

A Project Report

on

***Analysis of Cowrie Honeypot SSH Logs Using
Elastic Search, Logstash and Kibana***

by

AVINESH BENJAMIN

2017H1030080H

M.E Computer Science

SARADHI RAMAKRISHNA

2017H1030081H

M.E Computer Science

Under the supervision of

PROFESSOR GEETHA KUMARI

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS OF
BITS G513 NETWORK SECURITY



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
HYDERABAD CAMPUS
(APRIL 2018)**

ACKNOWLEDGMENTS

I owe my sincere gratitude to Prof. GEETHA KUMARI, a distinguished faculty at BITS Pilani ,Hyderabad Campus for her constant guidance, invaluable suggestions and advice at every stage of the project.



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
HYDERABAD CAMPUS

Certificate

This is to certify that the project report entitled “Analysis of Cowrie Honeypot SSH Logs Using Elastic Search, Logstash and Kibana” submitted by Mr AVINESH BENJAMIN(ID No. 2017H1030081H) and Mr.SARADHI RAMAKRISHNA (ID No. 2017H1030081H) in partial fulfilment of the requirements of the course BITS G513, Network Security, embodies the work done by them under my supervision and guidance.

(GEETHA KUMARI)

BITS- Pilani, Hyderabad Campus

Date : 10-04-2018

Contents

1	Introduction of Tools	1
1.1	ELK Stack	1
1.1.1	Elastic Search	1
1.1.2	Logstash	1
1.1.3	Kibana	1
1.2	Honeypots	1
1.2.1	Low-interaction honeypots	2
1.2.2	Medium-interaction honeypots	2
1.2.3	High-interaction honeypots	2
1.2.4	Cowrie Honeypot	2
1.3	Github	3
2	Installation of Tools	3
2.1	Elastic Search	3
2.2	Logstash	4
2.3	Kibana	4
2.4	Cowrie Honeypot	5
2.4.1	Install dependencies	5
2.4.2	Create a user account	5
2.4.3	Checkout the code	5
2.4.4	Setup Virtual Environment	6
2.4.5	Port redirection	6
2.4.6	Starting Cowrie	6
3	Monitoring System	6
3.1	Stochastic Process	12
4	Conclusion	12

List of Figures

1	Elastic Search Output	4
2	Kibana Dashboard	5
3	Monitoring System	6
4	Daily Attacks Count Area Graph	9
5	Unique Count of IP's	10
6	Global Attacks Map	11
7	Stochastic Process	12

1 Introduction of Tools

In this section, i will be introducing the tools ,languages and API's which will be used throughout the report.

1.1 ELK Stack

"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana.

1.1.1 Elastic Search

Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

1.1.2 Logstash

Logstash is an open source tool for collecting, parsing, and storing logs for future use.

1.1.3 Kibana

Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.

1.2 Honeypots

In computer terminology, a honeypot is a trap set to detect, deflect or in some manner counteract attempts at unauthorized use of information systems.

In other words, A server that is configured to detect an intruder by mirroring a real production system. It appears as an ordinary server doing work, but all the data and transactions are phony. Located either in or outside the firewall, these are used to learn about an intruder's techniques as well as determine vulnerabilities in the real system."

Based on design criteria, honeypots can be classified as

1. Low-interaction honeypots
2. Medium-interaction honeypots
3. High-interaction honeypots

1.2.1 Low-interaction honeypots

Low-interaction honeypots simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the security of the virtual systems.

Low-interaction honeypots present the hacker emulated services with a limited subset of the functionality they would expect from a server, with the intent of detecting sources of unauthorized activity. For example, the HTTP service on low-interaction honeypots would only support the commands needed to identify that a known exploit is being attempted.

1.2.2 Medium-interaction honeypots

Medium-interaction honeypots might more fully implement the HTTP protocol to emulate a well-known vendor's implementation, such as Apache. However, there are no implementations of a medium-interaction honeypots and for the purposes of this paper, the definition of low-interaction honeypots captures the functionality of medium-interaction honeypots in that they only provide partial implementation of services and do not allow typical, full interaction with the system as high-interaction honeypots.

1.2.3 High-interaction honeypots

High-interaction honeypots imitate the activities of the real systems that host a variety of services. It let the hacker interact with the system as they would any regular operating system, with the goal of capturing the maximum amount of information on the attacker's techniques. Any command or application an end-user would expect to be installed is available and generally, there is little to no restriction placed on what the hacker can do once he/she comprises the system.

According to recent researches in high interaction honeypot technology, by employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. Although high interaction honeypots provide more security by being difficult to detect, but it has the main drawback that it is costly to maintain.

1.2.4 Cowrie Honeypot

Cowrie is used for our research purposes. Cowrie is a medium interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker.

Interesting Features of Cowrie Honeypot

1. Fake filesystem with the ability to add/remove files. A full fake filesystem resembling a Debian 5.0 installation is included.
2. Possibility of adding fake file contents so the attacker can cat files such as `/etc/passwd`. Only minimal file contents are included.
3. Logging in JSON format for easy processing in log management solutions.

1.3 Github

GitHub (originally known as Logical Awesome LLC) is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

The code which is written for this project is entirely pushed to Github where it is bug free and version safe.

2 Installation of Tools

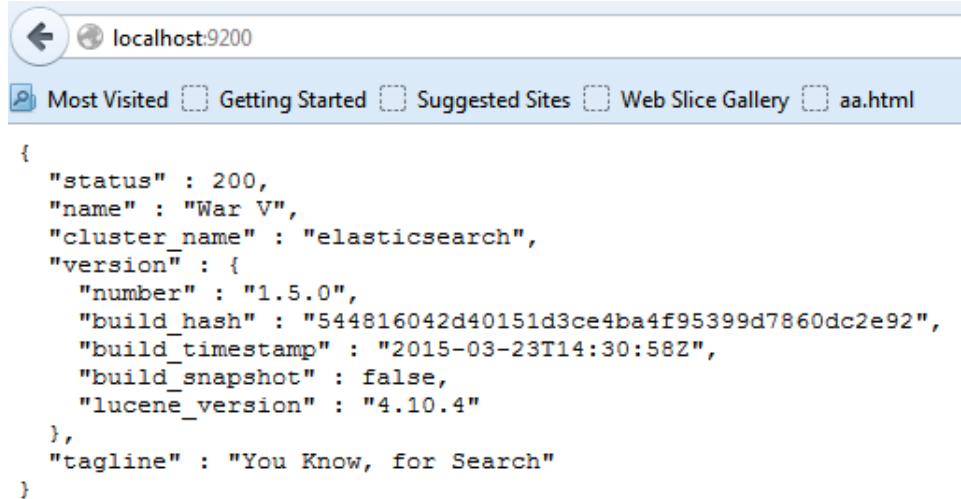
In this section, we will be mentioning about the installation of various tools which will be used in the project.

2.1 Elastic Search

1. Download the Debian Installation file from the below link
2. Install it using the below command
iotsys3@iotsys3-Precision-Tower-3420: dpkg -i elastic-search-6.2.0.deb
3. Now start the service using below command
iotsys3@iotsys3-Precision-Tower-3420: sudo service elasticsearch start

Open the link `http://localhost:9200` in the browser and check for the output which should look like below.

Figure 1: Elastic Search Output



We can check for the working of elastic search in the browser by typing the url `http://localhost:9200` and verifying the output.

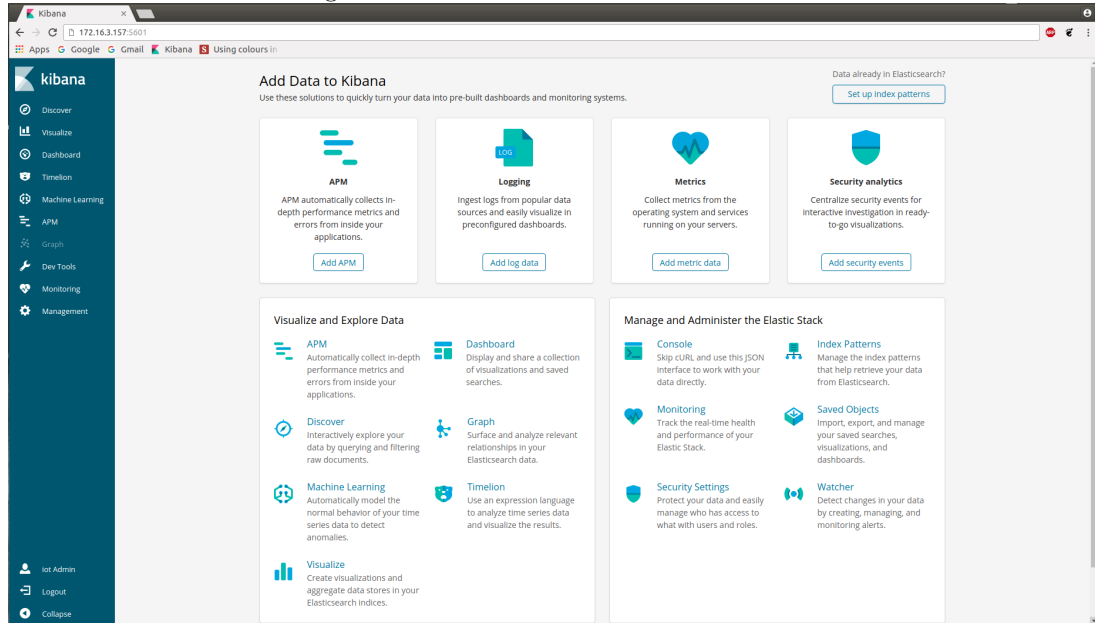
2.2 Logstash

1. Download the logstash debian installation file from below link
`https://www.elastic.co/downloads/logstash`
2. Install it using the below command
`iotsys3@iotsys3-Precision-Tower-3420: dpkg -i logstash-6.2.2.deb`
3. Now start the service using below command
`iotsys3@iotsys3-Precision-Tower-3420: sudo service logstash start`

2.3 Kibana

1. Download the Kibana debian installation file from below link
`https://www.elastic.co/downloads/kibana`
2. Install it using the below command
`iotsys3@iotsys3-Precision-Tower-3420: dpkg -i kibana-6.2.2.deb`
3. Now start the service using below command
`iotsys3@iotsys3-Precision-Tower-3420: sudo service kibana start`

Figure 2: Kibana Dashboard



Open the link (<http://localhost:5601>) in the browser and check for the Kibana Dashboard which looks like above.

2.4 Cowrie Honeypot

Cowrie honeypot is installed in a separate machine with public IP which is connected to the internet. It can be installed by following the steps given below.

2.4.1 Install dependencies

```
iotics3@iotics3-Precision-Tower-3420: sudo apt-get install git python-virtualenv  
libssl-dev libffi-dev build-essential libpython-dev python2.7-minimal authbind
```

2.4.2 Create a user account

```
iotics3@iotics3-Precision-Tower-3420: sudo adduser --disabled-password cowrie  
iotics3@iotics3-Precision-Tower-3420: sudo su - cowrie
```

2.4.3 Checkout the code

```
iotics3@iotics3-Precision-Tower-3420: git clone http://github.com/micheloosterhof/cowrie  
iotics3@iotics3-Precision-Tower-3420: cd cowrie
```

2.4.4 Setup Virtual Environment

```
iotsys3@iotsys3-Precision-Tower-3420: pwd  
iotsys3@iotsys3-Precision-Tower-3420: virtualenv cowrie-env
```

2.4.5 Port redirection

```
iotsys3@iotsys3-Precision-Tower-3420: sudo iptables -t nat -A PREROUTING  
-p tcp -dport 22 -j REDIRECT --to-port 2222
```

In the same folder, create a copy of *cowrie.cfg.dist* and modify the file according to your needs like changing the name, SSH port number etc.

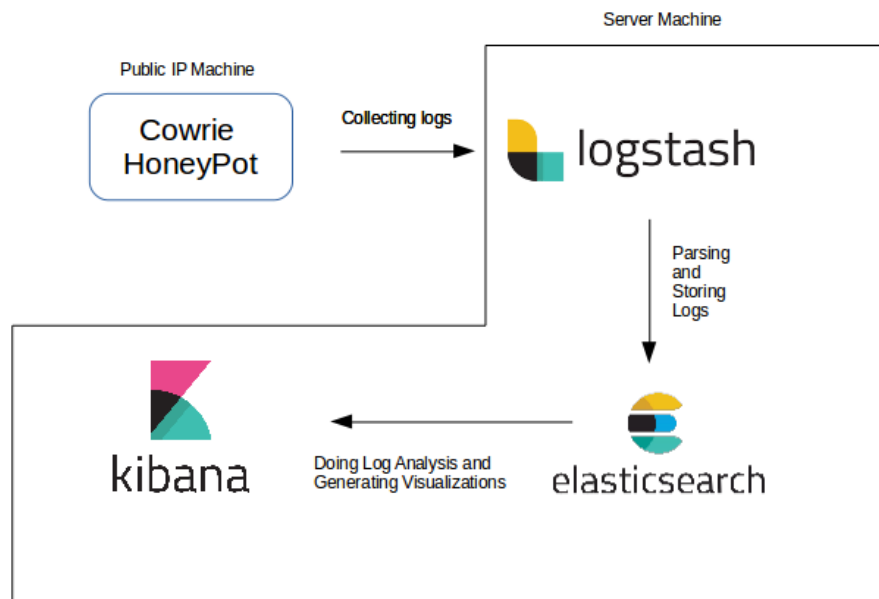
2.4.6 Starting Cowrie

```
iotsys3@iotsys3-Precision-Tower-3420: bin/cowrie start
```

Cowrie will be up and running on port 22 which will monitor all SSH logins and commands executed by the attackers or intruders.

3 Monitoring System

Figure 3: Monitoring System



Cowrie will be generating the logs in the form of JSON format which will be available in public IP machine as specified in the figure-3. In order to stream the logs from the public IP machine to Server machine, filebeat can be used which is an open source tool. Unfortunately we did not get to use that tool as there was no physical connection between public IP machine and server machine due to security reasons.

So all the logs are copied from public IP machine to server machine on daily basis. These log files are supplied as input to Logstash which will parse, add metadata and push them to Elastic Search database. Logstash works on config files. Each config file has three parameters namely input, filter and output. Our config file is given below.

Listing 1: Cowrie Config File

```

1 input {
2   file {
3     path => "/home/iotsys3/Documents/Cowrie-Logs/*"
4     start_position => "beginning"
5     ignore_older => 0
6   }
7 }
8 filter {
9
10  json {
11    source => message
12  }
13
14  date {
15    match => [ "timestamp", "ISO8601" ]
16  }
17
18  if [src_ip] {
19
20    dns {
21      reverse => [ "src_host", "src_ip" ]
22      action => "append"
23    }
24
25    geoip {
26      source => "src_ip" # With the src_ip field
27      target => "geoip" # Add the geoip one
28      # Using the database we previously saved
29      database => "/home/iotsys3/Downloads/GeoLite2-
30      City_20180206/GeoLite2-City.mmdb"
31      add_field => [ "[geoip][coordinates]", "%{[geoip][
32      longitude]}" ]
33      add_field => [ "[geoip][coordinates]", "%{[geoip][
34      latitude]}" ]
35    }
36
37    # Get the ASN code as well
38    geoip {
39      source => "src_ip"

```

```

37     database => "/home/iotsys3/Downloads/GeoLite2-
ASN.20180206/GeoLite2-ASN.mmdb"
38   }
39
40   mutate {
41     convert => [ "[geoip][coordinates]", "float" ]
42   }
43 }
44
45 }
46 output {
47
48   elasticsearch {
49     hosts => ["localhost:9200"]
50     sniffing => true
51     manage_template => false
52     index => "logstash-cowrie-%{+YYYY.MM.dd}"
53     user => "elastic"
54     password => "63TxZjWQNb4tArR7hri"
55   }
56
57   stdout {
58     codec => rubydebug
59   }
60 }

```

1. Input

It contains a parameter file which has options path, start-position and ignore-older. Path is where log files are present and start-position indicates from which point Logstash should start reading the logs from.

2. Filter

Filter is where entire processing happens in logstash. It has many options out of which json option is used as the logs are in JSON format. Some more information is added to the already present logs like country , Continent , latitude and longitude information which are specified in geoip option. Finally latitude and longitude are converted into float variables for plotting on map.

3. Output

It specifies options as to where the data should go after the processing is done. There are many options out of which we are using elastic search where data is stored in json format which will be in key value format.

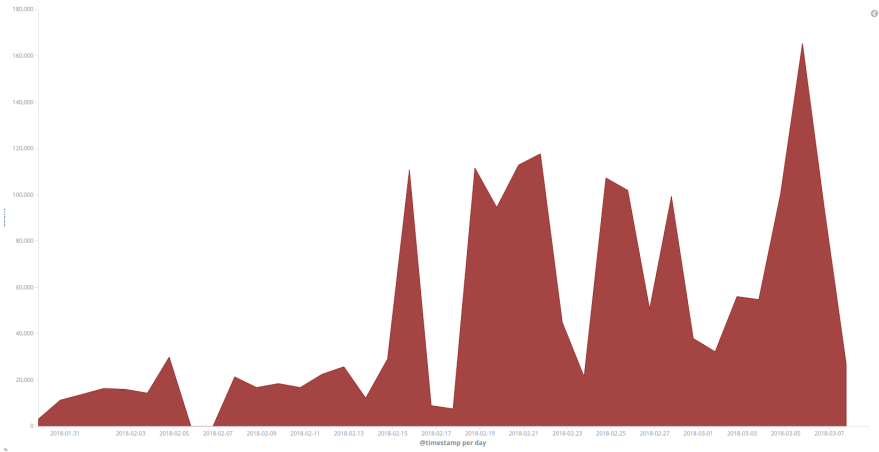
Data collected in the elastic search will be picked up by the Kibana by default as specified in the Kibana config files. Using the data visualizations are created for analyzing and monitoring the regular changes in data. Monitoring system was deployed for 40 days and collected nearly **2 Million Records**.

Using Kibana different visualizations are created which are listed below.

1. Daily Attacks Count Area Graph

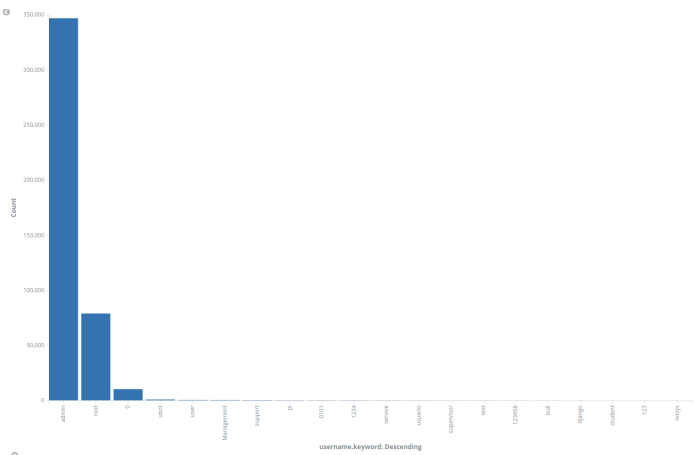
An area graph was plotted against the count of attacks on daily basis. X-axis represents the date and Y-axis represents the count of attacks. It can be observed that, attacks kept on increasing at the end.

Figure 4: Daily Attacks Count Area Graph



2. Top 20 Usernames Histogram

A Histogram was used to represent the top 20 usernames that have been used by the attackers and intruders.



From the graph, it can be observed that usernames "admin", "root" and "ubnt" are mostly used by attackers.

3. Unique Count Of IP's

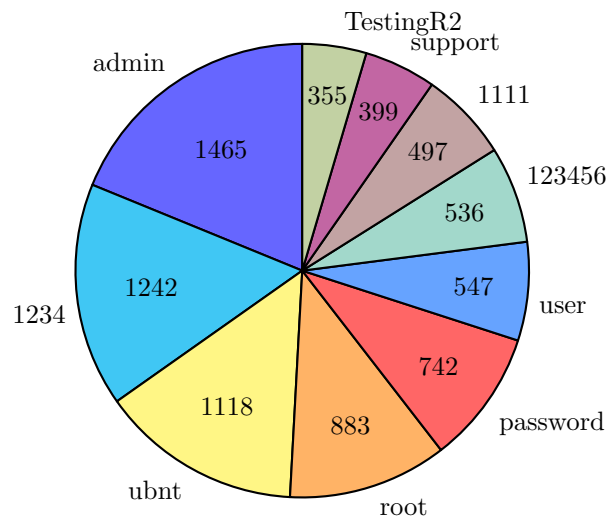
A Kibana Utility graph named Unique Count has been used for counting the Unique Number of IP's till now monitoring system has captured. It has found to be nearly 2000 IP's.

Figure 5: Unique Count of IP's



4. Top 10 Passwords Pie

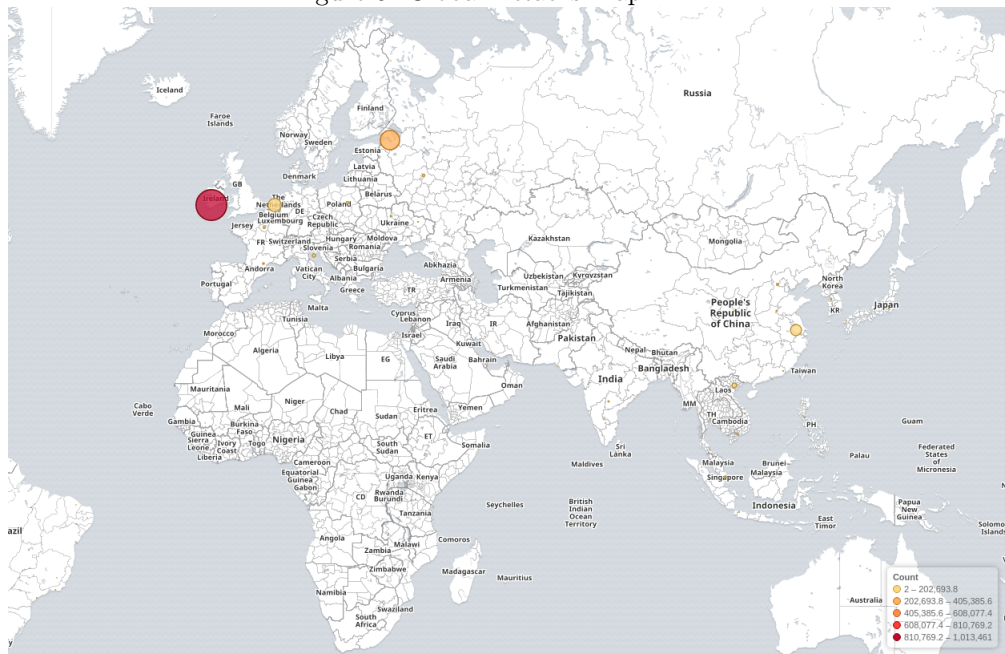
A pie chart has been drawn against the top 10 passwords used by the attackers and intruders. It can be observed that passwords "ubnt" and "admin" are mostly used.



5. Global Attacks Map

A map is also plotted against the attacks receiving from different countries and continents using the latitude and longitude values. Most of the attacks are received from European continent and China country.

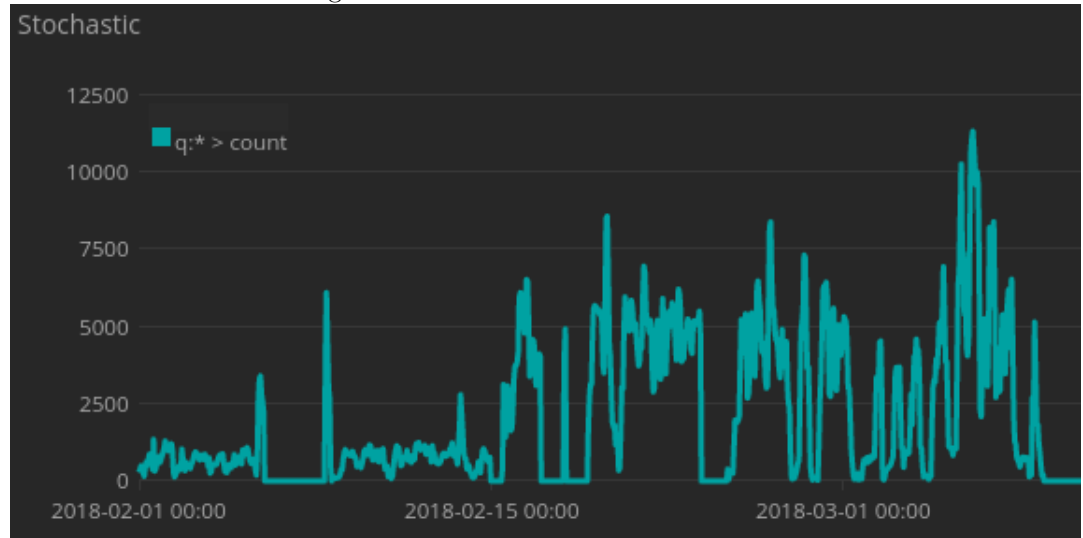
Figure 6: Global Attacks Map



3.1 Stochastic Process

Stochastic process is also called random process. It contains random variables with probabilities associated to it. While exact path cannot be inferred, assumptions can be made based on the probabilities of random variables. Attacks pattern observed is following random process or stochastic process as number of attacks on daily basis is not following any pattern.

Figure 7: Stochastic Process



4 Conclusion

We were able to design a monitoring system using dashboard from kibana by plotting the data from cowrie honeypot. This makes the security administrators work easy to handle as they will have complete control on the network.

References

- [1] Performance of ELK Stack and Commercial System in Security Log Analysis by *Sung Jun Son and Youngmi Kwon*
- [2] Building an IoT Data Hub with Elasticsearch, Logstash and Kibana by *Marcin Bajer*
- [3] Network security enhancement through effective log analysis using ELK by *Ibrahim Yahya Mohammed Al-Mahbashi and M. B. Potdar and Prashant Chauhan*

- [4] Malware capturing and detection in dionaea honeypot by *P Dilsheer Ali and T. Gireesh Kumar*
- [5] Integration of network intrusion detection systems and honeypot networks for cloud security by *Varan Mahajan and Sateesh K Peddoju*
- [6] Honeynet Data Analysis and Distributed SSH Brute-Force Attacks by *Gokul Kannan Sadasivam and Chittaranjan Hota and Anand Bhojan*
- [7] Detection of Severe SSH Attacks using Honeypot Servers and Machine Learning Techniques by *Gokul Kannan Sadasivam and Chittaranjan Hota and Anand Bhojan S*
- [8] Classification of SSH attacks using Machine learning algorithms by *Gokul Kannan Sadasivam and Chittaranjan Hota and and Anand Bhojan*
- [9] IoT honeypot: A multi-component solution for handling manual and Mirai-based attacks by *Sasa Mrdovic and Haris Šemić*
- [10] <https://www.virustotal.com>
- [11] Privacy and Security of Cloud-Based Internet of Things (IoT) by *Tanupriya Choudhury and Ayushi Gupta and Saurabh Pradhan and Praveen Kumar; Yogesh Singh Rathore*
- [12] Internet of Things (IoT) security platforms by *Ibrahim R. Waz and Mohamed Ali Sobh and Ayman M. Bahaa-Eldin*