IDP 3 basic operations

A few examples on the most basic operations and interactions one might have with the IDP: checking log files, updating the software, generating simple stats, managing local (non-federation) SPs, etc.

Paths and file names below assume a default installation location (/opt/shibboleth-idp) and unchanged logging/logback configuration. Though you might prefer to adjust your logging config or make /opt/shibboleth-idp/logs a symbolic link to another directory. You might also want to remove the "idp-" prefix from all the {process,warn,audit,consent-audit} log files since they'll likely end up in one IDP-specific logging directory anyway (and having all files start with the same letter isn't overly useful). But again, the examples below can't match local deployment decisions and so have been written to match a default IDP installation's behaviour. So please adjust as needed.

Who and how am I

What IDP version is currently installed

```
$ /opt/shibboleth-idp/bin/version.sh
3.4.6
```

What does the IDP think of its own state?

```
/opt/shibboleth-idp/bin/status.sh
```

Applying updates

See IDP 3 Updates for detailed instructions.

What's happening right now

Watch IDP und Webserver logs

```
multitail -f /opt/shibboleth-idp/logs/idp-process.log /var/log/tomcat9/access.log
```

Search for IDP Warnings and Errors

```
egrep 'WARN|ERROR' /opt/shibboleth-idp/logs/idp-process.log
```

Tomcat STDOUT/STDERR (formerly catalina.out)

```
journalctl -u tomcat9.service -e -f
```

Tail relevant logs at once

```
multitail -f /opt/shibboleth-idp/logs/idp-process.log /var/log/tomcat9/access.log -l 'journalctl -u tomcat9.
service -f'
```

Who logged in and where, with what attributes sent

Audit log

```
multitail -f /opt/shibboleth-idp/logs/idp-audit.log
```

Audit events for a given UserID

```
fgrep '|someuser99|' /opt/shibboleth-idp/logs/idp-audit.log
```

Failed logins in Jan 2019

```
zgrep ' failed$' /opt/shibboleth-idp/logs/idp-process.log.201901*
```

Successful logins today

```
fgrep succeeded /opt/shibboleth-idp/logs/idp-process.log
```

HTTP User-Agent IP address in audit and access log

```
fgrep 192.168.1.99 /opt/shibboleth-idp/logs/idp-audit.log /var/log/tomcat9/access.log
```

What data will go out for userid X to service Y

The aacli is a very useful tool to test what data the running IDP *would send* for a given subject (replace `SOME_USERID` below with the login name the subject would enter during authentication) to a given SP. Not only does that help verifying your attribute resolver and attribute filter configuration when you're making changes to either (or both), it can also be useful in debugging access problems someone experiences at a given SP as you can easily compare what data would go out for different subjects (e.g. in cases where access works for one person but fails for another) without needing the subjects' cooperation in this issue (or access to their password).

Attributes (and NameID) that would be sent

```
/opt/shibboleth-idp/bin/aacli.sh --saml2 -n SOME_USERID -r https://test-sp.aco.net/shibboleth
```

Statistics

ACOnet has contributed a log analysis tool for parsing the Shibboleth IDP's audit logs. For the current day use `/opt/shibboleth-idp/logs/idp-audit.log`.

Basic statistics for a given day

```
loganalysis.py -culn /opt/shibboleth-idp/logs/idp-audit.log.20190123.gz
2 unique relying parties
10 unique userids
25 logins

logins   | relyingPartyId
-------------------------
14       | https://sp.example.org/saml
11       | https://wiki.example.edu/shibboleth
```

Can be done for whole months or even years

```
loganalysis.py -cul /opt/shibboleth-idp/logs/idp-audit.log.201812*
21 unique relying parties
15 unique userids
406 logins
```

Maybe try one of the structured output formats for easy post-processing, e.g. JSON:

Can be done for whole months or even years

```
$ loganalysis.py -j /opt/shibboleth-idp/logs/idp-audit.log.20200[1-6]*
{
  "stats": {
    "logins": 8327,
    "rps": 211,
    "users": 150
  },
  "logins_per_rp": {
    "https://sp.example.org/saml": 29,
    "https://wiki.example.edu/shibboleth": 163,
    "usw.": "usf."
  }
}
```

For more see the built-in help (`loganalysis.py --help`) or the examples in the documentation.

Debugging

Log SAML Messages on DEBUG

```
$EDITOR /opt/shibboleth-idp/conf/logback.xml  # Set <logger name="PROTOCOL_MESSAGE" level="DEBUG"/> and save
/opt/shibboleth-idp/bin/reload-service.sh -id shibboleth.LoggingService
```

Make sure to undo this after you're done to avoid filling up file systems/volumes/disks with unnecessary DEBUG messages.

Locally managed Service Provider Metadata (non-eduID.at)

See our IDP 3 Metadata configuration documentation.