Pst-Extranct

Das Hauptprogramm befindet sich in der Datei PstMailJSON.java/class.

Compilation

Das Program benötigt die beiden Java Bibliotheken **jpst-1.0.jar** und **json-simple-1.1.1.jar**.

javac -cp jpst-1.0.jar:json-simple-1.1.1.jar PstMailJSON.java

jpst-1.0.jar ist die Bibliothek die benötigt wird um aus den .pst Dateien zu lesen. Die eingecheckte jpst-Bibliothek ist der Zeit aber nur eine Evaluations version dessen Support ausläuft. (http://www.independentsoft.de/jpst/index.html)

json-simple-1.1.1.jar wird benötigt um die extrahierten Daten in ein json zu Schreiben. (https://code.google.com/p/json-simple/)

Ausführung

Das Program braucht die beiden Bibliotheken, die ebenfalls zu Compilation nötig waren, zur Ausführung. Die .pst Datei wird als Parameter übergeben.

Wichtig: Es muss genügend Arbeitsspeicher zugeteilt werden (ungefähre Größe der Pst Datei) um Java-OutofMemory Errors zu verhindern.

java -Xmx12g -cp .:jpst-1.0.jar:json-simple-1.1.1.jar PstMailJSON Datei.pst

Die extrahierten Daten werden im json Format im Ordner data abgelegt.
Die E-Mail anhänge werden im Ordner Attachments abgelegt.
Die Ordner data und Attachments müssen vor der Ausführung des Programms erstellt werden. Das Program ist für UNIX ähnliche Maschinen konfiguriert, es müssen daher die Fade von / in \ geändert werden(Zeile 217 und 470), sollte es auf Windows Geräten zum Einsatz kommen.

Funktionsweise

Das Program kriegt die .pst Datei als Input. Initialisiert sie und geht alle Ordner und Mails (Items) durch. Die ids der Mails dienen als Name für das json. Da sich die id überschneiden können, wenn man mehrere pst Daten extrahiert, generiert das Program eine neue id. Überprüft wird dies in dem festgestellt wird ob die Datei schon im data Ordner existiert. Die Mail wird als Item dann an die Funktionen weiter gegeben die die Date extrahieren. Die Attachments enthalten im Namen die id des json/der Mail.

Weiterverarbeitung der Extrahierten daten

Die jsons werden an CloverETL übergeben und in einen Elasticsearch such Index geschrieben. Der Datenpath wird an CloverETL mit einem * übergeben (z.B: ./data/*.json). CloverETL schreibt die daten 1:1 in den Elasticsearch Suchindex http://hacking-team/mail.

Elasticsearch

Elasticsearch muss von der Herrstellerseite (https://www.elastic.co) heruntergeladen werden.

Es müssen folgende dinge an der datei config/elasticsearch.yml beigefügt werden:

http.cors.enabled: true
http.cors.allow-origin: "/./"*

Es muss vor dem ausführen des CloverETL-graph gestartet werden.

Kibana

Der Apache server muss konfiguriert werden. (Root Privilegien werden benötigt)

Mac OSX
/etc/apache/httpd.conf

Das php-Module muss aktiviert werden:

LoadModule php5_module libexec/apache2/libphp5.so

Und die Berechtigungen geändert werden

```
<Directory />
        AllowOverride none
        Order allow,deny
        allow from all
#       Require all denied
</Directory>

<Directory "/Library/WebServer/Documents">
        Options Indexes FollowSymLinks Multiviews
        MultiviewsMatch Any
        AllowOverride All
        Require all granted
</Directory>
```

Kibana muss aus dem Spiegel Hausinternen git-repository ausgecheckt werden. Der Inhalt muss dann den Ordner /Library/WebServer/Documents kopiert werden. (Sicherstellen das **.htaccess** mit kopiert wurde.)

Apache Server starten: *sudo apachectl start*

Kibana kann nun über http://localhost aufgerufen werden.