",
    "name": "TCP Rst"
  },
  {
    "id": "in_adaptive_baseline_rate.tcp_s",
    "name": "TCP Syn"
  },
  {
    "id": "in_adaptive_baseline_rate.tcp_sa",
    "name": "TCP Syn Ack"
  },
  {
    "id": "in_adaptive_baseline_rate.tcp_af",
    "name": "TCP Ack Fin"
  },
]
```

```
[
  {
    "id": "in_adaptive_baseline_rate.udp",
    "name": "UDP"
  },
  {
    "id": "in_adaptive_baseline_rate.icmp",
    "name": "ICMP"
  },
  {
    "id": "in_out_rate.in_out",
    "name": "Incoming / Outgoing Ratio"
  },
  {
    "id": "min_rate.in",
    "name": "Packed per second"
  },
  {
    "id": "min_rate.in_bps",
    "name": "Bits per second"
  },
  {
    "id": "in_baseline_rate.in",
    "name": "Manual threshold"
  }
]
```

3.2.3 Attacks

Method for obtaining all Attacks.

GET /rest/iad/attacks/

- Return Attacks
- @return 200 Attack (3)

```
GET /rest/iad/attacks/
```

```
Return 200:
[
```

```
{
  "id": 380,
  "segment": {
    "id": 22,
    "name": "Update by API"
  },
  "suspect": false,
  "falsePositive": true,
  "userComment": "",
  "start": "2017-12-01 14:27:30",
  "end": "2017-12-01 14:54:55",
  "status": {
    "id": "ENDED",
    "name": "Finished attack"
  },
  "activeMitigation": false,
  "actions": [
    {
      "action": "Detected",
      "performed": null,
      "date": "2017-12-01 14:28:33",
      "actionId": null,
      "info": [
        {
          "method": {
            "id": "in_adaptive_baseline_rate",
            "name": "Adaptive threshold"
          },
          "flags": [
            {
              "name": {
                "id": "in_adaptive_baseline_rate.icmp",
                "name": "ICMP"
              },
              "isSuspected": false
            }
          ]
        }
      ]
    },
    {
      "action": "Detected",
      "performed": null,
      "date": "2017-12-01 14:49:33",
      "actionId": null,
      "info": [
        {
          "method": {
```



```
        "id": "in_adaptive_baseline_rate",
        "name": "Adaptive threshold"
    },
    "flags": [
        {
            "name": {
                "id": "in_adaptive_baseline_rate.icmp",
                "name": "ICMP"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_r",
                "name": "TCP Rst"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_s",
                "name": "TCP Syn"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_af",
                "name": "TCP Ack Fin"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_sa",
                "name": "TCP Syn Ack"
            },
            "isSuspected": false
        }
    ]
}
],
{
    "action": "Detected",
    "performed": null,
    "date": "2017-12-01 14:54:33",
    "actionId": null,
```

```
"info": [
  {
    "method": {
      "id": "in_adaptive_baseline_rate",
      "name": "Adaptive threshold"
    },
    "flags": [
      {
        "name": {
          "id": "in_adaptive_baseline_rate.icmp",
          "name": "ICMP"
        },
        "isSuspected": false
      },
      {
        "name": {
          "id": "in_adaptive_baseline_rate.udp",
          "name": "UDP"
        },
        "isSuspected": false
      },
      {
        "name": {
          "id": "in_adaptive_baseline_rate.tcp_r",
          "name": "TCP Rst"
        },
        "isSuspected": false
      },
      {
        "name": {
          "id": "in_adaptive_baseline_rate.tcp_s",
          "name": "TCP Syn"
        },
        "isSuspected": false
      },
      {
        "name": {
          "id": "in_adaptive_baseline_rate.tcp_af",
          "name": "TCP Ack Fin"
        },
        "isSuspected": false
      },
      {
        "name": {
          "id": "in_adaptive_baseline_rate.tcp_sa",
          "name": "TCP Syn Ack"
        },
        "isSuspected": false
      }
    ]
  }
]
```

```
    }
  ]
}
],
{
  "action": "Ended",
  "performed": "manually",
  "date": "2017-12-01 14:54:55",
  "actionId": null,
  "info": []
},
{
  "action": "Mitigation Start (reverted)",
  "performed": "manually",
  "date": "2017-12-01 18:09:06",
  "actionId": 123,
  "info": []
}
]
}
```

3.2.4 Attack info

A method for acquisition of details about a specific attack searched by id.

GET /rest/iad/attacks/<id>

- Return attack information
- @return 200 Attack (3)

```
GET /rest/iad/attacks/380
```

```
Returns 200
```

```
{
  "id": 380,
  "segment": {
    "id": 22,
```

```
    "name": "Update by API"
  },
  "suspect": false,
  "falsePositive": true,
  "userComment": "",
  "start": "2017-12-01 14:27:30",
  "end": "2017-12-01 14:54:55",
  "status": {
    "id": "ENDED",
    "name": "Finished attack"
  },
  "activeMitigation": false,
  "actions": [
    {
      "action": "Detected",
      "performed": null,
      "date": "2017-12-01 14:28:33",
      "actionId": null,
      "info": [
        {
          "method": {
            "id": "in_adaptive_baseline_rate",
            "name": "Adaptive threshold"
          },
          "flags": [
            {
              "name": {
                "id": "in_adaptive_baseline_rate.icmp",
                "name": "ICMP"
              },
              "isSuspected": false
            }
          ]
        }
      ]
    }
  ],
  {
    "action": "Detected",
    "performed": null,
    "date": "2017-12-01 14:49:33",
    "actionId": null,
    "info": [
      {
        "method": {
          "id": "in_adaptive_baseline_rate",
          "name": "Adaptive threshold"
        },
        "flags": [
```

```
    {
      "name": {
        "id": "in_adaptive_baseline_rate.icmp",
        "name": "ICMP"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_r",
        "name": "TCP Rst"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_s",
        "name": "TCP Syn"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_af",
        "name": "TCP Ack Fin"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_sa",
        "name": "TCP Syn Ack"
      },
      "isSuspected": false
    }
  ]
}
],
{
  "action": "Detected",
  "performed": null,
  "date": "2017-12-01 14:54:33",
  "actionId": null,
  "info": [
    {
      "method": {
        "id": "in_adaptive_baseline_rate",
```

```
        "name": "Adaptive threshold"
    },
    "flags": [
        {
            "name": {
                "id": "in_adaptive_baseline_rate.icmp",
                "name": "ICMP"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.udp",
                "name": "UDP"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_r",
                "name": "TCP Rst"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_s",
                "name": "TCP Syn"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_af",
                "name": "TCP Ack Fin"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_sa",
                "name": "TCP Syn Ack"
            },
            "isSuspected": false
        }
    ]
}
]
```

```
    },
    {
      "action": "Ended",
      "performed": "manually",
      "date": "2017-12-01 14:54:55",
      "actionId": null,
      "info": []
    },
    {
      "action": "Mitigation Start (reverted)",
      "performed": "manually",
      "date": "2017-12-01 18:09:06",
      "actionId": 123,
      "info": []
    }
  ]
}
```

3.2.5 Attacks search

Method for search Attacks.

GET /rest/iad/attacks/

- Return Attacks
- @return 200 Attack (3)

```
GET /rest/iad/attacks/?search={
  "segment": [22],
  "status": "ENDED",
  "start": {
    "from": "2017-12-01 00:00:00",
    "to": "2017-12-01 23:59:59"
  }
}&limit=1&order=desc
```

Return 200:

```
[
  {
```

```
"id": 380,
"segment": {
  "id": 22,
  "name": "Update by API"
},
"suspect": false,
"falsePositive": true,
"userComment": "",
"start": "2017-12-01 14:27:30",
"end": "2017-12-01 14:54:55",
"status": {
  "id": "ENDED",
  "name": "Finished attack"
},
"activeMitigation": false,
"actions": [
  {
    "action": "Detected",
    "performed": null,
    "date": "2017-12-01 14:28:33",
    "actionId": null,
    "info": [
      {
        "method": {
          "id": "in_adaptive_baseline_rate",
          "name": "Adaptive threshold"
        },
        "flags": [
          {
            "name": {
              "id": "in_adaptive_baseline_rate.icmp",
              "name": "ICMP"
            },
            "isSuspected": false
          }
        ]
      }
    ]
  },
  {
    "action": "Detected",
    "performed": null,
    "date": "2017-12-01 14:49:33",
    "actionId": null,
    "info": [
      {
        "method": {
          "id": "in_adaptive_baseline_rate",
```



```
        "name": "Adaptive threshold"
    },
    "flags": [
        {
            "name": {
                "id": "in_adaptive_baseline_rate.icmp",
                "name": "ICMP"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_r",
                "name": "TCP Rst"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_s",
                "name": "TCP Syn"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_af",
                "name": "TCP Ack Fin"
            },
            "isSuspected": false
        },
        {
            "name": {
                "id": "in_adaptive_baseline_rate.tcp_sa",
                "name": "TCP Syn Ack"
            },
            "isSuspected": false
        }
    ]
}
]
},
{
    "action": "Detected",
    "performed": null,
    "date": "2017-12-01 14:54:33",
    "actionId": null,
    "info": [
```

```
{
  "method": {
    "id": "in_adaptive_baseline_rate",
    "name": "Adaptive threshold"
  },
  "flags": [
    {
      "name": {
        "id": "in_adaptive_baseline_rate.icmp",
        "name": "ICMP"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.udp",
        "name": "UDP"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_r",
        "name": "TCP Rst"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_s",
        "name": "TCP Syn"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_af",
        "name": "TCP Ack Fin"
      },
      "isSuspected": false
    },
    {
      "name": {
        "id": "in_adaptive_baseline_rate.tcp_sa",
        "name": "TCP Syn Ack"
      },
      "isSuspected": false
    }
  ]
}
```

```
    ]
  }
]
},
{
  "action": "Ended",
  "performed": "manually",
  "date": "2017-12-01 14:54:55",
  "actionId": null,
  "info": []
},
{
  "action": "Mitigation Start (reverted)",
  "performed": "manually",
  "date": "2017-12-01 18:09:06",
  "actionId": 123,
  "info": []
}
]
}
```

3.2.6 Set Whitelists to attack

Method for assign whitelists to attack

POST /rest/iad/attacks/set-whitelists

- Return Attacks
- @return 200 Attack (3)

```
POST /rest/iad/attacks/set-whitelists
BODY:
{
  "entity":
  {
    "attack": 220,
    "whitelists": [
      {
```

```
        "id": 1,
        "name": "#1 Every Day"
    }
]
}
}

Return 201
{
  "id": 220,
  "segment": {
    "id": 3,
    "name": "1.1.1.1",
    "group": null
  },
  "suspect": false,
  "falsePositive": false,
  "userComment": "",
  "attackSignature": "No attack signature available.",
  "attackDetection": [
    "General flood All",
    "Adaptive All",
    "Adaptive UDP"
  ],
  "attackState": [
    0,
    1
  ],
  "baseline": ...,
  "terminationCount": 3,
  "isMitigationFailed": false,
  "flags": [
    "whitelisted"
  ],
  "whitelists": [
    {
      "id": 2,
      "name": "#2 Day of week",
      "from": "2018-01-29 06:00:00",
      "to": "2018-02-02 14:00:00",
      "period": "week_day"
    }
  ],
  "start": "2020-02-18 23:41:42",
  "end": "2020-02-18 23:48:13",
  "status": {
    "id": "ENDED",
    "name": "Finished attack"
  }
}
```

```
},  
"activeMitigation": 0,  
"actions": [...]  
}
```

3.2.7 Unset Whitelists from attack

Method for remove whitelists from attack

POST /rest/iad/attacks/unset-whitelists

- Return Attacks
- @return 200 Attack (3)

```
GET /rest/iad/attacks/unset-whitelists/220
```

```
Return 200  
{  
  "id": 220,  
  "segment": {  
    "id": 3,  
    "name": "1.1.1.1",  
    "group": null  
  },  
  "suspect": false,  
  "falsePositive": false,  
  "userComment": "",  
  "attackSignature": "No attack signature available.",  
  "attackDetection": [  
    "General flood All",  
    "Adaptive All",  
    "Adaptive UDP"  
  ],  
  "attackState": [  
    0,  
    1  
  ],  
  "baseline": ...,  
  "terminationCount": 3,  
}
```

```
"isMitigationFailed": false,  
"flags": [],  
"whitelists": [],  
"start": "2020-02-18 23:41:42",  
"end": "2020-02-18 23:48:13",  
"status": {  
  "id": "ENDED",  
  "name": "Finished attack"  
},  
"activeMitigation": 0,  
"actions": [...]  
}
```

4 Alert

4.1 Data Types

Alert

```
Templates = {
  id: String("Alert id"),
  name: String("Alert name")
}

Alert = {
  id: Integer,
  name: String,
  sendSyslog: Boolean,
  sendSnmp: Boolean,
  snmpGroups: [Integer(snmpGroup.id)]
  sendEmail: Boolean,
  email: String,
  template: Templates("Available email templates"),
  emailEvents: {
    detected: Boolean,
    statistics: Boolean,
    ended: Boolean
  }
  ipTranslate: Boolean,
  runScript: Boolean,
  scriptEvents: {
    detected: Boolean,
    statistics: Boolean,
    ended: Boolean
  },
  scriptParam: String,
  scriptContent: String("base64")
  lang: Language("Language of template")
}
```

4.2 Endpoints

4.2.1 Templates

A templates consists of dials (arrays) of available Alerts.

GET /rest/iad/alerts/templates

- Return CodeList
- @return 200 Template[]

```
GET /rest/iad/alerts/templates
```

```
Returns 200
```

```
[
  {
    id: 1,
    name: "Default template"
  },
  {
    id: 2,
    name: "New Template by API"
  }
]
```

4.2.2 Alerts

4.2.2.1 Get all Alerts

This method serves for obtaining all alerts. The return is an array of available alerts.

GET /rest/iad/alerts/

- Return all Alerts
- @return 200 Alerts[] (4)

```
GET /rest/iad/alerts/
```

```
Returns 200
```

```
[
```



```
{
  "id": 1,
  "name": "Alert!",
  "sendSyslog": false,
  "sendSnmp": false,
  "snmpGroups": [],
  "sendEmail": true,
  "email": "test@test.cz",
  "emailEvents": {
    "detected": true,
    "statistics": false,
    "ended": true
  },
  "runScript": false,
  "scriptEvents": {
    "detected": false,
    "statistics": false,
    "ended": false
  },
  "template": {
    "id": 1,
    "name": "Default template"
  },
  "ipTranslations": false,
  "scriptParam": "",
  "scriptContent": ""
},
{
  "id": 2,
  "name": "Novy alert",
  "sendSyslog": false,
  "sendSnmp": false,
  "snmpGroups": [],
  "sendEmail": true,
  "email": "test@test.cz",
  "emailEvents": {
    "detected": true,
    "statistics": false,
    "ended": true
  },
  "runScript": true,
  "scriptEvents": {
    "detected": false,
    "statistics": false,
    "ended": true
  },
  "template": {
    "id": 19,
```

```
        "name": "New Template by API"
    },
    "ipTranslations": false,
    "scriptParam": "",
    "scriptContent": "IyEvYmluL2Jhc2gKZWNobyAiSGVsbG8sIFdvcmxkISIKZWNobyBEQVRFPWBkYXR1ICcrJVkt="
  }
]
```

4.2.2.2 Get one specific Alert

A method for acquisition information about specific Alert searched by id.

GET /rest/iad/alerts/<id>

- Return Alert
- @return 200 Alert (4)

```
GET /rest/iad/alerts/1
```

Returns 200

```
{
  "id": 1,
  "name": "Alert!",
  "sendSyslog": false,
  "sendSnmp": false,
  "snmpGroups": [],
  "sendEmail": true,
  "email": "test@test.cz",
  "emailEvents": {
    "detected": true,
    "statistics": false,
    "ended": true
  },
  "runScript": false,
  "scriptEvents": {
    "detected": false,
    "statistics": false,
    "ended": false
  },
  "template": {
```

```
    "id": 1,  
    "name": "Default template"  
  },  
  "ipTranslations": false,  
  "scriptParam": "",  
  "scriptContent": ""  
}
```

4.2.2.3 Add new Alert

A method for adding a new Alert. New alert must have an empty id.

POST /rest/iad/alerts/

- Return new Alert
- @return 200 Alert (4)

```
POST /rest/iad/alerts/  
BODY:  
{  
  "entity":  
  {  
    "id": null,  
    "name": "New Alert by API",  
    "sendSyslog": true,  
    "sendSnmp": false,  
    "snmpGroups": [],  
    "sendEmail": true,  
    "email": "flowmon@flowmon.com",  
    "emailEvents": {  
      "detected": true,  
      "statistics": false,  
      "ended": true  
    },  
    "runScript": true,  
    "scriptEvents": {  
      "detected": false,  
      "statistics": false,  
      "ended": true  
    }  
  },  
}
```

```
"template": {
  "id": 19,
  "name": "New Template by API"
},
"ipTranslations": false,
"scriptParam": "",
"scriptContent": "IyEvYmluL2Jhc2gKZWNoYAiSGVsbG8sIFdvcmxkISIKZWNoYBEQVRFPWBkYXR1ICcrJVkt="
}
}
```

Returns 201

```
{
  "id": 4,
  "name": "New Alert by API",
  "sendSyslog": true,
  "sendSnmp": false,
  "sendEmail": true,
  "snmpGroups": [],
  "email": "flowmon@flowmon.com",
  "emailEvents": {
    "detected": true,
    "statistics": false,
    "ended": true
  },
  "runScript": true,
  "scriptEvents": {
    "detected": false,
    "statistics": false,
    "ended": true
  },
  "template": {
    "id": 19,
    "name": "New Template by API"
  },
  "ipTranslations": false,
  "scriptParam": "",
  "scriptContent": "IyEvYmluL2Jhc2gKZWNoYAiSGVsbG8sIFdvcmxkISIKZWNoYBEQVRFPWBkYXR1ICcrJVkt="
}
```

4.2.2.4 Update existing Alert

A method for updating Alert. ScriptContent is not an obligatory item. The content is overwritten only when you type.

PUT /rest/iad/alerts/

- Return updated Alert
- @return 200 Alert (4)

```
PUT /rest/iad/alerts/
BODY:
{
  "entity":
  {
    "id": 4,
    "name": "Update Alert by API",
    "sendSyslog": true,
    "sendSnmp": false,
    "snmpGroups": [],
    "sendEmail": false,
    "email": "flowmon@flowmon.com",
    "emailEvents": {
      "detected": true,
      "statistics": false,
      "ended": true
    },
    "runScript": true,
    "scriptEvents": {
      "detected": false,
      "statistics": false,
      "ended": true
    },
    "template": {
      "id": 19,
      "name": "New Template by API"
    },
    "ipTranslations": false,
    "scriptParam": "",
    "scriptContent": "IyEvYmluL2Jhc2gKZWNobyAiSGVsbg8sIFdvcmxkISIKZWNobyBEQVRFPWBkYXR1ICcrJVkt="
  }
}
```

Returns 200

```
{
  "id": 4,
  "name": "Update Alert by API",
  "sendSyslog": true,
  "sendSnmp": false,
```

```
"snmpGroups": [],
"sendEmail": false,
"email": "flowmon@flowmon.com",
"emailEvents": {
  "detected": true,
  "statistics": false,
  "ended": true
},
"runScript": true,
"scriptEvents": {
  "detected": false,
  "statistics": false,
  "ended": true
},
"template": {
  "id": 19,
  "name": "New Template by API"
},
"ipTranslations": false,
"scriptParam": "",
"scriptContent": "IyEvYmluL2Jhc2gKZWNoYAiSGVsbG8sIFdvcmxkISIKZWNoYBEQVRFPWBkYXR1ICcrJVkt="
}
```

4.2.2.5 Delete specific Alert

Deletes specific Alert by id. Return code is 204 (No data).

DELETE /rest/iad/alerts/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/alerts/4
```

Returns 204

5 Router

5.1 Data Types

Router

```
Router = {
  id: Integer,
  name: String,
  ip: String(IPv4),
  plugin: String,
  osci: String << ["ACL", "BGP"],
  aclName: String,
  login: String,
  password: String,
  protocol: Strong,
  acl6Name: String,
  community: String,
  asNum: Integer,
  asNumIad: Integer,
  bgpMode: String,
  nextHopIp: String(IPv4),
  nextHopIp6: String(IPv6),
  ip6: String(IPv6),
  interface: String,
  bgpNeighbour: bgpNeighbour("Only with PLUGIN = bgp and OSCI = defensepro"),
  tieringEnabled: Boolean,
  tieringCommunity: String,
  tieringThreshold: Integer,
  tieringMaskIp4: Integer|null,
  tieringMaskIp6: Integer|null,
  connectionStatusInfo: String|null
}

bgpNeighBour = {
  ip: String,
  asNum: Integer,
  activated: Boolean
}
```

5.2 Endpoints

5.2.1 Routers

5.2.1.1 Get all Routers

This method serves for obtaining all Routers. The return is an array of available Routers.

GET /rest/iad/routers/

- Return all Routers
- @return 200 Router[] (5)

```
GET /rest/iad/routers/
```

```
Returns 200
```

```
[
  {
    "id": 1,
    "name": "First",
    "ip": "10.11.12.13",
    "aclName": "",
    "login": "test",
    "password": "password",
    "protocol": "ssh",
    "locked": false,
    "acl6Name": "",
    "community": "",
    "asNum": 1,
    "asNumIad": 1,
    "bgpMode": "ebgp",
    "nextHopIp": null,
    "nextHopIp6": null,
    "ip6": null,
    "interface": "auto",
    "bgpNeighbour": {
      "ip": "12.2.2.2",
      "asNum": 2,
      "activated": true
    }
  }
]
```



```
    },
    "osci": "exabgp",
    "plugin": "exabgp",
    "tieringEnabled": false,
    "tieringCommunity": "",
    "tieringThreshold": 0,
    "tieringMaskIp4": null,
    "tieringMaskIp6": null,
    "connectionStatusInfo": null
  },
  {
    "id": 6,
    "name": "test",
    "ip": "1.2.3.4",
    "aclName": "",
    "login": "",
    "password": "",
    "protocol": "ssh",
    "locked": false,
    "acl6Name": "",
    "community": "",
    "asNum": 1,
    "asNumIad": 1,
    "bgpMode": "ebgp",
    "nextHopIp": null,
    "nextHopIp6": null,
    "ip6": null,
    "interface": "auto",
    "bgpNeighbour": null,
    "osci": "exabgp",
    "plugin": "exabgp",
    "tieringEnabled": false,
    "tieringCommunity": "",
    "tieringThreshold": 0,
    "tieringMaskIp4": null,
    "tieringMaskIp6": null,
    "connectionStatusInfo": null
  }
]
```

5.2.1.2 Get one specific Router

A method for acquisition of informations about a specific Router searched by id.

GET /rest/iad/routers/<id>

- Return Router
- @return 200 Router (5)

```
GET /rest/iad/routers/1
```

```
Returns 200
```

```
{
  "id": 6,
  "name": "test",
  "ip": "1.2.3.4",
  "aclName": "",
  "login": "",
  "password": "",
  "protocol": "ssh",
  "locked": false,
  "acl6Name": "",
  "community": "",
  "asNum": 1,
  "asNumIad": 1,
  "bgpMode": "ebgp",
  "nextHopIp": null,
  "nextHopIp6": null,
  "ip6": null,
  "interface": "auto",
  "bgpNeighbour": null,
  "osci": "exabgp",
  "plugin": "exabgp",
  "tieringEnabled": false,
  "tieringCommunity": "",
  "tieringThreshold": 0,
  "tieringMaskIp4": null,
  "tieringMaskIp6": null,
  "connectionStatusInfo": null
}
```

5.2.1.3 Add new Router

A method for adding a new Router. New Router must have empty id.

POST /rest/iad/routers/

- Return new Router
- @return 201 Router (5)

```
POST /rest/iad/routers/
BODY:
{
  "entity":
  {
    "id": null,
    "name": "First",
    "ip": "10.11.12.15",
    "aclName": "",
    "login": "test",
    "password": "password",
    "protocol": "ssh",
    "acl6Name": "",
    "community": "",
    "asNum": 1,
    "asNumIad": 1,
    "bgpMode": "ebgp",
    "nextHopIp": null,
    "nextHopIp6": null,
    "ip6": null,
    "interface": "auto",
    "bgpNeighbour": {
      "ip": "12.2.2.2",
      "asNum": 2,
      "activated": true
    },
    "osci": "exabgp",
    "plugin": "exabgp",
    "tieringEnabled": false,
    "tieringCommunity": "",
    "tieringThreshold": 0,
    "tieringMaskIp4": null,
    "tieringMaskIp6": null
  }
}

Returns 200
{
  "id": 8,
```

```
"name": "First",
"ip": "10.11.12.15",
"aclName": "",
"login": "test",
"password": "password",
"protocol": "ssh",
"locked": false,
"acl6Name": "",
"community": "",
"asNum": 1,
"asNumIad": 1,
"bgpMode": "ebgp",
"nextHopIp": null,
"nextHopIp6": null,
"ip6": null,
"interface": "auto",
"bgpNeighbour": {
  "ip": "12.2.2.2",
  "asNum": 2,
  "activated": true
},
"osci": "exabgp",
"plugin": "exabgp",
"tieringEnabled": false,
"tieringCommunity": "",
"tieringThreshold": 0,
"tieringMaskIp4": null,
"tieringMaskIp6": null,
"connectionStatusInfo": null
}
```

5.2.1.4 Update existing Router

A method for updating a Router.

PUT /rest/iad/routers/

- Return updated Router
- @return 200 Router (5)

PUT /rest/iad/routers/

BODY:

```
{
  "entity":
  {
    "id": 8,
    "name": "First",
    "ip": "10.11.12.15",
    "aclName": "",
    "login": "",
    "password": "",
    "protocol": "ssh",
    "acl6Name": "",
    "community": "",
    "asNum": 1,
    "asNumIad": 1,
    "bgpMode": "ebgp",
    "nextHopIp": null,
    "nextHopIp6": null,
    "ip6": null,
    "interface": "auto",
    "bgpNeighbour": {
      "ip": "12.2.2.3",
      "asNum": 2,
      "activated": true
    },
    "osci": "defensepro",
    "plugin": "exabgp",
    "tieringEnabled": false,
    "tieringCommunity": "",
    "tieringThreshold": 0,
    "tieringMaskIp4": null,
    "tieringMaskIp6": null
  }
}
```

Returns 200

```
{
  "id": 8,
  "name": "First",
  "ip": "10.11.12.15",
  "plugin": "exabgp",
  "osci": "defensepro",
  "aclName": "",
  "login": "",
  "password": "",
  "protocol": "ssh",
  "locked": false,
  "acl6Name": "",
```

```
"community": "",
"asNum": 1,
"asNumIad": 1,
"bgpMode": "ebgp",
"nextHopIp": null,
"nextHopIp6": null,
"ip6": null,
"interface": "auto",
"bgpNeighbour": {
  "ip": "12.2.2.3",
  "asNum": 2,
  "activated": true
},
"tieringEnabled": false,
"tieringCommunity": "",
"tieringThreshold": 0,
"tieringMaskIp4": null,
"tieringMaskIp6": null,
"connectionStatusInfo": null
}
```

5.2.1.5 Delete specific Router

Deletes specific Router by id. Return code is 204 (No data).

DELETE /rest/iad/routers/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/routers/8
```

Returns 204

6 Scrubbing Center

6.1 Data Types

Scrubbing Center

```
ScrubbingCenter = {  
  id: Integer,  
  name: String,  
  ip: String(IPv4|IPv6),  
  osci: String,  
  login: String,  
  password: String,  
  locked: Boolean(ReadOnly),  
  options: String  
}
```

6.2 Endpoints

6.2.1 Scrubbing Centers

6.2.1.1 Get all Scrubbing Centers

This method serves for obtaining all Scrubbing Centers. The return is a array of available Scrubbing Centers.

GET /rest/iad/scrubbing-centers/

- Return all Scrubbing Centers
- @return 200 ScrubbingCenters[] (6)

```
GET /rest/iad/scrubbing-centers/
```

```
Returns 200
```

```
[  
  {
```

```
"id": 1,
  "name": "Moje AbsoluteVision3",
  "ip": "192.168.50.122",
  "osci": "APSolute Vision",
  "login": "aaaa",
  "password": "aaaa",
  "locked": false,
  "options": "keep_rtemplate"
},
{
  "id": 3,
  "name": "scrub c to fail",
  "ip": "192.168.55.58",
  "osci": "APSolute Vision",
  "login": "aaaa",
  "password": "aaaa",
  "locked": false,
  "options": "keep_rtemplate"
}
]
```

6.2.1.2 Get one specific Scrubbing Center

Method for acquisition of informations about specific Scrubbing Center searched by id.

GET /rest/iad/scrubbing-centers/<id>

- Return Scrubbing Center
- @return 200 ScrubbingCenter (6)

```
GET /rest/iad/scrubbing-centers/1
```

Returns 200

```
{
  "id": 1,
  "name": "Moje AbsoluteVision3",
  "ip": "192.168.50.122",
  "osci": "APSolute Vision",
  "login": "aaaa",
  "password": "aaaa",
```



```
"locked": false,  
"options": "keep_rtemplate"  
}
```

6.2.1.3 Add new Scrubbing Center

Method for adding a new Scrubbing Center. A new Scrubbing Center must have an empty id.

POST /rest/iad/scrubbing-centers/

- Return new Scrubbing Center
- @return 200 ScrubbingCenter (6)

```
POST /rest/iad/scrubbing-centers/  
BODY:  
{  
  "entity":  
  {  
    "id": null,  
    "name": "Add by Api",  
    "ip": "192.168.50.207",  
    "osci": "APSSolute Vision",  
    "login": "aaaa",  
    "password": "bbbb",  
    "options": "keep_rtemplate"  
  }  
}
```

Returns 200

```
{  
  "id": 6,  
  "name": "Add by Api",  
  "ip": "192.168.50.207",  
  "osci": "APSSolute Vision",  
  "login": "aaaa",  
  "password": "bbbb",  
  "locked": false,  
  "options": "keep_rtemplate"  
}
```

6.2.1.4 Update existing Scrubbing Center

A method for update a Scrubbing Center. "Locked" is read-only attribute.

PUT /rest/iad/scrubbing-centers/

- Return updated Scrubbing Center
- @return 200 ScrubbingCenter (6)

```
PUT /rest/iad/scrubbing-centers/  
BODY:  
{  
  "entity":  
  {  
    "id": 6,  
    "name": "Update by API",  
    "ip": "192.168.51.207",  
    "osci": "APbsolute Vision",  
    "login": "aaaa",  
    "password": "bbbb",  
    "options": "keep_rtemplate"  
  }  
}
```

Returns 200

```
{  
  "id": 6,  
  "name": "Update by API",  
  "ip": "192.168.51.207",  
  "osci": "APbsolute Vision",  
  "login": "aaaa",  
  "password": "bbbb",  
  "locked": false,  
  "options": "keep_rtemplate"  
}
```

6.2.1.5 Delete specific Scrubbing Center

Deletes specific Scrubbing Center by id. Return code is 204 (No data).

DELETE /rest/iad/scrubbing-centers/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/scrubbing-centers/6
```

```
Returns 204
```

7 Scrubbing Center Parameter

7.1 Data Types

Scrubbing Center Parameter

```
ScrubbingCenterParameter = {  
  option: String("Key of option"),  
  value: String  
}
```

7.2 Endpoints

7.2.1 Scrubbing Center Parameters

7.2.1.1 Get all defaults Scrubbing Center Parameters

This method serve for obtaining all default Scrubbing Center Parameters. Return is an array of available Scrubbing Center Parameters.

GET /rest/iad/scrubbing-center-parameters

- Return all Scrubbing Center Parameters
- @return 200 ScrubbingCenterParameters[] (7)

```
GET /rest/iad/scrubbing-center-parameters
```

```
Returns 200
```

```
[  
  {  
    "option": "dns_a_in_ipv4",  
    "value": "72"  
  },  
  {  
    "option": "dns_a_in_ipv6",  
    "value": "72"  
  }  
]
```

```
  },
  {
    "option": "dns_a_out_ipv4",
    "value": "0"
  },
  {
    "option": "dns_a_out_ipv6",
    "value": "0"
  },
  {
    "option": "dns_mx_in_ipv4",
    "value": "42"
  },
  {
    "option": "dns_mx_in_ipv6",
    "value": "42"
  },
  ...
  {
    "option": "con_limit_udp",
    "value": "30000"
  }
]
```

7.2.1.2 Update existing default Scrubbing Center Parameters

Method for updating Scrubbing Center Parameters. It is possible to update one or more parameters at the same time.

PUT /rest/iad/scrubbing-center-parameters

- Return updated Scrubbing Center Parameters
- @return 200 ScrubbingCenterParameters[] (7)

```
PUT /rest/iad/scrubbing-center-parameters
BODY:
{
  "entity":
```

```
[
  {
    "option": "dns_a_in_ipv4",
    "value": "42"
  },
  {
    "option": "dns_ptr_in_ipv6",
    "value": "72"
  }
]
```

Returns 200

```
[
  {
    "option": "dns_a_in_ipv4",
    "value": "72"
  },
  {
    "option": "dns_a_in_ipv6",
    "value": "72"
  },
  {
    "option": "dns_a_out_ipv4",
    "value": "0"
  },
  {
    "option": "dns_a_out_ipv6",
    "value": "0"
  },
  {
    "option": "dns_mx_in_ipv4",
    "value": "42"
  },
  {
    "option": "dns_mx_in_ipv6",
    "value": "42"
  },
  {
    "option": "dns_mx_out_ipv4",
    "value": "0"
  },
  {
    "option": "dns_mx_out_ipv6",
    "value": "0"
  },
  {
    "option": "dns_ptr_in_ipv4",
```

```
    "value": "42"  
  },  
  {  
    "option": "dns_ptr_in_ipv6",  
    "value": "42"  
  },  
  ...  
  {  
    "option": "con_limit_udp",  
    "value": "30000"  
  }  
]
```

8 Report Chapters

8.1 Data Types

CodeList

```
CodeList = {  
  id: Integer|String("Item id"),  
  name: String("Item name")  
}
```

Chapter

```
Chapter = {  
  id: Integer,  
  name: String,  
  description: String,  
  scope: ChapterScope("Whether it is overall or segment based chapter"),  
  
  //TODO - better name  
  template: ChapterTemplate("What (predefined) data is displayed")  
  parameters: ChapterParameters("Parameters this chapter is defined with, differ for various scopes")  
}  
  
ChapterParameters = OverallChapterParameters | PerSegmentChapterParameters  
  
OverallChapterParameters = {  
  segments: ProtectedSegment.{id, name}[]("Array of selected segments this chapter reports attacks from"),  
  attackedOnly: Integer(Boolean("If set to true, only attacked segments will be shown"))  
}  
  
PerSegmentChapterParameters = {  
  segment: ProtectedSegment.{id,name}("Explicit segment this chapter will display data for"),  
  nthTop: Integer("Index of segment this chapter will display data for")  
}  
  
ChapterScope extends CodeList << [  
  {id: 1, name: "Overall chapters"},  
  {id: 2, name: "Per-segment chapters"}  
]  
  
ReportedMetric extends CodeList << [  
  {id: "overall.attacksOverview", name: "Attacks overview"},  
  {id: "overall.ongoingAttacks", name: "Ongoing attacks"},  
  {id: "overall.overallStats", name: "Overall stats"},  
  {id: "overall.topAttacksPerSegment", name: "Top attacks per segment"},  
  {id: "overall.topAttacksPerType", name: "Top attacks per type"},
```



```
{id: "segment.attacksOverview", name: "Attacks overview for segment XXX"},  
{id: "segment.overallStats", name: "Overall stats for segment XXX"},  
{id: "segment.topAttacksPerType", name: "Top attacks per type for segment XXX"}  
]
```

8.2 Endpoints

8.2.1 Code List

Code Lists are consist of dials (arrays), where id can be integer or string.

GET /rest/iad/report-chapter/scopes

or

GET /rest/iad/report-chapter/templates

- Return Code List
- @return 200 CodeList[] (8.1)

```
GET /rest/iad/report-chapter/scopes
```

```
Returns 200
```

```
[  
  {  
    "id": 1,  
    "name": "Overall chapters"  
  },  
  {  
    "id": 2,  
    "name": "Per-segment chapters"  
  }  
]
```

```
GET /rest/iad/report-chapter/templates
```

```
Returns 200
```

```
[
```

```
{
  "id": "overall.attacksOverview",
  "name": "Attacks overview"
},
{
  "id": "overall.ongoingAttacks",
  "name": "Ongoing attacks"
},
{
  "id": "overall.overallStats",
  "name": "Overall stats"
},
{
  "id": "overall.topAttacksPerSegment",
  "name": "Top attacks per segment"
},
{
  "id": "overall.topAttacksPerType",
  "name": "Top attacks per type"
},
{
  "id": "segment.attacksOverview",
  "name": "Attacks overview for segment XXX"
},
{
  "id": "segment.overallStats",
  "name": "Overall stats for segment XXX"
},
{
  "id": "segment.topAttacksPerType",
  "name": "Top attacks per type for segment XXX"
}
]
```

8.2.2 Chapters

8.2.2.1 Get all Chapters

This method returns list od all Report Chapters.

GET /rest/iad/report-chapter/

- Return all Chapters
- @return 200 Chapter[] (8.1)
- @return 204 No data

```
GET /rest/iad/report-chapter/
```

```
Returns 200
```

```
[
  {
    "id": 1,
    "name": "Chapter",
    "description": "Description asd",
    "parameters": {
      "segments": [
        {
          "id": 7,
          "name": "Prvn"
        }
      ],
      "attackedOnly": true
    },
    "scope": {
      "id": 1,
      "name": "Overall chapters"
    },
    "template": {
      "id": "overall.overallStats",
      "name": "Overall stats"
    }
  },
  {
    "id": 2,
    "name": "Chapter no.2",
    "description": "Description new",
    "parameters": {
      "segment": {
        "id": 7,
        "name": "Prvn"
      },
      "nthBest": null
    },
    "scope": {
      "id": 2,
      "name": "Per-segment chapters"
    }
  }
]
```

```
    },
    "template": {
      "id": "segment.overallStats",
      "name": "Overall stats for segment XXX"
    }
  }
]
```

8.2.2.2 Get one specific Chapter

This method returns details of single Report Chapter given by its ID

GET /rest/iad/report-chapter/<id>

- @param <id> ReportChapter ID
- Return Chapter
- @return 200 Chapter (8.1)

```
GET /rest/iad/report-chapter/1
```

Returns 200

```
{
  "id": 1,
  "name": "Chapter",
  "description": "Popis",
  "parameters": {
    "segments": [
      {
        "id": 7,
        "name": "Prvn"
      }
    ],
    "attackedOnly": true
  },
  "scope": {
    "id": 1,
    "name": "Overall chapters"
  },
  "template": {
```

```
"id": "overall.overallStats",
"name": "Overall stats"
}
}
```

8.2.2.3 Add new Chapter

This method creates a new Report Chapter. New Report Chapter must have unique combination of name, type and selected Segments.

POST /rest/iad/report-chapter/

- Return new Chapter
- @return 405 ValidationResult[]
- @return 201 Chapter (8.1)

```
POST /rest/iad/report-chapter/
BODY:
{
  "entity":
  {
    "id": null,
    "name": "Chapter new REST",
    "description": "Description of chapter",
    "scope": {
      "id": 1,
      "name": "Overall chapters"
    },
    "template": {
      "id": "overall.overallStats",
      "name": "Overall stats"
    },
    "parameters": {
      "segments": [
        {
          "id": 7,
          "name": "Prvn"
        }
      ]
    }
  }
}
```

```
        "attackedOnly": true
      }
    }
  }
}

Returns 201
{
  "id": 1,
  "name": "Chapter new REST",
  "description": "Description of chapter",
  "parameters": {
    "segments": [
      {
        "id": 7,
        "name": "Prvn"
      }
    ],
    "attackedOnly": true
  },
  "scope": {
    "id": 1,
    "name": "Overall chapters"
  },
  "template": {
    "id": "overall.overallStats",
    "name": "Overall stats"
  }
}
```

8.2.2.4 Update existing Chapter

This method updates an existing Report Chapter.

PUT /rest/iad/report-chapter/

- Return an updated Chapter
- @return 404 Not found
- @return 405 ValidationResult[]
- @return 200 Chapter (8.1)

```
PUT /rest/iad/report-chapter/  
BODY:  
{  
  "entity":  
  {  
    "id": 1,  
    "name": "Chapter new REST",  
    "description": "New description",  
    "scope": {  
      "id": 1,  
      "name": "Overall chapters"  
    },  
    "template": {  
      "id": "overall.overallStats",  
      "name": "Overall stats"  
    },  
    "parameters": {  
      "segments": [  
        {  
          "id": 7,  
          "name": "Prvn"  
        }  
      ],  
      "attackedOnly": true  
    }  
  }  
}
```

Returns 200

```
{  
  "id": 1,  
  "name": "Chapter new REST",  
  "description": "New description",  
  "parameters": {  
    "segments": [  
      {  
        "id": 7,  
        "name": "Prvn"  
      }  
    ],  
    "attackedOnly": true  
  },  
  "scope": {  
    "id": 1,  
    "name": "Overall chapters"  
  },  
  "template": {
```

```
"id": "overall.overallStats",  
  "name": "Overall stats"  
}  
}
```

8.2.2.5 Delete specific Chapter

Delete specific Report Chapter by its id. Return code is 204 (No data).

DELETE /rest/iad/report-chapter/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/report-chapter/1
```

Returns 204

9 Email Templates

9.1 Data Types

Languages

```
Language = {  
  id: String("Language id"),  
  name: String("Language name")  
}
```

EmailTemplate

```
EmailTemplate = {  
  id: Integer,  
  name: String,  
  subject: String,  
  body: String,  
  assigned: Boolean,  
  lang: Language("Language of template")  
}
```

9.2 Endpoints

9.2.1 Languages

This method returns a list of available languages.

GET /rest/iad/email-templates/languages

- @return 200 Language[] (9.1)

```
GET /rest/iad/email-templates/languages
```

```
Returns 200
```

```
[  
  {  
    id: "cz",
```

```
    name: "Cesky"
  },
  {
    id: "en",
    name: "English"
  }
]
```

9.2.2 EmailTemplates

9.2.2.1 Get all Email Templates

This method serve for obtaining all Email Templates. The return is an array of available templates.

GET /rest/iad/email-templates/

- Return all EmailTemplates
- @return 200 EmailTemplate[] (9)

```
GET /rest/iad/email-templates/
```

Returns 200

```
[
  {
    "id": 1,
    "name": "Default template",
    "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
    "body": "Hi,
```

Segment: %SEGMENT (subnets: %SUBNETS)

Time: %TIME

Event type: %EVENT

Triggered detection methods: %METHODS

Mitigation configuration: %MITIGATION_TYPE

Mitigation status: %MITIGATION_STATUS

Flowmon device: %DEVICE

You can check the event here: %LINK

```
Your Flowmon DDoS Defender",
  "lang": {
    "id": "en",
    "name": "English"
  }
},
{
  "id": 2,
  "name": "Second template",
  "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
  "body": "Hi,

Segment: %SEGMENT (subnets: %SUBNETS)
Time: %TIME
Event type: %EVENT
Triggered detection methods: %METHODS
Mitigation configuration: %MITIGATION_TYPE
Mitigation status: %MITIGATION_STATUS
Flowmon device: %DEVICE

You can check the event here: %LINK

Your Flowmon DDoS Defender",
  "lang": {
    "id": "en",
    "name": "English"
  }
}
]
```

9.2.2.2 Get one specific Email Template

A method for acquisition of informations about a specific Email Template searched by its id.

GET /rest/iad/email-templates/<id>

- Return Email Template
- @return 200 EmailTemplate (g)

```
GET /rest/iad/email-templates/1
```

```
Returns 200
{
  "id": 1,
  "name": "Default template",
  "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
  "body": "Hi,

Segment: %SEGMENT (subnets: %SUBNETS)
Time: %TIME
Event type: %EVENT
Triggered detection methods: %METHODS
Mitigation configuration: %MITIGATION_TYPE
Mitigation status: %MITIGATION_STATUS
Flowmon device: %DEVICE
You can check the event here: %LINK
Your Flowmon DDoS Defender",
  "lang": {
    "id": "en",
    "name": "English"
  }
}
```

9.2.2.3 Add new Email Template

A method for adding a new Email Template. A new template must have empty id.

POST /rest/iad/email-template/

- Return new Email Template
- @return 201 EmailTemplate (9.1)

```
POST /rest/iad/email-template/
BODY:
{
  "entity":
  {
    "id": null,
    "name": "New Template by API",
    "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
```

```
        "body": "Hi,  
  
Segment: %SEGMENT (subnets: %SUBNETS)  
Time: %TIME  
Event type: %EVENT  
Triggered detection methods: %METHODS  
Mitigation configuration: %MITIGATION_TYPE",  
    "lang": {  
        "id": "en",  
        "name": "English"  
    }  
}  
  
Returns 201  
{  
    "id": 3,  
    "name": "New Template by API",  
    "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",  
    "body": "Hi,  
  
Segment: %SEGMENT (subnets: %SUBNETS)  
Time: %TIME  
Event type: %EVENT  
Triggered detection methods: %METHODS  
Mitigation configuration: %MITIGATION_TYPE",  
    "lang": {  
        "id": "en",  
        "name": "English"  
    }  
}
```

9.2.2.4 Update existing Email Template

A method for updating an Email Template.

PUT /rest/iad/email-template/

- Return updated Email Template
- @return 200 EmailTemplate (9.1)

```
PUT /rest/iad/email-template/
BODY:
{
  "entity":
  {
    "id": 3,
    "name": "Edit Template by API",
    "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
    "body": "Hi,

Segment: %SEGMENT (subnets: %SUBNETS)
Time: %TIME
Event type: %EVENT
Triggered detection methods: %METHODS
Mitigation configuration: %MITIGATION_TYPE",
    "lang": {
      "id": "en",
      "name": "English"
    }
  }
}

Returns 200
{
  "id": 3,
  "name": "Edit Template by API",
  "subject": "DDD Alert on %DEVICE: %TIME - %SEGMENT - %EVENT",
  "body": "Hi,

Segment: %SEGMENT (subnets: %SUBNETS)
Time: %TIME
Event type: %EVENT
Triggered detection methods: %METHODS
Mitigation configuration: %MITIGATION_TYPE",
  "lang": {
    "id": "en",
    "name": "English"
  }
}
```

9.2.2.5 Delete specific Email Template

Delete a specific Email Template by its id. Return code is 204 (No data). Assigned Email Template and with id 1 cannot be deleted.

DELETE /rest/iad/email-template/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/email-template/3
```

```
Returns 204
```

10 Rule

10.1 Data Types

Rule

```
Rule = {
  id: Integer,
  name: String,
  baselineSlots: Integer("Length of learning period"),
  anomalyLength: Integer("This option can be used to avoid false attack detections for short data bursts. (in second)", 10),
  historyWeek: Integer("Length of learning period for detection based on week day"),
  zeroBaseline: Boolean,
  subRules: SubRule[],
  customBaselines: CustomBaseline[]
}

SubRule = {
  id: Integer,
  ruleId: Integer,
  trafficType: TrafficType,
  detectionMethod: DetectionMethod,
  options: Option[]
}

TrafficType = {
  id: Integer,
  name: String
}

DetectionMethod = {
  id: Integer,
  name: String,
  options: String[]
}

Option = {
  id: Integer,
  option: String,
  value: Integer
}
```


10.2 Endpoints

10.2.1 DetectionMethod

This method returns a list of available detection methods.

GET /rest/iad/rules/detection-methods

- @return 200 DetectionMethod[]

```
GET /rest/iad/rules/detection-methods

Returns 200
[
  {
    "name": "AdaptiveAttack",
    "description": "Adaptive attack",
    "options": [
      "PpsNear",
      "BpsNear",
      "MultiplierNear",
      "PpsWeek",
      "BpsWeek",
      "MultiplierWeek"
    ]
  },
  {
    "name": "AdaptiveAttackSuspect",
    "description": "Adaptive attack suspect",
    "options": [
      "PpsNear",
      "BpsNear",
      "MultiplierNear",
      "PpsWeek",
      "BpsWeek",
      "MultiplierWeek"
    ]
  },
  {
    "name": "InOutRatio",
    "description": "in/out ratio",
    "options": [
```

```
        "InOutRatio"
    ]
},
{
    "name": "Manual",
    "description": "Manual",
    "options": [
        "ManualRatio"
    ]
},
{
    "name": "MinrateAttack",
    "description": "Min rate attack",
    "options": [
        "Pps",
        "Bps",
        "Pps;Bps"
    ]
},
{
    "name": "MinrateEvaluate",
    "description": "Min rate evaluate",
    "options": [
        "Pps",
        "Bps",
        "Pps;Bps"
    ]
},
{
    "name": "OverrideInMinrate",
    "description": "Override in min rate",
    "options": []
}
]
```

10.2.2 TrafficType

This method returns a list of available traffic types.

GET /rest/iad/rules/traffic-types

- @return 200 TrafficType[]

```
GET /rest/iad/rules/traffic-types
```

```
Returns 200
```

```
[
  {
    "id": 19,
    "name": "in",
    "description": "All"
  },
  {
    "id": 16,
    "name": "in_dns",
    "description": "DNS"
  },
  {
    "id": 17,
    "name": "in_http",
    "description": "HTTP"
  },
  {
    "id": 18,
    "name": "in_https",
    "description": "HTTPS"
  },
  {
    "id": 10,
    "name": "in_icmp",
    "description": "ICMP"
  },
  {
    "id": 3,
    "name": "in_tcp",
    "description": "TCP"
  },
  {
    "id": 6,
    "name": "in_tcp_a",
    "description": "TCP (flag ACK)"
  },
  {
    "id": 7,
    "name": "in_tcp_af",
    "description": "TCP (flag ACK, FIN)"
  },
  {
    "id": 9,
    "name": "in_tcp_ap",
```

```
    "description": "TCP (flag ACK, PUSH)"
  },
  {
    "id": 5,
    "name": "in_tcp_r",
    "description": "TCP (flag RESET)"
  },
  {
    "id": 4,
    "name": "in_tcp_s",
    "description": "TCP (flag SYN)"
  },
  {
    "id": 8,
    "name": "in_tcp_sa",
    "description": "TCP (flag SYN, ACK)"
  },
  {
    "id": 2,
    "name": "in_udp",
    "description": "UDP"
  },
  {
    "id": 11,
    "name": "in_usr1",
    "description": "User #1"
  },
  {
    "id": 12,
    "name": "in_usr2",
    "description": "User #2"
  },
  {
    "id": 13,
    "name": "in_usr3",
    "description": "User #3"
  },
  {
    "id": 14,
    "name": "in_usr4",
    "description": "User #4"
  },
  {
    "id": 15,
    "name": "in_usr5",
    "description": "User #5"
  },
  {
```

```
"id": 1,  
  "name": "out",  
  "description": "Outgoing all"  
}  
]
```

10.2.3 Rules

10.2.3.1 Get all Rules

This method serves for obtaining all Rules. Return is an array of available Rules.

GET /rest/iad/rules/

- Return all Rules
- @return 200 Rules[] (10)

```
GET /rest/iad/rules/
```

Returns 200

```
[  
  {  
    "id": 29,  
    "name": "low rule",  
    "baselineSlots": 288,  
    "anomalyLength": 10,  
    "historyWeeks": 4,  
    "zeroBaseline": false,  
    "subRules": [  
      {  
        "id": 492,  
        "ruleId": 29,  
        "trafficType": {  
          "id": 11,  
          "name": "in_usr1",  
          "filter": ""  
        },  
        "detectionMethod": {
```

```
    "id": 3,
    "name": "MinrateAttack",
    "options": [
      "Pps",
      "Bps",
      "Pps;Bps"
    ]
  },
  "options": [
    {
      "id": 1692,
      "option": "Pps",
      "value": 15000000
    },
    {
      "id": 1693,
      "option": "Bps",
      "value": 15000000
    },
    {
      "id": 1694,
      "option": "Pps;Bps",
      "value": 0
    }
  ]
},
{
  "id": 493,
  "ruleId": 29,
  "trafficType": {
    "id": 19,
    "name": "in",
    "filter": ""
  },
  "detectionMethod": {
    "id": 3,
    "name": "MinrateAttack",
    "options": [
      "Pps",
      "Bps",
      "Pps;Bps"
    ]
  },
  "options": [
    {
      "id": 1695,
      "option": "Pps",
      "value": 15000000
    }
  ]
}
```

```
    },
    {
      "id": 1696,
      "option": "Bps",
      "value": 15000000
    },
    {
      "id": 1697,
      "option": "Pps;Bps",
      "value": 0
    }
  ]
},
"customBaselines": [
  {
    "id": 21,
    "name": "p1000_all",
    "protocol": "both",
    "portFrom": "1000",
    "portTo": "1000",
    "trafficType": {
      "id": 11,
      "name": "in_usr1"
    }
  }
]
}
```

10.2.3.2 Get one specific Rule

A method for acquisition of information about a specific Rule searched by its id.

GET /rest/iad/rules/<id>

- Return Rule
- @return 200 Rule (10)

```
GET /rest/iad/rules/29
```

Returns 200

```
{
  "id": 29,
  "name": "low rule",
  "baselineSlots": 288,
  "anomalyLength": 10,
  "historyWeeks": 4,
  "zeroBaseline": false,
  "subRules": [
    {
      "id": 492,
      "ruleId": 29,
      "trafficType": {
        "id": 11,
        "name": "in_usr1",
        "filter": ""
      },
      "detectionMethod": {
        "id": 3,
        "name": "MinrateAttack",
        "options": [
          "Pps",
          "Bps",
          "Pps;Bps"
        ]
      },
      "options": [
        {
          "id": 1692,
          "option": "Pps",
          "value": 15000000
        },
        {
          "id": 1693,
          "option": "Bps",
          "value": 15000000
        },
        {
          "id": 1694,
          "option": "Pps;Bps",
          "value": 0
        }
      ]
    },
    {
      "id": 493,
      "ruleId": 29,
```



```
    "trafficType": {
      "id": 19,
      "name": "in",
      "filter": ""
    },
    "detectionMethod": {
      "id": 3,
      "name": "MinrateAttack",
      "options": [
        "Pps",
        "Bps",
        "Pps;Bps"
      ]
    },
    "options": [
      {
        "id": 1695,
        "option": "Pps",
        "value": 15000000
      },
      {
        "id": 1696,
        "option": "Bps",
        "value": 15000000
      },
      {
        "id": 1697,
        "option": "Pps;Bps",
        "value": 0
      }
    ]
  },
  "customBaselines": [
    {
      "id": 21,
      "name": "p1000_all",
      "protocol": "both",
      "portFrom": "1000",
      "portTo": "1000",
      "trafficType": {
        "id": 11,
        "name": "in_usr1"
      }
    }
  ]
}
```

10.2.3.3 Add new Rule

A method for adding a new Rule. A new Rule must have empty id.

POST /rest/iad/rules/

- Return new Rule
- @return 201 Rule (10)

```
POST /rest/iad/rules/
BODY:
{
  "entity":
  {
    "id": null,
    "name": "New Rule by API",
    "baselineSlots": 2,
    "anomalyLength": 10,
    "historyWeeks": 2,
    "zeroBaseline": false,
    "subRules": [
      {
        "trafficType": {
          "id": 19,
          "name": "in"
        },
        "detectionMethod": {
          "id": 3,
          "name": "MinrateAttack"
        },
        "options": [
          {
            "option": "Pps",
            "value": 150000000
          },
          {
            "option": "Bps",
            "value": 150000000
          },
          {
            "option": "Pps;Bps",
            "value": 0
          }
        ]
      }
    ]
  }
}
```

```
        }
      ]
    }
  ],
  "customBaselines": []
}
}
```

Returns 201

```
{
  "id": 30,
  "name": "New Rule by API",
  "baselineSlots": 2,
  "anomalyLength": 10,
  "historyWeeks": 2,
  "zeroBaseline": false,
  "subRules": [
    {
      "id": 494,
      "ruleId": 30,
      "trafficType": {
        "id": 19,
        "name": "in",
        "filter": ""
      },
      "detectionMethod": {
        "id": 3,
        "name": "MinrateAttack",
        "options": [
          "Pps",
          "Bps",
          "Pps;Bps"
        ]
      },
      "options": [
        {
          "id": 1698,
          "option": "Pps",
          "value": 150000000
        },
        {
          "id": 1699,
          "option": "Bps",
          "value": 150000000
        },
        {
          "id": 1700,
          "option": "Pps;Bps",
```

```
        "value": 0
      }
    ]
  }
],
"customBaselines": []
}
```

10.2.3.4 Update existing Rule

A method for updating a Rule.

PUT /rest/iad/rules/

- Return updated Rule
- @return 200 Rule (10)

```
PUT /rest/iad/rules/
BODY:
{
  "entity":
  {
    "id": 30,
    "name": "Update Rule by API",
    "baselineSlots": 2,
    "anomalyLength": 10,
    "historyWeeks": 2,
    "zeroBaseline": false,
    "subRules": [
      {
        "trafficType": {
          "id": 19,
          "name": "in"
        },
        "detectionMethod": {
          "id": 3,
          "name": "MinrateAttack"
        },
        "options": [
          {
```

```
        "option": "Pps",
        "value": 190000000
      },
      {
        "option": "Bps",
        "value": 100000000
      },
      {
        "option": "Pps;Bps",
        "value": 1
      }
    ]
  },
  "customBaselines": []
}
}
```

Returns 200

```
{
  "id": 30,
  "name": "Update Rule by API",
  "baselineSlots": 2,
  "anomalyLength": 10,
  "historyWeeks": 2,
  "zeroBaseline": false,
  "subRules": [
    {
      "id": 495,
      "ruleId": 30,
      "trafficType": {
        "id": 19,
        "name": "in",
        "filter": ""
      },
      "detectionMethod": {
        "id": 3,
        "name": "MinrateAttack",
        "options": [
          "Pps",
          "Bps",
          "Pps;Bps"
        ]
      },
      "options": [
        {
          "id": 1701,
          "option": "Pps",
```

```
        "value": 1900000000
      },
      {
        "id": 1702,
        "option": "Bps",
        "value": 1000000000
      },
      {
        "id": 1703,
        "option": "Pps;Bps",
        "value": 1
      }
    ]
  },
  "customBaselines": []
}
```

10.2.3.5 Delete specific Rule

Deletes specific Rule by its id. Return code is 204 (No data).

DELETE /rest/iad/rules/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/rules/4
```

Returns 204

11 Groups

11.1 Data types

Group

```
Group = {  
  id: Integer,  
  name: String,  
  comment: String or null  
}
```

11.2 Endpoints

11.2.1 Groups

11.2.1.1 Get all groups

This method serve for obtaining all Groups. The return is an array of available Groups.

GET /rest/iad/groups

- Return all Groups
- @return 200 Group[]

```
GET /rest/iad/groups/
```

```
Return 200
```

```
[  
  {  
    "id": 1,  
    "name": "#1 group",  
    "comment": null  
  }  
]
```

11.2.1.2 Get one specific Group

A method for acquisition of information about a specific Group searched by its id.

GET /rest/iad/groups/<id>

- Return Group
- @return 200 Group (11)

```
GET /rest/iad/groups/1
```

```
Return 200
```

```
{
  "id": 1,
  "name": "#1 group",
  "comment": null
}
```

11.2.1.3 Create new group

A method for adding new Group. New group can create only user with admin privilege.

POST /rest/iad/groups/

- Return new Group
- @return 201 Group (11)

```
POST /rest/iad/groups/
```

```
BODY:
```

```
{
  "entity": {
    "name": "Group by API",
  }
}
```



```
        "comment": "Group for segments"
    }
}

Return 201
{
    "id": 2,
    "name": "Group by API",
    "comment": "Group for segments"
}
```

11.2.1.4 Update existing Group

A method for updating a Group.

PUT /rest/iad/groups/

- Return updated Group
- @return 200 Group (11)

```
PUT /rest/iad/groups/
BODY:
{
    "entity": {
        "id": 2,
        "name": "Group by API UPDATED",
        "comment": "and updated comment"
    }
}

Return 200
{
    "id": 2,
    "name": "Group by API UPDATED",
    "comment": "and updated comment"
}
```

11.2.1.5 Delete specific group

Delete a specific Group by its id. Return code is 204 (No data).

DELETE /rest/iad/groups/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/groups/2
```

```
Returns 204
```

12 Segment

12.1 Data Types

Segment

```
Segment = {
  id: Integer,
  name: String,
  rule: Rule({id, name}) or null (for allways on),
  timeout: Integer("Positive integer, count of seconds, use -1 as "Never"),
  autoSubnet: Integer("minimal IPv4 subnet mask") or null,
  autoSubnetIpv6: Integer("minimal IPv6 subnet mask") or null,
  bandwidthDay: Integer("Prumerny bandwidth za predchozi den bez utoku TRANSLATE"),
  trafficSeen: Timestamp("First segment traffic"),
  status: [String] << [succ, error, warning],
  bandwidth: Integer("max bandwidth in bps"),
  bandwidthAuto: Boolean("is bandwidth determined automatically"),
  actionSubnet: ActionSubnets << ["subnets", "preferred", "bgp_community", "automatic"],
  type: SegmentTypes("subnets or as numbers") << [1, 2],
  asNumbers: [Integer] or null("AS numbers"),
  subnets: [String("Masked IP")],
  preferredSubnets: String[]("Masked IP"),
  enabled: Boolean,
  note: String,
  group: Group({id, name}) or null,
  selectedSources: [String] (id of source or '*' for all),
  whitelists: [Whitelist],
  sourceInterfaces: [SourceInterface],

  //measures to be taken to suppress attack
  measures: {
    mode: String("How actions should be triggered") << ["manual", "auto"],

    alert: Alert({id, name}, "Alert to be triggered") or null,
    reroute: [SegmentRouter, "Router to change route"] or null,
    mitigate: {
      scrubbingCenter: ScrubbingCenter({id,name}, "Scrubbing center to mittigate attack") or null
      *parameters: ScrubbingCenterParameter[] ( {"option", "value"} "Array of scrubbing center parameters,
                                                    available only if scrubbing center is
                                                    one of Vission scrubinnng centers."),
    }
  }
}

SegmentRouter = {
```

```
sommunity: String,
bgpMode: String or null,
tieringEnabled: Boolean,
tieringCommunity: String,
tieringThreshold: Integer,
tieringMaskIp4: Integer or null,
tieringMaskIp6: Integer or null,
routerId: Integer (Id of parent router),
routerName: String (Name of parent router)
}

CodeList = {
  id: Integer|String("Item id"),
  name: String("Item name")
}

SegmentTypes extends CodeList << [
  {id: 1, name: "Subnets"},
  {id: 2, name: "As numbers"}
]

ActionSubnets extends CodeList << [
  {id: "subnet", name: "Subnets"},
  {id: "preferred", name: "Preferred subnets"},
  {id: "bgp_community", name: "By BGP Community"},
  {id: "automatic", name: "Autodetected subnets"}
]

Whitelist = {
  id: Integer,
  whitelist: Whitelist({id, name})
}

SourceInterface = {
  source: String(id of source),
  direction: String << ["in", "out"],
  index: Integer
}
```

12.2 Endpoints

12.2.1 Segments

12.2.1.1 Get all Segments

This method serve for obtaining all Segments. The return is an array of available Segments.

GET /rest/iad/segments/

- Return all Segments
- @return 200 Segments[] (12)

```
GET /rest/iad/segments/
```

```
Returns 200
```

```
[
  {
    "id": 10,
    "name": "druhy",
    "rule": {
      "id": 1,
      "name": "Default rule"
    },
    "timeout": 120,
    "autoSubnet": null,
    "autoSubnetIPv6": null,
    "bandwidthDay": null,
    "trafficSeen": "2017-11-29 13:24:30",
    "status": [],
    "bandwidth": 0,
    "bandwidthAuto": false,
    "actionSubnet": "subnets",
    "type": 1,
    "asNumbers": null,
    "subnets": [
      "192.168.50.207/21"
    ],
    "preferredSubnets": [
```

```
    "12.15.16.10/32",
    "12.15.16.11/32"
  ],
  "whitelists": [],
  "selectedSources": [
    {
      "id": "127-0-0-1_p3000",
      "name": "127.0.0.1 (localhost)"
    }
  ],
  "sourceInterfaces": [],
  "measures": {
    "mode": "auto",
    "alert": null,
    "reroute": [
      {
        "router": {
          "id": 6,
          "name": "First",
          "osci": "exabgp",
          "community": ""
        },
        "community": null,
        "bgpMode": null
      }
    ],
    "mitigate": {
      "scrubbingCenter": {
        "id": 3,
        "name": "my APSolute Vision 3.00"
      },
      "parameters": [
        {
          "option": "dns_a_in_ipv4",
          "value": "72"
        },
        {
          "option": "dns_a_in_ipv6",
          "value": "72"
        },
        ...
        {
          "option": "con_limit_udp",
          "value": "30000"
        }
      ]
    }
  ]
}
```

```
    }
  }
},
{
  "id": 11,
  "name": "VivaLasVegas",
  "rule": {
    "id": 4,
    "name": "Baseline"
  },
  "timeout": 120,
  "autoSubnet": null,
  "autoSubnetIPv6": null,
  "bandwidthDay": null,
  "trafficSeen": "2017-11-24 11:04:30",
  "status": [],
  "bandwidth": 0,
  "bandwidthAuto": false,
  "actionSubnet": "subnets",
  "type": 1,
  "asNumbers": null,
  "subnets": [
    "12.2.2.2/32"
  ],
  "preferredSubnets": [],
  "whitelists": [],
  "selectedSources": [
    {
      "id": "127-0-0-1_p3000",
      "name": "127.0.0.1 (localhost)"
    }
  ],
  "sourceInterfaces": [
    {
      "source": "127-0-0-1_p3000",
      "direction": "in",
      "index": 8
    }
  ],
  "measures": {
    "mode": "manual",
    "alert": null,
    "reroute": null,
    "mitigate": {
      "scrubbingCenter": null,
      "parameters": null
    }
  }
}
```

```
    }  
  ]
```

12.2.1.2 Get one specific Segment

A method for acquisition of informations about a specific Segment searched by its id.

GET /rest/iad/segments/<id>

- Return Segment
- @return 200 Segment (12)

```
GET /rest/iad/segments/10
```

Returns 200

```
{  
  "id": 10,  
  "name": "Static",  
  "rule": {  
    "id": 28,  
    "name": "Static w/o",  
    "zeroBaseline": true  
  },  
  "timeout": 5,  
  "autoSubnet": null,  
  "autoSubnetIPv6": null,  
  "bandwidthDay": 0,  
  "trafficSeen": null,  
  "status": [],  
  "bandwidth": 1239,  
  "bandwidthAuto": true,  
  "actionSubnet": "subnets",  
  "type": 1,  
  "asNumbers": null,  
  "subnets": [  
    "10.20.20.45/32"  
  ],  
  "flowspecAction": {  
    "id": 19,  
    "action": "accept",
```



```
    "option": null
  },
  "preferredSubnets": [],
  "enabled": true,
  "note": "",
  "group": {
    "id": 1,
    "name": "#1 group"
  },
  "selectedSources": [
    "*"
  ],
  "whitelists": [],
  "sourceInterfaces": [],
  "measures": {
    "mode": "manual_auto",
    "alert": {
      "id": 1,
      "name": "Default alert"
    },
    "reroute": [
      {
        "id": 10,
        "segmentId": 10,
        "community": null,
        "bgpMode": null,
        "tieringEnabled": false,
        "tieringCommunity": "",
        "tieringThreshold": 0,
        "tieringMaskIp4": null,
        "tieringMaskIp6": null,
        "routerId": 7,
        "routerName": "iBGP DP ipv4"
      }
    ],
    "mitigate": {
      "scrubbingCenter": {
        "id": 2,
        "name": "DefensePro Vision 4.0"
      },
      "parameters": []
    }
  }
}
```

12.2.1.3 Add new Segment

A method for adding new Segment. A new Segment must have without ID (i.e. must be empty or unused).

POST /rest/iad/segments/

- Return new Segment
- @return 201 Segment (12)

```
POST /rest/iad/segments/
BODY:
{
  "entity":
  {
    "name": "Segment by API",
    "rule": {
      "id": 22,
      "name": "Adaptive threshold all"
    },
    "timeout": 120,
    "autoSubnet": null,
    "autoSubnetIPv6": null,
    "bandwidthDay": null,
    "bandwidth": 0,
    "status": [],
    "bandwidthAuto": false,
    "actionSubnet": "subnets",
    "type": 1,
    "asNumbers": [],
    "subnets": [
      "12.15.16.32/21"
    ],
    "preferredSubnets": null,
    "selectedSources": [
      {
        "id": "127-0-0-1_p3000",
        "name": "127.0.0.1 (localhost)"
      }
    ],
    "sourceInterfaces": [
      {
```

```
        "source": "127-0-0-1_p3000",
        "direction": "in",
        "index": 9
      }
    ],
    "measures": {
      "mode": "manual",
      "alert": null,
      "reroute": null,
      "mitigate": null
    }
  }
}
Returns 201
{
  "id": 15,
  "name": "Segment by API",
  "rule": {
    "id": 22,
    "name": "Adaptive threshold all",
    "zeroBaseline": true
  },
  "timeout": 120,
  "autoSubnet": null,
  "autoSubnetIPv6": null,
  "bandwidthDay": null,
  "trafficSeen": null,
  "status": [],
  "bandwidth": 0,
  "bandwidthAuto": false,
  "actionSubnet": "subnets",
  "type": 1,
  "asNumbers": [],
  "subnets": [
    "12.15.16.32/21"
  ],
  "flowspecAction": null,
  "preferredSubnets": [],
  "enabled": true,
  "note": "",
  "group": null,
  "selectedSources": [
    {
      "id": "127-0-0-1_p3000",
      "name": "127.0.0.1 (localhost)"
    }
  ],
  "sourceInterfaces": [
```

```
{
  "source": "127-0-0-1_p3000",
  "direction": "in",
  "index": 9
}
],
"whitelists": [],
"measures": {
  "mode": "manual",
  "alert": null,
  "reroute": [],
  "mitigate": {
    "scrubbingCenter": null,
    "parameters": []
  }
}
}
```

12.2.1.4 Update existing Segment

A method for updating a Segment.

PUT /rest/iad/segments/

- Return updated Segment
- @return 200 Segment (12)

```
PUT /rest/iad/segments/
BODY:
{
  "entity":
  {
    "id": 15,
    "name": "Segment by API",
    "rule": {
      "id": 22,
      "name": "Adaptive threshold all",
      "zeroBaseline": true
    },
    "timeout": 120,
```

```
"autoSubnet": null,
"autoSubnetIPv6": null,
"bandwidthDay": null,
"trafficSeen": null,
"status": [],
"bandwidth": 0,
"bandwidthAuto": false,
"actionSubnet": "subnets",
"type": 1,
"asNumbers": [],
"subnets": [
  "12.15.16.32/21"
],
"flowspecAction": null,
"preferredSubnets": [],
"enabled": true,
"note": "",
"group": {
  "id": 1,
  "name": "#1 group"
},
"selectedSources": [
  "127-0-0-1_p3000",
  "127-0-4-72_p3000",
  "127-0-4-242_p3000"
],
"whitelists": [],
"sourceInterfaces": [
  {
    "source": "127-0-4-242_p3000",
    "direction": "in",
    "index": 9
  },
  {
    "source": "127-0-4-242_p3000",
    "direction": "in",
    "index": 8
  }
],
"measures": {
  "mode": "manual",
  "alert": null,
  "reroute": [],
  "mitigate": {
    "scrubbingCenter": null,
    "parameters": []
  }
}
}
```

```
    }  
  }  
  
Returns 200  
{  
  "id": 15,  
  "name": "Segment by API",  
  "rule": {  
    "id": 22,  
    "name": "Adaptive threshold all",  
    "zeroBaseline": true  
  },  
  "timeout": 120,  
  "autoSubnet": null,  
  "autoSubnetIPv6": null,  
  "bandwidthDay": null,  
  "trafficSeen": null,  
  "status": [],  
  "bandwidth": 0,  
  "bandwidthAuto": false,  
  "actionSubnet": "subnets",  
  "type": 1,  
  "asNumbers": [],  
  "subnets": [  
    "12.15.16.32/21"  
  ],  
  "flowspecAction": null,  
  "preferredSubnets": [],  
  "enabled": true,  
  "note": "",  
  "group": {  
    "id": 1,  
    "name": "#1 group"  
  },  
  "selectedSources": [  
    {  
      "id": "127-0-0-1_p3000",  
      "name": "127.0.0.1 (localhost)"  
    },  
    {  
      "id": "127-0-4-72_p3000",  
      "name": "cypress.example.com"  
    },  
    {  
      "id": "127-0-4-242_p3000",  
      "name": "242.example.com"  
    }  
  ],  
}
```

```
"whitelists": [],
"sourceInterfaces": [
  {
    "source": "127-0-4-242_p3000",
    "direction": "in",
    "index": 9
  },
  {
    "source": "127-0-4-242_p3000",
    "direction": "in",
    "index": 8
  }
]
"measures": {
  "mode": "manual",
  "alert": null,
  "reroute": [],
  "mitigate": {
    "scrubbingCenter": null,
    "parameters": []
  }
}
}
```

12.2.1.5 Delete specific Segment

Delete a specific Segment by its id. Return code is 204 (No data).

DELETE /rest/iad/segments/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/segments/22
```

```
Returns 204
```

13 Whitelists

13.1 Data types

Whitelists

```
Whitelist = {
  id: Integer,
  name: String,
  sendAlert: Boolean,
  bandwidth: Integer|null,
  whitelistedSegments: WhitelistedSegment[],
  whitelistedGroups: WhitelistedGroup[],
  networks: Network[],
  asns: Asn[],
  whitelistedTrafficTypes: WhitelistedTrafficType[],
  recurrence: Recurrence,
  fromTime: String,
  toTime: String
}

WhitelistedSegment = {
  id: Integer,
  name: String
}

WhitelistedGroup = {
  id: Integer,
  name: String
}

network = {
  id: Integer,
  network: IP(with mask),
  direction: String(src|dst)
}

asn = {
  id: Integer,
  asn: Integer,
  direction: String(src|dst)
}

WhitelistedTrafficType = {
  id: Integer,
  name: String
}
```



```
}

Recurrence = {
  startOn: WeekDay|MonthDay|null,
  stopOn: WeekDay|Month|null,
  period: String(week_day|month_day|every_day)
}

WeekDay = {
  id: String,
  name: String
}

MonthDay = {
  id: Integer,
  name: String
}
```

13.2 Endpoints

13.2.1 Whitelists

13.2.1.1 Get all whitelists

This method serve for obtaining all Whitelists. The return is an array of available Whitelists.

GET /rest/iad/whitelists

- Return all Whitelists
- @return 200 Whitelist[]

```
GET /rest/iad/whitelists/
```

```
Return 200
```

```
[
  {
    "id": 1,
```

```
"name": "day",
"sendAlert": true,
"bandwidth": null,
"whitelistedSegments": [
  {
    "id": 122,
    "name": "low segment"
  },
  {
    "id": 101,
    "name": "second"
  }
],
"whitelistedGroups": [],
"networks": [
  {
    "id": 189,
    "network": "192.168.15.10/32",
    "direction": "src"
  },
  {
    "id": 190,
    "network": "192.168.120.10/32",
    "direction": "src"
  }
],
"asns": [],
"whitelistedTrafficTypes": [
  {
    "id": 19,
    "name": "in"
  }
],
"recurrence": {
  "startOn": {
    "id": "Mon",
    "name": "Monday"
  },
  "stopOn": {
    "id": "Fri",
    "name": "Friday"
  },
  "period": "week_day"
},
"fromTime": "06:00",
"toTime": "14:00"
}
```

]

13.2.1.2 Get one specific Whitelist

A method for acquisition of information about a specific Whitelist searched by its id.

GET /rest/iad/whitelists/<id>

- Return Whitelist
- @return 200 Whitelist (13)

```
GET /rest/iad/whitelists/1

Return 200

{
  "id": 1,
  "name": "day",
  "sendAlert": true,
  "bandwidth": null,
  "whitelistedSegments": [
    {
      "id": 122,
      "name": "low segment"
    },
    {
      "id": 101,
      "name": "second"
    }
  ],
  "whitelistedGroups": [],
  "networks": [
    {
      "id": 189,
      "network": "192.168.15.10/32",
      "direction": "src"
    },
    {
      "id": 190,
      "network": "192.168.120.10/32",
      "direction": "src"
    }
  ],
}
```

```
"asns": [],
"whitelistedTrafficTypes": [
  {
    "id": 19,
    "name": "in"
  }
],
"recurrence": {
  "startOn": null,
  "stopOn": null,
  "period": "every_day"
},
"fromTime": "06:00",
"toTime": "14:00"
}
```

13.2.1.3 Create new whitelist

A method for adding new Whitelist.

POST /rest/iad/whitelists/

- Return new Whitelist
- @return 201 Whitelist (13)

```
POST /rest/iad/whitelists/
BODY:
{
  "entity":
  {
    "id": null,
    "name": "#test 2",
    "sendAlert": false,
    "whitelistedSegments": [],
    "whitelistedGroups": [
      {
        "id": null,
        "name": "Ungrouped"
      }
    ]
  }
},
```

```
"networks": [
  {
    "id": 22,
    "network": "12.13.45.45/32",
    "direction": "src"
  }
],
"asns": [],
"whitelistedTrafficTypes": [
  {
    "id": 19,
    "name": "in"
  }
],
"recurrence": {
  "startOn": {
    "id": 20,
    "name": "20."
  },
  "stopOn": {
    "id": 20,
    "name": "2."
  },
  "period": "month_day"
},
"fromTime": "20:00",
"toTime": "19:00"
}
}
```

Return 201

```
{
  "id": 10,
  "name": "#test 2",
  "sendAlert": false,
  "bandwidth": null,
  "whitelistedSegments": [],
  "whitelistedGroups": [
    {
      "id": null,
      "name": "Ungrouped"
    }
  ],
  "networks": [
    {
      "id": 9,
      "network": "12.13.45.45/32",
      "direction": "src"
    }
  ]
}
```

```
    }
  ],
  "asns": [],
  "whitelistedTrafficTypes": [
    {
      "id": 19,
      "name": "in"
    }
  ],
  "recurrence": {
    "start0n": {
      "id": 20,
      "name": "20."
    },
    "stop0n": {
      "id": 20,
      "name": "20."
    },
    "period": "month_day"
  },
  "fromTime": "20:00",
  "toTime": "19:00"
}
```

13.2.1.4 Update existing Whitelist

A method for updating a Whitelist.

PUT /rest/iad/whitelists/

- Return updated Whitelist
- @return 200 Whitelist (13)

```
PUT /rest/iad/whitelists/
BODY:
{
  "entity":
  {
    "id": 10,
    "name": "#test 2",
```

```
"sendAlert": false,
"bandwidth": null,
"whitelistedSegments": [
  {
    "id": 3,
    "name": "1.1.1.1"
  }
],
"whitelistedGroups": [
  {
    "id": null,
    "name": "Ungrouped"
  }
],
"networks": [
  {
    "id": 9,
    "network": "12.13.45.45/32",
    "direction": "src"
  }
],
"asns": [],
"whitelistedTrafficTypes": [
  {
    "id": 19,
    "name": "in"
  }
],
"recurrence": {
  "start0n": {
    "id": 20,
    "name": "20."
  },
  "stop0n": {
    "id": 20,
    "name": "20."
  },
  "period": "month_day"
},
"fromTime": "20:00",
"toTime": "19:00"
}

Return 200
{
  "id": 10,
  "name": "#test 2",
```

```
"sendAlert": false,
"bandwidth": null,
"whitelistedSegments": [
  {
    "id": 3,
    "name": "1.1.1.1"
  }
],
"whitelistedGroups": [
  {
    "id": null,
    "name": "Ungrouped"
  }
],
"networks": [
  {
    "id": 11,
    "network": "12.13.45.45/32",
    "direction": "src"
  }
],
"asns": [],
"whitelistedTrafficTypes": [
  {
    "id": 19,
    "name": "in"
  }
],
"recurrence": {
  "startOn": {
    "id": 20,
    "name": "20."
  },
  "stopOn": {
    "id": 20,
    "name": "20."
  },
  "period": "month_day"
},
"fromTime": "20:00",
"toTime": "19:00"
}
```


13.2.1.5 Delete specific whitelist

Delete a specific Whitelist by its id. Return code is 204 (No data).

DELETE /rest/iad/whitelists/<id>

- Return empty answer
- @return 204 No data

```
DELETE /rest/iad/whitelists/1
```

Returns 204

13.2.1.6 Get available TrafficTypes

GET /rest/iad/whitelists/traffic-types

- Return available TrafficTypes
- @return 200

```
GET /rest/iad/whitelists/traffic-types
```

Return 200

```
[
  {
    "id": 19,
    "name": "in"
  },
  {
    "id": 16,
    "name": "in_dns"
  },
]
```

```
{
  "id": 17,
  "name": "in_http"
},
{
  "id": 18,
  "name": "in_https"
},
{
  "id": 10,
  "name": "in_icmp"
},
{
  "id": 3,
  "name": "in_tcp"
},
{
  "id": 6,
  "name": "in_tcp_a"
},
{
  "id": 7,
  "name": "in_tcp_af"
},
{
  "id": 9,
  "name": "in_tcp_ap"
},
{
  "id": 5,
  "name": "in_tcp_r"
},
{
  "id": 4,
  "name": "in_tcp_s"
},
{
  "id": 8,
  "name": "in_tcp_sa"
},
{
  "id": 2,
  "name": "in_udp"
},
{
  "id": 21,
  "name": "my HTTPS"
},
}
```

```
{
  "id": 1,
  "name": "out"
}
]
```

13.2.1.7 Get available Week days

GET /rest/iad/whitelists/week-days

- Return available WeekDays
- @return 200

```
GET /rest/iad/whitelists/week-days
```

```
Return 200
```

```
[
  {
    "id": "Mon",
    "name": "Monday"
  },
  {
    "id": "Tue",
    "name": "Tuesday"
  },
  {
    "id": "Wed",
    "name": "Wednesday"
  },
  {
    "id": "Thu",
    "name": "Thursday"
  },
  {
    "id": "Fri",
    "name": "Friday"
  },
  {
    "id": "Sat",
    "name": "Saturday"
  }
]
```

```
  },  
  {  
    "id": "Sun",  
    "name": "Sunday"  
  }  
]
```

13.2.1.8 Get available Month days

GET /rest/iad/whitelists/month-days

- Return available MonthDays
- @return 200

```
GET /rest/iad/whitelists/month-days
```

```
Return 200
```

```
[  
  {  
    "id": 1,  
    "name": "1."  
  },  
  {  
    "id": 2,  
    "name": "2."  
  },  
  ...  
  {  
    "id": 27,  
    "name": "27."  
  },  
  {  
    "id": 28,  
    "name": "28."  
  },  
  {  
    "id": 29,  
    "name": "29."  
  },  
  {
```

```
    "id": 30,  
    "name": "30."  
  },  
  {  
    "id": 31,  
    "name": "31."  
  }  
]
```

14 REST API changes

Version 4.0

- added Attacks methods (3)
- added Alerts methods (4)
- added Routers methods (5)
- added Scrubbing Center methods (6)
- added Report Chapters methods (8)
- added EmailTempaltes methods (9)
- added Rules methods (10)
- added Segments methods (12)

Version 5.0

- added Groups method (11)

Version 5.2

- added Whitelists methods (13)

Contacts

FLOWMON NETWORKS a.s.
Sochorova 3232/34
Brno 61600

Web: www.flowmon.com
Email: info@flowmon.com
Tel.: +420 530 510 600

Feedback

We would be pleased if you tell us your comments to this text (typing errors, incomplete or unclear information). Please, contact us via email support@flowmon.com.

Copyright

Except as stated here, none of the document may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of INVEA-TECH. Any unauthorized use of this specification may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

INVEA-TECH, the company logo, and other designated brands included herein are trademarks of INVEA-TECH a.s. All other trademarks are the property of their respective owners.

This product uses NfSen and NFDUMP software Copyright (c) 2004, SWITCH - Teleinformatikdienste fuer Lehre und Forschung.

This product uses compress library zlib Copyright (C) 1995-1998 Jean-loup Gailly and Mark Adler.

This product uses library libpcap Copyright (c) 1993, 1994, 1995, 1996, 1997, 1998 The Regents of the University of California and Copyright (c) 1999 - 2003 NetGroup, Politecnico di Torino (Italy).

This product includes sFlow(TM), freely available from <http://www.inmon.com/>.

Copyright (c) 2007 - 2014 INVEA-TECH a.s. All rights reserved.