CS158B Project 3 Final Report

Group Members

Name	Phone Number	Email Address
Sri Sreedharan (Coordinator)	925-699-3982	srisreed@gmail.com
Thanh Au	408-886-0391	thanhau123@yahoo.com
Kai - Ting (Danil) Ko	925-292-1299	KAI.TING.KO@GMAIL.COM
Wilson Yee	415-812-3665	wiyee88@gmail.com

Project Description

The Three Tier System
The project will be a three tier network management system.

1. **Client**
   The first tier will be implemented as a web page. The web page will convert user inputs into JSON and integrated results into URI to send to Web Services. It will also retrieve and parse resulted JSON to display results.

2. **Server**
   The second tier will be an application server that hosts web services and acts as an UDP client to communicate with network elements. The application server will be the Oracle Glassfish. The web services will be implemented with the RESTful architecture. This tier also has a connection to database to update and retrieve information about network elements. The database will be Microsoft SQL Server 2012.

3. **Network Elements**
   The third tier will be consisted of network elements that are SNMP enabled or network element simulators that simulate SNMP operations. These elements or simulators will have an UDP Server to receive operations and to return results. The simulator will be two types: a SNMP enabled real device and a simulator that is implemented as an UDP server.

Security

Overall Summary

The frontend and the application server will communicate through Security Socket Layer (SSL). The SSL will allow client to authenticate the application server through the usage of digital certificate. SSL establishes a secure session when a single transaction occurs through the use of public key system. However, the server will not authenticate the user since it will only perform necessary conversions for messages and forward them to particular targeted network elements. The frontend will construct URI by collecting user input information. It will then use correct SSL port, in this case, the server provides SSL services through 8181 port with SSL URI header, https. So the browser will be able to know to establish a SSL connection when it sends these particular URIs. However, since the digital certificate that is adopted in this project is not signed by Third Party Certificate Authority (CA), the connection will be blocked by browsers for invalid certificate. Users will first need to connect to the application directly and forced the browser to accept the invalid certificate before starting to use the frontend or frontend will not be able to communicate with the Application backend due to blocked connections. The message will be OID (Object Identifier), Community Strings, SNMP commands and computational results.Trudy (the attacker) will not be able to recover the key since all three tiers will dispose session keys once a transaction is complete. The backend and network elements will not have security implementations beside serialization of messages. The reason is because the project assumes the Application Server and the network elements are in the same local network, therefore it should be safer than the Internet network between the frontend and the backend.

1. **Encryption**
   SSL will encrypt messages with the public key system. The RSA public key system is used for the data encryption.

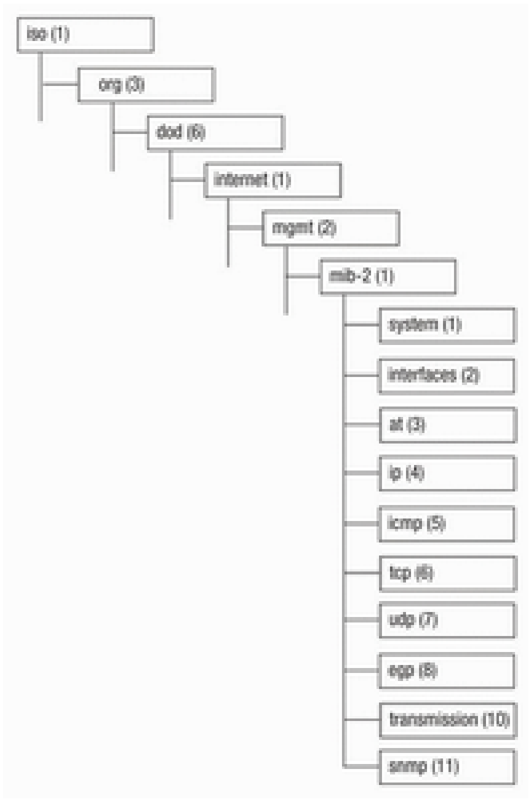
2. **Integrity**
   SSL will protect message integrity with the SHA-1 Hash Functions. It prevents the attacker from making modifications due to the difficulty in solving hash functions.

3. **Network Element Authentication**
   SNMPv1 community string is used for network element authentications since the project assumes all network elements only suppor SNMPv1.

Supported SNMP Configuration

- **Create or modify an SNMP View Record**
  The user will enter the host, community string and record. Then a URL string will be formulated into a single message for the Application Server. The Application Server will then compute the necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will retrieve the information from a local database according to inputted operations. The Application Server will parse returned results into JSON format and send back to the front end. Finally, the front end parses the JSON results to display to the user.
- **Create or modify access control (including the MIB view) for an SNMP Community**
  The user will enter the host, community string and record. Then a URL string will be formulated into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will find the particular record and only change permission if the inputted community string is authorized. The Network Element will respond the Server whether the operation is success or not. The Server will parse this result in JSON format and send to the FrontEnd which will then be displayed.
- **Enable the SNMP Agent Shutdown Mechanism**
  The user will enter the host, community string. Then a URL string will be formulated into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will only perform the shutdown if the inputted community string is authorized. The shutdown will stop the Network Element thread and starts a new thread. The Network Element will respond the Server whether the operation is success or not. The Server will parse this result in JSON format and send to the FrontEnd which will then be displayed.
- **Establish the Contact, Location, and Serial Number of the SNMP Agent**
  The user will enter the host, community string, contact value, location value and serial number. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will overwrite with new values only if the community string is authorized. It then return results. The Server will parse this result in JSON format and send to the FrontEnd which will then be displayed.

MIB for Network Element Simulator



IBM
http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.netcool_ssm.doc/rg/images/mib2MIB.gif

Supported RMON Configuration

- **Enable RMON**
  The user will enter host, community string and opinion to enable or disable rmon. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The network element will set the rmon object to be on/off status when the community string is authorized. The Application Server will parse returned results into JSON format and send back to the FrontEnd. Finally, the FrontEnd parses the JSON results to display to the user.
  This part is not working due to the connection issues between the Application Server and the network element.

- **Change the size of the RMON queue**

The user will enter host, community string and size of the queue. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will set the queue size if the community string is authorized and the queue is not full. The older alarm will be deleted if the queue is smaller than all alarms. The Application Server will parse returned results into JSON format and send back to the FrontEnd. Finally, the FrontEnd parses the JSON results to display to the user.
This part is not working due to the connection issues between the Application Server and the network element.

- **Set an alarm on a MIB object**
  Rmon alarm number variable interval
  Rising - threshold value [event - number]
  Falling - threshold value [event - number][owner string]
  The user will enter host, community string, variable, interval, rising threshold value, falling threshold value, event number and owner string. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will set the alarm if the community string is authorized and the queue is not full. The Application Server will parse returned results into JSON format and send back to the FrontEnd. Finally, the FrontEnd parses the JSON results to display to the user.
  This part is not working due to the connection issues between the Application Server and the network element.

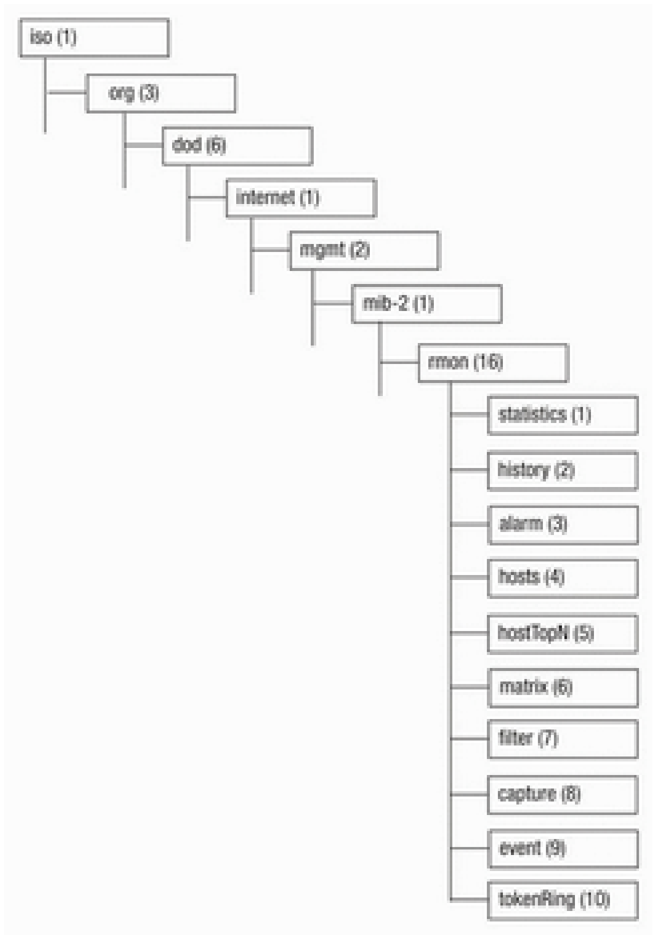
- **Rmon event number [log][trap community][description string][owner string]**
  The user will enter host, community string, event number, log, trap community, description string, owner string. The event number, log, trap community, description string and owner string are optional, which means the a message can be composed without them. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will set the event if the community string is authorized and the queue is not full. The Application Server will parse returned results into JSON format and send back to the FrontEnd. Finally, the FrontEnd parses the JSON results to display to the user.

This part is not working due to the connection issues between the Application Server and the network element.

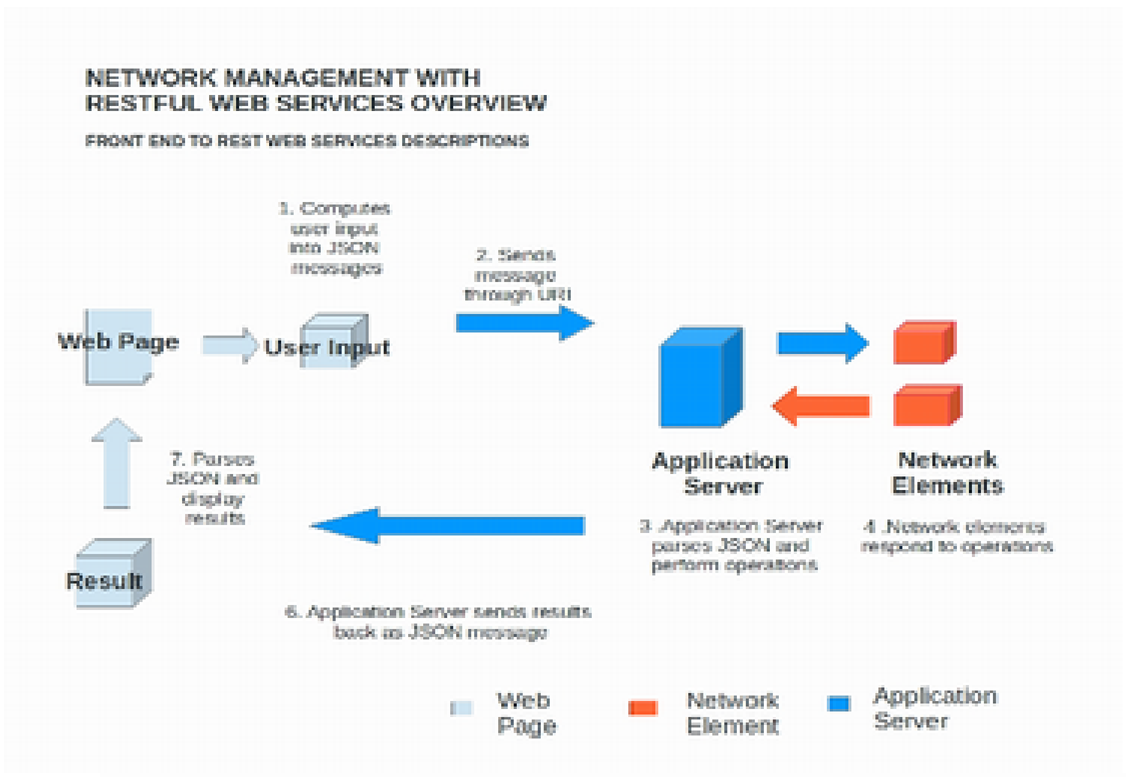
- **Show RMON Alarm**
  The user will enter host, community string. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will retrieve rmon information includes the number of packages and queue size if the community string is authorized. It then return results. The Server will parse this result in JSON format and send to the FrontEnd which will then be displayed.

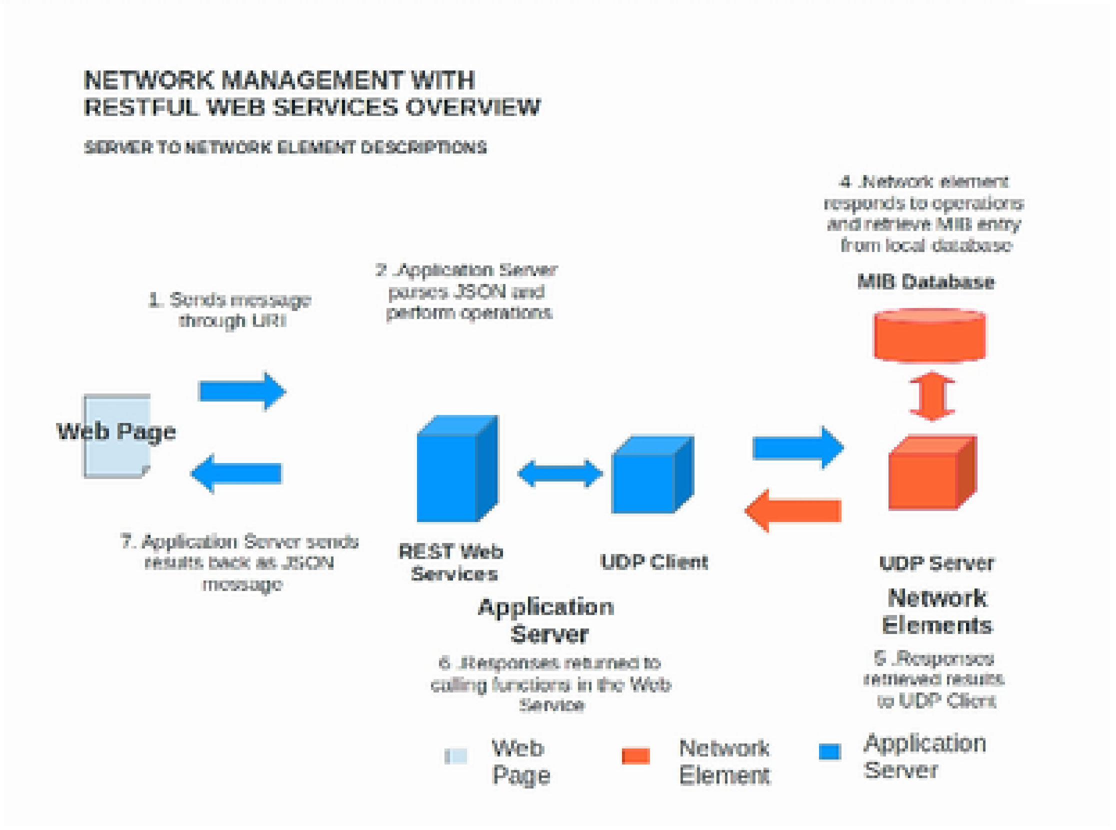
- **Show RMON Event**
  The user will enter host, community string. The FrontEnd will compute these information into a single message for the Application Server. The Application Server will compute necessary operations to match the inputted message. These operations will then be sent to Network Elements. The Network Element will retrieve rmon status includes the number of packages and queue size if the community string is authorized. It then return results. The Server will parse this result in JSON format and send to the FrontEnd which will then be displayed. This part is not working due to the connection issues between the Application Server and the network element.

MIB for RMON Simulator



IBM
http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.netcool_ssm.doc/rg/images/rmon1Grp.gif


NETWORK MANAGEMENT WITH RESTFUL WEB SERVICES OVERVIEW
FRONT END TO REST WEB SERVICES DESCRIPTIONS
1. Computes user input into JSON messages
2. Sends message through URI
Web Page
User Input
Application Server
Network Elements
3. Application Server parses JSON and perform operations
4. Network elements respond to operations
7. Parses JSON and display results
Result
6. Application Server sends results back as JSON message
Web Page
Network Element
Application Server


NETWORK MANAGEMENT WITH RESTFUL WEB SERVICES OVERVIEW
SERVER TO NETWORK ELEMENT DESCRIPTIONS
4 .Network element responds to operations and retrieve MIB entry from local database
MIB Database
2 .Application Server parses JSON and perform operations
1. Sends message through URI
Web Page
7. Application Server sends results back as JSON message
REST Web Services
UDP Client
Application Server
UDP Server
Network Elements
6 .Responses returned to calling functions in the Web Service
5 .Responses retrieved results to UDP Client
Web Page
Network Element
Application Server

Platform and Language

The platform that will be used are Windows 7 x64 bits and Oracle Glassfish Application Server. The language that will be used are JAVA, J2EE, SQL, JavaScript and JSON. We also plan on utilizing HTML/CSS for frontend with some help from Twitter's Bootstrap. Also, the JavaScript library jQuery will be used to retrieve and parse JSON formatted messages. Java keytool is used for digital certificate generations.

References

1. Orcale JAVA EE 6 Tutorial
   http://docs.oracle.com/javaee/6/tutorial/doc/giepu.html

2. IBM
   http://publib.boulder.ibm.com/

3. Generate Self - Signed Certificate with Keytool for Glassfish
   http://artur.ejsmont.org/blog/content/how-to-generate-self-signed-ssl-certificate-for-glassfish-v3-and-import-it-into-java-keyring