```
--
-- PyIDS Formal Specification
--
-- pyIDS_fs.use
--
-- This is a formal specification for the
-- Python Intrusion Detection System (PyIDS).
--
--
-- Author: Blake Knedler
-- Date: September 11, 2016
--

model PyIDS

-- ===========================================================
-- E N U M E R A T I O N S
-- ===========================================================

-- Enumeration list of packet types
enum PacketType {Valid, Malicious}



--
-- C L A S S E S
--

-- Packet class
-- - basic definition of what a packet will be
class Packet
attributes
        data : String
        type : PacketType
end

-- PacketReader class
-- - will read the packets as they are received
class PacketReader
attributes
        currentPacket : Packet
operations
        readPacket(p : Packet) : Boolean =
                self.currentPacket = p
end

-- NeuralNetwork class
-- - will make decisions based on the validity of the packet
class NeuralNetwork
attributes
        decision : Boolean
operations
        -- Decision will be true if the packet is malicious
        makeDecision(p : Packet) : Boolean =
                if (p.type = #Malicious) then
                        true
                else
                        false
                endif
end

-- Notifier class
-- - will notify user of the malicious packets
class Notifier
attributes
        notification : String
        numberOfMaliciousPackets : Integer
end


--
-- A S S O C I A T I O N S
--
```

```
-- Association relating the reader (PacketReader)
--      to reading Packets
association ReadPackets between
        PacketReader [1] role reader
        Packet [*] role packet
end

-- Association relating the reader (PacketReader)
--      and decision maker (NeuralNetwork)
association DetermineDecision between
        PacketReader [1] role reader
        NeuralNetwork [1] role decider
end

-- Association relating the decision maker (NeuralNetwork)
--      to the notificaiton system (Notifier)
association Notifications between
        NeuralNetwork [1] role decider
        Notifier [1] role notifier
end


--
-- C O N S T R A I N T S
--

constraints

-- There is only one Packet Reader
context PacketReader
        inv Only_One_PacketReader:
                PacketReader.allInstances->size = 1

-- There is only one Neural Network
context NeuralNetwork
```

```
        inv Only_One_NeuralNetwork:

                NeuralNetwork.allInstances->size = 1


-- There is only one Notification system

context Notifier

        inv Only_One_Notifier:

                Notifier.allInstances->size = 1


-- Invariant to report a malicious packet

context p:Packet

        inv ReportMalicousPacket:

                (NeuralNetwork.allInstances)->forAll(nn |

                ( if nn.decision = true then

                        nn.notifier.notification = p.data and

                        nn.notifier.numberOfMaliciousPackets = nn.notifier.numberOfMaliciousPackets + 1

                else

                        nn.notifier.numberOfMaliciousPackets = nn.notifier.numberOfMaliciousPackets

                endif ))
```