gTLV

API Documentation

October 9, 2009

Contents

Contents **1**

1 Module gtlv.network **2**
1.1 Class Target . . . 2
1.1.1 Methods . . . 2
1.2 Class Handler . . . 2
1.2.1 Methods . . . 3
1.3 Class GtlvServer . . . 3
1.3.1 Methods . . . 3
1.3.2 Class Variables . . . 4

2 Module gtlv.packet **5**
2.1 Functions . . . 5
2.2 Class HeaderFormat . . . 5
2.2.1 Instance Variables . . . 5
2.3 Class AttributeFormat . . . 6
2.3.1 Instance Variables . . . 6
2.4 Class ClassList . . . 6
2.4.1 Methods . . . 6
2.5 Class AttributeList . . . 7
2.5.1 Methods . . . 7
2.6 Class PacketList . . . 7
2.6.1 Methods . . . 8
2.7 Class Attribute . . . 8
2.7.1 Methods . . . 8
2.7.2 Instance Variables . . . 8
2.8 Class Packet . . . 9
2.8.1 Methods . . . 9
2.8.2 Instance Variables . . . 10

Index **12**

1 Module gtlv.network

gTLV networking

1.1 Class Target

Represents a client that targets a server and that can be used to send packets to that server.

1.1.1 Methods

__init__(*self*, *target_address*, *target_port*, *list_manager*)

Constructor. It only saves parameters to internal attributes.

Parameters

`target_address`: IP address of FQDN for the server.
(type=String.)

`target_port`: TCP listening port for the server.
(type=Integer.)

`list_manager`: List of available packets and attribute classes.
(type=Class that has a packet_list attribute of class PacketList and an attribute_list attribute of class AttributeList.)

send(*self*, *packet*)

Sends a given packet to the server and waits for a reply.

Parameters

`packet`: Packet to be sent.
(type=Packet instance.)

Return Value

Packet return by the server..
(type=Packet instance or None.)

1.2 Class Handler

SocketServer.BaseRequestHandler
└ **gtlv.network.Handler**

Request handler for the server.

1.2.1 Methods

handle(*self*)

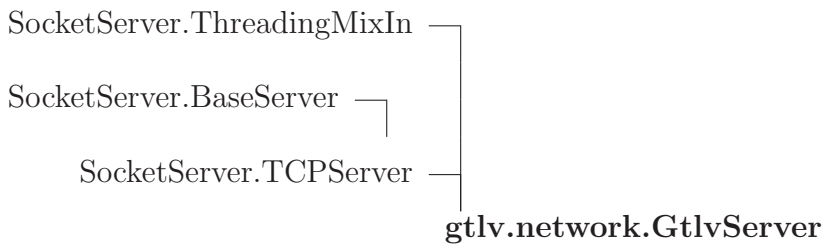
Waits until data is received, calls server's callback function and sends the result of the callback to the original sender.

Overrides: SocketServer.BaseRequestHandler.handle

Inherited from SocketServer.BaseRequestHandler

__init__(), finish(), setup()

1.3 Class GtlvServer



Represents a server that waits for incoming gTLV packets.

1.3.1 Methods

__init__(*self*, *ip*, *port*, *list_manager*)

Constructor. It only saves parameters to internal attributes and calls the constructor of its parent class.

Parameters

`ip`: Local IP address to bind to.

(type=String.)

`port`: Local TCP port to bind to.

(type=Integer.)

`list_manager`: List of available packets and attribute classes.

(type=Class that has a packet_list attribute of class PacketList and an attribute_list attribute of class AttributeList.)

Overrides: SocketServer.BaseServer.__init__

callback(*self*, *raw_packet*)

Callback function to process received packets. Needs to be overridden by subclasses.

Parameters

`raw_packet:` Raw packet as received on the socket.
(type=String.)

start(*self*)

Start the server in a new thread. Worker threads will be used to process incoming packets.

Inherited from SocketServer.ThreadingMixIn

process_request(), process_request_thread()

Inherited from SocketServer.TCPServer

close_request(), fileno(), get_request(), server_activate(), server_bind(), server_close()

Inherited from SocketServer.BaseServer

finish_request(), handle_error(), handle_request(), serve_forever(), verify_request()

1.3.2 Class Variables

Name	Description
Inherited from SocketServer.ThreadingMixIn	
daemon_threads	
Inherited from SocketServer.TCPServer	
address_family, allow_reuse_address, request_queue_size, socket_type	

2 Module gtlv.packet

gTLV packet

2.1 Functions

decode(*raw_packet*, *list_manager*)

Converts raw packet data from a socket into a Packet instance.

Parameters

`raw_packet:` Attribute to be added.

(type=Attribute instance.)

`list_manager:` List of available packets and attribute classes.

(type=Class that has a packet_list attribute of class PacketList and an attribute_list attribute of class AttributeList.)

Return Value

Packet instance that corresponds to raw packet data.

(type=Packet or None.)

2.2 Class HeaderFormat

Representation of gTLV packet header

2.2.1 Instance Variables

Name	Description
application	Format of application field. As of gTLV v1, 2 bytes. **Value:** `'H'` *(type=String that corresponds to struct definition of the format.)*
code	Format of code field. As of gTLV v1, 1 byte. **Value:** `'B'` *(type=String that corresponds to struct definition of the format.)*
length	Format of packet length field. As of gTLV v1, 2 bytes. **Value:** `'H'` *(type=String that corresponds to struct definition of the format.)*

2.3 Class AttributeFormat

Representation of gTLV packet attributes

2.3.1 Instance Variables

Name	Description
type	Format of type field. As of gTLV v1, 1 byte. **Value:** `'B'` *(type=String that corresponds to struct definition of the format.)*
length	Format of attribute length field. As of gTLV v1, 1 byte. **Value:** `'B'` *(type=String that corresponds to struct definition of the format.)*
value	Format of value field. Not used as it depends on attribute type. **Value:** `''` *(type=Empty string.)*

2.4 Class ClassList

Known Subclasses: gtlv.packet.AttributeList, gtlv.packet.PacketList

ClassList keeps a record of available classes to be used by the encoding/decoding engines.

2.4.1 Methods

__init__(*self*, *items*)

Constructor. Adds first items to the internal list.

Parameters

`items`: Initial list of available classes.

(type=Tuple of classes.)

append(*self*, *items*)

Adds items to the internal list.

Parameters

`items`: List of available classes to be added.

(type=Tuple of classes.)

find(*self*, *needle*)

Finds a given class in the internal list. Must be overridden by subclasses.

Parameters

`needle`: Class to be found.

(type=Class.)

2.5 Class AttributeList

gtlv.packet.ClassList
└ **gtlv.packet.AttributeList**

Keeps a record of available attribute classes to be used by the encoding/decoding engines.

2.5.1 Methods

find(*self*, *needle*)

Finds a given attribute type in the list of available attribute classes.

Parameters

`needle`: Attribute type to be found.

(type=Integer.)

Return Value

Attribute class found.

(type=Attribute class or None.)

Overrides: gtlv.packet.ClassList.find

Inherited from gtlv.packet.ClassList(Section 2.4)

__init__(), append()

2.6 Class PacketList

gtlv.packet.ClassList
└ **gtlv.packet.PacketList**

Keeps a record of available packet classes to be used by the encoding/decoding engines.

2.6.1 Methods

find(*self*, *needle*)

Finds a given packet type in the list of available packet classes, based on application and code fields..

Parameters

`needle`: Packet application and code combination to be found in the list.

(type=Tuple of two integers.)

Return Value

Packet class found.

(type=Packet class or None.)

Overrides: gtlv.packet.ClassList.find

Inherited from gtlv.packet.ClassList(Section 2.4)

__init__(), append()

2.7 Class Attribute

Generic placeholder for TLV attributes. Subclasses must fill `type`, `typedef` and `fields` on definition and `value` on instantiation.

2.7.1 Methods

__init__(*self*)

Constructor. Does nothing for now. Subclasses must call this constructor from their constructors.

encode(*self*)

Converts attribute data into a raw string that can be inserted in a raw packet.

Return Value

Raw attribute.

(type=String or None.)

2.7.2 Instance Variables

Name	Description
type	Attribute type. **Value:** `None` *(type=Integer.)*
value	Attribute value. **Value:** `None` *(type=Depends on the attribute type.)*
typedef	Value data type that corresponds to the attribute type. **Value:** `None` *(type=String. As of gTLV v1, one of 'integer', 'boolean', 'timestamp', 'string' or 'octets'.)*
fields	Value format for 'octets' attributes. **Value:** `None` *(type=Tuple. As of gTLV v1, one of 'integer', 'boolean' or 'timestamp'.)*

2.8 Class Packet

Generic placeholder for TLV packets. Subclasses must fill `application`, `code`, `mandatory_attributes` and `optional_attributes` (if any) on definition.

2.8.1 Methods

__init__(*self*)

Constructor. Subclasses must call this constructor from their constructors.

count(*self*, *attribute_class*)

Count how many attributes of a given class are currently in the packet.

Parameters

`attribute_class`: Attribute class for the search.

(type=Attribute class.)

Return Value

Number of occurrences.

(type=Integer.)

get_values(*self*, *attribute_class*)

Get all the values of all the attributes of a given class found in the packet.

Parameters

`attribute_class:` Attribute class for the search.

(type=Attribute class.)

Return Value

List of corresponding values.

(type=List.)

add_attribute(*self*, *attribute*)

Add an attribute to the packet.

Parameters

`attribute:` Attribute to be added.

(type=Attribute instance.)

Return Value

`True` on success, `False` on failure.

(type=`True` or `False`.)

encode(*self*)

Converts packet data into a raw string that can be sent over a socket.

Return Value

Raw packet.

(type=String or None.

 # TODO: Check that all mandatory attributes are present)

2.8.2 Instance Variables

Name	Description
application	Application identifier. **Value:** `None` *(type=Integer.)*
code	Packet code. **Value:** `None` *(type=Integer.)*
mandatory_attributes	List of mandatory attributes. **Value:** `None` *(type=Dictionary. Made up of `attribute_class: multiplicity` items.)*

continued on next page

Name	Description
optional_attributes	List of optional attributes. **Value:** `None` *(type=Dictionary. Made up of* `attribute_class: multiplicity` *items. Multiplicity means maximum number of occurrences, like* `1`, `23` *or* `float('inf')`*.)*

Index

gtlv *(package)*
- gtlv.network *(module)*, 2–4
  - gtlv.network.GtlvServer *(class)*, 3–4
  - gtlv.network.Handler *(class)*, 2–3
  - gtlv.network.Target *(class)*, 2
- gtlv.packet *(module)*, 5–11
  - gtlv.packet.Attribute *(class)*, 8–9
  - gtlv.packet.AttributeFormat *(class)*, 5–6
  - gtlv.packet.AttributeList *(class)*, 7
  - gtlv.packet.ClassList *(class)*, 6–7
  - gtlv.packet.decode *(function)*, 5
  - gtlv.packet.HeaderFormat *(class)*, 5
  - gtlv.packet.Packet *(class)*, 9–11
  - gtlv.packet.PacketList *(class)*, 7–8