

How-to implement SNMPv3 GETNEXT with SNMP4J?

The following code implements a simple SNMPv3 GETNEXT operation using USM and an user with SHA256 authentication protocol and AES128 privacy protocol. Feel free to extend and adapt this code in your project:

SNMPv3 GETNEXT

```
/* #####
_ ##
_ ##  SNMP4J - UsmGetNext.java
_ ##
_ ##  Copyright (C) 2019  Frank Fock (SNMP4J.org)
_ ##
_ ##  Licensed under the Apache License, Version 2.0 (the "License");
_ ##  you may not use this file except in compliance with the License.
_ ##  You may obtain a copy of the License at
_ ##
_ ##      http://www.apache.org/licenses/LICENSE-2.0
_ ##
_ ##  Unless required by applicable law or agreed to in writing, software
_ ##  distributed under the License is distributed on an "AS IS" BASIS,
_ ##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_ ##  See the License for the specific language governing permissions and
_ ##  limitations under the License.
_ ##
##### */

package org.snmp4j.getnext;

import org.snmp4j.*;
import org.snmp4j.event.ResponseEvent;
import org.snmp4j.event.ResponseListener;
import org.snmp4j.log.ConsoleLogFactory;
import org.snmp4j.log.LogFactory;
import org.snmp4j.log.LogLevel;
import org.snmp4j.mp.MPv3;
import org.snmp4j.mp.SnmpConstants;
import org.snmp4j.security.*;
import org.snmp4j.smi.*;
import org.snmp4j.transport.DefaultUdpTransportMapping;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class UsmGetNext {

    private Snmp snmp;
    private USM usm;

    public UsmGetNext() {
        ConsoleLogFactory consoleLogFactory = new ConsoleLogFactory();
        consoleLogFactory.getRootLogger().setLogLevel(LogLevel.DEBUG);
        LogFactory.setLogFactory(consoleLogFactory);
    }

    public void initSnmp() throws IOException {
        snmp = new Snmp();
        snmp.getMessageDispatcher().addCommandResponder(new CommandResponder() {
            @Override
            public <A extends Address> void processPdu(CommandResponderEvent<A> commandResponderEvent) {
                System.out.println(commandResponderEvent.toString());
            }
        });
        // Very important to add snmp as command responder which will finally process the PDU:
        snmp.getMessageDispatcher().addCommandResponder(snmp);
    }
}
```

```

snmp.addTransportMapping(new DefaultUdpTransportMapping(new UdpAddress(0)));
snmp.getMessageDispatcher().addMessageProcessingModel(new MPv3());
SecurityProtocols.getInstance().addDefaultProtocols();
OctetString localEngineID = new OctetString(MPV3.createLocalEngineID());
usm = new USM(SecurityProtocols.getInstance(), localEngineID, 0);
usm.setEngineDiscoveryEnabled(true);
SecurityModels.getInstance().addSecurityModel(usm);

snmp.getMessageDispatcher().addMessageProcessingModel(new MPv3(usm.getLocalEngineID().getValue()));
snmp.listen();
}

public void next(String address, String contextName, String securityName,
                String authPassphrase, String privPassphrase, String... oids) throws IOException {
    UsmUser usmUser = new UsmUser(new OctetString(securityName),
        AuthHMAC192SHA256.ID, new OctetString(authPassphrase),
        PrivAES128.ID, new OctetString(privPassphrase));
    usm.addUser(usmUser);

    List<VariableBinding> oidList = new ArrayList<>(oids.length);
    for (String objectID : oids) {
        oidList.add(new VariableBinding(new OID(objectID)));
    }

    Address targetAddress = GenericAddress.parse(address);
    Target<Address> userTarget = new UserTarget<>();
    userTarget.setAddress(targetAddress);
    userTarget.setRetries(1);
    // set timeout to 500 milliseconds: 2*500ms = 1s total timeout
    userTarget.setTimeout(500);
    userTarget.setVersion(SnmpConstants.version3);
    userTarget.setSecurityLevel(SecurityLevel.AUTH_PRIV);
    userTarget.setSecurityName(usmUser.getSecurityName());

    ScopedPDU scopedPDU = new ScopedPDU();
    scopedPDU.addAll(oidList);
    scopedPDU.setContextName(new OctetString(contextName));
    ResponseListener responseListener = new ResponseListener() {
        @Override
        public synchronized <A extends Address> void onResponse(ResponseEvent<A> responseEvent) {
            // Free resources we will not wait for further events
            snmp.cancel(responseEvent.getRequest(), this);
            // Process response here:
            if (responseEvent.getResponse() != null) {
                System.out.println("Received: "+responseEvent.getResponse());
            }
            else {
                if (responseEvent.getError() != null) {
                    System.err.println("Error: "+responseEvent.getError());
                }
                else {
                    System.err.println("Timed out.");
                }
            }
            notify();
        }
    };

    synchronized (responseListener) {
        snmp.getNext(scopedPDU, userTarget, null, responseListener);
        try {
            responseListener.wait(500000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public static void main(String[] args) {
    if (args.length < 5) {
        System.out.println("Usage: UsmGetNext <address> <secName> <authPassphrase> <privPassphrase>
<oid>...");
    }
}

```

```

        System.out.println("where <address> is of the form 'udp:<hostname>/<port>'");
    }
    String targetAddress = args[0];
    String context = "";
    String securityName = args[1];
    String authPassphrase = args[2];
    String privPassphrase = args[3];
    String[] oids = new String[args.length - 4];
    System.arraycopy(args, 4, oids, 0, args.length - 4);
    UsmGetNext usmGetNext = new UsmGetNext();
    try {
        usmGetNext.initSnmp();
        usmGetNext.next(targetAddress, context, securityName, authPassphrase, privPassphrase, oids);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```



Download the full UsmGetNext sample project from <https://snmp.app/dist/snapshot/org/snmp4j/snmp4j-getnext/1.1.0-SNAPSHOT/>

The following Unix Diff code block illustrates which changes to the above code have to be made to use the new DirectUserTarget (SNMP4J 3.4.0 or later) to directly specify an USM user without prior adding it to the local USM storage (table):

```

--- /var/folders/s_/7ltpzc55t19sn2_qnt9rm4h0000gn/T/fJLNBb_UsmGetNext.java      2020-03-06 22:29:22 +0000
+++ /var/folders/s_/7ltpzc55t19sn2_qnt9rm4h0000gn/T/DFlCAb_UsmGetNext.java      2020-03-06 22:29:22 +0000
@@ -69,39 +69,39 @@
    usm = new USM(SecurityProtocols.getInstance(), localEngineID, 0);
    usm.setEngineDiscoveryEnabled(true);
    SecurityModels.getInstance().addSecurityModel(usm);

    snmp.getMessageDispatcher().addMessageProcessingModel(new MPv3(usm.getLocalEngineID().getValue()));
    snmp.listen();
}

public void next(String address, String contextName, String securityName,
                 String authPassphrase, String privPassphrase, String... oids) throws IOException {
-    UsmUser usmUser = new UsmUser(new OctetString(securityName),
-    AuthHMAC192SHA256.ID, new OctetString(authPassphrase),
-    PrivAES128.ID, new OctetString(privPassphrase));
-    usm.addUser(usmUser);
-
    List<VariableBinding> oidList = new ArrayList<>(oids.length);
    for (String objectID : oids) {
        oidList.add(new VariableBinding(new OID(objectID)));
    }

    Address targetAddress = GenericAddress.parse(address);
-    Target<Address> userTarget = new UserTarget<>();
+    byte[] targetEngineID = snmp.discoverAuthoritativeEngineID(targetAddress, 1000);
+    if (targetEngineID != null) {
+        UsmUserEntry targetUser =
+            snmp.createLocalizedUsmUserEntry(targetEngineID, new OctetString(securityName),
+            AuthHMAC192SHA256.ID, OctetString.fromString(authPassphrase),
+            PrivAES128.ID, OctetString.fromString(privPassphrase));
+        DirectUserTarget<Address> userTarget = new DirectUserTarget<>();
+        userTarget.setAddress(targetAddress);
+        userTarget.setRetries(1);
+        // set timeout to 500 milliseconds: 2*500ms = 1s total timeout
+        userTarget.setTimeout(500);
+        userTarget.setVersion(SnmpConstants.version3);
-        userTarget.setSecurityLevel(SecurityLevel.AUTH_PRIV);
-        userTarget.setSecurityName(usmUser.getSecurityName());

```

```

+         snmp.setLocalizedUserCredentials(userTarget, targetUser);

        ScopedPDU scopedPDU = new ScopedPDU();
        scopedPDU.addAll(oidList);
        scopedPDU.setContextName(new OctetString(contextName));
        ResponseListener responseListener = new ResponseListener() {
            @Override
            public synchronized <A extends Address> void onResponse(ResponseEvent<A> responseEvent) {
                // Free resources we will not wait for further events
                snmp.cancel(responseEvent.getRequest(), this);
                // Process response here:
@@ -119,31 +119,35 @@
            };
            synchronized (responseListener) {
                snmp.getNext(scopedPDU, userTarget, null, responseListener);
                try {
                    responseListener.wait(500000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
+    } else {
+        System.err.println("Timeout on engine ID discovery for "+targetAddress+", GETNEXT not sent.");
+    }
+    }

    public static void main(String[] args) {
        if (args.length < 5) {
            System.out.println("Usage: UsmGetNext <address> <secName> <authPassphrase> <privPassphrase>
<oid>...");
            System.out.println("where <address> is of the form 'udp:<hostname>/<port>'");
        }
        String targetAddress = args[0];
        String context = "";
        String securityName = args[1];
        String authPassssphrase = args[2];
        String privPassssphrase = args[3];
-        String authPassssphrase = args[2].length() == 0 ? null : args[2];
+        String authPassssphrase = args[2].length() == 0 ? null : args[2];
+        String privPassssphrase = args[3].length() == 0 ? null : args[3];
        String[] oids = new String[args.length - 4];
        System.arraycopy(args, 4, oids, 0, args.length - 4);
        UsmGetNext usmGetNext = new UsmGetNext();
        try {
            usmGetNext.initSnmp();
            usmGetNext.next(targetAddress, context, securityName, authPassssphrase, privPassssphrase, oids);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Related articles

- [How-to implement SNMPv3 GETNEXT with SNMP4J?](#)