

Projektstudium - Softwareentwicklung am Beispiel eines internationalen "Big Data"-Projektes

Stefan Moll

Task: #10 (***)

There is a special log target in dCache, which produces xxx.access files. Some dCache components are using it, but not the ssh interface.

Description:

Update Ssh2Admin to log into access when admin is connected.

Starting point:

Ssh2Admin, NetLoggerBuilder classes and NetLogger docs <http://netlogger.lbl.gov/home>.

Topic:

Describe the use of the Java logging facility, (example : NetLogger and maybe something else)

Durchführung:

Basierend auf dieser Aufgabenstellung, habe ich mich zu aller erst damit beschäftigt, wo die beschriebene Ssh2Admin Klasse zu finden ist und wie das dCache eigene Dateiablage System organisiert ist, um mir einen Überblick über das System zu verschaffen.

Dabei habe ich auch das gewünschte logging Ziel gefunden, an dem Einträge im folgenden Format erstellt werden: `level=WARN ts=2017-08-14T14:39:46.762+0200`

`event=org.dcache.webdav.request transaction=door:WebDAV-S-lp-hrz-d207...`

`request.method=PUT`

Diese Einträge folgen den NetLogger Vorgaben, die als Sammlung von Best Practices zusammengetragen wurden. Sie geben Entwicklern einen Leitfaden zur einheitlichen Formatierung und Detaillierung, um auch ohne genaues Systemwissen lesbare und besser verarbeitbare Logeinträge zu erzeugen. Zur Erstellung dieser Einträge wird im dCache Projekt die Klasse NetLoggerBuilder genutzt, die im Wesentlichen einen StringBuilder kapselt und dabei die wesentlichen Formatierungsregeln umsetzt, um die gewünschte Vereinheitlichung zu schaffen. Diese mit dem Builder erstellten Einträge werden dann an das Logging Framework übergeben. dCache nutzt die beiden Nachfolger des ehemals populären *log4j* Frameworks *Simple Logging Facade for Java (SLF4J)*, sowie *logback*.

SLF4J stellt wie der Name vermuten lässt, nur eine Abstraktion in Form von Interfaces bereit, die dann von *logback* Implementiert werden. Dies ermöglicht dem Projekt später, ohne großen Aufwand, auch auf eine andere Implementierung zu wechseln. Logback wird wiederum mittels einer XML Datei, der `logback.xml` konfiguriert. In dieser ist definiert wie und wohin, abhängig vom Logging Ziel, Loglevel, etc., Einträge verarbeitet werden.

Die Ssh2Admin Klasse, in der nun das Logging hinzugefügt werden sollte, nutzt das *Apache Mina SSHD* Projekt um einen SSH Server zu erstellen. Die Schwierigkeit bei diesem Framework liegt in erster Linie darin, das es in bestimmten Teilen schlecht dokumentiert ist. Die Webseite selbst verweist auf eine ältere Version, die API Dokumentation auf eine noch ältere und in fragen wie, „Wird einem Session Objekt zu irgendeinem Zeitpunkt der Benutzername oder Authentifizierungsstatus zugewiesen?“ nicht hilfreich. Die Antwort dazu lautete für mich überraschend nein, also liegt es an der Implementierung solche Informationen abzulegen, die für das Logging interessant sind.

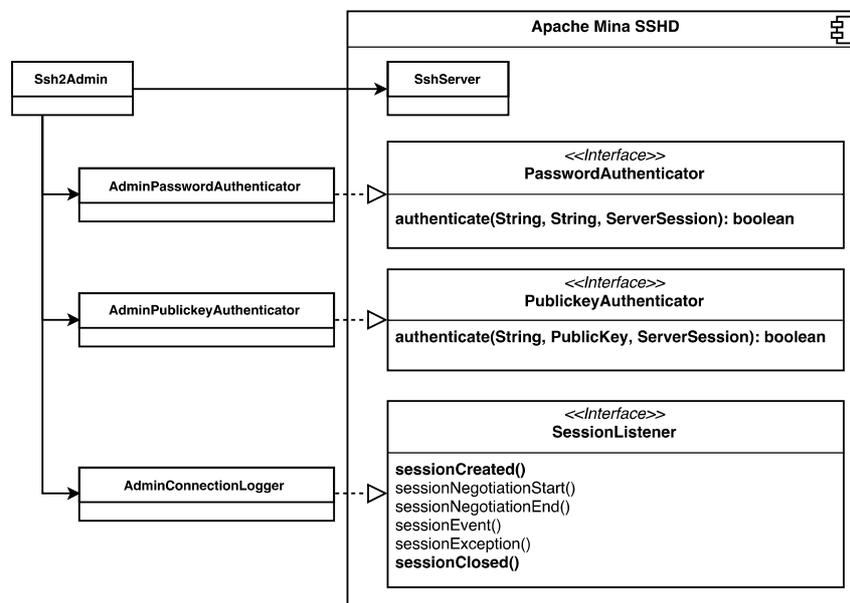
Die dCache Klasse besitzt 2 innere Klassen, den *AdminPasswordAuthenticator* (nutzt *gPlazma* zur User/Passwort Validierung und Gruppenabfrage) und *AdminPublickeyAuthenticator* (nutzt ein Keyfile zur Authentifizierung), die beide jeweils Schnittstellen des SSHD Frameworks zur Authentifizierung implementieren. An diesen beiden Stellen habe ich das Logging der

Anmeldeversuche implementiert. Dabei wird, bei Fehlschlag, unter anderem der Grund erfasst, wofür ein paar Anpassungen notwendig waren.

Im Zuge dieser Implementierung ist dem Kommilitonen Max (der auch am ssh Server entwickelte), sowie mir parallel aufgefallen, das es im PublicKey Authentifizierungsfall keine Überprüfung gibt, ob der verwendete Loginname zu einem existierenden Benutzer gehört, weshalb Max dafür ein Ticket erstellte. Nach kurzer Diskussion im Ticket, ob das ein Bug oder Feature sei, konnte ich dann davon überzeugen das es ein Bug ist, da es somit möglich war sich als anderen Benutzer Auszugeben. Dieses Problem habe ich dann in einem von der eigentlichen Aufgabe getrennten Pull Request gelöst und somit wird nun immer überprüft, ob der Loginname zum PublicKey im Keyfile gehört.

Damit es auch möglich ist nicht nur den eigentlichen Authentifizierungsvorgang im log zu sehen, habe ich noch eine neue Innere Klasse AdminConnectionLogger erstellt, die das SessionListener Interface für Session Auf-, sowie Abbau implementiert und in der Eventverarbeitung des SSH Server registriert.

Damit war die Aufgabe gelöst und die Struktur sieht nun wie folgt aus:



Diese Umsetzung ermöglicht es nun dCache Administratoren den gesamten Anmeldevorgang am SSH Server nachzuvollziehen und damit die Sicherheit des Systems zu steigern.