In [1]:

```
import subprocess
import json
from matplotlib import pyplot as plt


import findspark
findspark.init('/home/ubuntu/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7')
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext
from pyspark import sql
from pyspark.sql import functions as F
print('Hello bii')

SparkContext.setSystemProperty('spark.executor.memory', '30g')
conf = SparkConf().set("spark.executor.memory", "30G")
print(conf)
sc= SparkContext()
sc.setLogLevel("ERROR")
sqlContext = sql.SQLContext(sc)
```

```
Hello bii
<pyspark.conf.SparkConf object at 0x7f1b14e003c8>
```

In [2]:

```
from matplotlib import pyplot as plt
```

In [3]:

```
sc
```

Out[3]:

SparkContext

[Spark UI (http://45.113.233.20:4040)](http://45.113.233.20:4040)

Version

`v2.2.3`

Master

`spark://datacollect2.novalocal:7077`

AppName

`pyspark-shell`

In [4]:

```
files = !ls /home/ubuntu/MyVolumeStore/Virustotal_Responses/*.json
```

In [4]:

```
files
```

Out[4]:

```
['/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_307.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_308.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_309.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_310.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_311.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_312.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_313.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_314.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_315.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_316.json']
```

In [6]:

```
---------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-6-66c8e8b7b1b5> in <module>
----> 1 print ("file writen %s"%file)

NameError: name 'file' is not defined
```

In [23]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.filter(
        F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
        ).select(F.col("md5"),F.explode(
                        F.col("additional_info").getItem("sigcheck").getItem("counter signers details").getItem("cert issuer")
                ).alias("counter_signers_details"))
        df.repartition(4 if fsize > 100 else 2).write.mode("append").parquet("processed_parquet")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_virushashes_316.json
```

In [9]:

```
import os
```

In [25]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/processed_parquet').count()
```

Out[25]:

```
10332
```

In [29]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select(F.col("md5")).write.mode("append").parquet("md5_parquet_s
econd")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [30]:

```
sqlContext.read.json('/home/ubuntu/MyVolumeStore/md5_parquet_second').count()
```

Out[30]:

```
7312
```

In [39]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select(F.col("md5")).write.mode("append").parquet("type_parquet"
)
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [41]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/type_parquet').select().show
()
```

```
+--------------------+
|                 md5|
+--------------------+
|d822e8ce21bef84ca...|
|0dba64a8b3a6da6c9...|
|a29f9383ab57c5fb6...|
|d9085bc83c9e5ad9e...|
|021253f13b7b61a42...|
|d918693704383eeea...|
|d94660310686da66b...|
|d94bebaa012c10f69...|
|d990259151b769511...|
|d8edecb902b98ce17...|
|da103663a071ef016...|
|da5f53e6cc44c680c...|
|da57fbe63064240f4...|
|f1c19fd27ead96167...|
|2baf58ab708dfafe1...|
|da815fa364bda8953...|
|da9e54c9560928b9f...|
|f7e223e9004aed80e...|
|048cdd9f9c2703a89...|
|0475d1faf81f7fd49...|
+--------------------+
only showing top 20 rows
```

In [47]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select(F.col("md5"),F.explode(
                        F.col("additional_info").getItem("sigcheck").getItem("co
unter signers details").getItem("cert issuer")
                ).alias("counter_signers_details")).write.mode("append").parquet
("analysis/countersigners_parquest")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [50]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parq
uest').show(truncate=False)
```

```
+--------------------------------+----------------------------------
-----+
|md5                             |counter_signers_details
|
+--------------------------------+----------------------------------
-----+
|984ccc3d8bebe798bdb07f0eb7af707b|Symantec Time Stamping Services CA
- G2|
|984ccc3d8bebe798bdb07f0eb7af707b|Thawte Timestamping CA
|
|984ccc3d8bebe798bdb07f0eb7af707b|Thawte Timestamping CA
|
|df3174a5a87cd8ecbc7aa989bc16807e|GlobalSign Timestamping CA - G2
|
|df3174a5a87cd8ecbc7aa989bc16807e|GlobalSign Root CA
|
|df3174a5a87cd8ecbc7aa989bc16807e|GlobalSign Root CA
|
|907d3510c4cc87ea1d7cdec202f5d183|GlobalSign Timestamping CA - G2
|
|907d3510c4cc87ea1d7cdec202f5d183|GlobalSign Root CA
|
|907d3510c4cc87ea1d7cdec202f5d183|GlobalSign Root CA
|
|005fbb5538daacf13a447e9fa4fa7abe|Symantec Time Stamping Services CA
- G2|
|005fbb5538daacf13a447e9fa4fa7abe|Thawte Timestamping CA
|
|005fbb5538daacf13a447e9fa4fa7abe|Thawte Timestamping CA
|
|a17032ed2687dc9f3c6a1ffe66ff30d6|GlobalSign Timestamping CA - G2
|
|a17032ed2687dc9f3c6a1ffe66ff30d6|GlobalSign Root CA
|
|a17032ed2687dc9f3c6a1ffe66ff30d6|GlobalSign Root CA
|
|f618e4c8d420fe8866076d232bbace10|Symantec Time Stamping Services CA
- G2|
|f618e4c8d420fe8866076d232bbace10|Thawte Timestamping CA
|
|f618e4c8d420fe8866076d232bbace10|Thawte Timestamping CA
|
|a3b005981f882b90259d4dfb1cf7316e|GlobalSign Timestamping CA - G2
|
|a3b005981f882b90259d4dfb1cf7316e|GlobalSign Root CA
|
+--------------------------------+----------------------------------
-----+
only showing top 20 rows
```

In [64]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquest')\
.where("lower(counter_signers_details) LIKE '%time%stamping%'")\
.select("counter_signers_details").distinct().count()
```

Out[64]:

```
11
```

In [17]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquest').count()
```

Out[17]:

```
10506
```

In [66]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquest')\
.where("lower(counter_signers_details) LIKE '%time%stamping%'")\
.select("counter_signers_details").distinct().show(truncate=False)
```

```
+----------------------------------------+
|counter_signers_details                 |
+----------------------------------------+
|GlobalSign Timestamping CA - G2         |
|Symantec SHA256 TimeStamping CA         |
|Symantec Time Stamping Services CA - G2 |
|DigiCert SHA2 Assured ID Timestamping CA|
|Entrust Timestamping CA - TS1           |
|WoSign Time Stamping Services CA G2     |
|GlobalSign Timestamping CA - SHA256 - G2|
|Microsoft Timestamping PCA              |
|Thawte Timestamping CA                  |
|VeriSign Time Stamping Services CA      |
|GlobalSign Timestamping CA              |
+----------------------------------------+
```

In [70]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquest')\
.where("lower(counter_signers_details) LIKE '%time%stamping%'")\
.groupby("counter_signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+----------------------------------------+----+
|counter_signers_details                 |md5 |
+----------------------------------------+----+
|GlobalSign Timestamping CA - G2         |518 |
|Symantec SHA256 TimeStamping CA         |113 |
|Symantec Time Stamping Services CA - G2 |1867|
|DigiCert SHA2 Assured ID Timestamping CA|17  |
|WoSign Time Stamping Services CA G2     |25  |
|Entrust Timestamping CA - TS1           |31  |
|GlobalSign Timestamping CA - SHA256 - G2|61  |
|Microsoft Timestamping PCA              |2   |
|Thawte Timestamping CA                  |2089|
|VeriSign Time Stamping Services CA      |222 |
|GlobalSign Timestamping CA              |4   |
+----------------------------------------+----+
```

In [6]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquest')\
.where("lower(counter_signers_details) LIKE '%time%stamping%'")\
.withColumn("Signers",F.split(F.col("counter_signers_details")," ").getItem(0))\
.groupby("Signers").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+----------+----+
|Signers   |md5 |
+----------+----+
|GlobalSign|583 |
|DigiCert  |17  |
|Entrust   |31  |
|WoSign    |25  |
|Symantec  |1980|
|Microsoft |2   |
|Thawte    |2089|
|VeriSign  |222 |
+----------+----+
```

In [78]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select(F.col("md5"),F.explode(
                        F.col("additional_info").getItem("sigcheck").getItem("si
gners details").getItem("cert issuer")
                ).alias("signers_details")).write.mode("append").parquet("analys
is/signers_parquest")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [79]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest').
show(truncate=False)
```

```
+--------------------------------+-----------------------------------
--------------------------+
|md5                             |signers_details
|
+--------------------------------+-----------------------------------
--------------------------+
|d822e8ce21bef84ca1096038a4e2aad3|VeriSign Class 3 Code Signing 2010
CA                        |
|d822e8ce21bef84ca1096038a4e2aad3|VeriSign Class 3 Public Primary Ce
rtification Authority - G5|
|d822e8ce21bef84ca1096038a4e2aad3|VeriSign Class 3 Public Primary Ce
rtification Authority - G5|
|a29f9383ab57c5fb6b24948c022ef89a|GlobalSign CodeSigning CA - SHA256
- G3                      |
|a29f9383ab57c5fb6b24948c022ef89a|GlobalSign
|
|a29f9383ab57c5fb6b24948c022ef89a|GlobalSign
|
|021253f13b7b61a42bb78e98d5118eda|GlobalSign CodeSigning CA - SHA256
- G3                      |
|021253f13b7b61a42bb78e98d5118eda|GlobalSign
|
|021253f13b7b61a42bb78e98d5118eda|GlobalSign
|
|d94660310686da66b7b660e045d4c33b|GlobalSign CodeSigning CA - SHA256
- G3                      |
|d94660310686da66b7b660e045d4c33b|GlobalSign
|
|d94660310686da66b7b660e045d4c33b|GlobalSign
|
|d94bebaa012c10f69ef5d6a7dbd11d30|GlobalSign CodeSigning CA - SHA256
- G3                      |
|d94bebaa012c10f69ef5d6a7dbd11d30|GlobalSign
|
|d94bebaa012c10f69ef5d6a7dbd11d30|GlobalSign
|
|d990259151b7695114f1625582e27e75|VeriSign Class 3 Code Signing 2010
CA                        |
|d990259151b7695114f1625582e27e75|VeriSign Class 3 Public Primary Ce
rtification Authority - G5|
|d990259151b7695114f1625582e27e75|VeriSign Class 3 Public Primary Ce
rtification Authority - G5|
|da103663a071ef0162d95e93e95d6944|VeriSign Class 3 Code Signing 2010
CA                        |
|da103663a071ef0162d95e93e95d6944|VeriSign Class 3 Public Primary Ce
rtification Authority - G5|
+--------------------------------+-----------------------------------
--------------------------+
only showing top 20 rows
```

In [80]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%code%signing%'")\
.groupby("signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+-----------------------------------------------------------+-----+
|signers_details                                            |md5  |
+-----------------------------------------------------------+-----+
|VeriSign Class 3 Code Signing 2009 CA                      |5    |
|thawte SHA256 Code Signing CA                              |211  |
|GlobalSign CodeSigning CA - SHA256 - G2                    |34   |
|DigiCert EV Code Signing CA (SHA2)                         |82   |
|WoSign Class 3 Code Signing CA                             |100  |
|Symantec Class 3 Extended Validation Code Signing CA - G3  |10   |
|Microsoft Code Signing PCA                                 |14   |
|WoSign Class 3 Code Signing CA G2                          |10   |
|Symantec Class 3 Extended Validation Code Signing CA       |6    |
|DigiCert EV Code Signing CA                                |7    |
|Thawte Code Signing CA                                     |11   |
|Entrust Code Signing CA - OVCS1                            |32   |
|Symantec Class 3 SHA256 Code Signing CA - G2               |1    |
|GlobalSign CodeSigning CA - G2                             |70   |
|VeriSign Class 3 Code Signing 2001 CA                      |2    |
|GlobalSign Extended Validation CodeSigning CA - SHA256 - G3|12   |
|GlobalSign CodeSigning CA - SHA256 - G3                    |11879|
|VeriSign Class 3 Code Signing 2009-2 CA                    |68   |
|DigiCert SHA2 Assured ID Code Signing CA                   |61   |
|COMODO Code Signing CA                                     |3    |
+-----------------------------------------------------------+-----+
only showing top 20 rows
```

In [81]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%code%signing%'")\
.withColumn("Signers",F.split(F.col("signers_details")," ").getItem(0))\
.groupby("Signers").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+----------+-----+
|Signers   |md5  |
+----------+-----+
|GlobalSign|12031|
|DigiCert  |223  |
|thawte    |211  |
|Entrust   |32   |
|COMODO    |1221 |
|Certum    |9    |
|WoSign    |121  |
|Symantec  |643  |
|Microsoft |14   |
|Thawte    |176  |
|VeriSign  |1204 |
+----------+-----+
```

In [7]:

```
file_316 = '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_vi
rushashes_316.json'
```

In [115]:

```
sqlContext.read.json(file_316).select(F.col("scans").getItem("Avg")).schema
```

Out[115]:

```
StructType(List(StructField(scans.Avg,StructType(List(StructField(de
tected,BooleanType,true),StructField(result,StringType,true),StructF
ield(update,StringType,true),StructField(version,StringType,true))),
true)))
```

In [8]:

```
from pyspark.sql import types as t
```

In [9]:

```
StructField=t.StructField
StringType=t.StringType
List=list
BooleanType=t.BooleanType
StructType=t.StructType
dataType = t.DataType
```

In [125]:

```
sqlContext.read.json(file_316).select(F.col("scans").cast(t.MapType(t.StringType
(),StructType(List(StructField("detected",BooleanType,True),StructField("result"
,StringType,True),StructField("update",StringType,True),

StructField("version",StringType,True)))))).schema
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent cal
l last)
<ipython-input-125-f843df19bda4> in <module>
----> 1 sqlContext.read.json(file_316).select(F.col("scans").cast(t.
MapType(t.StringType(),StructType(List(StructField("detected",Boolea
nType,True),StructField("result",StringType,True),StructField("updat
e",StringType,True),
      2 
StructField("version",StringType,True)))))).schema

~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/pyspark/sql/t
ypes.py in __init__(self, name, dataType, nullable, metadata)
    401         False
    402         """
--> 403         assert isinstance(dataType, DataType), "dataType sho
uld be DataType"
    404         assert isinstance(name, basestring), "field name sho
uld be string"
    405         if not isinstance(name, str):

AssertionError: dataType should be DataType
```

In [8]:

```
df = sqlContext.read.json(file_316)
```

In [21]:

```
dir(df.scans)
```

Out[21]:

```
['__add__',
 '__and__',
 '__bool__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__div__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattr__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__invert__',
 '__iter__',
 '__le__',
 '__lt__',
 '__mod__',
 '__module__',
 '__mul__',
 '__ne__',
 '__neg__',
 '__new__',
 '__nonzero__',
 '__or__',
 '__pow__',
 '__radd__',
 '__rand__',
 '__rdiv__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__ror__',
 '__rpow__',
 '__rsub__',
 '__rtruediv__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__sub__',
 '__subclasshook__',
 '__truediv__',
 '__weakref__',
 '_endswith_doc',
 '_isNotNull_doc',
 '_isNull_doc',
 '_jc',
 '_like_doc',
 '_rlike_doc',
 '_startswith_doc',
 'alias',
 'asc',
 'astype',
 'between',
 'bitwiseAND',
 'bitwiseOR',
 'bitwiseXOR',
 'cast',
 'contains',
 'desc',
 'endswith',
 'getField',
 'getItem',
 'isNotNull',
 'isNull',
 'isin',
 'like',
 'name',
 'otherwise',
 'over',
 'rlike',
 'startswith',
 'substr',
 'when']
```

In [15]:

```
new_df = df.select(
    F.col("md5"), F.col("scans.*")
).where("positives > 2")
new_df.columns[1:]

new_df.withColumn(
    "detected_count",
    sum([
        F.when(F.col(cl).getItem("detected"), 1).otherwise(0) for cl in new_df.c
olumns[1:]
    ])
).select("md5", "detected_count").show()
```

```
+--------------------+--------------+
|                 md5|detected_count|
+--------------------+--------------+
|7ef3c7993f5d30075...|            41|
|aad2e37a5e733c140...|             8|
|729edc69880f27262...|            52|
|2863d3061e289bc50...|            47|
|7f30bd792da3934b6...|            47|
|38de2a133934dc5ef...|            45|
|8cc09e049d9a0ea1f...|            44|
|9bcb0bd9a5ac1d166...|             6|
|ebe776c97f7caba70...|            42|
|eeaf12c14e62afcc9...|            43|
|9e4a85b46c4fcb2a9...|            42|
|775542926871b5889...|            49|
|1c2d1528ee1e52407...|            27|
|84ed8a005edb039c2...|            49|
|b91799507c63792e5...|            42|
|27fc3df80771bd0ce...|            48|
|d9ccd82673815df0d...|            34|
|ce2a5974ae17e9d7c...|            43|
|dcefbad6923989cf1...|            42|
|a75e132050f5c7058...|            43|
+--------------------+--------------+
only showing top 20 rows
```

In [3]:

```
df = sqlContext.read.json(file_316)
```

```
---------------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-3-b04d96d74fd7> in <module>
----> 1 df = sqlContext.read.json(file_316)

NameError: name 'sqlContext' is not defined
```

In [16]:

```
new_df.withColumn(
    "file_type",
    F.array([
        F.col(cl).getItem("result") for cl in new_df.columns[1:]
    ])
).select(
    "md5",
    F.explode("file_type").alias("file_type")
).where("file_type != 'null'").show(truncate=False)
```

```
+--------------------------------+---------------------------------
---+
|md5                             |file_type
|
+--------------------------------+---------------------------------
---+
|7ef3c7993f5d30075432172cdd0c21da|Gen:Variant.Ursu.365454
|
|7ef3c7993f5d30075432172cdd0c21da|Win32:Adware-gen [Adw]
|
|7ef3c7993f5d30075432172cdd0c21da|suspicious
|
|7ef3c7993f5d30075432172cdd0c21da|Gen:Variant.Ursu.365454
|
|7ef3c7993f5d30075432172cdd0c21da|Adware/Win32.Adposhel.R226766
|
|7ef3c7993f5d30075432172cdd0c21da|Trojan.Ursu.D5938E
|
|7ef3c7993f5d30075432172cdd0c21da|Win32:Adware-gen [Adw]
|
|7ef3c7993f5d30075432172cdd0c21da|HEUR/AGEN.1003948
|
|7ef3c7993f5d30075432172cdd0c21da|Gen:Variant.Ursu.365454
|
|7ef3c7993f5d30075432172cdd0c21da|win/malicious_confidence_100% (D)
|
|7ef3c7993f5d30075432172cdd0c21da|malicious.93f5d3
|
|7ef3c7993f5d30075432172cdd0c21da|W32/Adware.BENU-8236
|
|7ef3c7993f5d30075432172cdd0c21da|Trojan.Adposhel.83
|
|7ef3c7993f5d30075432172cdd0c21da|a variant of Win32/Adware.Adposhe
l.AW|
|7ef3c7993f5d30075432172cdd0c21da|Gen:Variant.Ursu.365454 (B)
|
|7ef3c7993f5d30075432172cdd0c21da|malicious (high confidence)
|
|7ef3c7993f5d30075432172cdd0c21da|Heuristic.HEUR/AGEN.1003948
|
|7ef3c7993f5d30075432172cdd0c21da|W32/Adposhel.AW
|
|7ef3c7993f5d30075432172cdd0c21da|Win32.Application.OneSysCare.A
|
|7ef3c7993f5d30075432172cdd0c21da|PUA.Adposhel
|
+--------------------------------+---------------------------------
---+
only showing top 20 rows
```

In [23]:

```
new_df = df.where("positives > 2").select(
    F.col("md5"), F.col("positives"), F.col("scans.*")
)

new_df.withColumn(
    "file_type_count",
    sum([
        F.when(
            F.instr(F.lower(F.col(cl).getItem("result")), "adware") > 0,
            1
        ).when(
            F.instr(F.lower(F.col(cl).getItem("result")), "pup") > 0,
            1
        ).otherwise(0) for cl in new_df.columns[2:]
    ])
).select(
    "md5", "positives",
    F.col("file_type_count")
).show(truncate=False)
```

```
+--------------------------------+---------+---------------+
|md5                             |positives|file_type_count|
+--------------------------------+---------+---------------+
|7ef3c7993f5d30075432172cdd0c21da|41       |11             |
|aad2e37a5e733c140b3e02f9d793a572|8        |1              |
|729edc69880f2726288b973cded25880|52       |2              |
|2863d3061e289bc5092cc3dedda9e25e|47       |6              |
|7f30bd792da3934b6f9519a5a1af624e|47       |6              |
|38de2a133934dc5ef1988df54b8054a9|45       |0              |
|8cc09e049d9a0ea1fc3355292d10ce85|44       |16             |
|9bcb0bd9a5ac1d166ebbafd1879b3675|6        |1              |
|ebe776c97f7caba708f4695fcf907873|42       |2              |
|eeaf12c14e62afcc9ea898e2c2d489e6|43       |3              |
|9e4a85b46c4fcb2a950d186bbe20304d|42       |2              |
|77554292687 1b5889bc98c5c059f27f3|49       |17             |
|1c2d1528ee1e524077b21373405ababd|27       |9              |
|84ed8a005edb039c20b7bc0ad82a77f5|49       |11             |
|b91799507c63792e5e7375c458015544|42       |8              |
|27fc3df80771bd0cec791e00b6f9ed66|48       |8              |
|d9ccd82673815df0db5394032c8d6916|34       |1              |
|ce2a5974ae17e9d7c140f7ea0d4eecce|43       |0              |
|dcefbad6923989cf1501b3c85ffdc6f3|42       |15             |
|a75e132050f5c7058f0c2ed5a655b40d|43       |15             |
+--------------------------------+---------+---------------+
only showing top 20 rows
```

In [25]:

```
new_df = df.where("positives > 2").select(
    F.col("md5"), F.col("positives"), F.col("scans.*")
)

new_df.withColumn(
    "file_type_count",
    sum([
        F.when(
            F.instr(F.lower(F.col(cl).getItem("result")), "adware") > 0,
            1
        ).when(
            F.instr(F.lower(F.col(cl).getItem("result")), "pup") > 0,
            1
        ).otherwise(0) for cl in new_df.columns[2:]
    ])
).select(
    "md5", "positives",
    F.col("file_type_count")
).withColumn(
    "type",
    F.when(
        F.col("file_type_count") > (F.col("positives")/10),
        "pup"
    ).otherwise("virus")
).show(truncate=False)
```

```
+--------------------------------+---------+---------------+-----+
|md5                             |positives|file_type_count|type |
+--------------------------------+---------+---------------+-----+
|7ef3c7993f5d30075432172cdd0c21da|41       |11             |pup  |
|aad2e37a5e733c140b3e02f9d793a572|8        |1              |pup  |
|729edc69880f2726288b973cded25880|52       |2              |virus|
|2863d3061e289bc5092cc3dedda9e25e|47       |6              |pup  |
|7f30bd792da3934b6f9519a5a1af624e|47       |6              |pup  |
|38de2a133934dc5ef1988df54b8054a9|45       |0              |virus|
|8cc09e049d9a0ea1fc3355292d10ce85|44       |16             |pup  |
|9bcb0bd9a5ac1d166ebbafd1879b3675|6        |1              |pup  |
|ebe776c97f7caba708f4695fcf907873|42       |2              |virus|
|eeaf12c14e62afcc9ea898e2c2d489e6|43       |3              |virus|
|9e4a85b46c4fcb2a950d186bbe20304d|42       |2              |virus|
|77554292687 1b5889bc98c5c059f27f3|49       |17             |pup  |
|1c2d1528ee1e524077b21373405ababd|27       |9              |pup  |
|84ed8a005edb039c20b7bc0ad82a77f5|49       |11             |pup  |
|b91799507c63792e5e7375c458015544|42       |8              |pup  |
|27fc3df80771bd0cec791e00b6f9ed66|48       |8              |pup  |
|d9ccd82673815df0db5394032c8d6916|34       |1              |virus|
|ce2a5974ae17e9d7c140f7ea0d4eecce|43       |0              |virus|
|dcefbad6923989cf1501b3c85ffdc6f3|42       |15             |pup  |
|a75e132050f5c7058f0c2ed5a655b40d|43       |15             |pup  |
+--------------------------------+---------+---------------+-----+
only showing top 20 rows
```

In [26]:

```
files = !ls /home/ubuntu/MyVolumeStore/Virustotal_Responses/*.json
```

In [38]:

```
for file in files:
    df = sqlContext.read.json(file)
    new_df = df.where("positives > 2").select(
        F.col("md5"), F.col("positives"), F.col("scans.*")
    )
    #dropping additional info and other columns

    new_df.withColumn(
        "file_type_count",
        sum([
            F.when(
                F.instr(F.lower(F.col(cl).getItem("result")), "adware") > 0,
                1
            ).when(
                F.instr(F.lower(F.col(cl).getItem("result")), "pup") > 0,
                1
            ).otherwise(0) for cl in new_df.columns[2:]
        ])
    ).select(
        "md5", "positives",
        F.col("file_type_count")
    ).withColumn(
        "type",
        F.when(
            F.col("file_type_count") > (F.col("positives")/10),
            "pup"
        ).otherwise("virus")
    ).write.mode("append").parquet("analysis/pup_virus_parquet")
    print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [39]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/pup_virus_parquet')
.show(truncate=False)
```

```
+--------------------------------+---------+---------------+-----+
|md5                             |positives|file_type_count|type |
+--------------------------------+---------+---------------+-----+
|a29f9383ab57c5fb6b24948c022ef89a|45       |14             |pup  |
|d9085bc83c9e5ad9e9ce833eb0614ab7|46       |10             |pup  |
|021253f13b7b61a42bb78e98d5118eda|44       |15             |pup  |
|d918693704383eeea8d4ac89542d491b|46       |10             |pup  |
|d94660310686da66b7b660e045d4c33b|52       |17             |pup  |
|d94bebaa012c10f69ef5d6a7dbd11d30|49       |17             |pup  |
|d990259151b7695114f1625582e27e75|42       |11             |pup  |
|d8edecb902b98ce170041bccd6130c9e|43       |2              |virus|
|da103663a071ef0162d95e93e95d6944|37       |12             |pup  |
|da5f53e6cc44c680c8c1eefdd7204a20|47       |10             |pup  |
|da57fbe63064240f48d56628b5333e58|47       |10             |pup  |
|f1c19fd27ead96167ccaa7cd92b4e15a|44       |14             |pup  |
|2baf58ab708dfafe1850ec270cf9edcd|48       |16             |pup  |
|da815fa364bda89538b696ff515210da|51       |9              |pup  |
|da9e54c9560928b9f732bc3be22028e5|48       |10             |pup  |
|f7e223e9004aed80e2fdd91819c3afd4|26       |9              |pup  |
|048cdd9f9c2703a8961bb5a9aa85233f|3        |0              |virus|
|b54e372c781a7db66b6421588c29498e|47       |15             |pup  |
|db3b0149d23b54b9128ccc0b7e10e799|48       |11             |pup  |
|04f5b24332c3f8d309512259e4b481aa|3        |0              |virus|
+--------------------------------+---------+---------------+-----+
only showing top 20 rows
```

In [40]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/pup_virus_parquet')
\
.groupby("type").agg(F.count("md5").alias("frequency"))\
.show(truncate=False)
```

```
+-----+---------+
|type |frequency|
+-----+---------+
|pup  |32331    |
|virus|15417    |
+-----+---------+
```

In [3]:

```
df = sqlContext.read.json('/home/ubuntu/MyVolumeStore/Virustotal_Responses/respo
nses_windows_virushashes_316.json')
```

In [4]:

```
new_df = df.filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("counter signers details")
        ).alias("counter signers details")
).select(
        F.col("md5"),
        F.col("counter signers details").getItem("cert issuer").alias("cert issuer"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter signers details").getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter signers details").getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_to"),
    F.col("counter signers details").getItem("valid from").alias("valid from"),
    F.col("counter signers details").getItem("valid to").alias("valid to")
).withColumn("difference",F.datediff(("valid_to"),("valid_from")))
```

In [5]:

```
new_df.show(truncate=False)
```

```
+--------------------------------+-----------------------------------
-------------+----------+----------+-------------------+------------
-------+----------+
|md5                             |cert issuer
|valid_from|valid_to  |valid from         |valid to           |diffe
rence|
+--------------------------------+-----------------------------------
-------------+----------+----------+-------------------+------------
-------+----------+
|aad2e37a5e733c140b3e02f9d793a572|Symantec Time Stamping Services CA
- G2        |2012-10-17|2020-12-29|11:00 PM 10/17/2012|11:59 PM 12/2
9/2020|2995      |
|aad2e37a5e733c140b3e02f9d793a572|Thawte Timestamping CA
|2012-12-21|2020-12-30|12:00 AM 12/21/2012|11:59 PM 12/30/2020|2931
|
|aad2e37a5e733c140b3e02f9d793a572|Thawte Timestamping CA
|1997-01-01|2020-12-31|12:00 AM 01/01/1997|11:59 PM 12/31/2020|8765
|
|9bcb0bd9a5ac1d166ebbafd1879b3675|Symantec SHA256 TimeStamping CA
|2017-01-02|2028-04-01|12:00 AM 01/02/2017|10:59 PM 04/01/2028|4107
|
|9bcb0bd9a5ac1d166ebbafd1879b3675|VeriSign Universal Root Certificat
ion Authority|2016-01-12|2031-01-11|12:00 AM 01/12/2016|11:59 PM 01/
11/2031|5478      |
|9bcb0bd9a5ac1d166ebbafd1879b3675|VeriSign Universal Root Certificat
ion Authority|2008-04-01|2037-12-01|11:00 PM 04/01/2008|11:59 PM 12/
01/2037|10836     |
|a9eda36c8c9d981e525378499e363bc2|VeriSign Time Stamping Services CA
|2007-06-14|2012-06-14|11:00 PM 06/14/2007|10:59 PM 06/14/2012|1827
|
|a9eda36c8c9d981e525378499e363bc2|Thawte Timestamping CA
|2003-12-04|2013-12-03|12:00 AM 12/04/2003|11:59 PM 12/03/2013|3652
|
|a9eda36c8c9d981e525378499e363bc2|Thawte Timestamping CA
|1997-01-01|2020-12-31|12:00 AM 01/01/1997|11:59 PM 12/31/2020|8765
|
|cce94b9791ce6afa89288333c06ce731|Symantec Time Stamping Services CA
- G2        |2012-10-18|2020-12-29|12:00 AM 10/18/2012|11:59 PM 12/2
9/2020|2994      |
|cce94b9791ce6afa89288333c06ce731|Thawte Timestamping CA
|2012-12-21|2020-12-30|12:00 AM 12/21/2012|11:59 PM 12/30/2020|2931
|
|cce94b9791ce6afa89288333c06ce731|Thawte Timestamping CA
|1997-01-01|2020-12-31|12:00 AM 01/01/1997|11:59 PM 12/31/2020|8765
|
|293f3a9a9d7c2f7c55bb5e4426b19527|UTN-USERFirst-Object
|2015-12-31|2019-07-09|12:00 AM 12/31/2015|06:40 PM 07/09/2019|1286
|
|293f3a9a9d7c2f7c55bb5e4426b19527|UTN-USERFirst-Object
|1999-07-09|2019-07-09|06:31 PM 07/09/1999|06:40 PM 07/09/2019|7305
|
|fa58b1b0e6a722ff87a7da84419353d5|Symantec Time Stamping Services CA
- G2        |2012-10-17|2020-12-29|11:00 PM 10/17/2012|11:59 PM 12/2
9/2020|2995      |
|fa58b1b0e6a722ff87a7da84419353d5|Thawte Timestamping CA
|2012-12-21|2020-12-30|12:00 AM 12/21/2012|11:59 PM 12/30/2020|2931
|
|fa58b1b0e6a722ff87a7da84419353d5|Thawte Timestamping CA
|1997-01-01|2020-12-31|12:00 AM 01/01/1997|11:59 PM 12/31/2020|8765
|
|aed4ecd9a76700265118609b65321489|GlobalSign Timestamping CA - SHA25
```

```
6 - G2       |2016-05-23|2027-06-23|11:00 PM 05/23/2016|11:00 PM 06/
23/2027|4048      |
|aed4ecd9a76700265118609b65321489|GlobalSign
|2011-08-02|2029-03-29|09:00 AM 08/02/2011|09:00 AM 03/29/2029|6449
|
|aed4ecd9a76700265118609b65321489|GlobalSign Root CA
|2009-11-18|2019-03-18|10:00 AM 11/18/2009|10:00 AM 03/18/2019|3407
|
+--------------------------------+----------------------------------
-------------+----------+----------+------------------+------------
-------+----------+
only showing top 20 rows
```

In [6]:

```
new_df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("counter si
gners details")
        ).alias("counter_signers_details")
).select(
        F.col("md5"),
        F.col("counter_signers_details").getItem("cert issuer").alias("cert_issu
er"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_d
etails").getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid
_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_d
etails").getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_t
o"),
).where("lower(cert_issuer) LIKE '%time%stamping%'").withColumn("difference",F.d
atediff(("valid_to"),("valid_from")))
```

In [7]:

```
new_df.show(truncate=False)
```

```
+--------------------------------+----------------------------------
------+----------+----------+----------+
|md5                             |cert_issuer
|valid_from|valid_to  |difference|
+--------------------------------+----------------------------------
------+----------+----------+----------+
|aad2e37a5e733c140b3e02f9d793a572|Symantec Time Stamping Services CA
- G2 |2012-10-17|2020-12-29|2995      |
|aad2e37a5e733c140b3e02f9d793a572|Thawte Timestamping CA
|2012-12-21|2020-12-30|2931      |
|aad2e37a5e733c140b3e02f9d793a572|Thawte Timestamping CA
|1997-01-01|2020-12-31|8765      |
|9bcb0bd9a5ac1d166ebbafd1879b3675|Symantec SHA256 TimeStamping CA
|2017-01-02|2028-04-01|4107      |
|a9eda36c8c9d981e525378499e363bc2|VeriSign Time Stamping Services CA
|2007-06-14|2012-06-14|1827      |
|a9eda36c8c9d981e525378499e363bc2|Thawte Timestamping CA
|2003-12-04|2013-12-03|3652      |
|a9eda36c8c9d981e525378499e363bc2|Thawte Timestamping CA
|1997-01-01|2020-12-31|8765      |
|cce94b9791ce6afa89288333c06ce731|Symantec Time Stamping Services CA
- G2 |2012-10-18|2020-12-29|2994      |
|cce94b9791ce6afa89288333c06ce731|Thawte Timestamping CA
|2012-12-21|2020-12-30|2931      |
|cce94b9791ce6afa89288333c06ce731|Thawte Timestamping CA
|1997-01-01|2020-12-31|8765      |
|fa58b1b0e6a722ff87a7da84419353d5|Symantec Time Stamping Services CA
- G2 |2012-10-17|2020-12-29|2995      |
|fa58b1b0e6a722ff87a7da84419353d5|Thawte Timestamping CA
|2012-12-21|2020-12-30|2931      |
|fa58b1b0e6a722ff87a7da84419353d5|Thawte Timestamping CA
|1997-01-01|2020-12-31|8765      |
|aed4ecd9a76700265118609b65321489|GlobalSign Timestamping CA - SHA25
6 - G2|2016-05-23|2027-06-23|4048      |
|867106bc27c3c464e14874695a1ffab0|Symantec SHA256 TimeStamping CA
|2017-12-23|2029-03-22|4107      |
|08bbe07ad85f4eb10167bf522c9eb4fe|Symantec Time Stamping Services CA
- G2 |2012-10-17|2020-12-29|2995      |
|08bbe07ad85f4eb10167bf522c9eb4fe|Thawte Timestamping CA
|2012-12-21|2020-12-30|2931      |
|08bbe07ad85f4eb10167bf522c9eb4fe|Thawte Timestamping CA
|1997-01-01|2020-12-31|8765      |
|5b50c2fe7c55a6a00a16cdb3bc008897|Symantec SHA256 TimeStamping CA
|2017-01-02|2028-04-01|4107      |
|7134299c38eef0797a7cf18f83b990ad|Symantec Time Stamping Services CA
- G2 |2012-10-17|2020-12-29|2995      |
+--------------------------------+----------------------------------
------+----------+----------+----------+
only showing top 20 rows
```

In [8]:

```
pd_frame = new_df.select(F.year("valid_from").alias("year")).groupby("year").agg
(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [9]:

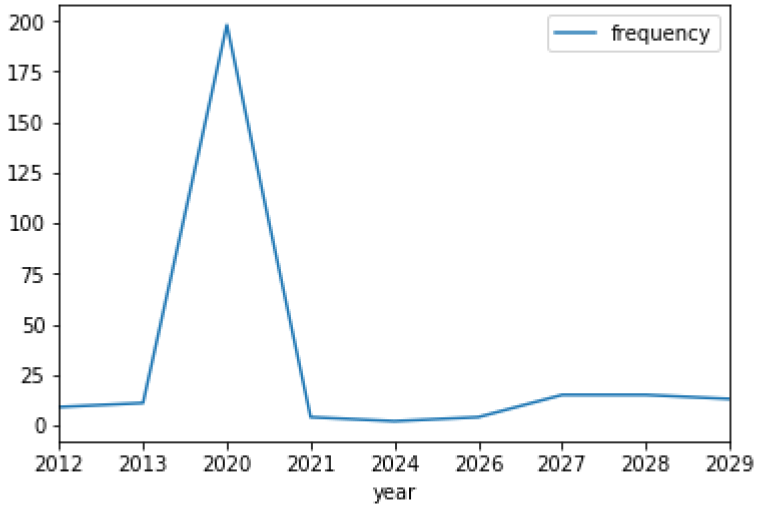
```
%matplotlib inline
```

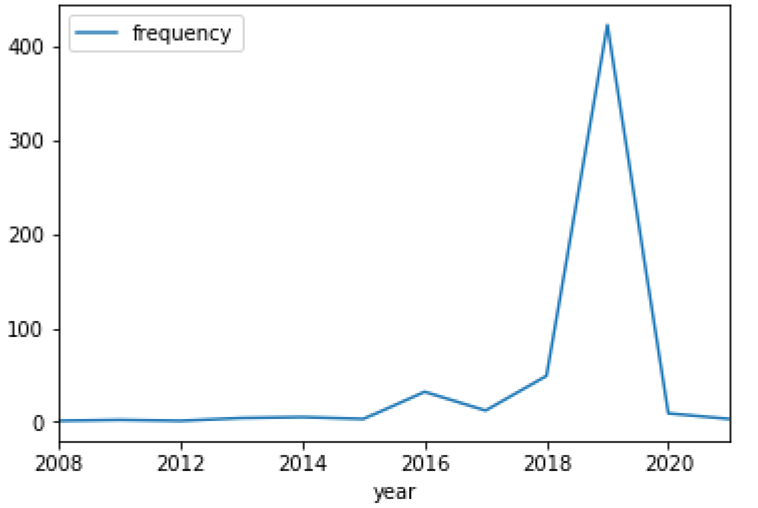
In [31]:

```
pd_frame.sort_values("year").astype({"year":str}).plot(x="year")
pd_frame_codesign.sort_values("year").astype({"year":str}).plot(x="year")
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbb041dcd68>
```





In [34]:

```
pd_frame.rename({"frequency":"timecheck_frequency"})
```

Out[34]:

	year	frequency
0	2027	15
1	2013	11
2	2026	4
3	2029	13
4	2020	198
5	2012	9
6	2028	15
7	2024	2
8	2021	4

In [40]:

```
pd_frame.rename(columns={"frequency":"timestamp_frequency","year":"timestamp_year"})\
.join(pd_frame_codesign.rename(columns={"frequency":"codesign_frequency","year":"codesign_year"})\
      ,on=["timestamp_year","codesign_year"], how="outer")
```

```
---------------------------------------------------------------------
-------
ValueError                                Traceback (most recent cal
l last)
<ipython-input-40-6bb2e4cb7eef> in <module>
      1 pd_frame.rename(columns={"frequency":"timestamp_frequency",
"year":"timestamp_year"})\
      2 .join(pd_frame_codesign.rename(columns={"frequency":"codesig
n_frequency","year":"codesign_year"})\
----> 3         ,on=["timestamp_year","codesign_year"], how="outer")

/usr/local/lib/python3.5/dist-packages/pandas/core/frame.py in join
(self, other, on, how, lsuffix, rsuffix, sort)
   6813         # For SparseDataFrame's benefit
   6814         return self._join_compat(other, on=on, how=how, lsuf
fix=lsuffix,
-> 6815                                  rsuffix=rsuffix, sort=sort)
   6816
   6817     def _join_compat(self, other, on=None, how='left', lsuff
ix='', rsuffix='',

/usr/local/lib/python3.5/dist-packages/pandas/core/frame.py in _join
_compat(self, other, on, how, lsuffix, rsuffix, sort)
   6828             return merge(self, other, left_on=on, how=how,
   6829                          left_index=on is None, right_index=
True,
-> 6830                          suffixes=(lsuffix, rsuffix), sort=s
ort)
   6831         else:
   6832             if on is not None:

/usr/local/lib/python3.5/dist-packages/pandas/core/reshape/merge.py
 in merge(left, right, how, on, left_on, right_on, left_index, right
_index, sort, suffixes, copy, indicator, validate)
     45                          right_index=right_index, sort=sort,
suffixes=suffixes,
     46                          copy=copy, indicator=indicator,
---> 47                          validate=validate)
     48     return op.get_result()
     49

/usr/local/lib/python3.5/dist-packages/pandas/core/reshape/merge.py
 in __init__(self, left, right, how, on, left_on, right_on, axis, le
ft_index, right_index, sort, suffixes, copy, indicator, validate)
    522             warnings.warn(msg, UserWarning)
    523
--> 524         self._validate_specification()
    525
    526         # note this function has side effects

/usr/local/lib/python3.5/dist-packages/pandas/core/reshape/merge.py
 in _validate_specification(self)
   1045             if self.right_index:
   1046                 if len(self.left_on) != self.right.index.nle
vels:
-> 1047                     raise ValueError('len(left_on) must equa
l the number '
   1048                                      'of levels in the index
of "right"')
   1049                 self.right_on = [None] * n
```

```
ValueError: len(left_on) must equal the number of levels in the inde
x of "right"
```

In [41]:

```
pd_frame.join(pd_frame_codesign,on="year",lsuffix = "_left", rsuffix= "_right",
how="outer")
```

Out[41]:

	year	year_left	frequency_left	year_right	frequency_right
0	2027	2027.0	15.0	NaN	NaN
1	2013	2013.0	11.0	NaN	NaN
2	2026	2026.0	4.0	NaN	NaN
3	2029	2029.0	13.0	NaN	NaN
4	2020	2020.0	198.0	NaN	NaN
5	2012	2012.0	9.0	NaN	NaN
6	2028	2028.0	15.0	NaN	NaN
7	2024	2024.0	2.0	NaN	NaN
8	2021	2021.0	4.0	NaN	NaN
8	0	NaN	NaN	2018.0	49.0
8	1	NaN	NaN	2015.0	3.0
8	2	NaN	NaN	2013.0	4.0
8	3	NaN	NaN	2014.0	5.0
8	4	NaN	NaN	2019.0	423.0
8	5	NaN	NaN	2020.0	9.0
8	6	NaN	NaN	2012.0	1.0
8	7	NaN	NaN	2016.0	32.0
8	8	NaN	NaN	2011.0	2.0
8	9	NaN	NaN	2008.0	1.0
8	10	NaN	NaN	2017.0	12.0
8	11	NaN	NaN	2021.0	3.0

In [60]:

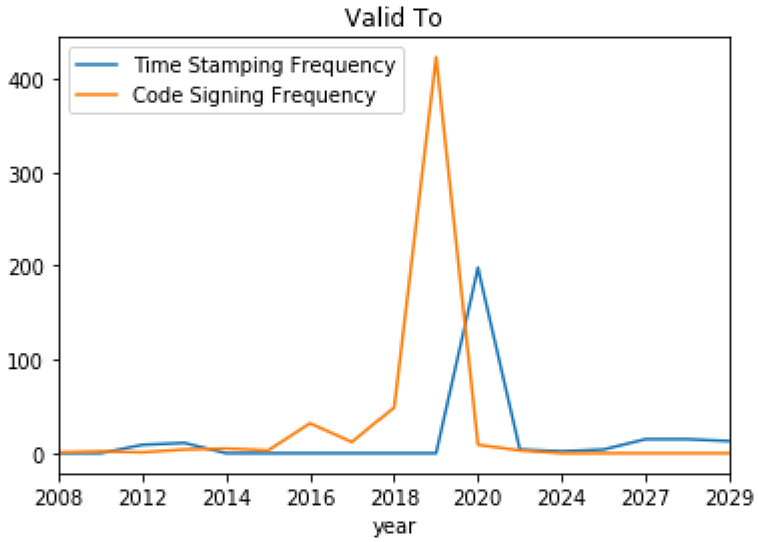
```
combined_data = pd_frame.astype({"year":str}).set_index("year").join(pd_frame_co
design.astype({"year":str}).set_index("year"), \
                                lsuffix = "_left", rsuffix= "_right", how="oute
r")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequenc
y_right":"Code Signing Frequency"})
```

In [65]:

```
combined_data.plot().set_title("Valid To")
```

Out[65]:

```
Text(0.5, 1.0, 'Valid To')
```



In [53]:

```
combined_data.index.astype(str)
```

Out[53]:

```
Index(['2008', '2011', '2012', '2013', '2014', '2015', '2016', '201
7', '2018',
       '2019', '2020', '2021', '2024', '2026', '2027', '2028', '202
9'],
      dtype='object', name='year')
```

In [59]:

```
combined_data
```

Out[59]:

	frequency_left	frequency_right
year		
2008	0.0	1.0
2011	0.0	2.0
2012	9.0	1.0
2013	11.0	4.0
2014	0.0	5.0
2015	0.0	3.0
2016	0.0	32.0
2017	0.0	12.0
2018	0.0	49.0
2019	0.0	423.0
2020	198.0	9.0
2021	4.0	3.0
2024	2.0	0.0
2026	4.0	0.0
2027	15.0	0.0
2028	15.0	0.0
2029	13.0	0.0

In [18]:

```
pd_frame.sort_values("year")
```

Out[18]:

	year	frequency
5	1997	74
0	2003	10
1	2007	6
6	2012	132
4	2013	2
3	2015	4
7	2016	15
8	2017	24
2	2018	4

In [19]:

```
pd_frame = new_df.select(F.year("valid_to").alias("year")).groupby("year").agg(F
.count(F.lit(1)).alias("frequency")).toPandas()
```

In [5]:

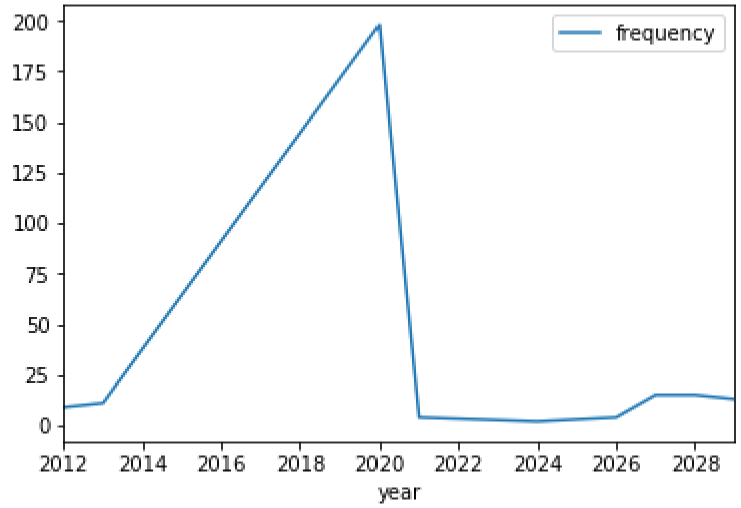
```
%matplotlib inline
```

In [22]:

```
pd_frame.sort_values("year").plot(x="year")
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbb042b5c50>
```



In [23]:

```
pd_frame.sort_values("year")
```

Out[23]:

	year	frequency
5	2012	9
1	2013	11
4	2020	198
8	2021	4
7	2024	2
2	2026	4
0	2027	15
6	2028	15
3	2029	13

In [26]:

```
new_df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("signers details")
        ).alias("counter_signers_details")
).select(
        F.col("md5"),
        F.col("counter_signers_details").getItem("cert issuer").alias("cert_issuer"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_details").getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_details").getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_to"),
).where("lower(cert_issuer) LIKE '%code%signing%'").withColumn("difference",F.datediff(("valid_to"),("valid_from")))
```

In [27]:

```
new_df.show(truncate=False)
```

```
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
|md5                             |cert_issuer
|valid_from|valid_to  |difference|
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
|aad2e37a5e733c140b3e02f9d793a572|VeriSign Class 3 Code Signing 2010
CA  |2013-06-04|2016-09-03|1187      |
|8cc09e049d9a0ea1fc3355292d10ce85|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|9bcb0bd9a5ac1d166ebbafd1879b3675|Symantec Class 3 SHA256 Code Signi
ng CA|2017-10-02|2018-10-03|366       |
|775542926871b5889bc98c5c059f27f3|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|a9eda36c8c9d981e525378499e363bc2|VeriSign Class 3 Code Signing 2009
-2 CA|2009-12-16|2012-12-15|1095      |
|dcefbad6923989cf1501b3c85ffdc6f3|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|a75e132050f5c7058f0c2ed5a655b40d|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|ee98d649b7162e886bacd702e1574746|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|896215bea9826a68bcff5c8fe15af8dc|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b730628b7e7c9ef1e1215096267a8e6f|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|fb64bfe1795a309733abde4fbdc0bd54|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|cce94b9791ce6afa89288333c06ce731|VeriSign Class 3 Code Signing 2010
CA  |2015-08-28|2017-09-26|760       |
|cea23408db4d74f79f646dcf88eafa20|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|d68fc15c50ecfea3fba0e13d240c212d|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b20adacce6da81c4a8a765c9eaf35c70|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|5f6449899a3986fd6d70f48eeb394202|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|53059b04972664743b9dc1dc1e2bc342|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b87bccdc1b1b43e9c446b534f5e02006|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|5b0ccd97eeed5b21fbd091ff50c97f45|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|9ce3cddf87d09f9ad352f98c3fe1c65b|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
only showing top 20 rows
```

In [30]:

```
pd_frame_codesign = new_df.select(F.year("valid_to").alias("year")).groupby("yea
r").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [ ]:

```
pd_frame_codesign.
```

In [4]:

```
files = !ls /home/ubuntu/MyVolumeStore/Virustotal_Responses/*.json
```

In [5]:

```
files
```

Out[5]:

```
['/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_307.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_308.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_309.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_310.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_311.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_312.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_313.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_314.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_315.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_316.json']
```

In [14]:

```
for file in files:
    df = sqlContext.read.json(file)
    new_df = df.where("positives > 2").select(
        F.col("md5"), F.col("positives"), F.col("scans.*")
    )
    #dropping additional info and other columns

    new_df.withColumn(
        "file_type_count",
        sum([
            F.when(
                F.instr(F.lower(F.col(cl).getItem("result")), "trojan") > 0,
                1
            ).otherwise(0) for cl in new_df.columns[2:]
        ])
    ).select(
        "md5", "positives",
        F.col("file_type_count")
    ).withColumn(
        "type",
        F.when(
            F.col("file_type_count") > (F.col("positives")/10),
            "trojan"
        ).otherwise("nottrojan")
    ).write.mode("append").parquet("analysis/trojan_parquet")
    print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [15]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/trojan_parquet').sh
ow(truncate=False)
```

```
+--------------------------------+---------+---------------+--------
-+
|md5                             |positives|file_type_count|type
|
+--------------------------------+---------+---------------+--------
-+
|a29f9383ab57c5fb6b24948c022ef89a|45       |11             |trojan
|
|d9085bc83c9e5ad9e9ce833eb0614ab7|46       |14             |trojan
|
|021253f13b7b61a42bb78e98d5118eda|44       |10             |trojan
|
|d918693704383eeea8d4ac89542d491b|46       |12             |trojan
|
|d94660310686da66b7b660e045d4c33b|52       |12             |trojan
|
|d94bebaa012c10f69ef5d6a7dbd11d30|49       |10             |trojan
|
|d990259151b7695114f1625582e27e75|42       |8              |trojan
|
|d8edecb902b98ce170041bccd6130c9e|43       |17             |trojan
|
|da103663a071ef0162d95e93e95d6944|37       |1              |nottroja
n|
|da5f53e6cc44c680c8c1eefdd7204a20|47       |14             |trojan
|
|da57fbe63064240f48d56628b5333e58|47       |14             |trojan
|
|f1c19fd27ead96167ccaa7cd92b4e15a|44       |11             |trojan
|
|2baf58ab708dfafe1850ec270cf9edcd|48       |12             |trojan
|
|da815fa364bda89538b696ff515210da|51       |14             |trojan
|
|da9e54c9560928b9f732bc3be22028e5|48       |12             |trojan
|
|f7e223e9004aed80e2fdd91819c3afd4|26       |2              |nottroja
n|
|048cdd9f9c2703a8961bb5a9aa85233f|3        |0              |nottroja
n|
|b54e372c781a7db66b6421588c29498e|47       |12             |trojan
|
|db3b0149d23b54b9128ccc0b7e10e799|48       |11             |trojan
|
|04f5b24332c3f8d309512259e4b481aa|3        |3              |trojan
|
+--------------------------------+---------+---------------+--------
-+
only showing top 20 rows
```

In [20]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/trojan_parquet')\
.groupby("type").agg(F.count("md5").alias("frequency"))\
.show(truncate=False)
```

```
+----------+---------+
|type      |frequency|
+----------+---------+
|nottrojan|2813     |
|trojan    |44935    |
+----------+---------+
```

In [28]:

```
for file in files:
    df = sqlContext.read.json(file)
    new_df = df.where("positives > 2").select(
        F.col("md5"), F.col("positives"), F.col("scans.*")
    )
    #dropping additional info and other columns

    new_df.withColumn(
        "file_type_count",
        sum([
            F.when(
                F.instr(F.lower(F.col(cl).getItem("result")), "FakeAV") > 0,
                1
            ).otherwise(0) for cl in new_df.columns[2:]
        ])
    ).select(
        "md5", "positives",
        F.col("file_type_count")
    ).withColumn(
        "type",
        F.when(
            F.col("file_type_count") > (F.col("positives")/10),
            "FakeAV"
        ).otherwise("notFakeAV")
    ).write.mode("append").parquet("analysis/FakeAV_parquet")
    print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [29]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/FakeAV_parquet')\
.groupby("type").agg(F.count("md5").alias("frequency"))\
.show(truncate=False)
```

```
+----------+---------+
|type      |frequency|
+----------+---------+
|notFakeAV|47748    |
+----------+---------+
```

In [3]:

```
files = !ls /home/ubuntu/MyVolumeStore/Virustotal_Responses/*.json
```

In [ ]:

```

```

In [4]:

```
files
```

Out[4]:

```
['/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_307.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_308.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_309.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_310.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_311.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_312.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_313.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_314.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_315.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_316.json']
```

In [13]:

```
new_df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("signers de
tails")
        ).alias("signers_details")
).select(
        F.col("md5"),
        F.col("signers_details").getItem("cert issuer").alias("cert_issuer"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("signers_details")
.getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("signers_details")
.getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_to"),
).where("lower(cert_issuer) LIKE '%code%signing%'").withColumn("difference",F.da
tediff(("valid_to"),("valid_from")))
```

In [6]:

```
df = sqlContext.read.json('/home/ubuntu/MyVolumeStore/Virustotal_Responses/respo
nses_windows_virushashes_316.json')
```

In [15]:

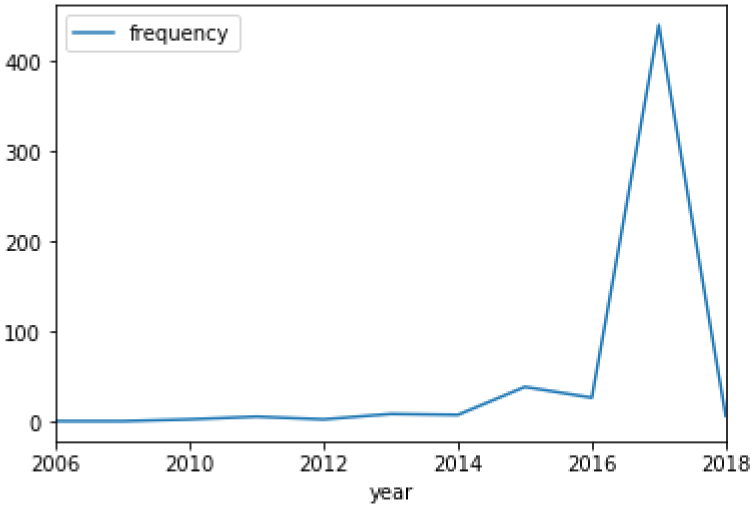
```
pd_frame = new_df.select(F.year("valid_from").alias("year")).groupby("year").agg
(F.count(F.lit(1)).alias("frequency")).toPandas()
```

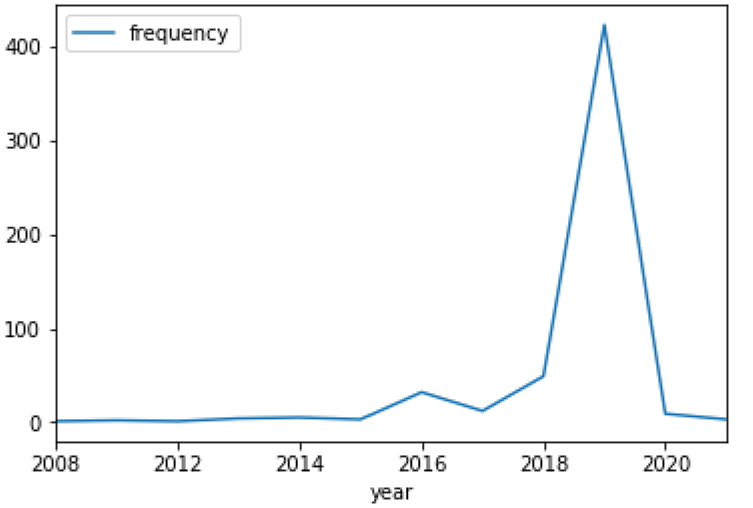
In [18]:

```
pd_frame.sort_values("year").astype({"year":str}).plot(x="year")
pd_frame_codesign.sort_values("year").astype({"year":str}).plot(x="year")
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbfa8cdb1d0>
```





In [14]:

```
new_df.show(truncate=False)
```

```
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
|md5                             |cert_issuer
|valid_from|valid_to  |difference|
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
|aad2e37a5e733c140b3e02f9d793a572|VeriSign Class 3 Code Signing 2010
CA  |2013-06-04|2016-09-03|1187      |
|8cc09e049d9a0ea1fc3355292d10ce85|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|9bcb0bd9a5ac1d166ebbafd1879b3675|Symantec Class 3 SHA256 Code Signi
ng CA|2017-10-02|2018-10-03|366       |
|775542926871b5889bc98c5c059f27f3|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|a9eda36c8c9d981e525378499e363bc2|VeriSign Class 3 Code Signing 2009
-2 CA|2009-12-16|2012-12-15|1095      |
|dcefbad6923989cf1501b3c85ffdc6f3|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|a75e132050f5c7058f0c2ed5a655b40d|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|ee98d649b7162e886bacd702e1574746|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|896215bea9826a68bcff5c8fe15af8dc|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b730628b7e7c9ef1e1215096267a8e6f|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|fb64bfe1795a309733abde4fbdc0bd54|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|cce94b9791ce6afa89288333c06ce731|VeriSign Class 3 Code Signing 2010
CA  |2015-08-28|2017-09-26|760       |
|cea23408db4d74f79f646dcf88eafa20|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|d68fc15c50ecfea3fba0e13d240c212d|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b20adacce6da81c4a8a765c9eaf35c70|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|5f6449899a3986fd6d70f48eeb394202|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|53059b04972664743b9dc1dc1e2bc342|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|b87bccdc1b1b43e9c446b534f5e02006|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|5b0ccd97eeed5b21fbd091ff50c97f45|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|9ce3cddf87d09f9ad352f98c3fe1c65b|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
+--------------------------------+-----------------------------------
-----+----------+----------+----------+
only showing top 20 rows
```

In [30]:

```
pd_frame_codesign = new_df.select(F.year("valid_from").alias("year")).groupby("y
ear").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [62]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("signers de
tails")
        ).alias("signers_details")
).select(
        F.col("md5"),
        F.col("signers_details").getItem("cert issuer").alias("cert_issuer"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("signers_details")
.getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("signers_details")
.getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_to"),
).where("lower(cert_issuer) LIKE '%code%signing%'").withColumn("difference",F.da
tediff(("valid_to"),("valid_from"))).write.mode("append").parquet("analysis/date
_codesigning_parquet")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [20]:

```
import os
```

In [40]:

```
df = sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/date_codesigni
ng_parquet/*')
```

In [41]:

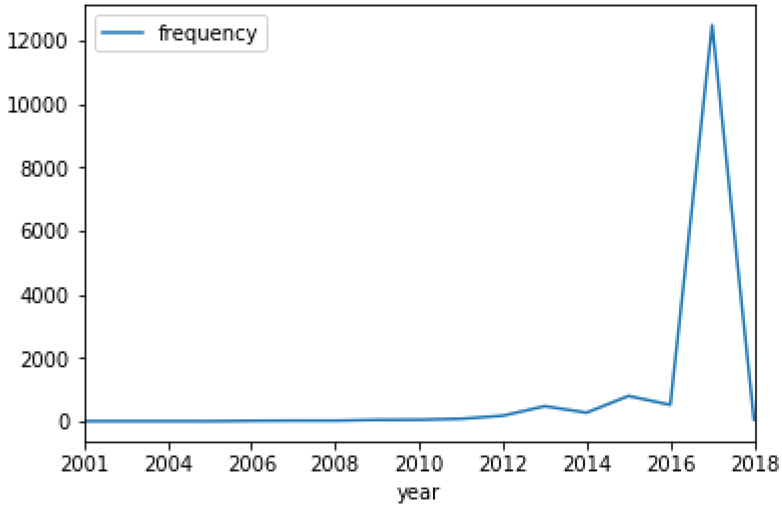
```
pd_frame_codesigning_valid_from = df.select(F.year("valid_from").alias("year")).
groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [42]:

```
pd_frame_codesigning_valid_from.sort_values("year").astype({"year":str}).plot(x=
"year")
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbfa8ba0c18>
```



In [ ]:

```
combined_data = pd_frame.astype({"year":str}).set_index("year").join(pd_frame_co
design.astype({"year":str}).set_index("year"), \
                                    lsuffix = "_left", rsuffix= "_right", how="oute
r")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequenc
y_right":"Code Signing Frequency"})
```

In [31]:

```
df.show(truncate=False)
```

```
+--------------------------------+----------------------------------
-----+----------+----------+----------+
|md5                             |cert_issuer
|valid_from|valid_to  |difference|
+--------------------------------+----------------------------------
-----+----------+----------+----------+
|d822e8ce21bef84ca1096038a4e2aad3|VeriSign Class 3 Code Signing 2010
CA  |2016-03-08|2018-02-10|704       |
|a29f9383ab57c5fb6b24948c022ef89a|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|021253f13b7b61a42bb78e98d5118eda|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|d94660310686da66b7b660e045d4c33b|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|d94bebaa012c10f69ef5d6a7dbd11d30|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|d990259151b7695114f1625582e27e75|VeriSign Class 3 Code Signing 2010
CA  |2015-06-14|2017-09-13|822       |
|da103663a071ef0162d95e93e95d6944|VeriSign Class 3 Code Signing 2010
CA  |2017-08-03|2019-10-01|789       |
|f1c19fd27ead96167ccaa7cd92b4e15a|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|2baf58ab708dfafe1850ec270cf9edcd|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|0475d1faf81f7fd498db1334a75decef|COMODO Code Signing CA 2
|2013-03-25|2016-03-24|1095      |
|b54e372c781a7db66b6421588c29498e|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|ae339d0d3018baef34d67f643c50f51c|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|05eabcd00b9592c589ad3db10b9fd190|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|065e5f246a03573de8e53e76e6996c96|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|dcb445e32d0f50bb11bdf9be92489176|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|7668b7fad4231a519cd923ba75924795|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|dc53b7feae0be413aaf2e2facdef0a54|GlobalSign CodeSigning CA - G3
|2016-10-13|2019-11-30|1143      |
|082e975282c6b1b5a27be3c2a708004b|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|de58778e795b99c0fceccfab6bf7a7e4|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
|de5e53e60e403566d621ec833e5f58ba|GlobalSign CodeSigning CA - SHA256
- G3|2017-07-26|2019-08-26|761       |
+--------------------------------+----------------------------------
-----+----------+----------+----------+
only showing top 20 rows
```

In [45]:

```
df = sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/date_timestamp
ing_parquet/*')
```

In [46]:

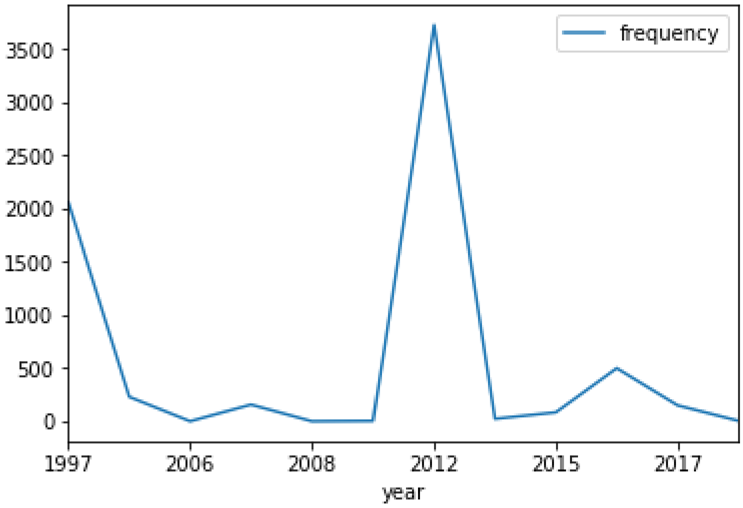
```
pd_frame_timestamping_valid_from = df.select(F.year("valid_from").alias("year")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [47]:

```
pd_frame_timestamping_valid_from.sort_values("year").astype({"year":str}).plot(x="year")
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbfa8b252e8>
```



In [48]:

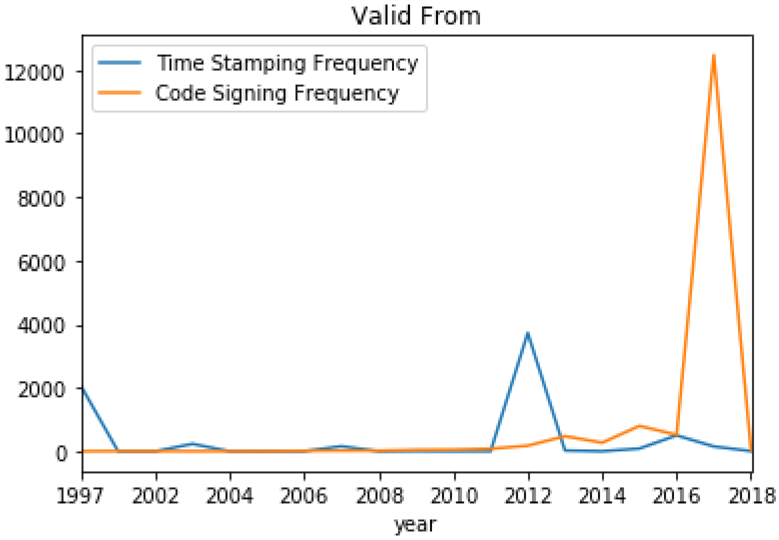
```
combined_data = pd_frame_timestamping_valid_from.astype({"year":str}).set_index("year").join(pd_frame_codesigning_valid_from.astype({"year":str}).set_index("year"), \
                                             lsuffix = "_left", rsuffix= "_right", how="outer")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequency_right":"Code Signing Frequency"})
```

In [49]:

```
combined_data.plot().set_title("Valid From")
```

Out[49]:

```
Text(0.5, 1.0, 'Valid From')
```



In [50]:

```
pd_frame_codesigning_valid_to = df.select(F.year("valid_to").alias("year")).grou
pby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [56]:

```
pd_frame_timestamping_valid_to = df_timestamp.select(F.year("valid_to").alias("y
ear")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [55]:

```
df_timestamp = sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/date
_timestamping_parquet/*')
```

In [57]:

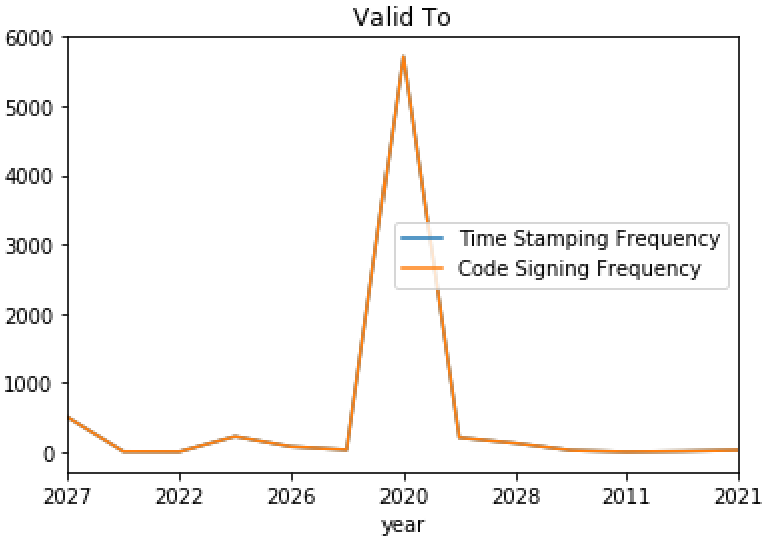
```
combined_data = pd_frame_timestamping_valid_to.astype({"year":str}).set_index("year").join(pd_frame_codesigning_valid_to.astype({"year":str}).set_index("year"),\
                                    lsuffix = "_left", rsuffix= "_right", how="outer")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequency_right":"Code Signing Frequency"})
```

In [58]:

```
combined_data.plot().set_title("Valid To")
```

Out[58]:

```
Text(0.5, 1.0, 'Valid To')
```



In [61]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("md5"),
        F.explode(
                F.col("additional_info").getItem("sigcheck").getItem("counter si
gners details")
        ).alias("counter_signers_details")
).select(
        F.col("md5"),
        F.col("counter_signers_details").getItem("cert issuer").alias("cert_issu
er"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_d
etails").getItem("valid from")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid
_from"),
    F.from_unixtime(F.unix_timestamp(F.substring(F.trim(F.col("counter_signers_d
etails").getItem("valid to")),10,10), "MM/dd/yyyy"),"yyyy-MM-dd").alias("valid_t
o"),
).where("lower(cert_issuer) LIKE '%time%stamping%'").withColumn("difference",F.d
atediff(("valid_to"),("valid_from"))).write.mode("append").parquet("analysis/dat
e_timestamping_parquet")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [3]:

```
df_timestamp = sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/date
_timestamping_parquet/*')
```

In [4]:

```
df_codesigning = sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/da
te_codesigning_parquet/*')
```

In [5]:

```
pd_frame_timestamping_valid_to = df_timestamp.select(F.year("valid_to").alias("y
ear")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [6]:

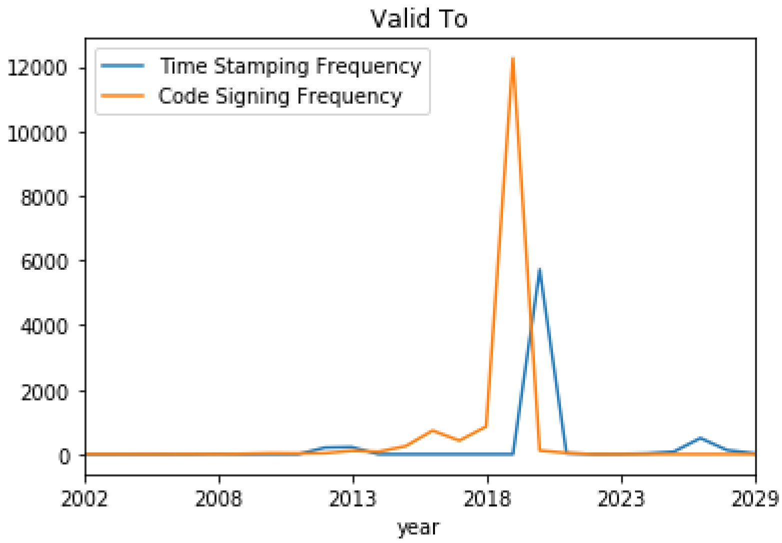
```
pd_frame_codesigning_valid_to = df_codesigning.select(F.year("valid_to").alias(
"year")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [7]:

```
combined_data = pd_frame_timestamping_valid_to.astype({"year":str}).set_index("y
ear").join(pd_frame_codesigning_valid_to.astype({"year":str}).set_index("year"),
\
                                lsuffix = "_left", rsuffix= "_right", how="oute
r")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequenc
y_right":"Code Signing Frequency"})
```

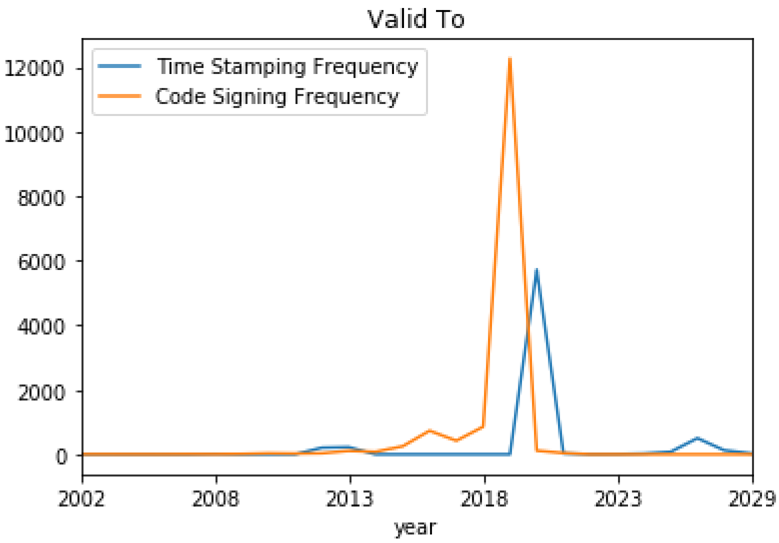
In [18]:

```
combined_data.plot().set_title("Valid To")
```



In [28]:

```
fig =combined_data.plot().set_title("Valid To").get_figure()
```



In [29]:

```
type(fig)
```

Out[29]:

```
matplotlib.figure.Figure
```

In [31]:

```
fig.savefig('ValidTo.eps')
```

In [23]:

```
combined_data.savefig("figure_67.eps", format="eps", dpi=1000)
```

```
---------------------------------------------------------------------
-------
AttributeError                            Traceback (most recent cal
l last)
<ipython-input-23-e1515c8f105f> in <module>
----> 1 combined_data.savefig("figure_67.eps", format="eps", dpi=100
0)

/usr/local/lib/python3.5/dist-packages/pandas/core/generic.py in __g
etattr__(self, name)
   5065             if self._info_axis._can_hold_identifiers_and_hol
ds_name(name):
   5066                 return self[name]
-> 5067             return object.__getattribute__(self, name)
   5068 
   5069     def __setattr__(self, name, value):

AttributeError: 'DataFrame' object has no attribute 'savefig'
```

In [32]:

```
pd_frame_timestamping_valid_from = df_timestamp.select(F.year("valid_from").alia
s("year")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

In [33]:

```
pd_frame_codesigning_valid_from = df_codesigning.select(F.year("valid_from").ali
as("year")).groupby("year").agg(F.count(F.lit(1)).alias("frequency")).toPandas()
```

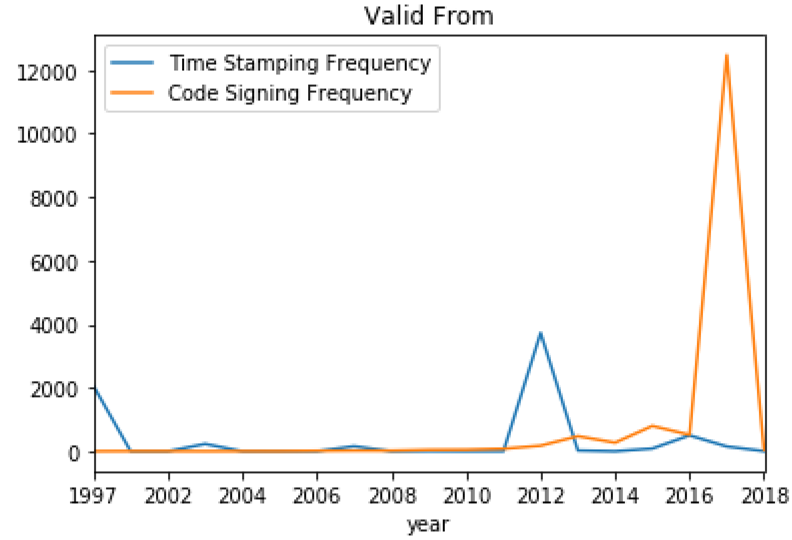
In [34]:

```
combined_data = pd_frame_timestamping_valid_from.astype({"year":str}).set_index(
"year").join(pd_frame_codesigning_valid_from.astype({"year":str}).set_index("yea
r"), \
                                lsuffix = "_left", rsuffix= "_right", how="oute
r")\
.fillna(0).rename(columns={"frequency_left":"Time Stamping Frequency", "frequenc
y_right":"Code Signing Frequency"})
```

```
In [35]:
```

```
combined_data.plot().set_title("Valid From")
```

```
Out[35]:

Text(0.5, 1.0, 'Valid From')
```



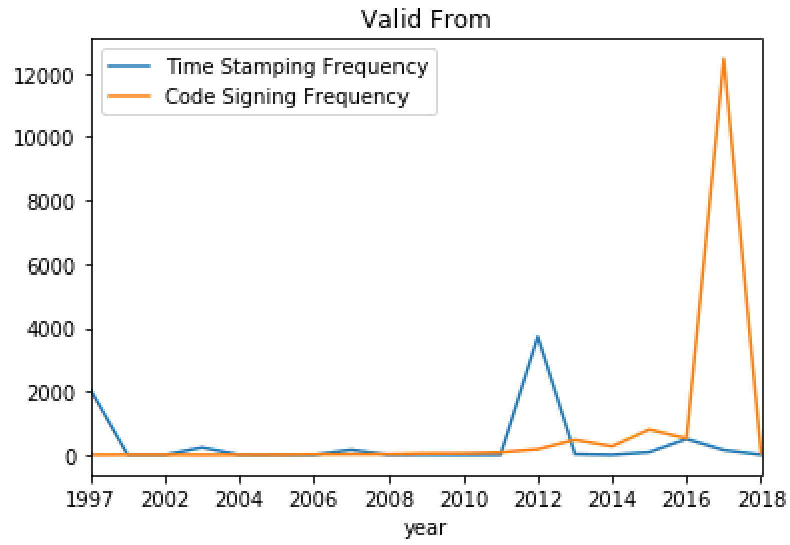
```
In [77]:
```

```
combined_data.savefig("figure_67.eps", format="eps", dpi=1000)
```

```
---------------------------------------------------------------------
-------
AttributeError                            Traceback (most recent cal
l last)
<ipython-input-77-e1515c8f105f> in <module>
----> 1 combined_data.savefig("figure_67.eps", format="eps", dpi=100
0)

/usr/local/lib/python3.5/dist-packages/pandas/core/generic.py in __g
etattr__(self, name)
   5065             if self._info_axis._can_hold_identifiers_and_hol
ds_name(name):
   5066                 return self[name]
-> 5067             return object.__getattribute__(self, name)
   5068
   5069     def __setattr__(self, name, value):

AttributeError: 'DataFrame' object has no attribute 'savefig'
```

In [36]:

```
fig =combined_data.plot().set_title("Valid From").get_figure()
```



In [37]:

```
fig.savefig('ValidFrom.eps')
```

In [76]:

```
import matplotlib.pyplot as plt
```

In [39]:

```
import seaborn as sns
```

In [41]:

```
df = pd.DataFrame(dict(time=np.arange(500),
                       value=np.random.randn(500).cumsum()))
g = sns.relplot(x="time", y="value", kind="line", data=df)
g.fig.autofmt_xdate()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-41-62337f2fd05d> in <module>
----> 1 df = pd.DataFrame(dict(time=np.arange(500),
      2                        value=np.random.randn(500).cumsum()))
      3 g = sns.relplot(x="time", y="value", kind="line", data=df)
      4 g.fig.autofmt_xdate()

NameError: name 'pd' is not defined
```

In [42]:

```
sns.combined_data.plot().set_title("Valid From").get_figure()
```

```
---------------------------------------------------------------------
-------
AttributeError                            Traceback (most recent cal
l last)
<ipython-input-42-30008421fb97> in <module>
----> 1 sns.combined_data.plot().set_title("Valid From").get_figure(
)

AttributeError: module 'seaborn' has no attribute 'combined_data'
```

In [43]:

```
import pandas as pd
```

In [45]:

```
df = pd.DataFrame()
```

In [47]:

```
pd.DataFrame(combined_data)
```

Out[47]:

year	Time Stamping Frequency	Code Signing Frequency
1997	2061.0	0.0
2001	0.0	3.0
2002	0.0	3.0
2003	231.0	0.0
2004	0.0	2.0
2005	0.0	1.0
2006	1.0	14.0
2007	158.0	20.0
2008	1.0	19.0
2009	4.0	44.0
2010	0.0	49.0
2011	0.0	78.0
2012	3731.0	176.0
2013	24.0	475.0
2014	0.0	271.0
2015	85.0	798.0
2016	501.0	519.0
2017	151.0	12478.0
2018	5.0	51.0

In [5]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/processed_parquet').count()
```

Out[5]:

```
10332
```

In [8]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/countersigners_parquet')\
.select("md5").write.option("header", "true").mode('overwrite').text("timestampingMD5.txt")
```

In [10]:

```
pwd
```

Out[10]:

```
'/home/ubuntu/MyVolumeStore'
```

In [18]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select(F.col("md5"),F.explode(
                        F.col("additional_info").getItem("sigcheck").getItem("si
gners details").getItem("cert issuer")
                ).alias("signers_details")).write.mode("append").parquet("analys
is/codesigners_parquest")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [20]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest').
count()
```

Out[20]:

```
49526
```

In [21]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.select("md5").write.option("header", "true").mode('overwrite').text("codesignin
gMD5.txt")
```

In [ ]:

```
new_df = df.select(
    F.col("md5"), F.col("scans.*")
).where("positives > 2")
new_df.columns[1:]

new_df.withColumn(
    "detected_count",
    sum([
        F.when(F.col(cl).getItem("detected"), 1).otherwise(0) for cl in new_df.c
olumns[1:]
    ])
).select("md5", "detected_count").show()
```

In [22]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.filter(
        F.col("additional_info").getItem("sigcheck").getItem("verified") == "Sig
ned"
        ).select(F.col("md5"),F.col("positives").alias("positives_details"))
        df.repartition(4 if fsize > 100 else 2).write.mode("append").parquet("an
alysis/positives_parquet")
        print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [33]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/positives_parquet')
.groupby("positives_details")\
.agg(F.countDistinct("md5").alias("md5")).show(100)
```

```
+-----------------+----+
|positives_details| md5|
+-----------------+----+
|               26|  59|
|               29|  28|
|               19|  26|
|               54|   1|
|                0|1039|
|               22|  82|
|                7|  37|
|               34|  15|
|               50| 459|
|               43|1613|
|               32|  17|
|               31|  11|
|               39|  31|
|               25| 101|
|                6|  80|
|                9|  22|
|               27|  62|
|               51| 436|
|               52| 178|
|               17|  24|
|               41| 304|
|               33|  17|
|               28|  33|
|                5|  50|
|                1| 620|
|               10|  26|
|               48| 195|
|               44|2657|
|                3| 110|
|               37|  13|
|               12|  36|
|               55|   1|
|                8|  34|
|               11|  27|
|               49| 324|
|               35|  21|
|                2| 252|
|                4|  85|
|               13|  33|
|               36|  16|
|               18|  25|
|               14|  34|
|               21|  38|
|               15|  19|
|               38|  19|
|               42| 682|
|               30|  19|
|               23|  90|
|               46|1734|
|               20|  43|
|               40| 134|
|               16|  16|
|               45|2735|
|               47| 653|
|               53|  21|
|               24|  87|
+-----------------+----+
```

In [30]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/positives_parquet') \
.select("positives_details").write.option("header", "true").mode('overwrite').text("positives_parquet.txt")
```

```
--------------------------------------------------------------------
-------
Py4JJavaError                             Traceback (most recent cal
l last)
~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/pyspark/sql/u
tils.py in deco(*a, **kw)
     62         try:
---> 63             return f(*a, **kw)
     64         except py4j.protocol.Py4JJavaError as e:

~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/lib/py4j-0.1
0.7-src.zip/py4j/protocol.py in get_return_value(answer, gateway_cli
ent, target_id, name)
    327                     "An error occurred while calling {0}{1}
{2}.\n".
--> 328                     format(target_id, ".", name), value)
    329             else:

Py4JJavaError: An error occurred while calling o983.text.
: org.apache.spark.sql.AnalysisException: Text data source supports
 only a string column, but you have bigint.;
        at org.apache.spark.sql.execution.datasources.text.TextFileF
ormat.verifySchema(TextFileFormat.scala:51)
        at org.apache.spark.sql.execution.datasources.text.TextFileF
ormat.prepareWrite(TextFileFormat.scala:66)
        at org.apache.spark.sql.execution.datasources.FileFormatWrit
er$.write(FileFormatWriter.scala:135)
        at org.apache.spark.sql.execution.datasources.InsertIntoHado
opFsRelationCommand.run(InsertIntoHadoopFsRelationCommand.scala:145)
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.sideEffectResult$lzycompute(commands.scala:58)
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.sideEffectResult(commands.scala:56)
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.doExecute(commands.scala:74)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
$1.apply(SparkPlan.scala:117)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
$1.apply(SparkPlan.scala:117)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
Query$1.apply(SparkPlan.scala:138)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOper
ationScope.scala:151)
        at org.apache.spark.sql.execution.SparkPlan.executeQuery(Spa
rkPlan.scala:135)
        at org.apache.spark.sql.execution.SparkPlan.execute(SparkPla
n.scala:116)
        at org.apache.spark.sql.execution.QueryExecution.toRdd$lzyco
mpute(QueryExecution.scala:92)
        at org.apache.spark.sql.execution.QueryExecution.toRdd(Query
Execution.scala:92)
        at org.apache.spark.sql.execution.datasources.DataSource.wri
teInFileFormat(DataSource.scala:435)
        at org.apache.spark.sql.execution.datasources.DataSource.wri
te(DataSource.scala:471)
        at org.apache.spark.sql.execution.datasources.SaveIntoDataSo
urceCommand.run(SaveIntoDataSourceCommand.scala:48)
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.sideEffectResult$lzycompute(commands.scala:58)
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.sideEffectResult(commands.scala:56)
```

```
        at org.apache.spark.sql.execution.command.ExecutedCommandExe
c.doExecute(commands.scala:74)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
$1.apply(SparkPlan.scala:117)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
$1.apply(SparkPlan.scala:117)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute
Query$1.apply(SparkPlan.scala:138)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOper
ationScope.scala:151)
        at org.apache.spark.sql.execution.SparkPlan.executeQuery(Spa
rkPlan.scala:135)
        at org.apache.spark.sql.execution.SparkPlan.execute(SparkPla
n.scala:116)
        at org.apache.spark.sql.execution.QueryExecution.toRdd$lzyco
mpute(QueryExecution.scala:92)
        at org.apache.spark.sql.execution.QueryExecution.toRdd(Query
Execution.scala:92)
        at org.apache.spark.sql.DataFrameWriter.runCommand(DataFrame
Writer.scala:609)
        at org.apache.spark.sql.DataFrameWriter.save(DataFrameWrite
r.scala:233)
        at org.apache.spark.sql.DataFrameWriter.save(DataFrameWrite
r.scala:217)
        at org.apache.spark.sql.DataFrameWriter.text(DataFrameWrite
r.scala:554)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Metho
d)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodA
ccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(Delegatin
gMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:2
44)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.
java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractComman
d.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:748)


During handling of the above exception, another exception occurred:

AnalysisException                         Traceback (most recent cal
l last)
<ipython-input-30-1f18699d711f> in <module>
      1 sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysi
s/positives_parquet')\
----> 2 .select("positives_details").write.option("header", "true").
mode('overwrite').text("positives_parquet.txt")

~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/pyspark/sql/r
eadwriter.py in text(self, path, compression)
    704         """
    705         self._set_opts(compression=compression)
--> 706         self._jwrite.text(path)
    707
    708     @since(2.0)

~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/lib/py4j-0.1
0.7-src.zip/py4j/java_gateway.py in __call__(self, *args)
   1255         answer = self.gateway_client.send_command(command)
   1256         return_value = get_return_value(
-> 1257             answer, self.gateway_client, self.target_id, sel
f.name)
   1258 
   1259         for temp_arg in temp_args:

~/MyVolumeStore/spark/spark-2.2.3-bin-hadoop2.7/python/pyspark/sql/u
tils.py in deco(*a, **kw)
     67                                              e.java_exceptio
n.getStackTrace()))
     68             if s.startswith('org.apache.spark.sql.AnalysisEx
ception: '):
---> 69                 raise AnalysisException(s.split(': ', 1)[1],
stackTrace)
     70             if s.startswith('org.apache.spark.sql.catalyst.a
nalysis'):
     71                 raise AnalysisException(s.split(': ', 1)[1],
stackTrace)

AnalysisException: 'Text data source supports only a string column,
 but you have bigint.;'
```

In [5]:

```
for file in files:
    df = sqlContext.read.json(file)
    new_df = df.where("positives > 2").filter(
    F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
    ).select(
        F.col("md5"), F.col("positives"), F.col("scans.*")
    )
    #dropping additional info and other columns

    new_df.withColumn(
        "file_type_count",
        sum([
            F.when(
                F.instr(F.lower(F.col(cl).getItem("result")), "adware") > 0,
                1
            ).when(
                F.instr(F.lower(F.col(cl).getItem("result")), "pup") > 0,
                1
            ).otherwise(0) for cl in new_df.columns[2:]
        ])
    ).select(
        "md5", "positives",
        F.col("file_type_count")
    ).withColumn(
        "type",
        F.when(
            F.col("file_type_count") > (F.col("positives")/10),
            "pup"
        ).otherwise("virus")
    ).write.mode("append").parquet("analysis/signed_pup_virus_parquet")
    print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

In [6]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signed_pup_virus_parquet')\
.groupby("type").agg(F.count("md5").alias("frequency"))\
.show(truncate=False)
```

```
+-----+---------+
|type |frequency|
+-----+---------+
|pup  |13218    |
|virus|395      |
+-----+---------+
```

In [7]:

```
for file in files:
    df = sqlContext.read.json(file)
    new_df = df.where("positives > 2").filter(
    F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
    ).select(
        F.col("md5"), F.col("positives"), F.col("scans.*")
    )
    #dropping additional info and other columns

    new_df.withColumn(
        "file_type_count",
        sum([
            F.when(
                F.instr(F.lower(F.col(cl).getItem("result")), "trojan") > 0,
                1
            ).otherwise(0) for cl in new_df.columns[2:]
        ])
    ).select(
        "md5", "positives",
        F.col("file_type_count")
    ).withColumn(
        "type",
        F.when(
            F.col("file_type_count") > (F.col("positives")/10),
            "trojan"
        ).otherwise("nottrojan")
    ).write.mode("append").parquet("analysis/signed_trojan_parquet")
    print ("file writen %s"%file)
```

```
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_307.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_308.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_309.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_310.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_311.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_312.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_313.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_314.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_315.json
file writen /home/ubuntu/MyVolumeStore/Virustotal_Responses/response
s_windows_virushashes_316.json
```

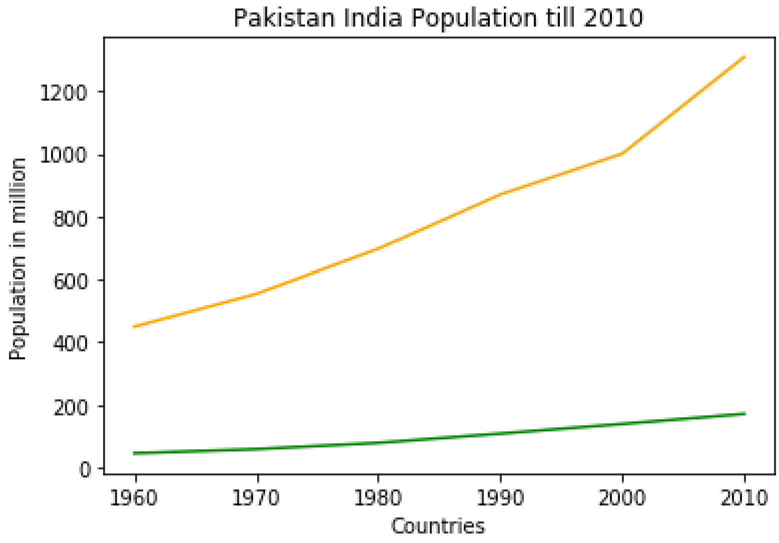
In [8]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signed_trojan_parqu
et')\
.groupby("type").agg(F.count("md5").alias("frequency"))\
.show(truncate=False)
```

```
+---------+---------+
|type     |frequency|
+---------+---------+
|nottrojan|1036     |
|trojan   |12577    |
+---------+---------+
```

In [8]:

```
year = [1960, 1970, 1980, 1990, 2000, 2010]
pop_pakistan = [44.91, 58.09, 78.07, 107.7, 138.5, 170.6]
pop_india = [449.48, 553.57, 696.783, 870.133, 1000.4, 1309.1]
plt.plot(year, pop_pakistan, color='g')
plt.plot(year, pop_india, color='orange')
plt.xlabel('Countries')
plt.ylabel('Population in million')
plt.title('Pakistan India Population till 2010')
plt.show()
```

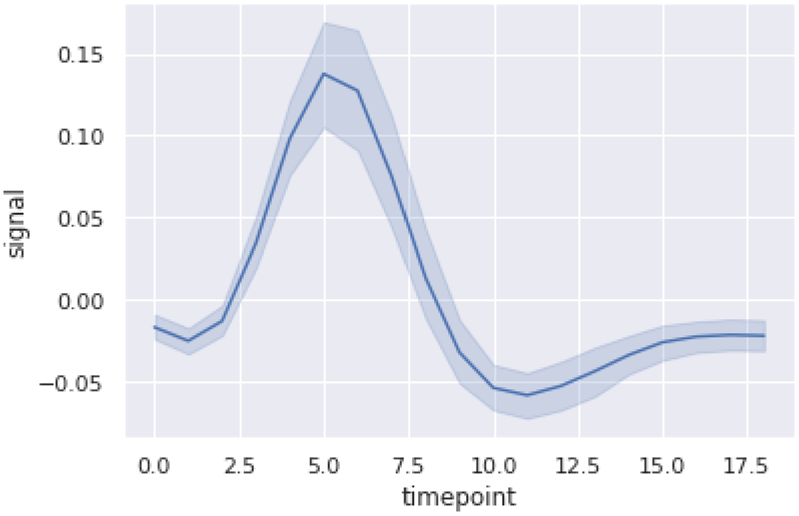


In [7]:

```
from matplotlib import pyplot as plt
```

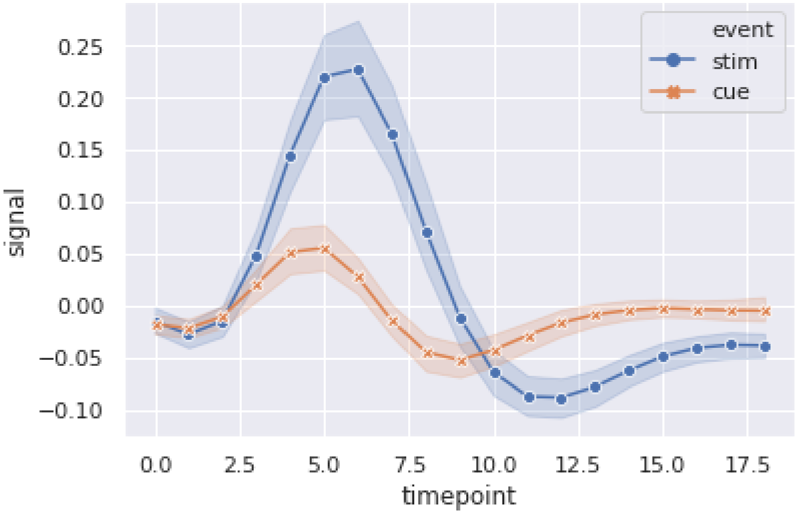
In [9]:

```
import seaborn as sns; sns.set()
import matplotlib.pyplot as plt
fmri = sns.load_dataset("fmri")
ax = sns.lineplot(x="timepoint", y="signal", data=fmri)
```



In [10]:

```
ax = sns.lineplot(x="timepoint", y="signal",hue="event", style="event", markers=
True, dashes=False, data=fmri)
```



In [3]:

```
import seaborn as sns
import numpy as np
import pandas as pd


# inputs
num = np.array([1, 2, 3, 4, 5])
sqr = np.array([1, 4, 9, 16, 25])

# convert to pandas dataframe
d = {'num': num, 'sqr': sqr}
pdnumsqr = pd.DataFrame(d)

# plot using lineplot
sns.set(style='darkgrid')
sns.lineplot(x='num', y='sqr', data=pdnumsqr)
```

Out[3]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe1b4010898>
```

In [15]:

```
census_data = pd.read_csv('Book5.csv')
```

In [16]:

```
census_data.describe()
```

Out[16]:

	Unnamed: 4	Unnamed: 5	Unnamed: 6
count	0.0	0.0	0.0
mean	NaN	NaN	NaN
std	NaN	NaN	NaN
min	NaN	NaN	NaN
25%	NaN	NaN	NaN
50%	NaN	NaN	NaN
75%	NaN	NaN	NaN
max	NaN	NaN	NaN

In [17]:

```
census_data.head()
```

Out[17]:

	Entity	Gas Price	TX Fees	Network	Unnamed: 4	Unnamed: 5	Unnamed: 6
0	Root CA	542,908	271,422	Goerli	NaN	NaN	NaN
1	Intermidiate CA	1,422,170	711085	Goerli	NaN	NaN	NaN
2	Dictionary	74,748	37374	Goerli	NaN	NaN	NaN
3	Certificate Signing Request	365,139	182569	Goerli	NaN	NaN	NaN
4	TimeStamping	221,449	11070	Goerli	NaN	NaN	NaN

In [18]:

```
census_data.info()
```

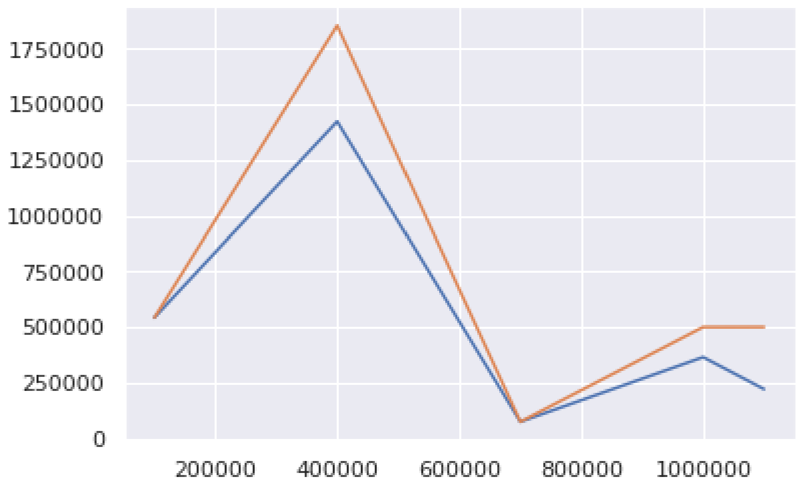
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
Entity          10 non-null object
Gas Price       10 non-null object
TX Fees         10 non-null object
Network         10 non-null object
Unnamed: 4      0 non-null float64
Unnamed: 5      0 non-null float64
Unnamed: 6      0 non-null float64
dtypes: float64(3), object(4)
memory usage: 640.0+ bytes
```

In [16]:

```
plt.plot([100000, 400000, 700000, 1000000, 1100000], [542908, 1422170, 74748, 36
5139, 221449],
            [100000, 400000, 700000, 1000000, 1100000], [542844, 1853048, 74748,
500139, 500449])
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x7fe1afafe630>,
 <matplotlib.lines.Line2D at 0x7fe1afafe828>]
```



In [3]:

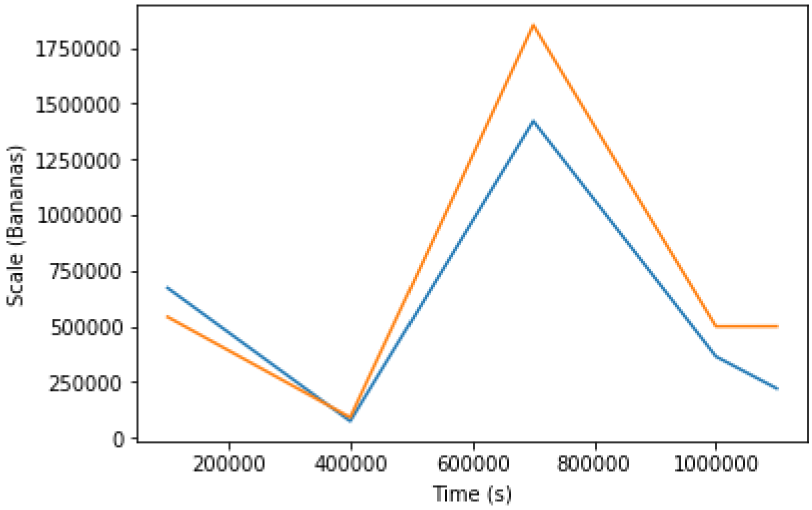
```
from matplotlib import pyplot as plt
```

In [7]:

```
plt.plot([100000, 400000, 700000, 1000000, 1100000], [ 672908,74748, 1422170, 36
5139, 221449], )
plt.plot([100000, 400000, 700000, 1000000, 1100000], [ 542844,91748, 1853048, 50
0139, 500449])
plt.xlabel("Time (s)")
plt.ylabel("Scale (Bananas)")
```

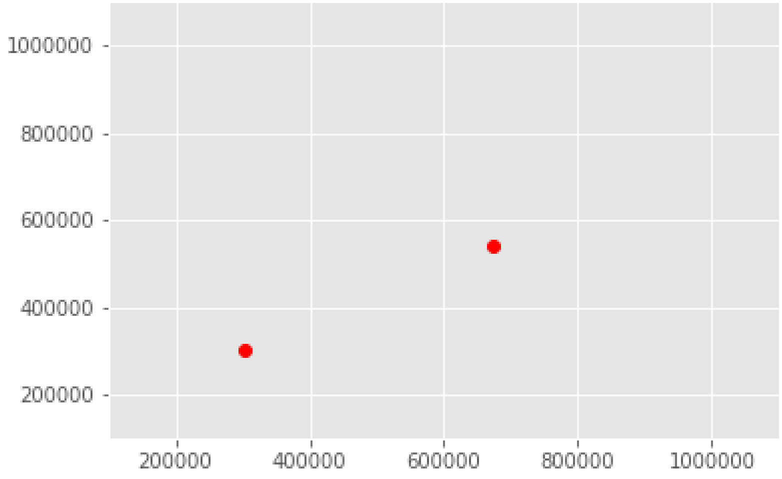
Out[7]:

```
Text(0, 0.5, 'Scale (Bananas)')
```



In [42]:

```
plt.plot([672908, 2, 3, 4], [542844, 4, 9, 16], 'ro')
plt.plot([302908, 2, 3, 4], [300844, 4, 9, 16], 'ro')
plt.axis([100000, 1100000, 100000, 1100000])
plt.show()
```

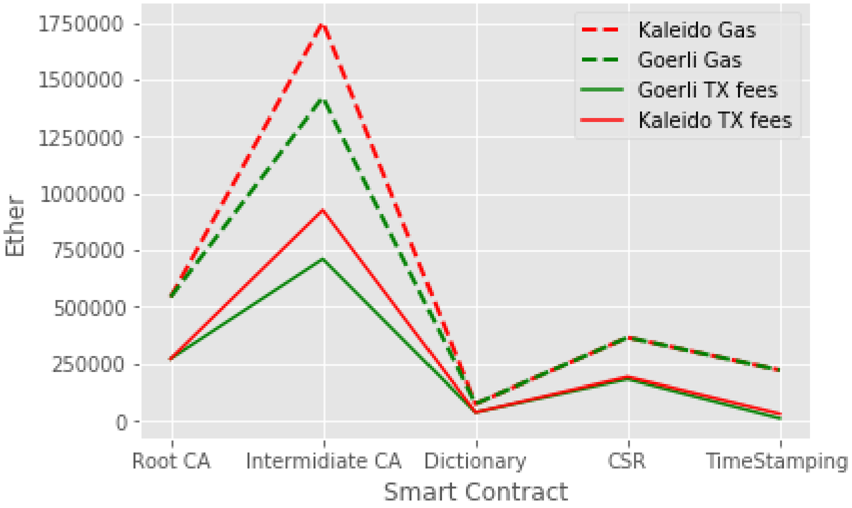


In [17]:

```
plt.plot?
```

In [7]:

```
from matplotlib import pyplot as plt
plt.style.use('ggplot')

years_hr = [ 542908, 1422170 , 74748, 365139, 221449]
kaleido_gasprice = [542844,1753048,74748,365139,221449]


years_lr = [271422,711085,37374,182569,11070]
kaleido_txfees = [271400,926500,37482,192611,31070]

years = ["Root CA", "Intermidiate CA", "Dictionary", "CSR", "TimeStamping "]

plt.plot(years, kaleido_gasprice, label='Kaleido Gas', linewidth=2, color='red', linestyle='dashed')
plt.plot(years, years_hr, label='Goerli Gas', linewidth=2, color='green',linestyle='dashed')

plt.plot(years, years_lr, label='Goerli TX fees', color='green')
plt.plot(years, kaleido_txfees,label='Kaleido TX fees', color='red')

plt.xlabel("Smart Contract")
plt.ylabel("Ether")
plt.legend()

plt.xticks(rotation=0)
plt.savefig('Goerli.png', bbox_inches='tight',dpi=500)

plt.show()

plt.figure(figsize=(13,14))
```



Out[7]:

```
<Figure size 936x1008 with 0 Axes>

<Figure size 936x1008 with 0 Axes>
```

In [7]:

```
type(fig)
```

```
---------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-7-d3bd7867356c> in <module>
----> 1 type(fig)

NameError: name 'fig' is not defined
```

In [21]:

```
fig.savefig('SmartC.eps')
```

In [23]:

```
plt.show()
```

In [24]:

```
plt.figure(figsize=(200, 2))
```

Out[24]:

```
<Figure size 14400x144 with 0 Axes>

<Figure size 14400x144 with 0 Axes>
```

In [27]:

```
plt.show()
```

In [8]:

```
plt.show()
```

In [ ]:

```
for file in files:
        statinfo = os.stat(file)
        fsize = (statinfo.st_size/1024)/1024
        df = sqlContext.read.json(file)
        df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).where("lower(cert_issuer) LIKE '%code%signing%'").withColumn("difference",F.da
tediff(("valid_to"),("valid_from"))).write.mode("append").parquet("analysis/date
_codesigning_parquet")
        print ("file writen %s"%file)
```

In [4]:

```
files = !ls /home/ubuntu/MyVolumeStore/Virustotal_Responses/*.json
```

In [6]:

```
files
```

Out[6]:

```
['/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_307.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_308.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_309.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_310.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_311.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_312.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_313.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_314.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_315.json',
 '/home/ubuntu/MyVolumeStore/Virustotal_Responses/responses_windows_
virushashes_316.json']
```

In [ ]:

```
new_df = df.select("md5", "additional_info").filter(
F.col("additional_info").getItem("sigcheck").getItem("verified") == "Signed"
).select(
        F.col("first_seen").alias("counter_signers_details")
)
```

In [2]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%symantec%class%3%'")\
.groupby("signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+------------------------------------------------------------+----+
|signers_details                                             |md5 |
+------------------------------------------------------------+----+
|VeriSign Class 3 Code Signing 2009 CA                       |5   |
|WoSign Class 3 Code Signing CA                              |100 |
|Symantec Class 3 Extended Validation Code Signing CA - G3   |10  |
|StartCom Class 3 Object CA                                  |5   |
|Class 3 Public Primary Certification Authority              |160 |
|WoSign Class 3 Code Signing CA G2                           |10  |
|Symantec Class 3 Extended Validation Code Signing CA        |6   |
|Symantec Class 3 SHA256 Code Signing CA - G2                |1   |
|VeriSign Class 3 Code Signing 2001 CA                       |2   |
|VeriSign Class 3 Code Signing 2009-2 CA                     |68  |
|StartCom Class 3 Primary Intermediate Object CA             |68  |
|VeriSign Class 3 Public Primary Certification Authority - G5|1671|
|Symantec Class 3 SHA256 Code Signing CA                     |570 |
|Symantec Class 3 Extended Validation Code Signing CA - G2   |56  |
|VeriSign Class 3 Code Signing 2010 CA                       |1064|
|VeriSign Class 3 Code Signing 2004 CA                       |65  |
+------------------------------------------------------------+----+
```

In [6]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%symantec%class%3%'")\
.groupby("signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+---------------------------------------------------------+---+
|signers_details                                          |md5|
+---------------------------------------------------------+---+
|Symantec Class 3 Extended Validation Code Signing CA - G3|10 |
|Symantec Class 3 Extended Validation Code Signing CA     |6  |
|Symantec Class 3 SHA256 Code Signing CA - G2             |1  |
|Symantec Class 3 SHA256 Code Signing CA                  |570|
|Symantec Class 3 Extended Validation Code Signing CA - G2|56 |
+---------------------------------------------------------+---+
```

In [4]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%class%1%'")\
.groupby("signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+-------------------------------------+----+
|signers_details                      |md5 |
+-------------------------------------+----+
|VeriSign Class 3 Code Signing 2001 CA|2   |
|Class 1 Primary CA                   |1   |
|VeriSign Class 3 Code Signing 2010 CA|1064|
+-------------------------------------+----+
```

In [5]:

```
sqlContext.read.parquet('/home/ubuntu/MyVolumeStore/analysis/signers_parquest')\
.where("lower(signers_details) LIKE '%class%2%'")\
.groupby("signers_details").agg(F.countDistinct("md5").alias("md5")).show(truncate=False)
```

```
+---------------------------------------------------------+----+
|signers_details                                          |md5 |
+---------------------------------------------------------+----+
|VeriSign Class 3 Code Signing 2009 CA                    |5   |
|WoSign Class 3 Code Signing CA G2                        |10  |
|Symantec Class 3 SHA256 Code Signing CA - G2             |1   |
|VeriSign Class 3 Code Signing 2001 CA                    |2   |
|VeriSign Class 3 Code Signing 2009-2 CA                  |68  |
|Go Daddy Class 2 Certification Authority                 |42  |
|Starfield Class 2 Certification Authority                |5   |
|Symantec Class 3 SHA256 Code Signing CA                  |570 |
|Symantec Class 3 Extended Validation Code Signing CA - G2|56  |
|StartCom Class 2 Object CA                               |357 |
|StartCom Class 2 Primary Intermediate Object CA          |8   |
|WoSign Class 2 Code Signing CA                           |3   |
|VeriSign Class 3 Code Signing 2010 CA                    |1064|
|VeriSign Class 3 Code Signing 2004 CA                    |65  |
+---------------------------------------------------------+----+
```

In [ ]: