

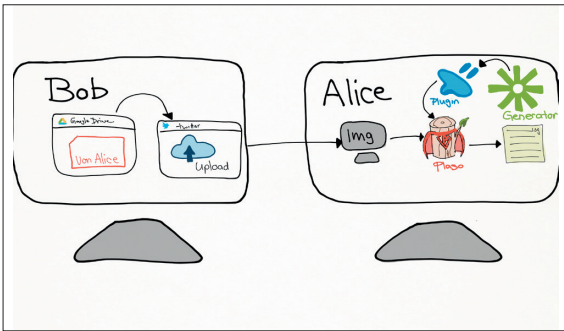


Claudia Saxer

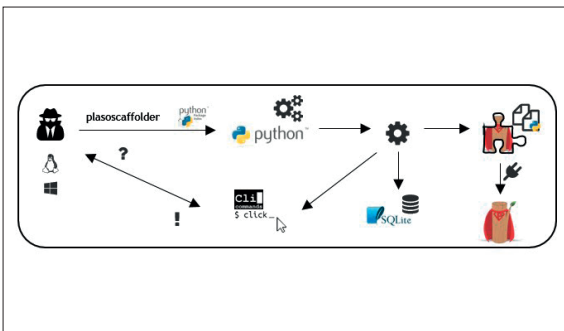
Diplomandin	Claudia Saxer
Examinator	Jürg Jucker
Experte	Maurin Egler, Synpulse Schweiz AG, Zürich, ZH
Themengebiet	Software
Projektpartner	Google Switzerland GmbH, Zürich, ZH

## Plaso SQLite Plugin Scaffolder

### Command Line Interface in Python



**Ausgangslage:** Plaso ist die Backend-Engine für das Tool Log2timeline, das Google für die forensische Datenanalyse verwendet. Log2timeline ist ein Tool, das aus einem Image alle Informationen sammelt, um eine Super Timeline zu erstellen, das ein riesiges Logfile darstellt. Plaso ist zu einem grossen Opensource Framework gewachsen, das von vielen Security Engineers verwendet wird und beinhaltet eine Ansammlung von Plugins. Plugins werden von Log2timeline verwendet, um Informationen von einem File zu extrahieren. Ein Grossteil der Plugins wurden für verschiedene SQLite Datenbanken entwickelt, und die Nachfrage für neue Plugins für SQLite-Datenbanken steigt. Im Moment involviert das Hinzufügen eines Plugins das manuelle Verändern von mehreren Files in vorgegebener Reihenfolge.



**Vorgehen/Technologien:** In einer ersten Phase soll deshalb das Erstellen von SQLite Plugins mit einem Gerüst vereinfacht und teilweise automatisiert werden. Somit sollen auch Security Engineers, die nicht viel Erfahrung mit der Entwicklung haben, schnell und einfach ein Plugin zu Plaso hinzufügen können. Dies verringert den manuellen Aufwand für das Schreiben der Plugins, sowie für die Codereviews. Das Tool soll ein ähnliches Konzept verwenden, das in anderen Frameworks als «scaffolding» bekannt ist. Das Tool soll vom Benutzer Eingaben entgegennehmen und daraus Files generieren sowie bestehende editieren, um schnell ein neues SQLite Plugin zu erstellen.

**Ergebnis:** Im Rahmen dieses Projektes wurde ein Command Line Interface Tool erstellt, das interaktiv Daten des Benutzers entgegennimmt und validiert sowie Daten einer SQLite-Datenbank auswertet. Die Applikation erlaubt durch ihren Aufbau das einfache Hinzufügen von weiteren Plugin Scaffolders. Der SQLite Scaffolder besitzt einerseits einen interaktiven Vorgang und erlaubt auch das Parametrisieren von einigen Eingaben. Die Kommunikation zwischen User und Applikation läuft mit dem Command Line Interface Tool Click, das auf optparse aufbaut. Die Files Templates wurden mit Jinja2 erstellt und sind einfach anpassbar. Das Tool beinhaltet das Editieren, Erstellen sowie Kopieren von Files. Die generierten Files sowie das Programm sind mit Pylint validiert.

```

C:\plascoscaffolder>sqlite
What's the path to the plaso project? C:/temp/plaso
What's the name of the plugin? the_plugin
What's the path to your test files? C:/temp/twitter_test.db
Do you want to have an output example for your SQL Query? [Y/n]: y
Please write your SQL script for the plugin. select id, name, updatedat from users
Your query output could look like this:
[16], name', 'updatedat']
[5402612, 'BBC Breaking News', 1449070544.333328]
[3334762, 'Globe', 144907072.57184]
[14388264, 'Tom Hohl', 144907072.562435]
Do you want to add this query? [Y/n]: y
Do you want to name the query base row: users ? [Y/n]: y
Is the column a time event? updatedat [Y/n]: y
Enter (additional) timestamp events from the query [columnName,aliasName...] or [abort]: abort
Does the event users need customizing? [Y/N]: y
Enter columns that are customizable (columnName,aliasName...) or [abort]: name
Added: name
[23166]
Do you want to add more columns that are customizable? [Y/N]: n
Do you want to add another query? [Y/n]: n
Do you want to generate the files? [Y/n]: y
create C:/temp/plaso/formatters/the_plugin.py
create C:/temp/plaso/parsers/sqlite_plugin/the_plugin.py
create C:/temp/plaso/tests/formatters/the_plugin.py
create C:/temp/plaso/tests/parsers/sqlite_plugin/the_plugin.py
copy C:/temp/plaso/test_data/the_plugin.db
edit C:/temp/plaso/parsers/sqlite_plugins/__init__.py
edit C:/temp/plaso/formatters/__init__.py
  
```