Cybersecurity analysis of Web portals

Zubair Khan

2020-10-17

Contents

INTRODUCTION

This project focuses on the conducting root cause analysis on what factors attract potential hackers to exploit web portal, specifically state or governmet sponsored and respresneted institute. There is a general perception that the weakness of a institute can be reviewed in the way they are represented online. By exploiting such weekneses a hacker reinforces that perception.

Information Gathering

The primary source of information was captured from www.zone-h.org. It's a resource used to catalog successful cyber-attacks on web portals. The information is not properly structured as a data set for analysis. Any subsequent information will have to be first extracted by filtering through current data set and then captured from other sources using Python scripts. The final data set is a composite of all information gathered in this process.

Project Ovierview

We will attempt to build a reliable data set from the noisy data that was extracted from a web portal. Using the primary data set we will build an additional dataset that will enrich the final dataset and allow us to perform analytics and ML algorithms. The project will flow in the following order

- Loading required libraries
- Acquisition and analysis of the initial Dataset
- Extracting relevant information for further data gathering
- Acquiring resulting data & merging with original dataset
- Data preparation by analyzing attributes
- Variable selection.
- Data exploration & Analysis
- Run ML algorithm on final dataset
- Splitting the dataset to a training and test dataset
- Run another ML algorithm on both training & test datasets
- Present result and findings

Goals

The final objective will be to analyze Hacker's pattern of behavior and identify how targets are picked for attack. We will further attempt to identify alternate course of action that should be considered during the initial planning process in an attempt to decrease probable risk factors.

METHOD

Loading Libraries

We will be utilizing a combination of geo map libraries to visualize data on wrold map. All libraries are configured to be installed upon execution with their respective dependencies.

We will be utilizing maps libraries to visually display geo locations that have been affected.

Data Acquizition

The original data set was extracted from the following path:

Path: "http://www.zone-h.org"

The data was extracted in CSV format as **Zone-h.csv** file attached to the Github portal below. The specified CSV contains a Domain column that is further filtered and extracted into a new CSV file **SiteCheck.CSV**. A python script is used to first verify if the respective domain can be resolved to an IP address else it is ignored. Next the script connects to online IP resolution service, using a webscrappting library it attempts to extract additional hosting service information and is eventually saved into the **webscrape.csv** file.

Both dataset were then merged into a Final dataset for further analysis. Any record that did not exist in the second data set was excluded from the final.

Github: "https://github.com/zubairmk83/CyberTest"

Table 1: Sample Data

S No.	Date	Notifier	H	M	R	L	Domain	OS	View
1	9/16/2020	Clash Hackers	NA	M	NA	NA	www.productive.pk/cl.html	Linux	mirror
2	9/15/2020	The3x	NA	NA	NA	NA	daroodcircle.pk/7PpDqcr.php	Linux	mirror
3	9/15/2020	The3x	NA	M	NA	NA	oxygonpakistan.pk/J61YkOM.php	Linux	mirror
4	9/15/2020	The3x	NA	M	NA	NA	medics.com.pk/8NrJ4P5.php	Linux	mirror
5	9/15/2020	Bla3k_D3vil	NA	M	NA	NA	acl.org.pk/a.html	Win 2012	mirror
6	9/15/2020	Bla3k_D3vil	NA	M	NA	NA	hospiramedicalsystem.com.pk/a....	Win 2012	mirror

The dataset consists of 2465 rows and 10 attributes. We notice that although the **H, M, R, L** columns appear promising. We will need to investigate and find out which attributes will contribute towards our analytics and which can be dropped.

Table 2: Initial data Summary statistics

S No.	Date	Notifier	H	M	R	L	Domain	OS	View
Min. : 1	Length:2465	Length:2465	Length:2465	Length:2465	Length:2465	Mode:logical	Length:2465	Length:2465	Length:2465
1st Qu.: 617	Class :character	Class :character	Class :character	Class :character	Class :character	NA's:2465	Class :character	Class :character	Class :character
Median :1233	Mode :character	Mode :character	Mode :character	Mode :character	Mode :character	NA	Mode :character	Mode :character	Mode :character
Mean :1233	NA	NA	NA	NA	NA	NA	NA	NA	NA
3rd Qu.:1849	NA	NA	NA	NA	NA	NA	NA	NA	NA
Max. :2465	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 2 depicts that most of the attributes are structured as characters. The Date column is also in character format which needs to be converted. Further analysis is required to be able to identify potential attributes.

Reviewing Table 2 we can see that the **L** attribute contains only 1 unique value. This is of no use to us therefore dropping redundant attributes.

We can safely remove the **H, M, R, L, View** attributes from our original dataset as they will not be able to contribute. There are more than 400 records where **H, M & R** attributes are all **NA**. Utilizing these

Table 3: Summarizing unique values in all attributes

S No.	Date	Notifier	H	M	R	L	Domain	OS	View
2465	855	563	2	2	2	1	2382	12	2

Table 4: Reviewing attributes that can be used as data factors

H	M	R	L	View	n
H	M	R	NA	mirror	92
H	M	NA	NA	mirror	240
H	NA	R	NA	mirror	108
H	NA	NA	NA	mirror	178
NA	M	R	NA	mirror	300
NA	M	NA	NA	mirro	1
NA	M	NA	NA	mirror	918
NA	NA	R	NA	mirror	204
NA	NA	NA	NA	mirror	424

attributes will mean that we will have to drop these records from our dataset. We will continue reviewing remaining attributes.

Table 5: Reviewing domain attribute

Domain
www.productive.pk/cl.html
daroodcircle.pk/7PpDqcr.php
oxygonpakistan.pk/J61YkOM.php
medics.com.pk/8NrJ4P5.php
acl.org.pk/a.html

Table 5 represents Domain column. The values are saved as a string format that needs to be split. This can be done by scanning through the string and splitting the columns using on the first occurrence of "/" symbol.

Table 6: Domain after fileration

Domains
www.productive.pk
daroodcircle.pk
oxygonpakistan.pk
medics.com.pk
acl.org.pk

The filtered Domain attribute is extracted to a **SiteCheck.csv** file for further processing.

Further Data Gathering

The filtered domain attribute saved in **SiteCheck.csv** is used as a data source to be first evaluated if it can be resolved. The evaluation process involves contacting a DNS service provider like "www.google.com" to verify if the DNS address can be matched to an IP address. This is important as all forms of online communication and traceability is only possible if we can identify the IP address. The result is saved in **SiteResult.csv**.

Once we can match the IP address of respective domain we will utilize webscraping libraries to automate and extract further information like geolocation, ISP and hosting services etc. The entire flow of the script can be better understood in the following diagram.

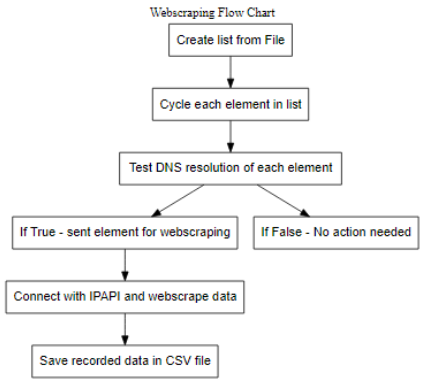

Figure 1: Webscraping flow chart

Merging Datasets

The resulting data acquired is first filtered to remove unneeded attributes and then merged with the original data set to create our Final dataset. Any source data that does not have trailing data in our second dataset is removed from our analysis.

Table 7: Duplicate Rows

	a	Site	IP.Address	IP.Decimel	Hostname	ASN	ISP	Organization	Services	Assignment	Blacklist	Continent	Country	Region	City	Latitude	Longitude	Postal.Code
2	2	15icge-9aygec.uet.edu.pk	104.28.9.226	1746668002	104.28.9.226	13335	Cloudflare	Cloudflare	None detected	Likely Static IP	None	North America	United States			37.751Â Â (37Â° 45â€² 3.606â€³ N)	-97.822Â Â (97Â° 49â€² 19.206â€³ W)	

The remaining dataset is merged with the source data set.

Table 8: Summary of imported dataset

a	Site	IP.Address	IP.Decimel	Hostname	ASN	ISP	Organization	Services	Assignment	Blacklist	Continent	Country	Region	City	Latitude	Longitude	Postal.Code
2085	2085	658	658	652	113	107	101	3	3	2	5	19	48	70	109	108	92

Data Preperation

In this section we will be filtering and cleaning the Final data set. Converting relevant attributes to needed format and removing attributes that will not be required.

Table 9: Merged Dataset

Domains	Date	Notifier	OS	IP.Address	Hostname	ASN	ISP	Organization	Continent	Country	Region	City	Lat	Lon
2063	841	559	12	655	649	113	107	101	5	19	48	69	108	107

The Dates column is converted into its respective format. Any records that do not have an IP address are removed and the **Notifier** attribute that represents Hacker name is cleared of any anomaly data that was captured during initial import. Additionally, the first letter of **Notifier** attribute is converted to lower case to clear duplicate name entries.

Table 10: Sample of Final Dataset

Domains	Notifier	OS	IP.Address	Hostname	ASN	ISP	Organization	Continent	Country	Region	City	Lat	Lon	Dates
15icge-9aygcc.uet.edu.pk	hunter bajwa	Linux	172.67.165.42	172.67.165.42	13335	Cloudflare	Cloudflare	North America	United States			37.7510	-97.8220	2020-06-29
3dtechnologies.com.pk	bla3k_d3vil	Win 2012	37.187.76.99	windows.websouls.net	16276	OVH SAS	OVH SAS	Europe	France			48.8582	2.3387	2020-09-15
3i.com.pk	royal battler bd	Linux	107.190.137.147	ns23.hostingcare.net	33182	HostDime.com	HostDime.com	North America	United States	Florida	Orlando	28.4647	-81.2468	2019-10-14
4yaarinc.pk	bla3k_d3vil	Win 2012	37.187.76.99	windows.websouls.net	16276	OVH SAS	OVH SAS	Europe	France			48.8582	2.3387	2020-09-15
7stripe.com.pk	chinafans	Linux	144.91.115.46	server4142.skyhost.pk	51167	Contabo GmbH	Contabo GmbH	Europe	Germany	Bavaria	Nuremberg	49.4050	11.1617	2020-08-06

Variable Selection

While reviewing the remaining variables we can so far summarize the following:

- Total number of hackers in the dataset is 452 .
- Total number of web portal affected are 1665.
- Total number of unique IP addresses are 654.
- Total countries in the dataset are 18 distributed within 4 continents.
- Total variations of Operating system are 12.
- Total number of records are 1973.

Data Exploration

We will try to visualize the finalized attributes to identify pattern. The below map displays overall distribution of attacks captured in our dataset.

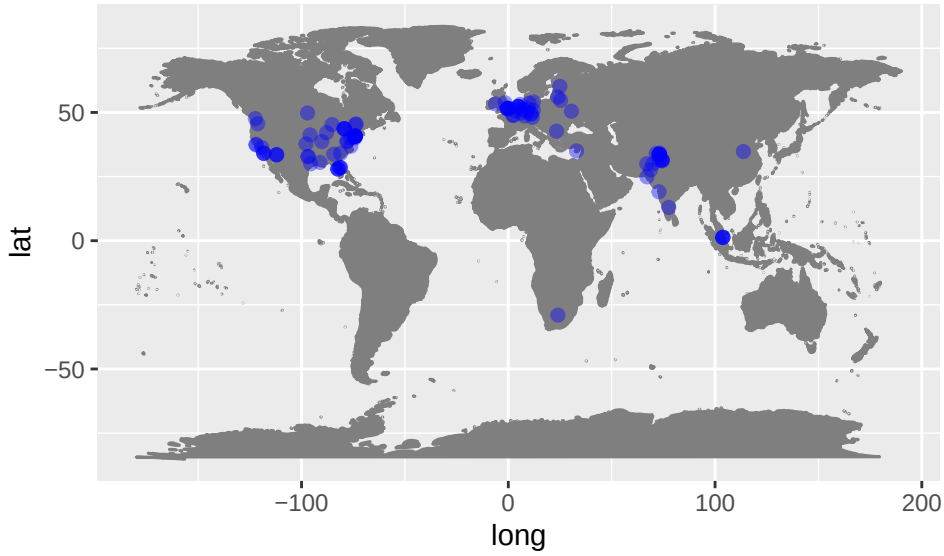

Figure 2: Hackers attack distribution

We notice that the attacks are focused on 3 regions. It is still unclear as to the reason of distribution. We will start by filtering result of on top 5 Hackers based on their number of attacks recorded.

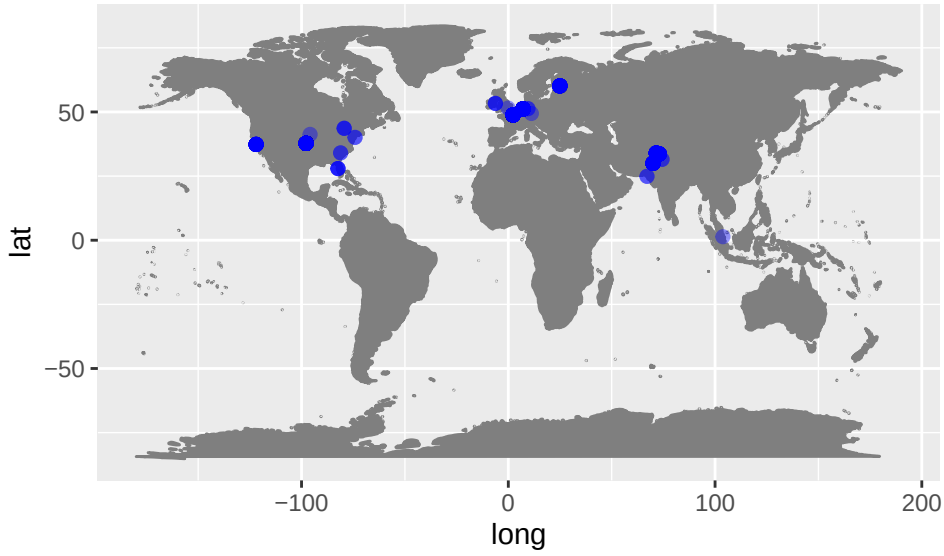

Figure 3: Top 5 Hacker attack distribution

There is a clear indication on 3 cluster formation based on region. We will now attempt to identify this cluster's contributing attribute or attributes.

From the above table we notice that the top recorded hacker in this dataset has the highest contribution towards a specific ISP.

Filtering the dataset based on selected ISP from table **Top hacker dataset** we observe that this ISP has been the victim of 33 successful hacks.

Table 11: Top hacker dataset

Notifier	ISP	n
bla3k_d3vil	OVH SAS	106
bla3k_d3vil	Hetzner Online GmbH	31
bla3k_d3vil	Limestone Networks	15
bla3k_d3vil	HIVELOCITY	6
bla3k_d3vil	Cloudflare	1

Table 12: Filtered ISP dataset

Notifier	ISP
bla3k_d3vil	OVH SAS
fakesmile	OVH SAS
mohamed.xo	OVH SAS
the3x	OVH SAS
krdsec	OVH SAS

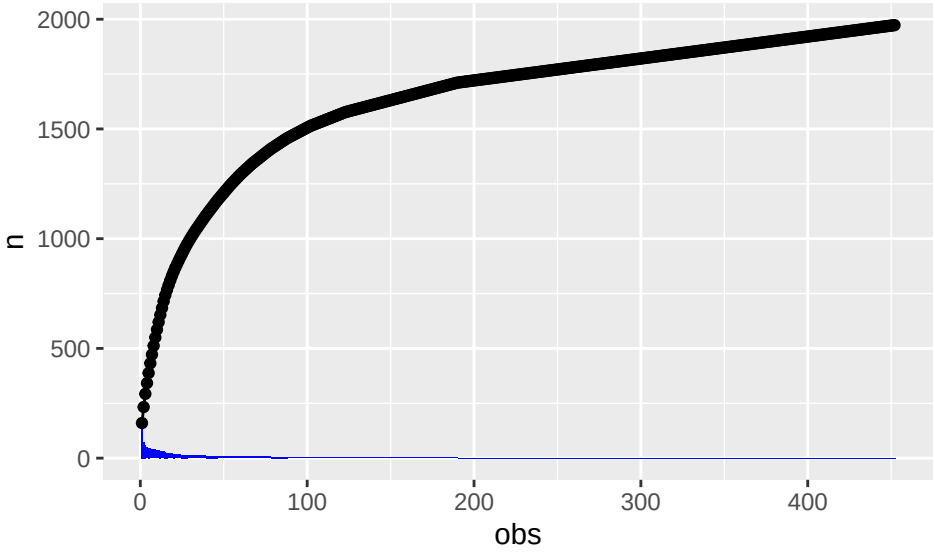

Figure 4: Attack distribution

Further analyzing the distribution of attacks we notice that majority of attacked are contributions of the first top 100 hackers arranged in descending.

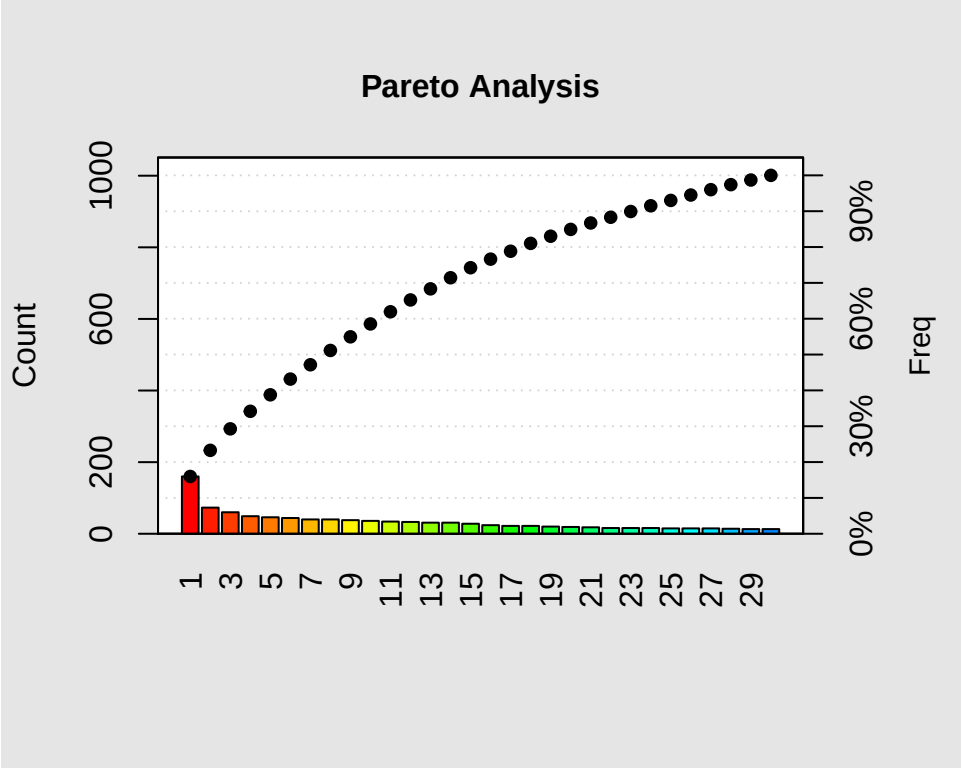

Figure 5: Top 30 attack distributions

Analyzing the pareto chart based on an 80/20 distribution indicates that 80% of data contributions are from less than top 20 hackers. For an even distribution of data, we will select the top 30 hackers in our proceeding analysis. Based on our finding from the above pareto analysis we can refine our world map to represent top 30 hackers.

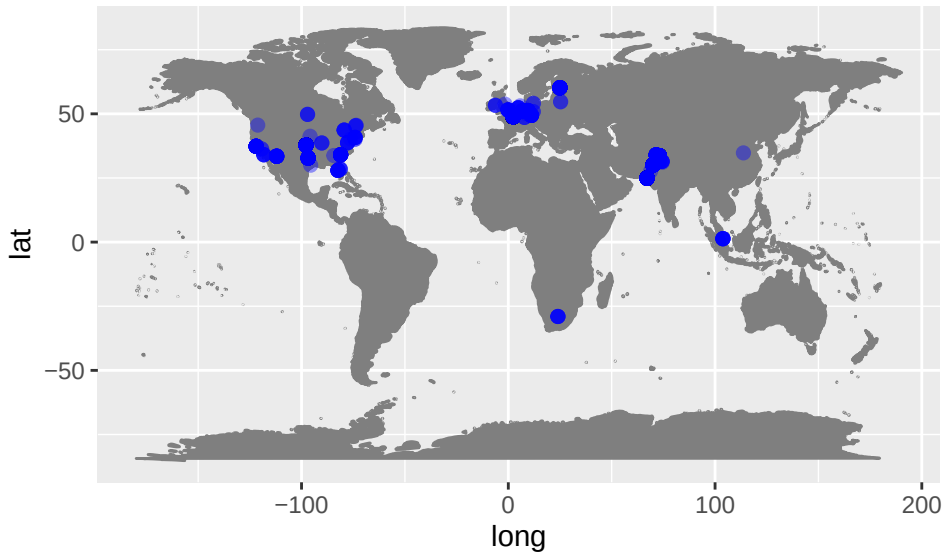

Figure 6: Analysing potential clusters

Focusing our analysis on Operating system we observer that there are a total of 12 operating system represented in the dataset. Among these 8 are the ones exploited by top 30 hackers. Looking at distribution of Operating system on world map

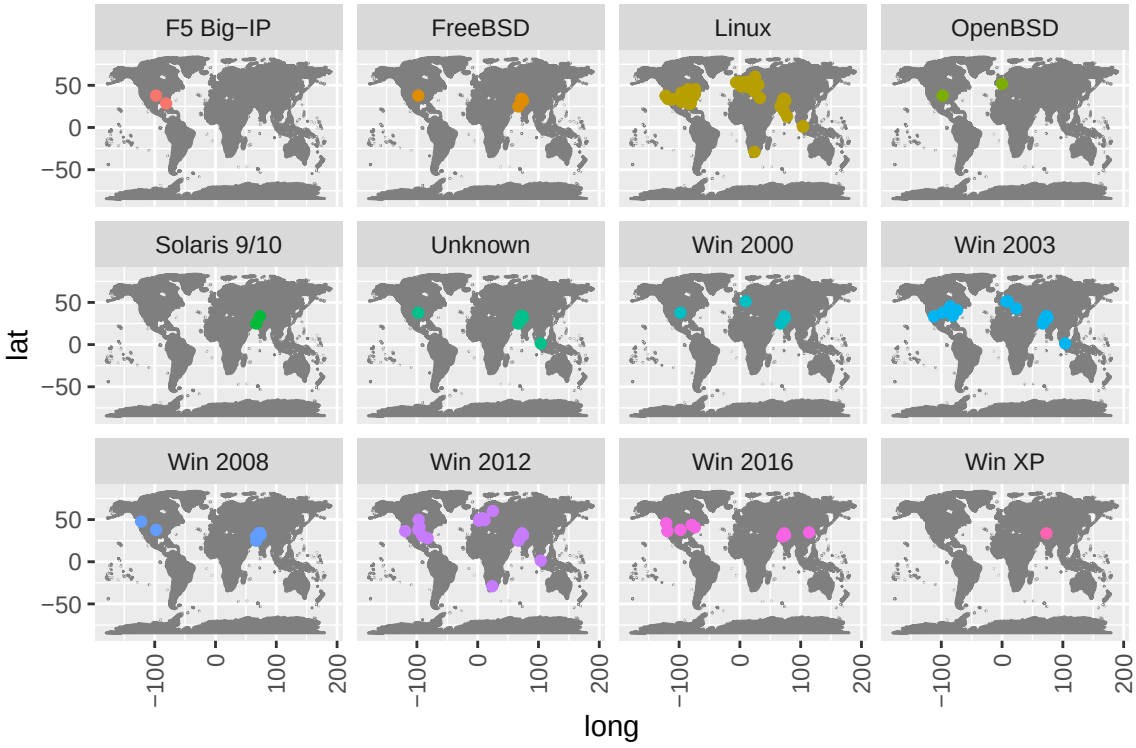

Figure 7: OS distribution

We observe that the **Linux, win 2003 & win 2012** Operating systems have a saturated distribution from the rest. The **Linux** term used here represents custom Linux distributions that are not identifiable from the web scrap data source. The **unknown** Operating system can represent appliance that is specifically build to host web portals. It is surprising to see that there are a number of Microsoft based operating systems that are no longer supported but are still in use.

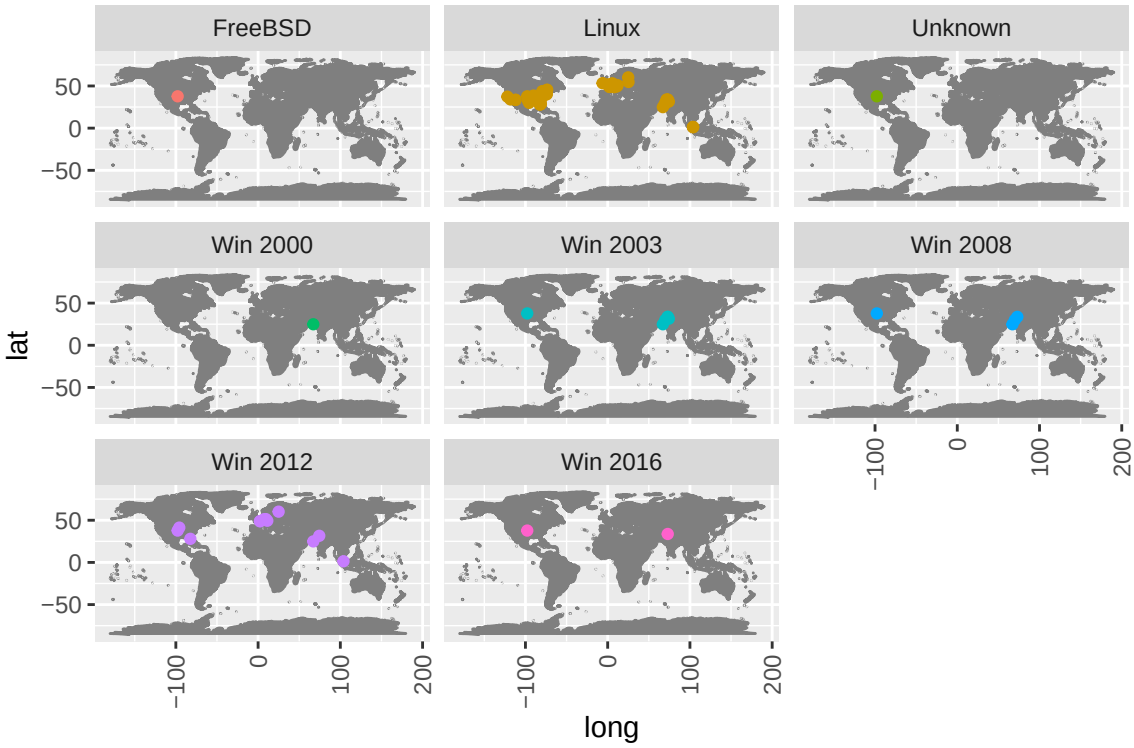

Figure 8: OS targeted by top 30 hackers

Reviewing the attack distribution based on country by using pareto analysis we find that 80% of attacks are concentrated in **Pakistan** & **USA**.

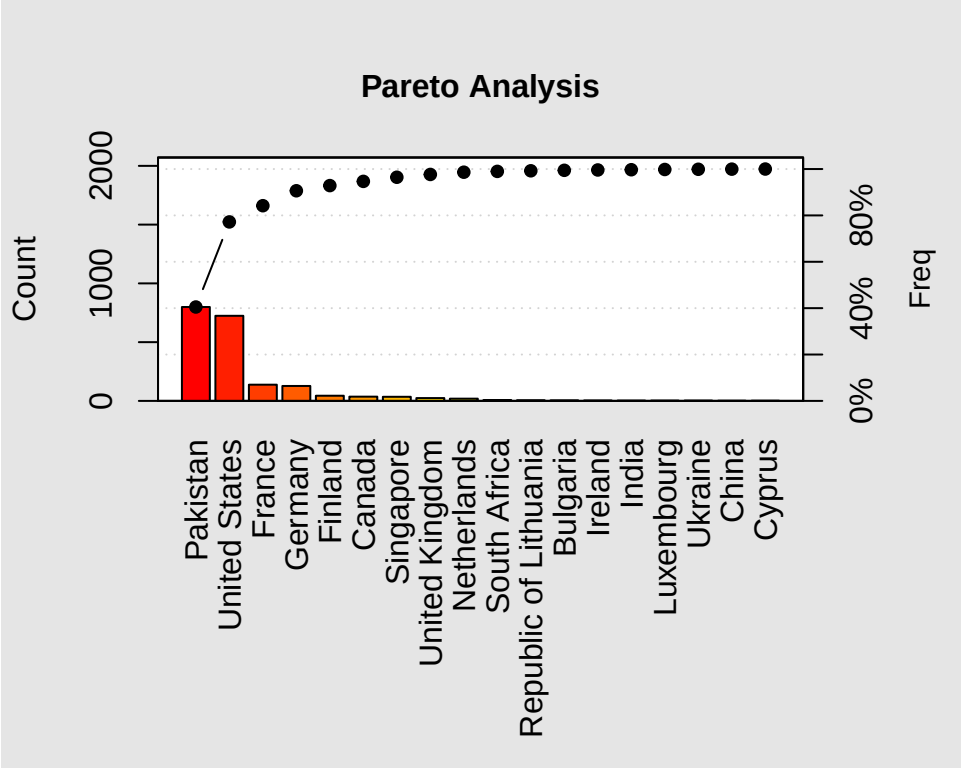

Figure 9: Pareto analysis of OS

This matches out initial cluster assessment but still leaves the primary question unanswered. Which attribute has the most contribution towards the creation of the below represented 3 clusters.

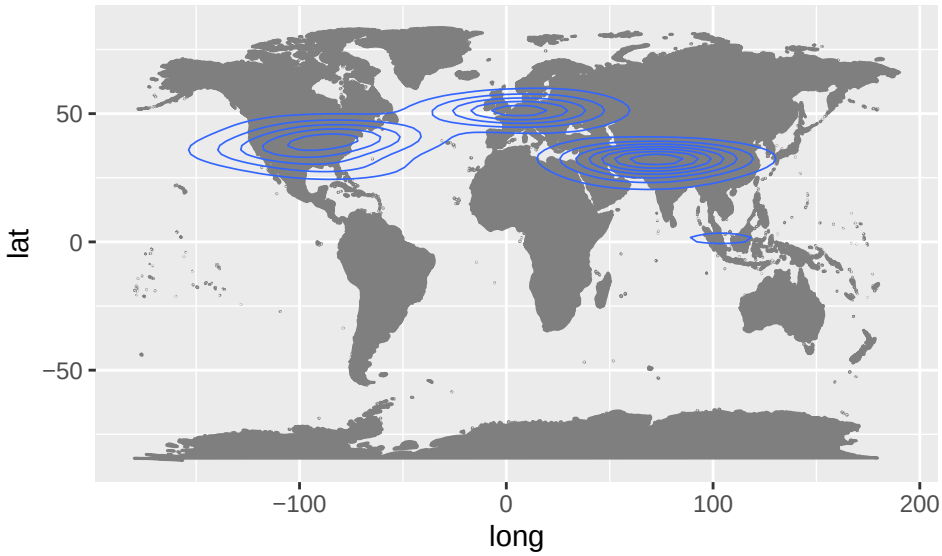

Figure 10: 3 Clusters identification

Kmeans Algorithm

We will be utilizing Kmeans machine learning algorithm to narrow down the contributing attributes. In order to achieve this, we will be converting the **Notifier, OS, ISP & Country** attributes to factor forms by first sequencing and labeling them in descending order of occurrence and then replacing the allocated labels to their respective sequence. The code that will carry out this activity is given below.

Code for this process is represented below

```
#Arranging Hackers in decending order
a <- Final %>% group_by(Notifier) %>% count(Notifier) %>% arrange(desc(n)) %>% select(Notifier)
#creating a sequence that is of lenght a
b <- seq.int(nrow(a))
#creating a dataframe and combining both entries
Final1 <- data.frame(a,Hacker = b)
#Joining our dataframe to every occurance in the initial dataset. Sort of how vlookup works in excel.
Final <- left_join(Final,Final1, by="Notifier")


#repeating the process for remaining attributes.
a <- Final %>% group_by(OS) %>% count(OS) %>% arrange(desc(n)) %>% select(OS)
b <- seq.int(nrow(a))
Final1 <- data.frame(a,OpSys = b)
Final <- left_join(Final,Final1, by="OS")

a <- Final %>% group_by(ISP) %>% count(ISP) %>% arrange(desc(ISP)) %>% select(ISP)
b <- seq.int(nrow(a))
Final1 <- data.frame(a,ISPs = b)
Final <- left_join(Final,Final1, by="ISP")

a <- Final %>% group_by(Country) %>% count(Country) %>% arrange(desc(n)) %>% select(Country)
b <- seq.int(nrow(a))
Final1 <- data.frame(a,Ctry = b)
Final <- left_join(Final,Final1, by="Country")
remove(Final1,a,b)
```

Once that is done, we will execute the Kmeans ML algorithm and represent it in a graphical format.

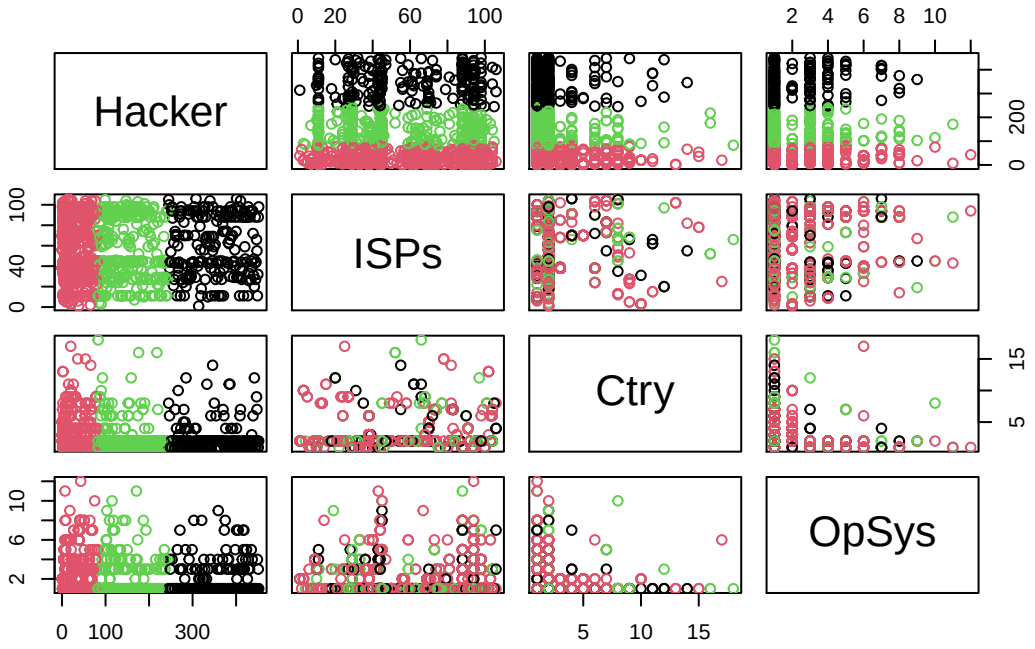

Figure 11: Kmeans visual representation

Using the chart above we can conclude that the primary approach a hacker takes is not what is generally perceived. Based on our analysis, a hacker targets a potential ISP for weakness. Once identified, they will then move to affecting all available hosted services in that specific ISP. Their selection of potential target least involves Operating system exploitation or a hosting country. The list of portals in the domains attribute are potential victims to a flawed infrastructure. This is opposite to what the general approach when we plan on hosting web portal. The general planning that is involved while making key decisions on creating web portals is as follows:

- Finalizing on the content type (static or dynamic).
- Selecting the easiest language this can be programmed in or whose expertise is readily available.
- Selecting the easiest OS that it will be hosted on.
- Reviewing the cost of hosting the portal.
- Selecting the cheapest one that meets management requirement while keeping operational cost as low as possible.

It is generally perceived that the ISP is responsible for the security aspect of the web portals and in most cases they are as stipulated in the terms and conditions agreements. But having said that, the impact a defacement has does not affect the image of the ISP but rather the web portal owner. Due diligence is needed especially when such online portals represent an institute that reflects service presented by a state or country. Delivering services in a secured and reliable manner is more important and should be considered at top priority when making such decisions.

Data Split

For our next analysis we will approach the problem by proposing an insight into the current planning behaviors and what should be the proposed method to approach. We will be using the **decision tree**

algorithm on our training and test datasets. We will also exclude the Hacker attribute from the dataset as the effect a hacker has is a byproduct of a cybersecurity weakness.

```
remove(Final1)
Final1 <- Final %>% select(OpSys, ISPs, Ctry)
Final1$OpSys <- as.factor(Final1$OpSys)
#Splitting the dataset to a training and test data set at 80:20 ration
pd <- sample(2, nrow(Final1), replace = TRUE, prob = c(0.8,0.2))
train <- Final1[pd==1,]
test <- Final1[pd==2,]
```

We will be sticking with the standard 80 / 20 ratio where 80% of the data is used in our training algorithm and the 20% remaining data is used to test our hypothesis.

Decision Tree Algorithm

Using the remaining 3 attributes we will use Decision tree method to propose a better selection approach to our planning phase. We know that the primary three elements in building a web portal are:

- Finalizing content
- Selecting programing language
- Selecting hosting platform

The dataset does not contain information on the first two attributes but has information on operating system its hosted on. Using **OpSys** attribute as factorial variable we will run decision tree analysis on our remaining dataset.

```
#building a Decision tree
tree <- ctree(OpSys~., data=train, controls = ctree_control(mincriterion = 0.9, minsplit = 300) )
#there are a total of 19 nodes in this tree. the tree is set to 90 % confidence level and split is kept
#this diagram is basically upside down with root at the top and leaves at the bottom.
#checking the probability distribution of our dataset we observer that all variations belong to top 4 c
#of OpSys
predict(tree, test)
```

```
##   [1] 1 1 2 1 1 1 1 1 3 1 1 1 1 2 1 1 2 1 1 1 1 1 2 4 4 1 1 1 1 1 2 1 1 1 1 1 1
##  [38] 2 1 1 1 1 1 1 3 1 4 2 1 1 1 2 1 2 1 1 1 1 3 1 1 2 1 1 1 1 1 1 3 4 4 2 1 1
##  [75] 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 1 2 2 1 2 1 1 1 2 1 1 1 1 1 4 1 1 1 1 1 1 1
## [112] 1 4 1 4 1 1 1 1 2 1 1 1 1 1 2 1 2 4 2 1 1 1 1 1 1 1 1 1 4 1 1 1 2 2 1 2 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 2 2 2 4 1 4 4 2 2 1 1 2 2 1 1 2 1 1 1 1
## [186] 1 1 1 1 4 1 4 1 1 4 1 1 1 1 1 4 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1
## [223] 1 1 1 4 1 4 4 1 1 1 1 2 1 1 2 1 1 1 4 1 1 1 1 2 1 1 1 3 1 4 1 1 1 2 1 1 1
## [260] 1 2 1 1 2 1 2 1 1 2 1 1 1 2 1 2 1 1 1 2 2 1 1 1 1 2 1 2 1 2 1 2 2 1 4 1 2
## [297] 1 2 1 2 1 2 2 1 1 1 1 4 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1
## [334] 3 1 1 1 3 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1
## Levels: 1 2 3 4 5 6 7 8 9 10 11 12
```

Using the decision tree method, we observer that our prediction is only limited to the top 4 classes out of a total of 12. The **Ctry** attribute is at the top of the tree which depicts it as a root node. We will read the tree in reverse since lower numbers portray higher risk factory. We will follow the decision tree in our planning phase focusing on higher numbers when choosing branches.

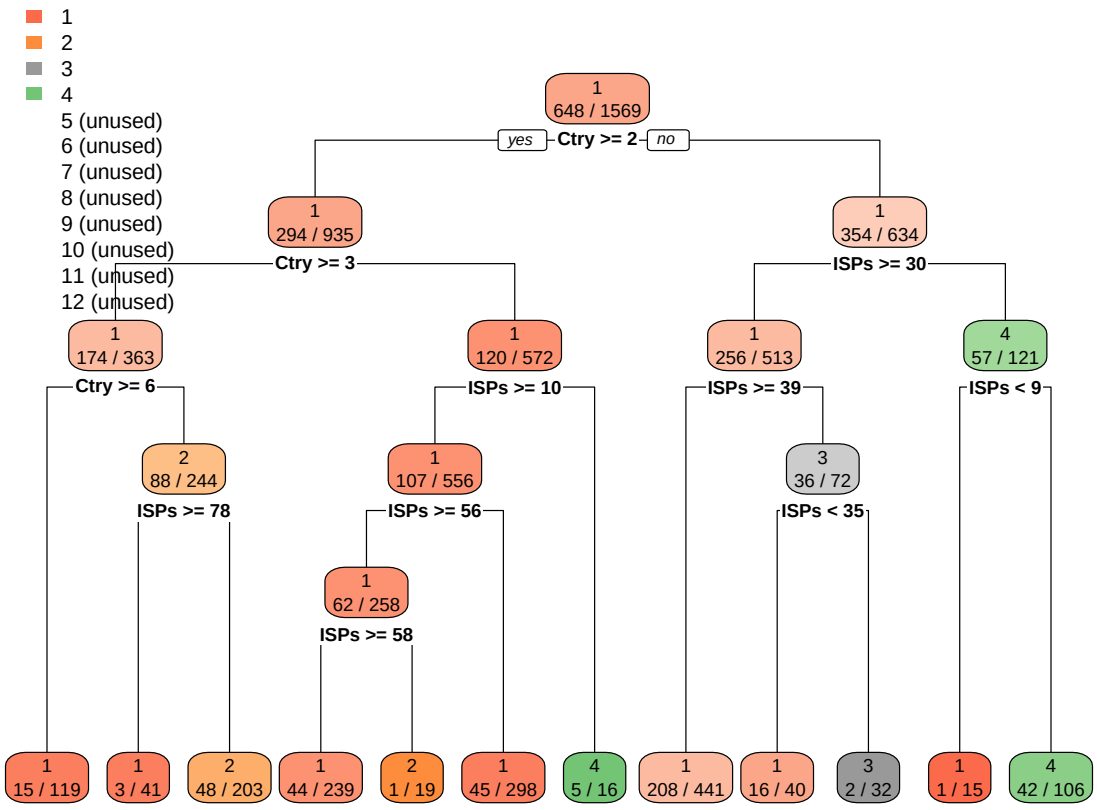

Figure 12: Decision tree analysis

As an observation if we refer to the **Country Pareto Chart** and we pick **Germany** for hosting our web portal, as it is not the top two countries we satisfy the **root** condition and move to the next observation towards the left. On our next node our initial choice is still higher than the prescribed condition, therefore we move to the left again and so forth.

```
#Misclassification error for "train" data
tab <- table(predict(tree), train$OpSys)
trainResult <- 1-sum(diag(tab))/sum(tab)
#misclassification error is about 33% based on training data and is
#focused on top 4 used operating systems


testPrediction <- predict(tree, newdata=test)
tab <- table(testPrediction, test$OpSys)
testResult <- 1-sum(diag(tab))/sum(tab)
#misclassification error is about 28% based on test data and is also
#focused on top 4 used operating systems
```

Results

Finally, we use the train dataset to calculate the misclassification error and run it against our test dataset. We observe the training dataset contains 1569 observations and works with **77%** accuracy. While the test

dataset contains 404 observations and works with **81%** accuracy. Although the dataset is quite limited, but we can safely state that following the specified model of approach in our decision making process we can lower the potential risk factor of our web portal by more then 70%.

CONCLUSION

Cybersecurity is a broad spectrum and has its own limitations. The most prominent limitation that is observed is of a personal image status rather than of technical nature. The willingness of an institute to be transparent and open in such scenarios can empower cyber defense bodies to build better frameworks. It's a process of collecting factorial data, cross validating it and eventually identifying emerging patterns from it.

The construct of this report was both challenging and insightful. Consolidating such information into a collective dataset where hosting services are categorized and graded based on the strength of their cybersecurity framework and that information is available to the masses will be the next phase of this project.