Managing and Securing Computer Networks

Project Report

General overview:

The application provided queries switches and routers in a network using SNMP(Simple Network Management Protocol). The program uses SNMP4J library for Java to gather information about the devices.

The program consists of a main class Grabber and several auxiliary classes.

- the Grabber collects the information on the devices using Lexer and Parser, then sends the data to SnmpAgent to provide the required output
- the Lexer, Parser and Sym classes are generated by Jflex and Cup and are used for parsing the configuration file
- helper class ConfigFile stores two List<SnmpAgent> -- one for switches, another for routers.Both could be initialized with or without credentials, v1, v2c, v3 as well
- We create only one connection for all requests, TransportMapping and Snmp objects, then we reuse them to retrieve all required data
- Instantiate Switch and Router objects from appropriate lists, invoke discover()method and print result on standart output
- Switch contains a field for Vlans, which are simple data structure, used to represent grabbed data. The same about classes Iface and Route, when Router is asked about interfaces and routes
- Other auxiliary classes: SnmpConstants – a Helper class, containing required OIDs for making queries; Utils – a utilities class

Example of the application output:

```
Switches:
139.165.222.2/161 v1 public
139.165.223.254/161 v2c public
Routers:
139.165.222.5/161 v2c Zork!69RUN
139.165.222.161/161 v2c Zork!69RUN
139.165.222.193/161 v2c Zork!69RUN

switch falcon {
  vlan 3 222.160/27 { 3 15 16 24 }
  vlan 2 222.128/27 { 1 2 }
  vlan 1 "Default VLAN" { 4 5 6 7 8 9 12 13 17 18 19 20 21 22 23 25 26 27 28
29 30 }
  vlan 4 222.192/27 { 10 11 14 }
  vlan 8 "223.0/24 2230::/64" {  }
}

switch vulture {
  vlan 1  { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 29 30 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50 56 }
}

router hawk.run.montefiore.ulg.ac.be {
  interface 139.165.222.129(<UNKNOWN>) 255.255.255.224 00:07:85:a8:83:21
  interface <UNKNOWN>(<UNKNOWN>) <UNKNOWN>
  interface 139.165.222.5(<UNKNOWN>) 255.255.255.128 00:07:85:a8:83:20
  route 139.165.222.0 255.255.255.128 139.165.222.5(<UNKNOWN>)
  route 139.165.222.192 255.255.255.224 139.165.222.193(<UNKNOWN>)
  route 0.0.0.0 0.0.0.0 139.165.222.1(<UNKNOWN>)
  route 139.165.222.160 255.255.255.224 139.165.222.130(<UNKNOWN>)
  route 139.165.222.128 255.255.255.224 139.165.222.129(<UNKNOWN>)
}

router owl.run.montefiore.ulg.ac.be {
  interface 139.165.222.161(<UNKNOWN>) 255.255.255.224 00:09:43:3a:06:41
  interface <UNKNOWN>(<UNKNOWN>) <UNKNOWN>
  interface 139.165.222.130(<UNKNOWN>) 255.255.255.224 00:09:43:3a:06:40
  route 139.165.222.0 255.255.255.128 139.165.222.129(<UNKNOWN>)
  route 139.165.222.192 255.255.255.224 139.165.222.129(<UNKNOWN>)
  route 0.0.0.0 0.0.0.0 139.165.222.129(<UNKNOWN>)
  route 139.165.222.160 255.255.255.224 139.165.222.161(<UNKNOWN>)
  route 139.165.222.128 255.255.255.224 139.165.222.130(<UNKNOWN>)
}

router sparrow.run.montefiore.ulg.ac.be {
  interface <UNKNOWN>(<UNKNOWN>) <UNKNOWN>
  interface 139.165.222.193(<UNKNOWN>) 255.255.255.224 00:b0:64:34:0f:c0
  route 139.165.222.0 255.255.255.128 139.165.222.5(<UNKNOWN>)
  route 139.165.222.192 255.255.255.224 139.165.222.193(<UNKNOWN>)
  route 0.0.0.0 0.0.0.0 139.165.222.5(<UNKNOWN>)
  route 139.165.222.160 255.255.255.224 139.165.222.5(<UNKNOWN>)
  route 139.165.222.128 255.255.255.224 139.165.222.5(<UNKNOWN>)
}
```