# ExtraHop 7.5

# Build a trigger to monitor responses to NTP monlist requests

Published: 2019-01-08

Machines in your environment synchronize clocks through the Network Time Protocol (NTP), but NTP has some security vulnerabilities such as amplification attacks that lead to denial of service.

For example, an attacker can spoof the IP address of your NTP server, and then repeatedly send a monlist command through the spoofed address. The monlist command requests a list of the last 600 hosts that connected to the NTP server, but because the requesting IP address is spoofed, the server actually sends the list to the spoofed address. The response is considerably larger than the request, and the spoofed client becomes overloaded, which can lead to denial of legitimate requests.

In this walkthrough, you will write a trigger that checks UDP traffic on your NTP server for responses to monlist commands. The trigger also sends an alert-level message to a remote syslog server when a monlist response occurs.

## Prerequisites

- You must have access to an ExtraHop Discover appliance with a user account that has unlimited privileges.
- Your must have at least one NTP server that you want to monitor.
- You must have a remote syslog server that can receive data from the ExtraHop system.
- You must have familiarity with JavaScript.
- Familiarize yourself with the concepts in this walkthrough by reading the Open Data Streams section in the ExtraHop Admin UI Guide and the Get started with triggers section in the ExtraHop Web UI Guide.
- Familiarize yourself with the processes of creating triggers and configuring open data streams by completing the Trigger Walkthrough and the ODS Walkthrough.

## Configure an open data stream to a syslog target

In the following steps, you will configure the host, port, and protocol for the open data stream target.

1. Log into the ExtraHop Discover appliance that you want to send data from with an account that has unlimited privileges.
2. Click the System Settings icon, and then click **Administration**.
3. In the System Configuration section, click **Open Data Streams**.
4. Click **Add Target**.
5. Select **Syslog** from the Target Type drop-down list.
6. In the Name field, type `NTP Syslog` unless this is the first syslog target you have created. In that case, the target is automatically named “default” and cannot be renamed.
7. In the Host field, type the IP address or hostname of the syslog server you want to send data to.
8. In the Port field, type the port number you want to send data to.
9. From the protocol list, select **UDP**.

**Tip:** Click **Test** to establish a connection and send a test message from the Discover appliance to the remote syslog server.

10. Select **Local** if you want to send syslog information with timestamps in the local timezone of the ExtraHop appliance. Otherwise, timestamps are sent in GMT.
11. Click **Save**.
    The target is added to the Syslog table on the Open Data Stream page.

## Write a trigger to parse NTP payloads

In the following steps, you will write a trigger that specifies what data to examine from NTP server responses and whether to send the data to a remote syslog server.

1. Click the ExtraHop logo in the upper left corner to return to the ExtraHop Web UI.
2. Click the System Settings icon, and then click **Triggers**.
3. Click **New** to open the Trigger Configuration window.
4. In the Name field, type `Parse UDP payload for NTP responses`.
5. Click **Enable Debugging** to enable the debug log and trigger performance metrics.
6. In the Events field, select **UDP_PAYLOAD**.
7. Click **Show advanced options** and specify the following payload settings to look for NTP traffic only on UDP port 123:
   a) Select **All UDP Datagrams**.
   b) In the Server port min field, type `123`.
   c) In the Server port max field, type `123`.
8. Click the **Editor** tab.
9. Add the following trigger code to enable access to the NTP server response payload:

```
//Capture the NTP server response
let buf = Flow.server.payload;
//Exit the trigger if the NTP server response cannot be captured
if (buf === null) {
    return;
}
```

10. Add the following trigger code to the existing script to specify which fields the trigger will extract from the header of the payload and which fields to ignore:

```
//Define the format of the NTP response
let fmt = ('B' + // Flags (LI, Version, Mode)
           'x' + // Auth + Seq (ignore)
           'x' + // Implementations (ignore)
           'B' + // Request code
           'B'); // Error
```

11. Add the following trigger code to the existing script to extract the fields from the payload:

```
//Analyze the NTP response based on the defined format
let values = buf.unpack(fmt);
let mode = values[0] & 0x7;
```

12. Add the following trigger code to the existing script to check the values of the following header fields:

```
// Exit the trigger if the mode value is not 7.
if (mode !== 7) {
    return;
}
let reqCode = values[1];

//Save the last four bits of the error code as a variable
```

```
let errorCode = values[2] >> 4;
```

The mode, located in the last three bits of the field, indicates the NTP mode of operation. A value of 7 specifies that the NTP server is responding to a private mode command, which includes the monlist command.

The request code field indicates the request type. A value of 20 or 42 specifies a monlist request.

The error code field, located in the last four bits, indicates the error type. A value of 0 specifies that the response is not an error.

13. Add the following trigger code to the existing script to send an alert-level message to the remote syslog server if the NTP server is responding to a monlist command and if the response is not an error.

```
//Check that there is no error and that the monlist command has been run
if ((errorCode === 0) && ((reqCode === 20) || (reqCode === 42))) {
    //If the monlist command has been run, send an alert level message
 with
    //the NTP server IP address to the Syslog server
    Remote.Syslog('NTP Syslog').alert('monlist enabled on ' +
 Flow.server.ipaddr);
}
```

The trigger sends messages that include the NTP server's IP address to the remote syslog server that you configured earlier. If the target you configured was automatically named, replace `'NTP Syslog'` with `'default'` in the code.

14. Add the following trigger code to the existing script to check whether debugging is enabled and send the specified output to the runtime log.

```
//Print the IP address, request code, and error code in the runtime log
debug('NTP Server ' + Flow.server.ipaddr +
      ' responded to mode 7 command ' + reqCode +
      ' with error code '  + errorCode + '.');
```

15. Click **Save and Close**.

## Assign the UPA trigger to a device

Before the trigger can examine UDP response payloads, you must assign the trigger to at least one device. For this walkthrough, you will assign the trigger to NTP servers on your network.

> **Important:** Assign triggers only to the specific devices that you need to collect metrics from to minimize the performance impact of your triggers on the ExtraHop system.

1. Click **Metrics** from the top menu.
2. From the left pane, click **Devices**.
3. In the Name column, locate at least one NTP server and select the checkbox.
4. Click the **Assign Trigger** icon from the top of the page.
5. Click the checkbox next to the **Parse UDP payload for NTP responses** trigger, and then click **Assign Triggers**.

After the trigger is assigned, it runs continuously until disabled.

## Check your syslog server and the runtime log for trigger results

When a response to a monlist command is sent by the NTP server, the trigger sends an alert-level message to your remote syslog server. The message contains the IP address of the NTP server that sent the response, similar to the following message:

```
1 2017-01-11T22:14:15.003Z mymachine.example.com monlist enabled on
    198.51.100.0
```

In addition, the trigger sends output to the runtime log if debugging is enabled. To view the results of the debug statement, return to the Trigger Configuration window and click **Runtime Log**. The output includes the IP address of the NTP server, the monlist request code, and the error code, similar to the following output:

```
NTP Server 198.51.100.0 responded to mode 7 command 42 with error code 0.
```

If the trigger results indicate that your NTP server has responded to a monlist command, you can take one of the following actions:

- Upgrade your NTP server to version 4.2.7 or later, which disallows monlist commands by default. Downloads are available from the NTP Software Downloads page at www.ntp.org.
- Modify the `ntp.conf` file on the NTP server to disable the monitoring function that allows monlist commands. Instructions are available on the Access Restrictions page at www.ntp.org.
- If your security and monitoring workflow requires that your NTP server responds to monlist commands, you can leverage this trigger to tighten controls around NTP responses. For example, you can create custom metrics based on information extracted with the trigger. With those custom metrics, you can create a dashboard to track NTP server activity or configure an alert that notifies you of responses to monlist commands.

If your NTP server is already configured to disallow monlist commands, you will not receive any syslog messages or see output in the runtime log. You can still check that the trigger is running through one of the following actions:

- Return to the Trigger Configuration window and click **Performance**. The graph shows activity as long as there is UDP traffic on the NTP server.
- Click **System Health** from the System Settings page and double-click the Trigger executes chart. The chart shows activity that indicates the trigger is running.
- Test for monlist commands from the client-side. Modify the trigger by setting the buf variable to `Flow.client.payload`, and then send a monlist command through a program such as `ntpdc` to the NTP server. This code change in conjunction with the monlist command extracts the request payload and the trigger sends a message to syslog and shows results in the output log.

By running this trigger, you learn whether your NTP servers are vulnerable to amplification attacks and what you can do to either monitor for attacks or disable the NTP commands that open the door to attacks.