IdPProdLogging

Production IdP Logging

The IdP, by default, writes to three log files. Before proceeding you should familiarize yourself with these files, how to configure them, and logback, the underlying logging framework.

Analyzing Logs

The IdP Audit Log Analysis Tool can analyze one or more audit log files and produce statistics such number of authentications per relying party, total number of relying parties access, etc.

Reducing Log Files

The default logging configuration causes all IdP logs to be rolled daily. In order to save space a deployer may wish to remove logs of a certain age, compress the log files, or both. Either, or both, of these methods may be used with any of the rolling file appenders in the default IdP logging configuration. The examples just happened to refer to the process log.

Removing Aged Log Files

To remove logs of a certain again add the element `MaxHistory`, with a value equal to the number of days worth of logs (including today) that should be kept, to the `rollingPolicy` element. For example, keeping a weeks worth of logs would provide result in:

```
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <MaxHistory>7</MaxHistory>
    <FileNamePattern>$IDP_HOME$/logs/idp-process-%d{yyyy-MM-dd}.log</FileNamePattern>
</rollingPolicy>
```

Compressing Log Files

To compress the log files add either the `.gz` (for gzip compression) or `.zip` (for zip compression) to the end of the value of the `FileNamePattern` element in the `rollingPolicy`. For example, the following configuration enables gzip compression:

```
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <MaxHistory>7</MaxHistory>
    <FileNamePattern>$IDP_HOME$/logs/idp-process-%d{yyyy-MM-dd}.log.gz</FileNamePattern>
</rollingPolicy>
```

Emailing Error Messages

Administrators should be informed immediately if an error has occurred within the IdP. The logging framework supports the ability to send email messages under such conditions.

To do this a deployer must, in their `logging.xml` configuration file:

1. Define an SMTP appender
2. Add that appender to the `root` logging category

Example Logging Configuration to Email Errors

```
<appender name="IDP_EMAIL" class="ch.qos.logback.classic.net.SMTPAppender">
   <SMTPHost>smtp.example.org</SMTPHost>
   <To>jsmith@example.org</To>
   <To>jdoe@example.org</To>
   <From>idp@example.org</From>
   <Subject>[IdP Error] %msg</Subject>
   <layout class="ch.qos.logback.classic.html.HTMLLayout"/>
</appender>

<!-- add email appender at root logger level -->
<root>
   <appender-ref ref="IDP_EMAIL" />
</root>
```


Prior to release 2.1.3 of the IdP a number of log messages were logged as `ERROR` instead of `WARN`. This can result in a large number of emails being sent.

Centralized Logs

Deployments may wish to send their IdP logs to a central logging service. This is especially useful when running an IdP cluster. The logging framework has the ability to send its log messages to a remote syslog server.

To do this a deployer must, in their `logging.xml` configuration file:

1. Define an syslog appender
2. Add that appender to the `root` logging category

```
<appender name="IDP_SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">
   <SyslogHost>syslog.example.org</SyslogHost>
   <Port>514</Port>
   <Facility>AUTH</Facility>
   <SuffixPattern>[%logger:%line] %msg</SuffixPattern>
</appender>

<!-- add syslog appender at root logger level -->
<root>
   <appender-ref ref="IDP_SYSLOG" />
</root>
```

This doesn't introduce a single point of failure because syslog is a send-and-forget protocol over UDP, so if a log message is never recorded on the central log server for some reason, the IdP will not be aware of this failure. For this reason, it can be wise to still log locally in addition to centrally with prudent rotation policies.

Logging Behind a Reverse Proxy

The Shibboleth IdP will log whatever IP address is given to it by the servlet container. If there is a load balancer in front of the IdP, then the logs may display the load balancer's IP rather than real client addresses. Logging the real IP address requires changing the servlet container to set the context IP address to a variable set by the load balancer like X-Forwarded-For.