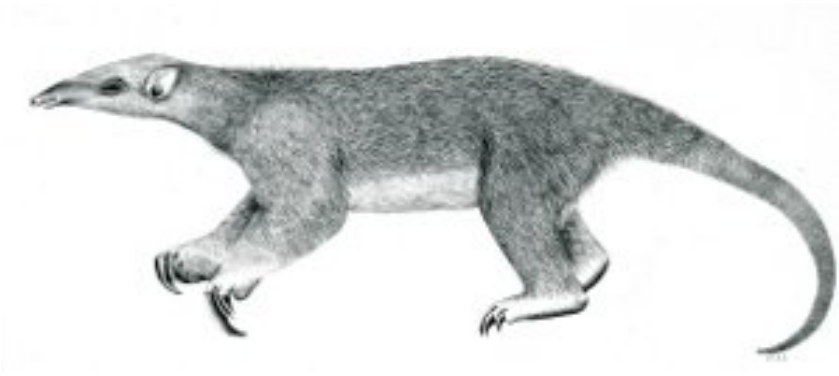
Tranalyzer2

ospfDecode

Open Shortest Path First (OSPF)



Tranalyzer Development Team

Contents

1 ospfDecode

1.1 Description

This plugin analyzes OSPF traffic and provides absolute and relative statistics to the `PREFIX_ospfStats.txt` file. In addition, the `rospf` script extracts the areas, networks and netmasks, along with the routers and their interfaces (Section 1.5).

1.2 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description
`OSPF_OUTPUT_DBD`	0	Output routing tables
`OSPF_OUTPUT_MSG`	0	Output all messages
`OSPF_MASK_AS_IP`	0	How to display netmasks: 0: hex, 1: IP
`OSPF_AREA_AS_IP`	0	How to display areas: 0: int, 1: IP, 2: hex

1.3 Flow File Output

The ospfDecode plugin outputs the following columns:

Column	Type	Description
`ospfStat`	H8	Status
`ospfType`	H8	Message type
`ospfAuType`	H16	Authentication type
`ospfAuPass`	RS	Authentication password (if `ospfAuType == 0x4`)
`ospfArea`	U32/H32	Area ID (see `OSPF_AREA_AS_IP` in Section 1.2)

1.3.1 ospfStat

The hex based status variable (`ospfStat`) is defined as follows:

ospfStat	Description
2^0 (=`0x01`)	OSPF message had invalid TTL ($\neq 1$)
2^1 (=`0x02`)	OSPF message had invalid destination
2^2 (=`0x04`)	OSPF message had invalid type
2^3 (=`0x08`)	OSPF message had invalid checksum
2^4 (=`0x10`)	OSPF message was malformed

The invalid checksum status `0x08` is currently not implemented.
The malformed status `0x10` is currently used to report cases such as possible covert channels, e.g., `authfield` used when `auType` was `NULL`.

1.3.2 ospfType

The hex based message type variable `ospfType` is defined as follows:

ospfType	Description
2^1 (=0x02)	Hello
2^2 (=0x04)	Database Description
2^3 (=0x08)	Link State Request
2^4 (=0x10)	Link State Update
2^5 (=0x20)	Link State Acknowledgement

1.3.3 ospfAuType

The hex based authentication type variable `ospfAuType` is defined as follows:

ospfAuType	Description
2^1 (=0x0002)	Null authentication
2^2 (=0x0004)	Simple password
2^3 (=0x0008)	Cryptographic authentication

1.4 Additional Output

- `PREFIX_ospfStats.txt`: global statistics about OSPF traffic
- `PREFIX_ospfHello.txt` Hello messages (see Section 1.5)
- `PREFIX_ospfDBD.txt`: Routing tables (see Section 1.2)
- `PREFIX_ospfMsg.txt`: All other messages (see Section 1.2)

1.5 Post-Processing

1.5.1 rospf

Hello messages can be used to discover the network topology and are stored in the `PREFIX_ospfHello.txt` file. The script `rospf` extracts the areas, networks, netmasks, routers and their interfaces:

```
./scripts/rospf PREFIX_ospfHello.txt
```

1.5.2 dbd

If `OSPF_OUTPUT_DBD` is activated (Section 1.2), database description messages are stored in a file `PREFIX_ospfDBD.txt`. The `dbd` script formats this file to produce an output similar to that of standard routers:

```
./scripts/dbd PREFIX_ospfDBD.txt
```

```
Name    Area    Network          Netmask
N1      0       192.168.21.0     0xffffff00
N2      1       192.168.16.0     0xffffff00
N3      1       192.168.22.0     0xfffffffc
...

Router    Interface_n      Network_n
R1        192.168.22.29    N11    192.168.21.4    N5    192.168.22.25    N10
R2        192.168.22.5     N12    192.168.16.1    N0    192.168.22.1     N6
R3        192.168.22.10    N13    192.168.21.2    N5    192.168.22.6     N12
...

Router    Connected Routers
R0        R2    R4    R6    R7    R8
R1        R2    R4
R2        R0    R1    R4    R8
...
```

```
OSPF Router with ID (192.168.22.10)

Router Link States (Area 1)

Link ID          ADV Router       Age    Seq#          Checksum
192.168.22.5     192.168.22.5     4      0x80000002    0x38ce
192.168.22.10    192.168.22.10    837    0x80000002    0x6b0f
192.168.22.9     192.168.22.9     837    0x80000002    0x156c

Net Link States (Area 1)

Link ID          ADV Router       Age    Seq#          Checksum
192.168.22.6     192.168.22.10    4      0x80000001    0x150b
192.168.22.9     192.168.22.9     838    0x80000001    0x39e0

Summary Net Link States (Area 1)

Link ID          ADV Router       Age    Seq#          Checksum
192.168.17.0     192.168.22.9     735    0x80000001    0x5dd9
192.168.17.0     192.168.22.10    736    0x80000001    0x57de
192.168.18.0     192.168.22.9     715    0x80000001    0x52e3
...
```