Refactoring :: Hintergründe, Designfragen und Ziele

`mapping_old` funktioniert im Prinzip, hat aber den Schönheitsfehler, dass alle Dateien für den Vergleich komplett in den Speicher geladen werden; und das gleich mehrmals. Das lauffähige Script ist samt Konfigurationsfiles im Repository enthalten. Auf Seite 2, Bedienung und Ausgabe, sind Beispiele mit Aufruf und Ausgabe zu finden.

Das Refactoring zielt darauf ab, nur Zeilen die »matchen« im Speicher als Hash zu hinterlegen und die fehlenden Daten als anonymes Array dem jeweiligen Schlüssel hinzuzufügen (»mergen«).
`mapping_neu` ist ein erster Entwurf mittels **`grep`** die passenden Zeilen aus zwei Datensätzen zu finden; allerdings mit dem Schönheitsfehler, dass die Zuordnung der Daten zum Hash innerhalb des BLOCKS von **`grep`** erfolgt.

Meine Frage zum »Design/Best Practice« an den Channel:
Ist es eleganter innerhalb eines **`map`** BLOCKS { **`m/$pattern/`** } übereinstimmende Zeilen zu selektieren und bei entsprechenden Treffern die Daten zu »mappen«?

Bedienung und Ausgabe

Die folgenden Beispiele veranschaulichen die Bedienung samt Ausgabe. Es können für **`--vpn_id`** und **`--ftam_id`** mehrere Werte übergeben werden; auch ist eine Kombination, beispielsweise zusätzlich mit **`--err_log`**, möglich.

```
mapping_old -v 1076
FTJUH1  172.16.200.74   102     1076    J & H Limited           StrongSwan
```

```
mapping_old -t FTXXX1 -v 1006 4001
FTBAR1  172.16.196.186  4800    1006    Bar Limited             Palo Alto
FTYYY1  172.16.205.234  4800    4001    Dummy Yummy Dataset     Cisco ASR 1002
FTXXX1  172.16.128.10   102     1001    Ritter Testing          Cisco 866
```

```
mapping_old -t FTBLA1 --err_log 0_FTYYY1.tst
FTBLA1  172.16.192.58   4800    4111    Bla Blubb & Fun         Lancom 1781
FTYYY1  172.16.205.234  4800    4001    Dummy Yummy Dataset     Cisco ASR 1002
```

Was in `mapping_old` noch nicht möglich ist:

1) Die Suche mit **`--ip_addr`** anhand der IP Adresse (wie auf Seite fünf beschrieben)
2) **`sub print_list`** wirft Fehler, wenn Datensätze unvollständig sind (in Folge nicht vorhanden seins)

Übergabe der Argumente beim Starten des Scripts

Welche Datensätze »zusammengeführt« werden sollen, können als Kommandozeilenargumente und/oder als Datei übergeben werden. Beim Starten des Scripts sollen über Getopt::Long vier Argumente für die zu verarbeitenden Datensätze übergeben werden können:

```
GetOptions (\%OPT,
    'run|r',
    'debug|x',
    'man',
    'map=s',
    'help',
    'lsped=s',
    'err_log|f=s'                               # error_log im Format @hw_list
    'vpn_id|v=i{1,}'    => \&shift_values,      # vpn_id
    'ftam_id|t=s{1,}'   => \&shift_values,      # ftam_id
    'ip|i=s{1,}'        => \&shift_values       # ip_address
) or pod2usage(2);
```

Verarbeitung der übergebenen Argumente :: Matching I

Zeilen mit passender **--vpn_id** und/oder **--ftam_id** aus @hw_list extrahieren und die entsprechenden Zeilen aus @logsped über die **$id** extrahieren. Beispiel: —vpn_id **1001 1129** —ftam_id **FTBAZ1**

@hw_list

```
# Format kann nötigenfalls angepasst werden
#  $id,$vpn_id,$participants,$hw
FTXXX1,1001,Ritter Testing,Cisco 866
FTFOO1,1129,Foo Limited,CheckPoint
FTBAR1,1006,Bar Limited,Palo Alto
FTBAZ1,1074,Baz Limited,SonicWall NSA 3200
FTJUH1,1076,J & H Limited,StrongSwan
```

@logsped

```
# Format der Datenquelle ist unveränderbar! undef == nicht benötigte Spalten
# $id    undef $ip_address      undef undef                              $port  undef
FTABB1   2630  172.16.134.138   X     oT1G7CbPbXOX                       102    FTAM
FTZYX1   5661  172.16.130.106   X     vXuIfNLH                           4800   SNI-FTAM
FTFOO1   5377  172.16.142.90    X     592w4Rts3R                         4800   SNI-FTAM
FTXYZ1   2528  172.16.128.117   X     9gVhAzIpVS                         102    FTAM
FTAAA1   5858  172.16.129.58    X     1bCrqNXvTF00                       4800   SNI-FTAM
FTBBB1   2884  172.16.197.27    X     gJIHNTyp                           102    FTAM
FTBAZ1   9940  172.16.192.10    X     Rb5EI3fm2t                         102    SNI-FTAM
FTBLA1   1596  172.16.192.58    X     glm4azkpute                        4800   SNI-FTAM
FTBLB1   3033  172.16.141.186   X     XsjLfOm99M                         4800   SNI-FTAM
FTXXX1   4711  172.16.128.10    X     axOdwigwZJWT                       102    ANY-FTAM
```

Verarbeitung der übergebenen Argumente :: Matching II

Zeilen mit passender **--ip_address** aus @logspeed extrahieren und die entsprechenden Zeilen aus @hw_list über die **$id** extrahieren. Beispiel: --ip_address 172.16.128.10 172.16.192.10 172.16.129.58

@logsped

```
# Format der Datenquelle ist unveränderbar! undef == nicht benötigte Spalten
# $id    undef $ip_address      undef undef                              $port  undef
FTABB1   2630  172.16.134.138   X     oT1G7CbPbXOX                       102    FTAM
FTZYX1   5661  172.16.130.106   X     vXuIfNLH                           4800   SNI-FTAM
FTFOO1   5377  172.16.142.90    X     592w4Rts3R                         4800   SNI-FTAM
FTXYZ1   2528  172.16.128.117   X     9gVhAzIpVS                         102    FTAM
FTAAA1   5858  172.16.129.58    X     1bCrqNXvTF00                       4800   SNI-FTAM
FTBBB1   2884  172.16.197.27    X     gJIHNTyp                           102    FTAM
FTBAZ1   9940  172.16.192.10    X     Rb5EI3fm2t                         102    SNI-FTAM
FTBLA1   1596  172.16.192.58    X     glm4azkpute                        4800   SNI-FTAM
FTBLB1   3033  172.16.141.186   X     XsjLfOm99M                         4800   SNI-FTAM
FTXXX1   4711  172.16.128.10    X     axOdwigwZJWT                       102    ANY-FTAM
```

@hw_list

```
# Format kann nötigenfalls angepasst werden
#  $id,$vpn_id,$participants,$hw
FTXXX1,1001,Ritter Testing,Cisco 866
FTFOO1,1129,Foo Limited,CheckPoint
FTBAR1,1006,Bar Limited,Palo Alto
FTBAZ1,1074,Baz Limited,SonicWall NSA 3200
FTJUH1,1076,J & H Limited,StrongSwan
FTAAA1,4211,Tripple A Ltd.,Fortigate 20D
```

Zu verarbeitenden Datensätze als File übergeben

Wenn die @hw_list als Argument übergeben wird, dann sollen alle entsprechenden Zeilen aus @logsped über die **$id** extrahiert werden.

@hw_list

```
# Format kann nötigenfalls angepasst werden
#  $id,$vpn_id,$participants,$hw
FTXXX1,1001,Ritter Testing,Cisco 866
FTFOO1,1129,Foo Limited,CheckPoint
FTBAR1,1006,Bar Limited,Palo Alto
FTBAZ1,1074,Baz Limited,SonicWall NSA 3200
```

Variante 4:
Die zu verarbeitenden Datensätze werden im »aufbereiteten Format« als Datei übergeben (siehe Option err_log).

@logsped

```
# Format der Datenquelle ist unveränderbar! undef == nicht benötigte Spalten
# $id    undef $ip_address      undef undef                          $port   undef
FTABB1   2630  172.16.134.138   X     oT1G7CbPbXOX                   102     FTAM
FTZYX1   5661  172.16.130.106   X     vXuIfNLH                       4800    SNI-FTAM
FTFOO1   5377  172.16.142.90    X     592w4Rts3R                     4800    SNI-FTAM
FTXYZ1   2528  172.16.128.117   X     9gVhAzIpVS                     102     FTAM
FTAAA1   5858  172.16.129.58    X     1bCrqNXvTF00                   4800    SNI-FTAM
FTBBB1   2884  172.16.197.27    X     gJIHNTyp                       102     FTAM
FTBAZ1   9940  172.16.192.10    X     Rb5EI3fm2t                     102     SNI-FTAM
FTBLA1   1596  172.16.192.58    X     glm4azkpute                    4800    SNI-FTAM
FTBAR1   3033  172.16.141.186   X     XsjLfOm99M                     4800    SNI-FTAM
FTXXX1   4711  172.16.128.10    X     axOdwigwZJWT                   102     ANY-FTAM
```

Übereinstimmende Zeilen beider Dateien »zusammenführen«

Daten der übereinstimmenden Zeilen aus **@logsped** und **@hw_list** in einem Hash zusammenführen

@logsped

```
# Format der Datenquelle ist unveränderbar! undef == nich
# $id   undef $ip_address     undef undef
FTZYX1  5661  172.16.130.106  X  vXuIfNLH
FTFOO1  5377  172.16.142.90   X  592w4Rts3R
FTBBB1  2884  172.16.197.27   X  gJIHNTyp
FTBAZ1  9940  172.16.192.10   X  Rb5EI3fm2t
FTBLA1  1596  172.16.192.58   X  glm4azkpute
FTBAR1  3033  172.16.141.186  X  XsjLfOm99M
FTXXX1  4711  172.16.128.10   X  axOdwigwZJWT
```

@hw_list

```
# Format kann nötigenfalls angepasst werden
#  $id,$vpn_id,$participant,$hw
FTXXX1,1001,Ritter Testing,Cisco 866
FTFOO1,1129,Foo Limited,CheckPoint
FTBAR1,1006,Bar Limited,Palo Alto
FTJUH1,1076,J & H Limited,StrongSwan
FTBAZ1,1074,Baz Limited,SonicWall NSA 3200
FTZYX1,4211,RIP Zyx Records,Fortigate 220D
```

%Merged_Data

```
# Datenstruktur und zusammengeführte Inhalte aus den Datenquellen @logsped und @hw_list
# Die Reihenfolge der Indizes im Anonymen Array spielt keine Rolle
%Merged_Data = (
    'FTXXX1' => [
        '172.16.128.10',          # $ip_address from @logsped
        '102',                    # $port from @logsped
        '1001',                   # $vpn_id from @hw_list
        'Ritter Testing,          # $participant from @hw_list
        'Cisco 866'               # $hw from @hw_list
    ]
);
```

Erweiterungen nach dem Refactoring

In der finalen Version möchte ich noch eine Statistik in folgender Art implementieren (nur falls das schon beim Design berücksichtigt werden muss):

Von 1200 Einträgen waren 75 % (700) erreichbar

Von 1200 Einträgen waren 25 % (300) nicht erreichbar

Verarbeitungsdauer: nn Sekunden

Zusätzlich sollen zwei Logfiles mit den jeweils erfolgreichen und missglückten Tests erstellt werden.