MySQL Queries on "Nmap Results"

Ivan Bütler – 31. August 2009

SQL Abfragen auf Nmap Ergebnisse

Wer den Portscanner "NMAP" häufig benutzt weiss, dass die Auswertung von grossen Scans mit vielen C- oder sogar B- Netzen viel Zeit und etwas Geschick mit Regexp verlangt.

Dieser Artikel beschreibt die Idee, die Daten von NMAP in eine MySQL Datenbank zu speichern und dort mit komfortablen SQL Queries anstatt der langen Regexp zu analysieren. In diesem kleinen Artikel lernen Sie mehr über dieses Unterfangen.

Scanning mit NMAP

Im ersten Schritt steht natürlich das Scanning mit nmap. Für diesen Bericht habe ich folgenden Scan durchgeführt. Es geht jetzt weniger um den Scan, sondern vielmehr um die Ergebnisverarbeitung.

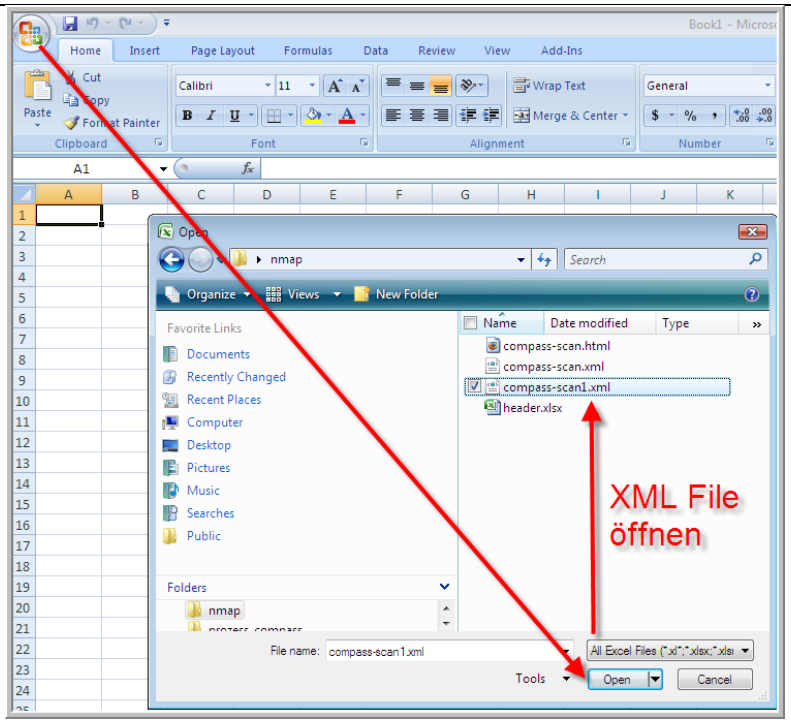
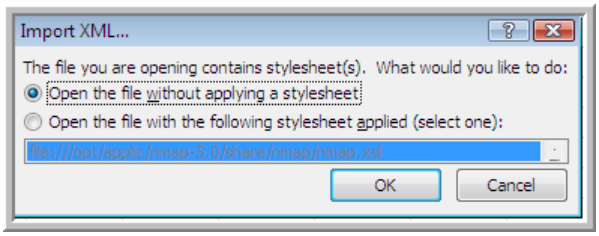
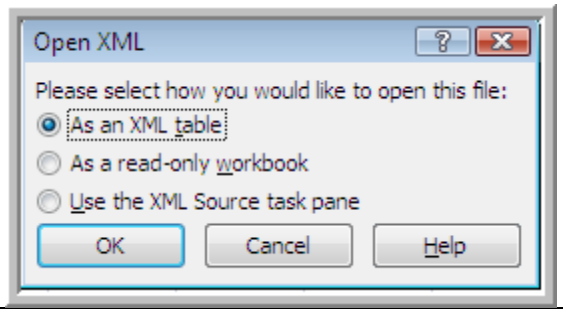
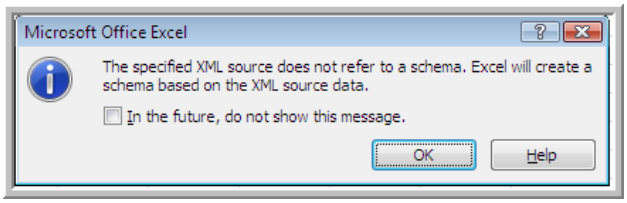
```
nmap -iL hostlist -T3 -oA /opt/data/nmap/compass-security/compass-scan1 --reason -P0 -sV -sC -O
```

Dieser Befehl mit der Option "-A" erzeugt folgende Files

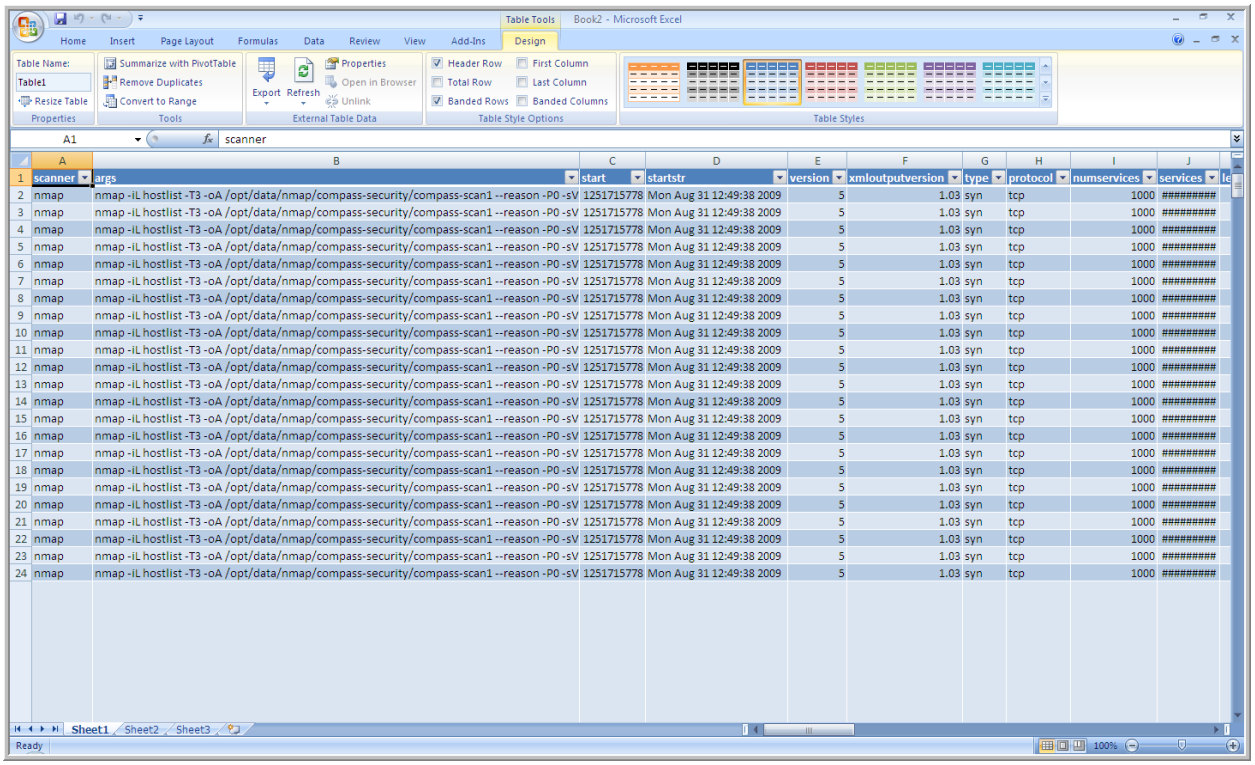
```
ibuetler@lagas compass $ ls -al compass-scan1.*

-rw-r--r--  1 ibuetler  staff    6693 Aug 31 11:45 compass-scan1.gnmap
-rw-r--r--  1 ibuetler  staff   65200 Aug 31 11:45 compass-scan1.nmap
-rw-r--r--  1 ibuetler  staff  109660 Aug 31 11:45 compass-scan1.xml
```

Konvertieren des XML Output in ein CSV

Für den Import der NMAP Ergebnisse in die MySQL Datenbank habe ich das CSV Format gewählt. Komfortabler wäre möglicherweise eine Transformation via XSLT – hier wäre ich dankbar für Feedback der Leserschaft. Für die Umwandlung von XML in CSV verwende ich Microsoft Excel 2007.

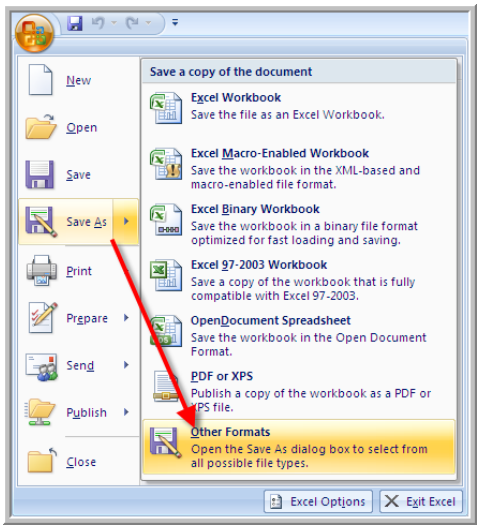
	Die XML Daten des NMAP Scans habe ich mit Microsoft Office 2007 gelöst. Dazu muss man lediglich Excel starten und die XML Datei öffnen. Es erscheint daraufhin folgender Dialog:
	Hier sollte man den Default belassen und einfach OK drücken (Ohne die Wahl des NMAP.xsl)
	Beim nächsten Dialog ebenfalls "As an XML table" belassen.
	Nun erstellt Excel automatisch ein Schema anhand der XML Datei und das bedeutet einen Zeile pro Ereigniss.

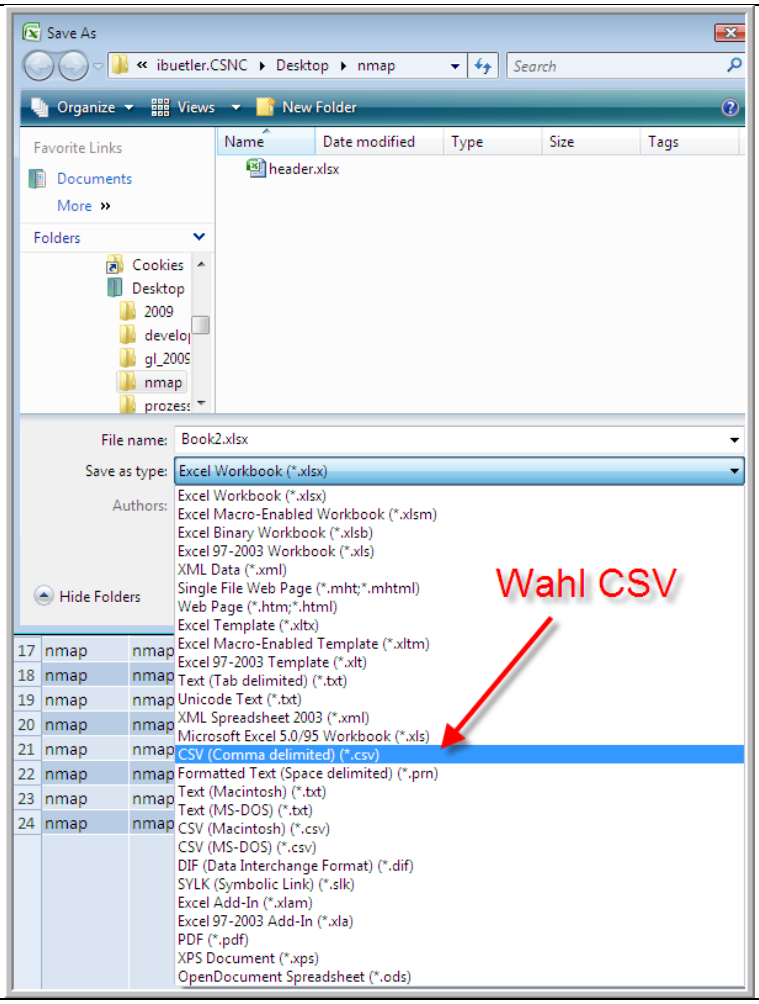
Daraufhin öffnet Excel die XML Datei. Unten ein Screenshot der geöffneten Datei.



scanner	args	start	startstr	version	xmloutputversion	type	protocol	numservices	services
nmap	nmap -iL hostlist -T3 -oA /opt/data/nmap/compass-security/compass-scan1 --reason -P0 -sV	1251715778	Mon Aug 31 12:49:38 2009	5	1.03	syn	tcp	1000	##########

Wer NMAP Ergebnisse als Excel wünscht, der ist jetzt fertig mit lesen. Für den Import in die MySQL Datenbank muss als nächstes die Excel Datei als CSV abgespeichert werden.

	Dazu wählt man "Save As" und wählt "Other Formats"



Man wähle das CSV und speichere die Datei als "compass-scan1.csv" ab.

Bezeichnung der Spalten

In der ersten Zeile des CSV stehen nun die Bezeichnungen der Spalten – und diese gilt es als Spalten in MySQL zu erzeugen. Um an die Spaltennamen zu kommen verwende ich folgenden Befehl:

```
ibuetler@lagas compass $ head -1 compass-scan1.csv

scanner,args,start,startstr,version,xmloutputversion,type,protocol,numservices,services,level,level2,starttime,endtime,state,reason,addr,addrtype,name,type3,state4,count,reason5,count6,protocol7,portid,state8,reason9,reason_ttl,name10,servicefp,method,conf,tunnel,product,extrainfo,devicetype,ostype,version11,hostname,id,output,state12,proto,portid13,type14,vendor,osfamily,accuracy,osgen,name15,accuracy16,line,srtt,rttvar,to,value,time,timestr,elapsed,up,down,total
```

Damit sind die Spalten des CSV bekannt, welche für die Erstellung der MySQL Tabellen notwendig sind.

MySQL Datenbank Erzeugung

Nun geht es darum, die neue Datenbank und NMAP Tabelle in MySQL zu erfassen. Das folgende Script übernimmt die Hauptarbeit (aber nicht alles) und nimmt als Grundlage die vorher generierte Datei compass-scan1.csv.

ibuetler@lagas compass $ cat do_create_final.sh

```
rm compass-scan1-header.txt
rm create_compass.sql
rm final.sql

head -1 compass-scan1.csv > compass-scan1-header.txt
perl -p -i -e 's/,/\n/g' compass-scan1-header.txt
cat compass-scan1-header.txt | awk '{print "csnc"$1 " varchar(100), \\"}' >> create_compass.sql
echo "create database compass;" > final.sql
echo "use compass;" >> final.sql
echo "create table compass (\\" >> final.sql
cat create_compass.sql >> final.sql
echo ");" >> final.sql
```

Nun hat das final.sql noch einen Fehler auf der zweitletzten Zeile. Diese Zeile muss man noch manuell korrigieren indem man das "**^M**" und "**,**" am Schluss korrigiert entfernt.

FALSCH: csnctotal^M varchar(100), \
RICHTIG: csnctotal varchar(100)

Nun steht eigentlich der Erzeugung der DB nichts mehr im Wege unter Anwendung des neu generierten final.sql

```
ibuetler@lagas compass $ mysql -u root -p < final.sql
Enter password:
ibuetler@lagas compass $
```

Import CSV in MySQL

Nachdem die Datenbank und Tabelle "compass" erzeugt wurde, können nun die Daten aus dem CSV in die MySQL geladen werden. Bevor wir die CSV jedoch laden, sollte man die erste Zeile des CSV entfernen (Titel der Spalten).

Danach das CSV mit folgenden Befehlen importiert werden.

ibuetler@lagas compass $ cat do_import.sh

```
zeilen=`wc -l compass-scan1.csv | awk '{print $1}'`

tail -$((zeilen-1)) compass-scan1.csv > compass-scan1.1.csv

mysql -u root -p compass -e "load data local infile '/Users/ibuetler/compass/compass-scan1.1.csv' into table compass fields terminated by ',' lines terminated by '\r\n';";
```

Erste Tests mit den NMAP Ergebnisse

Um den korrekten Import zu prüfen, empfiehlt sich folgender Befehl

```
mysql> use compass
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select distinct csncscanner from compass;

+-------------+
| csncscanner |
+-------------+
| nmap        |
+-------------+

1 row in set (0.00 sec)

mysql>
```

Wenn es "nur" eine Zeile gibt, dann scheint auf den ersten Blick alles ok zu sein.

Suche nach Apache Servern

```
mysql> select csncaddr,csnchostname,csncportid,csncproduct from
compass where csncstate8="open" and csncproduct like '%apache%' order
by csncproduct;
```

```
+-----------------+--------------+------------+--------------+
| csncaddr        | csnchostname | csncportid | csncproduct  |
+-----------------+--------------+------------+--------------+
| 212.254.178.XXX |              | 80         | Apache httpd |
| 212.254.178.XXX |              | 443        | Apache httpd |
+-----------------+--------------+------------+--------------+
2 rows in set (0.04 sec)
```

Suche nach Open SSH Ports

```
mysql> select csncaddr,csnchostname,csncportid,csncproduct from
compass where csncstate8="open" and csncproduct like '%ssh%' order by
csncproduct;
```

```
+-----------------+--------------+------------+-------------+

| csncaddr        | csnchostname | csncportid | csncproduct |

+-----------------+--------------+------------+-------------+

| 212.254.178.XXX |              | 22         | OpenSSH     |
| 212.254.178.XXX |              | 22         | OpenSSH     |
| 212.254.178.XXX |              | 22         | OpenSSH     |
+-----------------+--------------+------------+-------------+
```

More SQL Queries

Nun sind die Daten in der MySQL Datenbank und man kann diese selbstverständlich auch noch mit

Fremd-Daten wie GeoIP korrellieren. Dies aber ein anderes mal...

Tipp im Nachtrag

Nur noch für diejenigen, die am liebsten einen HTML Output von NMAP haben würden. Dies ist mit den Standard-Funktionen von NMAP bereits sehr einfach realisierbar mit xsltproc.

http://nmap.org/book/output-formats-output-to-html.html

```
xsltproc compass-scan1.xml -o /tmp/compass-scan1.html
```

Thank You

Vielen Dank für Ihr Interesse. Feedback as always – is welcomed.

31. August 2009 by Ivan Bütler, alias e1