An approach to malware collection log visualization

Jaime Blasco

Aitsec
C/Gobelas, 19
28023 Madrid, Spain
E-mail:jaime.blasco@aitsec.com

An approach to malware collection log visualization

jaime.blasco@aitsec.com

This paper focus on visualization of the logs files generated by malware collectors, we'll learn to generate some graphs with the data collected by Nepenthes sensors.

Nepenthes

Nepenthes is a low interaction Honeypot that emulate "known" vulnerabilities to collect information about potential attacks, the goal of this tool is to emulate vulnerabilities to collect information and capture the worms that takes advantage of the vulnerabilities to spread.

1) Installation

We have install all the stuff in a Ubuntu Server, first we have to install Nepenthes and the required libraries:

```
root@ubuntu# apt-get install  libcurl3-dev libmagic-dev libpcre3-dev
libadns1-dev libpcap0.8-dev iptables-dev

root@ubuntu# apt-get install  nepenthes
```

Once we have installed Nepenthes we realize Nepenthes open several ports on the machine:

```
root@ubuntu:~# lsof -i

COMMAND    PID      USER   FD   TYPE DEVICE SIZE NODE NAME
nepenthes 3795 nepenthes   6u  IPv4  12903       TCP *:smtp (LISTEN)
nepenthes 3795 nepenthes   7u  IPv4  12904       TCP *:pop3 (LISTEN)
nepenthes 3795 nepenthes   8u  IPv4  12905       TCP *:imap2 (LISTEN)
nepenthes 3795 nepenthes   9u  IPv4  12906       TCP *:imap3 (LISTEN)
nepenthes 3795 nepenthes  10u  IPv4  12907       TCP *:ssmtp (LISTEN)
nepenthes 3795 nepenthes  11u  IPv4  12908       TCP *:imaps (LISTEN)
nepenthes 3795 nepenthes  12u  IPv4  12909       TCP *:pop3s (LISTEN)
nepenthes 3795 nepenthes  13u  IPv4  12920       TCP *:2745 (LISTEN)
nepenthes 3795 nepenthes  14u  IPv4  12922       TCP *:6129 (LISTEN)
nepenthes 3795 nepenthes  15u  IPv4  12924       TCP *:loc-srv (LISTEN)
nepenthes 3795 nepenthes  16u  IPv4  12925       TCP *:microsoft-ds (LISTEN)
nepenthes 3795 nepenthes  17u  IPv4  12926       TCP *:1025 (LISTEN)
nepenthes 3795 nepenthes  18u  IPv4  12928       TCP *:ftp (LISTEN)
  nepenthes 3795 nepenthes   19u  IPv4  12929       TCP *:https (LISTEN)
  nepenthes 3795 nepenthes   20u  IPv4  12932       TCP *:17300 (LISTEN)
  nepenthes 3795 nepenthes   21u  IPv4  12936       TCP *:2103 (LISTEN)
  nepenthes 3795 nepenthes   22u  IPv4  12937       TCP *:eklogin (LISTEN)
  nepenthes 3795 nepenthes   23u  IPv4  12938       TCP *:2107 (LISTEN)
  nepenthes 3795 nepenthes   24u  IPv4  12940       TCP *:3372 (LISTEN)
  nepenthes 3795 nepenthes   25u  IPv4  12941       UDP *:ms-sql-m
  nepenthes 3795 nepenthes   26u  IPv4  12943       TCP *:3127 (LISTEN)
  nepenthes 3795 nepenthes   27u  IPv4  12947       TCP *:netbios-ssn (LISTEN)
  nepenthes 3795 nepenthes   28u  IPv4  12948       TCP *:3140 (LISTEN)
  nepenthes 3795 nepenthes   29u  IPv4  12951       TCP *:5554 (LISTEN)
  nepenthes 3795 nepenthes   30u  IPv4  12952       TCP *:1023 (LISTEN)
  nepenthes 3795 nepenthes   31u  IPv4  12957       TCP *:27347 (LISTEN)
  nepenthes 3795 nepenthes   32u  IPv4  12961       TCP *:5000 (LISTEN)
  nepenthes 3795 nepenthes   33u  IPv4  12965       TCP *:webmin (LISTEN)
  nepenthes 3795 nepenthes   34u  IPv4  12968       TCP *:nameserver (LISTEN)

  nepenthes 3795 nepenthes   35u  IPv4  12970       TCP *:www (LISTEN)
```

The ports listed emulates services with known vulnerabilities that will be “exploited” by the worms that try to attack our honeypot.

2) Log Files:

Nepenthes generate several files that contain information about the received attacks, file downloads and binary downloads.

Once Nepenthes receive an attack in one of the active services, Nepenthes examine the type of attack and interpret the shellcode with the goal of set if the worm is trying to download a file and which one method (http/link/ftp..).
When Nepenthes detect an attempt to download a binary file, Nepenthes has the ability of download the file and store it, so we can analyze the file in the future.
We can find the general log file of Nepenthes at /var/log/nepenthes.log
We can extract information about incoming connections, exploiting attempts, payloads and so on analyzing this log file.

Another interesting log file is var/log/nepenthes/logged_submissions and var/log/nepenthes/logged_downloads, these files contains information about the worm´s download attempts; who is trying to download the corresponding file.

```
[2008-06-18T16:14:06] 193.227.**.** -> 80.231.**.**
link://193.227. **.**:15383/nu0f+A== 47952bf18443c458b2792798c2433ec4
[2008-06-18T16:40:43] 220.224. **.**-> 80.231.**.**
link://220.224. **.**:52776/dOoZcA== cbed16069043a0bf3c92fff9a99cccdc
[2008-06-18T16:54:55] 89.223. **.**-> 80.231. **.**
link://89.223. **.**:64954/dOoZcA== cbed16069043a0bf3c92fff9a99cccdc
```

We can find the download binary files in /var/lib/nepenthes/binaries/:

```
root@ubuntu:/var/lib/nepenthes/binaries# ls
005472c686a5f84ad8e2dea597f50e1d  6fa0cd44b0664049f6a0ec5ffc6e1f07
0190d56cb7fe4e4094bbd8dc5dc2a65c  740f43e5daa7bf8b699776bbb12468ba
01f82083828db9ac4a898de52f5c6f5e  75eaf21b4df1218ad07d210860383dec
034680ae512fcd183a5a1e75e5b34986  77634b4e8669ea212138ec0931b4cedd
```

ClamAv

1) Installation:

At this time we have used the Open Source Virus Scanner “ClamAv”, with this tool we can scan the binary files obtained.

To install ClamAv:

```
root@ubuntu# apt-get install clamav
```

We can easily scan “binaries” directory:

```
root@ubuntu# clamscan ./ -l nep.log

./8a7b16ac83afbc89dd14885eea04fd64: W32.Bobax FOUND
./8ee8619debba32adbb40045316559dde: Trojan.SdBot-6673 FOUND
./18b3e69b9ba5b0cad8a04d329f34a94c: Trojan.SdBot-6301 FOUND
./6439ad20608e07380428ca0dc7574c41: Trojan.SdBot-6777 FOUND
….
….
….
----------- SCAN SUMMARY -----------
Known viruses: 315678
Engine version: 0.92.1
Scanned directories: 1
Scanned files: 159
Infected files: 99
Data scanned: 14.66 MB
Time: 6.176 sec (0 m 6 s)
```

Obtaining the required information

In this paper we will generate a graph representing information such as country, attacker Ip and detected malware.

For this reason we have to represent the necessary data in a CSV file like:
country, IP, malware

We can obtain the attacker ip from Nepenthes's logged_submissions log file, this file associate the attacker with the name of the binary file, we have to find the malware´s name searching the log file generated previously with clamscan.

On the other hand to determine the country of the attacker Ip we will use the Python Module ip2country,this module is able to determine the associated country of a given Ip at APNIC database. We have to install the specific module in our system:

```
wget http://www.freenet.org.nz/python/ip2country/ip2country.tar.gz
root@ubuntu # tar xvzf ip2country.tar.gz
root@ubuntu # cd ip2country
root@ubuntu # python setup.py install
```

Then we will try if the recent installed module works:

```
root@ubuntu #python
>>> import ip2country
>>> ip2c = ip2country.IP2Country(verbose=1)
>>> ip2c.lookup("218.77.84.186")
('CN', 'China')
```

At this point we have all the necessary data for generate the CSV file, then we have to write an script that correlate all the data sources:

```
import re
import fileinput
import ip2country

downloadsLog = "logged_submissions"
antivirusLog = "nep.log"

ip2c = ip2country.IP2Country(verbose=0)
```

```
#(clamAv) related work
#./129c66470181d226e6548a34b21479d4: Trojan.SdBot-5489 FOUND
#Regex: "\./(?P<base>\S+):\s(?P<name>\S+)\s"
p = re.compile(r"\./(?P<base>\S+):\s(?P<name>\S+)\s")
malware = dict()

for line in fileinput.input(antivirusLog):
        m = p.match(line)
        try:
                malware[m.group(1)] = m.group(2)
        except:
                pass

#logged_submissions related work
#[2007-08-07T17:42:18] 60.195.**.** -> 80.231.**.**
#link://60.195.**.**:23893/RfEgUA== f7a0ca139560fe8dfd246f546a45aa7f
#Regex: "\[\S*?\]\s(?P<source>\S+)\s->\s\S*?
#(?P<dest>\S+)\s(?P<link>\S+)\s(?P<base>\S+)"

p = re.compile(r'\[\S*?\]\s(?P<source>\S+)\s->\s\S*?
(?P<dest>\S+)\s(?P<link>\S+)\s(?P<base>\S+)')
for line in fileinput.input(downloadsLog):
        m = p.match(line)
        try:
                attacker = m.group(1)
                malw = malware[m.group(4)]
                country = ip2c.lookup(attacker)

                print country[0] + "," + attacker + "," + malw
        except:
                pass
```

Then we can execute the script and save the CSV file:

```
root@ubuntu #python genCSV.py > datos.csv
TW,210.209.**.**,Trojan.SdBot-5787
CN,61.161. **.**,Trojan.Vanbot-69
IN,58.68. **.**,Trojan.Vanbot-162
IN,58.68. **.**,Trojan.Vanbot-162
JP,222.73. **.**,Trojan.Vanbot-59
CN,218.228. **.**,Trojan.IRCBot-1063
JP,220.227. **.**,Worm.Gaobot-442
…
…
```

Generating the Graph

To generate the graphs we will use AfterGlow and Graphviz. First we have to download AfterGlow 1.5.X from SourceForge (**afterglow**.**sourceforge**.net), then extract the files and install the necessary perl modules if required (Text::CSV).

Install GraphViz:

```
root@ubuntu #apt-get install graphviz
...
```

The last step is to write a color.properties file, this file define the way AfterGlow represent the information we want to represent.

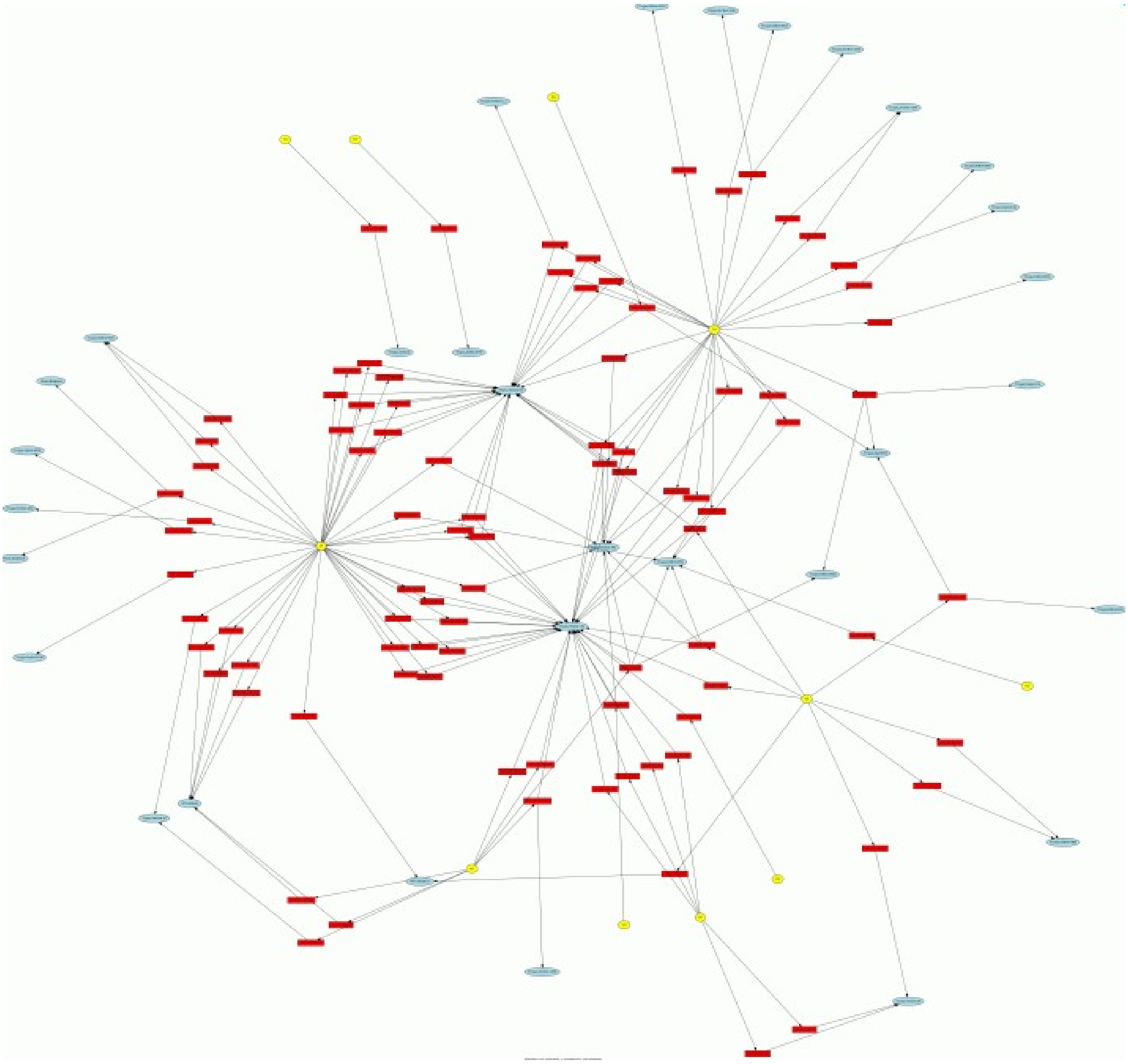
```
root@ubuntu #cat > color.properties
color.source="yellow"
color.event="red"
color.target="lightblue"
```

In this way the graph will represent the information:
country-yellow
attacker-red
malware-blue

Once we have all the necessary tools and data, we are ready to generate the graph:

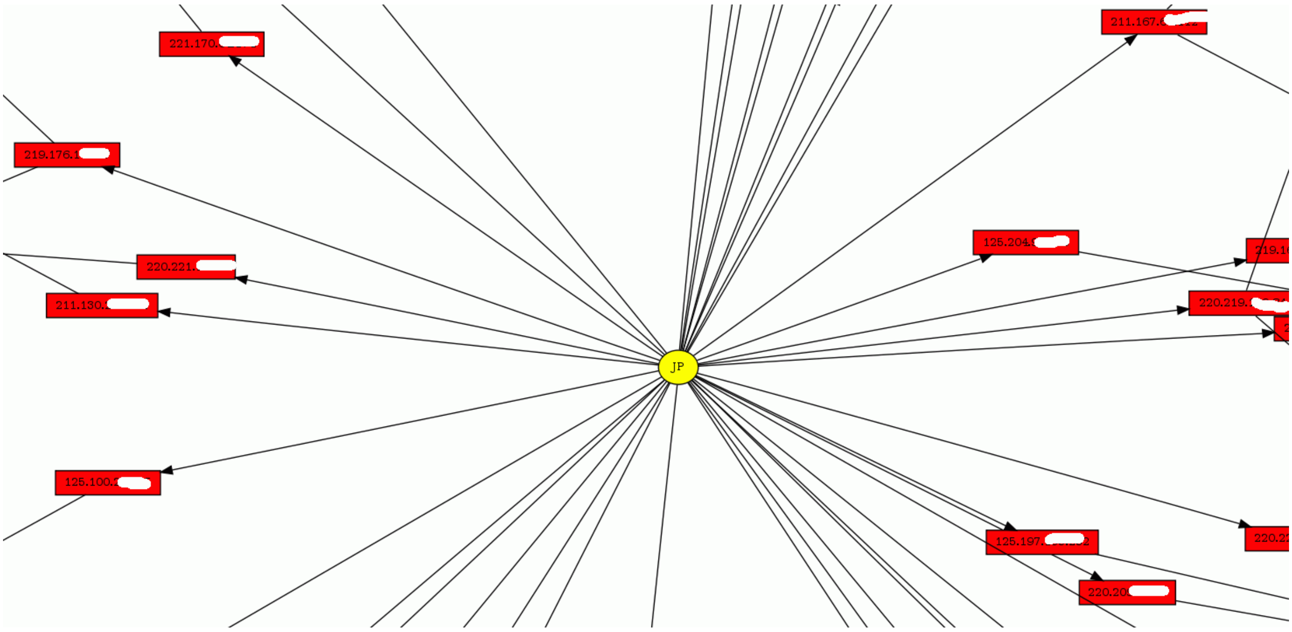
```
root@ubuntu # cat datos.csv  | perl afterglow/src/perl/graph/afterglow.pl
-c color.properties -e 6 -p 1 > img.dot

root@ubuntu # cat img.dot | neato -Tgif -o test.gif
```

And finally we have the graph generated at test.gif

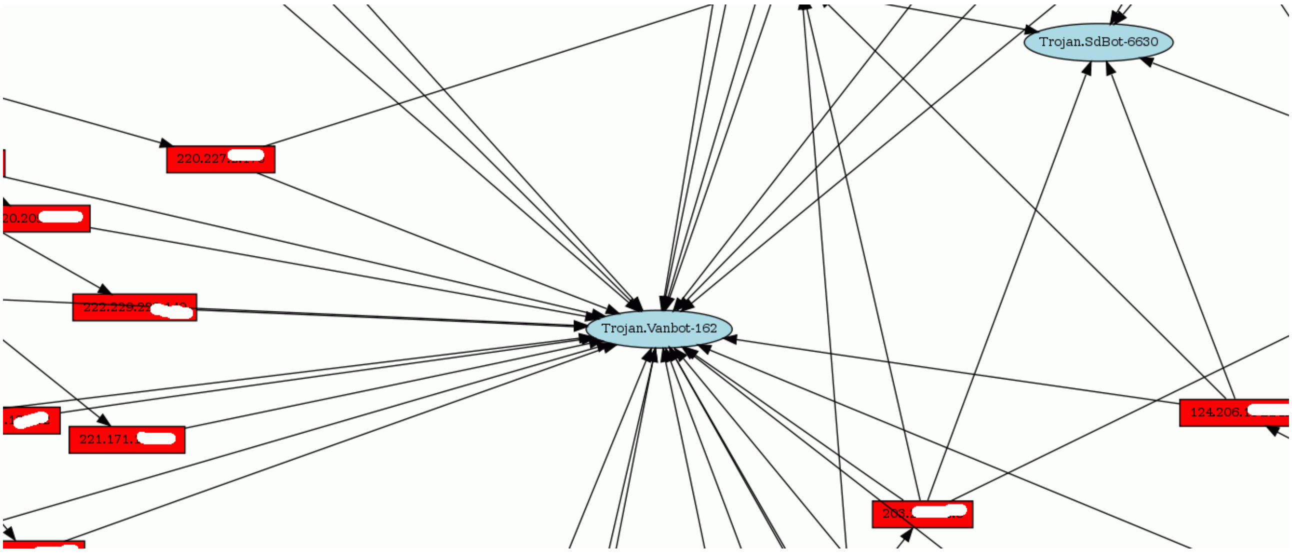


With the help of the graph we can visualize some interesting information:

- The country with more attacks:



- The more present malware:



Another Example

We will use the data provided by Nepenthes related to SQLSlammer to represent another graph. When an infected host try to attack Nepenthes host, Nepenthes will generate a log entry in nepenthes.log:
[18062008 21:46:57 info dia] 218.26.**.**:1064 asked us to join his SQLSlammer Party

Once again, we will parse this data and create a CSV file, this time with the following format:
country,attacker

First, we have to extract the SQLSlammer related data from Nepenthes log:

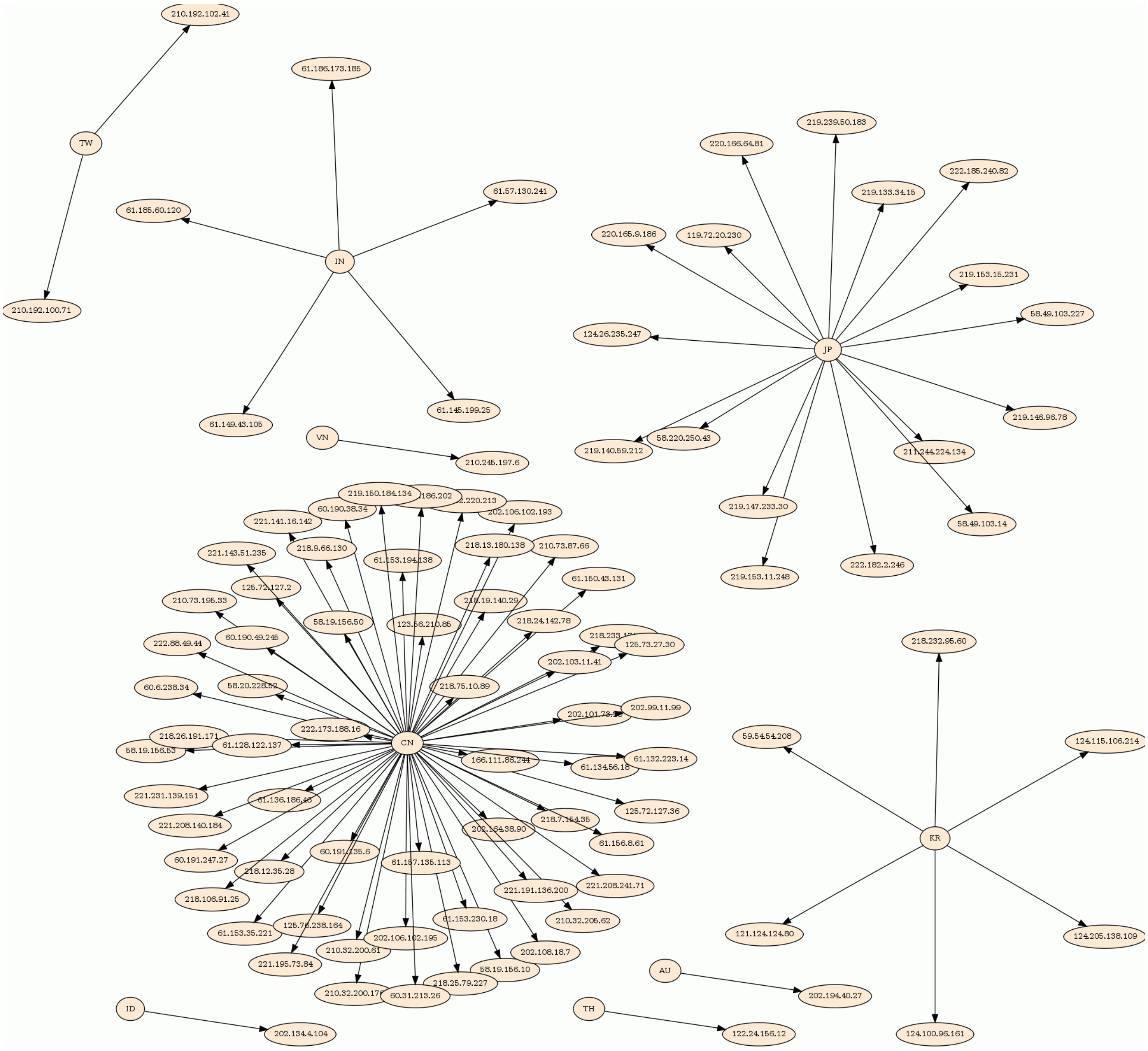
```
root@ubuntu # grep -i "party" nepenthes.log > /tmp/sqlslammer.log
```

And then use the following script to make the CSV file:

```
import re
import fileinput
import ip2country

slammerLog = "slammer.log"

ip2c = ip2country.IP2Country(verbose=0)

#[26072007 06:57:08 info dia] 202.106.**.**:1058 asked us to
#join his SQLSlammer Party
p = re.compile(r'\[.*?\]\s(?P<source>\S+):\S*? asked us
to join his SQLSlammer Party')
for line in fileinput.input(slammerLog):
        m = p.match(line)
        try:
                attacker = m.group(1)
                country = ip2c.lookup(attacker)
      #print attacker
                print country[0] + "," + attacker
        except:
                pass
```

Finally we save the CSV data and execute the necessary commands:

```
root@ubuntu # python slammer.py > slammer.csv
root@ubuntu # cat slammer.csv | perl afterglow/src/perl/graph/afterglow.pl
-c color.properties2 -t | > img.dot

root@ubuntu #cat img.dot | neato -Tgif -o slammer.gif
```



Conclusion

Log visualization is a very interesting way of representing large amount of data, with this method we can obtain some interesting data of the log files in a quick way.
With the tools described in this paper we can generate a lot of interesting graphs related to malware, we can correlate different sensors, countries and generate visual representations that may give us a general vision of the malware´s spread.

References

- **Nepenthes: http://nepenthes.mwcollect.org**
- **AfterGlow: http://afterglow.sourceforge.net**
- **GraphViz: http://www.research.att.com/sw/tools/graphviz**
- **Ip2country: http://www.freenet.org.nz/python/ip2country/ip2country.tar.gz**
- **ClamAv: http://www.clamav.net**