Andy Lee  
500163559

CCPS 633

Contents

[1. From which countries do the attacks originate from (Top 25)? 3](#_Toc511435639)

[2. What ports are the most common to be probed or attacked (Top 25)? 4](#_Toc511435640)

[3. For each day of the week, what hour is the most active? 5](#_Toc511435641)

[4. What attacks were successful? 15](#_Toc511435642)

[Dionaea 15](#_Toc511435643)

[Cowrie 15](#_Toc511435644)

[ElasticHoney 16](#_Toc511435645)

[Glastopf 17](#_Toc511435646)

[Wordpot 17](#_Toc511435647)

[Shockpot 17](#_Toc511435648)

[5. What attacking IPs performed an Internet wide scan/attack? 18](#_Toc511435649)

[6. Identify three IP scan/attack pairs. A scan/attack pair is where one IP is used to scan the honeypots and a second IP or more are used to attack the honeypots. 19](#_Toc511435650)

[Set 1 19](#_Toc511435651)

[Set 2 20](#_Toc511435652)

[Set 3 20](#_Toc511435653)

[7. Do scans and attack pairings follow any trends over time? 21](#_Toc511435654)

[Option E – Attack Method Detection Part 1 22](#_Toc511435655)

[Search for strings containing the characters which are known to be used in a Shellshock exploit. 22](#_Toc511435656)

[Search for attempts where a website/webserver is being used that potentially hosts malware. 23](#_Toc511435657)

[Search for strings containing the characters which are known to be used in a directory traversal attack 27](#_Toc511435658)

[Search for cross site scripting (XSS) attacks. What types of threat actors are using the above attacks? Where are these attacks coming from? 32](#_Toc511435659)

[In your conclusions you need to write a report and answer the following questions: 35](#_Toc511435660)

[References 35](#_Toc511435661)

You must submit a report detailing your findings, observations, any screenshots and answers to questions. This report must not be zipped. All the programs, scripts, source codes, files created/required, together with a README file of detailed instructions on how your lab environment was setup and how to run your scripts/programs can be sent in a zipped file.

Please DO NOT use Pie Charts. Graphs are to be in the form of Line or Bar Charts only.

## From which countries do the attacks originate from (Top 25)?

1. What type of data do you include/exclude?
2. What data (if any) would throw your results off?
3. Support your decision(s).

Below listed the top 25 countries where attack originates. Data from AllTraffic.csv data were used. Compared JOIN with Tor CSV removing any matches on IP and Date. Removing all private IPs and fake IPs. The remainder dataset will be the attack where each row equals to a recorded attack. A count of values on the column country will return the number of attacks from each country.

There are certain cases where our results might be off. For example, the attacker might be attacking through a machine that has been compromised; or maybe the attack is carried through VPN, Proxies or Tor Networks will throw the results off.

China 310602

UnitedStates 208761

Vietnam 128860

Russia 91795

Taiwan 73040

Brazil 71806

Netherlands 56372

Ukraine 55692

RepublicofKorea 48895

Turkey 46790

India 35253

Germany 31660

RepublicofLithuania 26355

UnitedKingdom 26266

Romania 25064

France 22654

HongKong 22224

Mexico 22160

Argentina 19376

Colombia 17973

Poland 17956

Canada 17272

Indonesia 11559

Iran 11015

Italy 9852

## What ports are the most common to be probed or attacked (Top 25)?

1. What type of data do you include/exclude?
2. Support your decision(s).

Data from AllTraffic.csv data were used. Since these honeypots does not provide meaningful services, any attempts to communicate with them can only be seen as an attempt to scout and/or attack. The port number is extracted from the destination\_port column.

The 3rd largest attack is on port 2323. This attack should be meant for Telnet’s Port 23 scanning for IoT from Mirai botnet.[[1]](#footnote-1)

**Port Description Total attack**

23 Telnet 737873

22 SSH Remote Login Protocol 237174

2323 3D-NFSD 101705

5060 Session Initiation Protocol (SIP) 101617

3389 Microsoft Terminal Server (RDP) 59390

5900 Virtual Network Computing (VNC) 45547

80 HTTP 22581

7547 CPE WAN Management Protocol (CWMP) 21005

3306 MySQL 14931

5358 WSDAPI Secure Channel 13784

23231 UNKNOWN 10015

0 ICMP 8772

6789 Logger Net/Bucky's Inatand Msg 8378

81 Torpark Onion Routing 5178

8080 HTTP Alternate 3884

3390 UNKNOWN 3683

443 HTTPS 2985

222 UNKNOWN 2274

110 POP3 1675

5800 UNKNOWN 1605

5432 PostgreSQL 1590

8888 MAMP Server 1481

8000 Splunk 1450

9000 UNKNOWN 1392

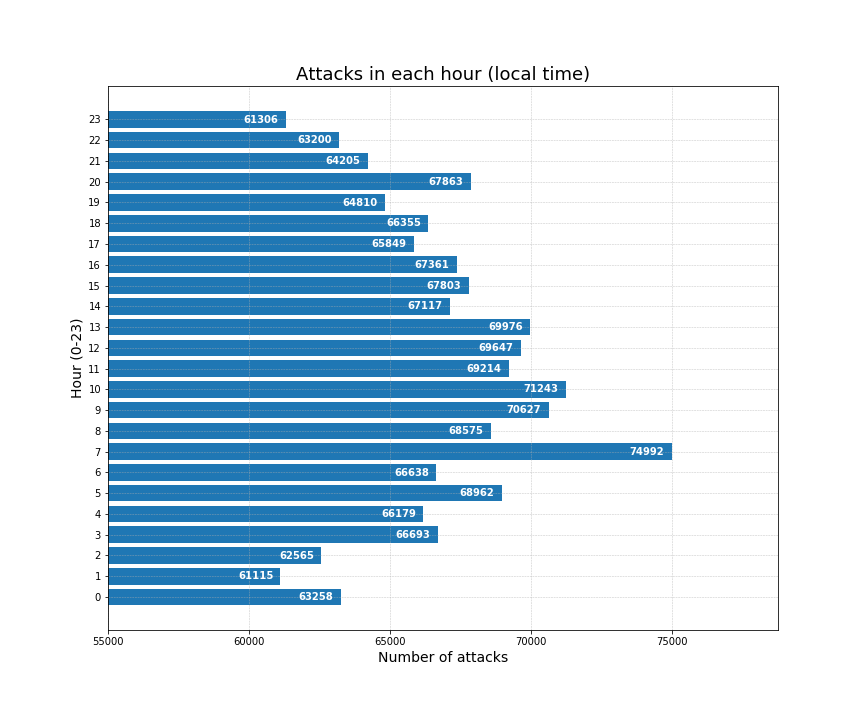
1723 Point-to-Point Tunneling Protocol (PPTP) 1354

## For each day of the week, what hour is the most active?

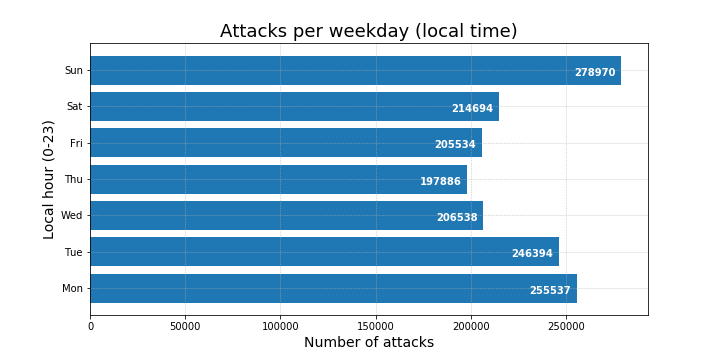
1. What reasons can you give for any difference between the days?
2. What type of data do you include/exclude? Explain.

|  |  |
| --- | --- |
| Monday | Peak attacks during 7am, with afternoons higher than overnight hours |
| Tuesday | Peak attack at 3am, with mornings more active than evenings |
| Wednesday | Peak attacks between 1-3am with relatively low traffic at other times |
| Thursday | 10pm has highest attacks, and evening attacks are higher than other times of the day |
| Friday | Active attacks spread through the entire day with spikes in random hours day and late night. No discernable pattern in terms of attacks throughout the day |
| Saturday | Midnight starts off with 8k with a steady gradual increase throughout the entire day eventually hitting 10k. Random spikes at 7am and 8pm. |
| Sunday | Highest activity between 4am to 8pm averaging close to 14k attacks per hour. After 8pm attack drops sharply into the midnight. |

All times have been localized[[2]](#footnote-2) therefore the hours share a common “day and night”, “weekdays and weekends”. Based on the graph "Attacks in each hour (local time)" which combines all attacks grouping by the hour, day time generally has more attacks than night time. With a out of pattern spike at 7am and 8pm.

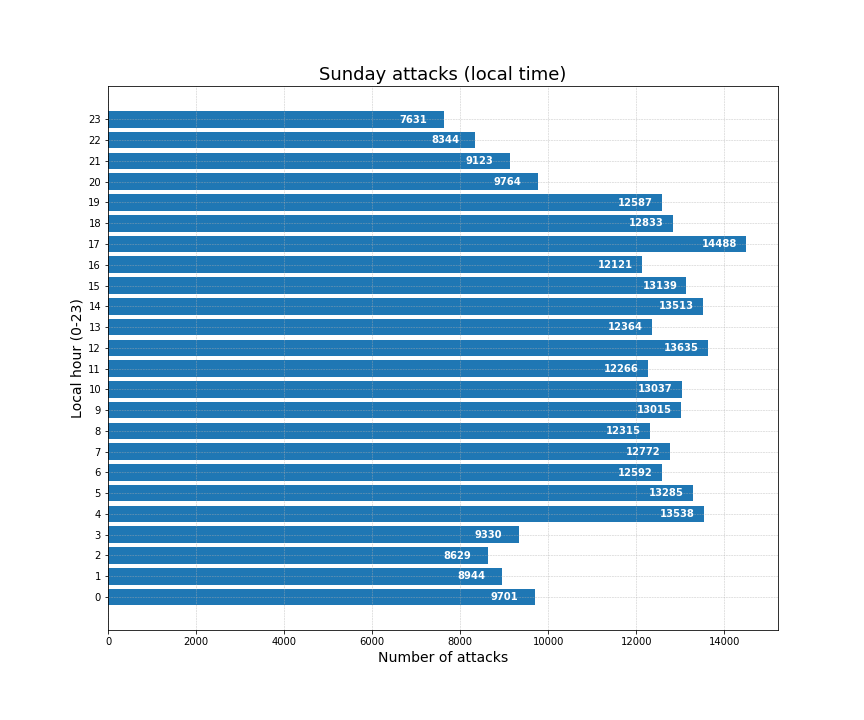
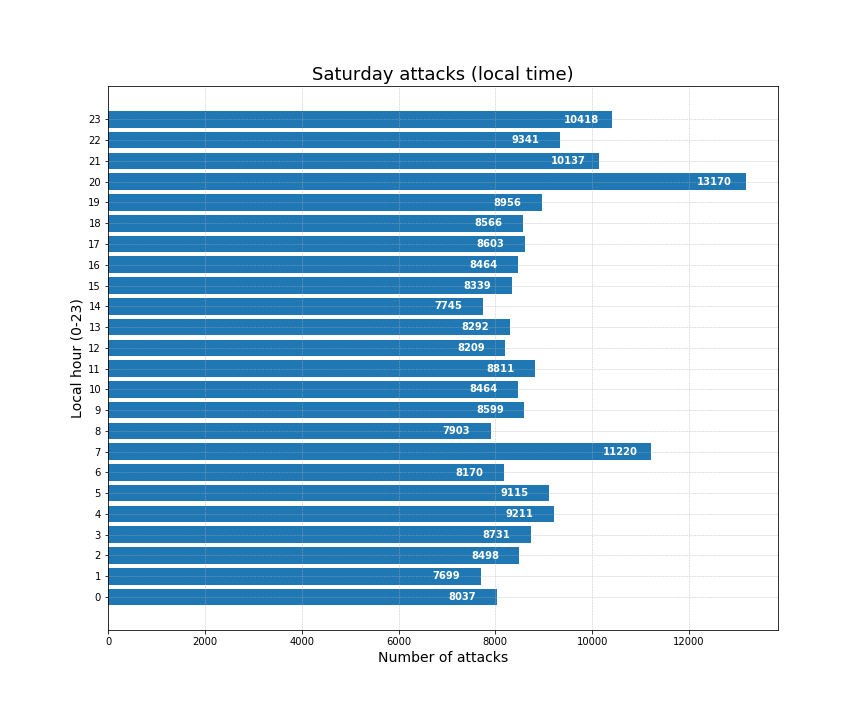
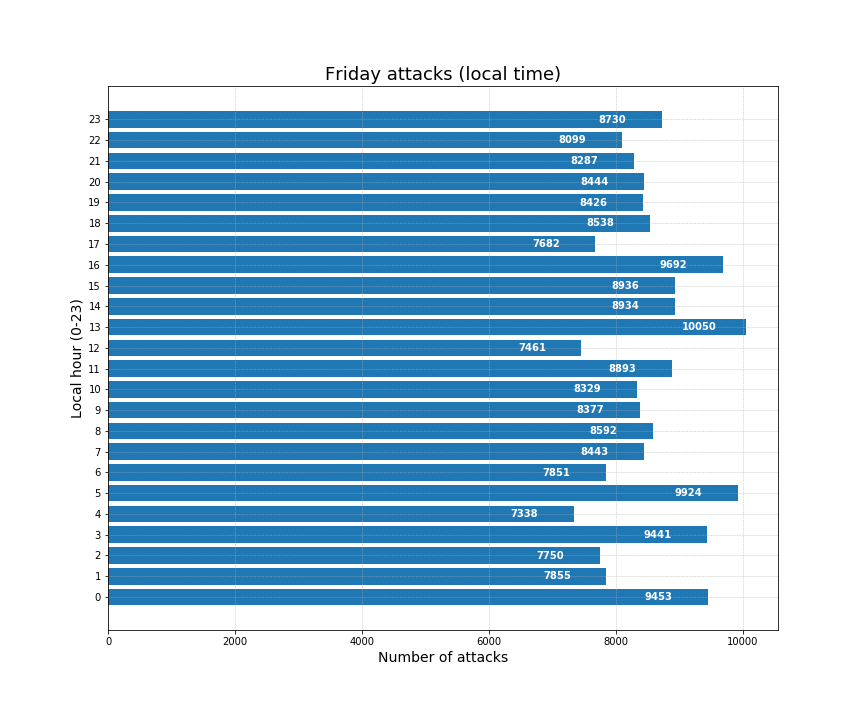
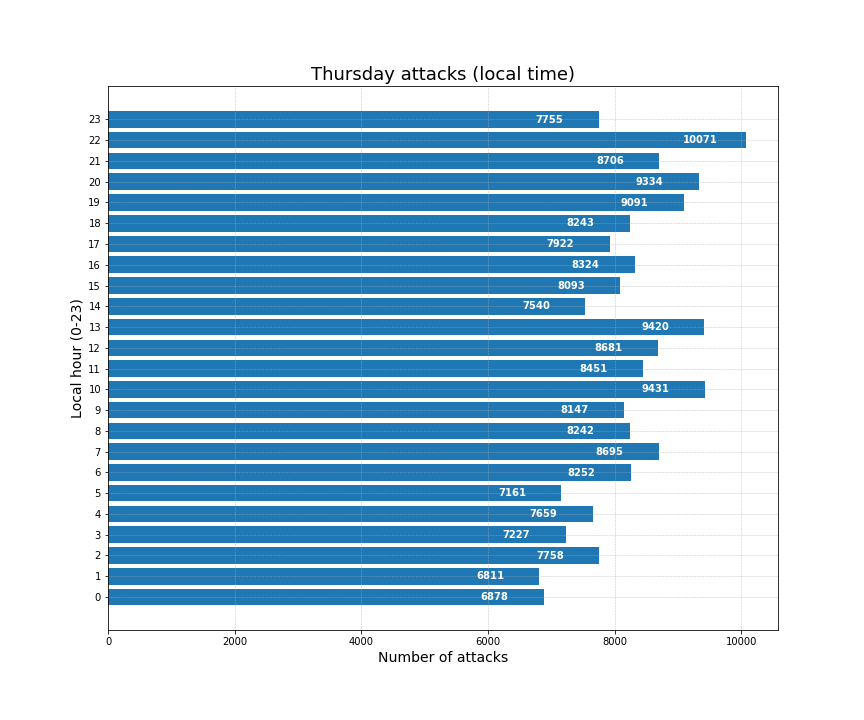
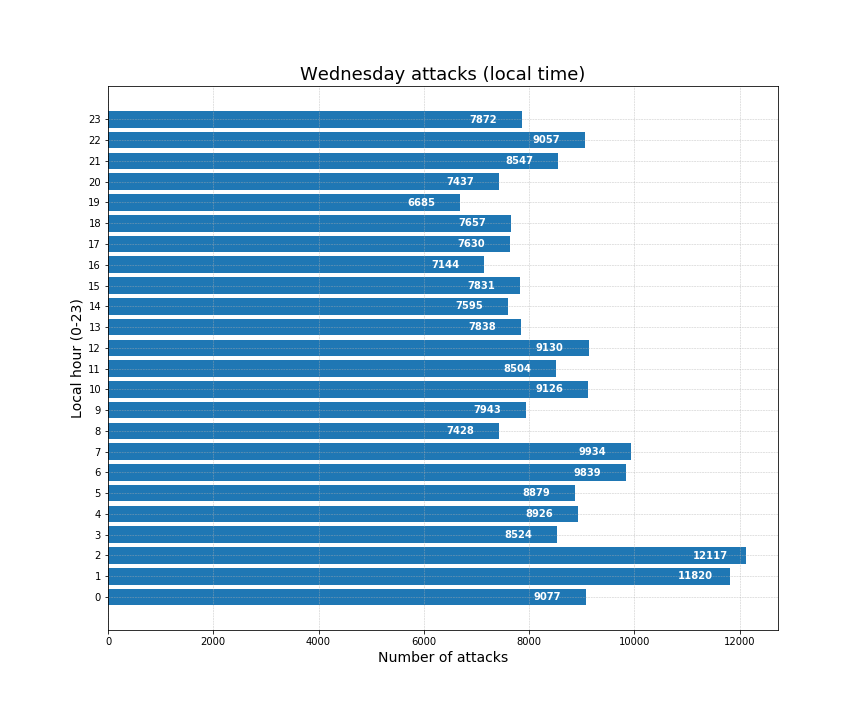
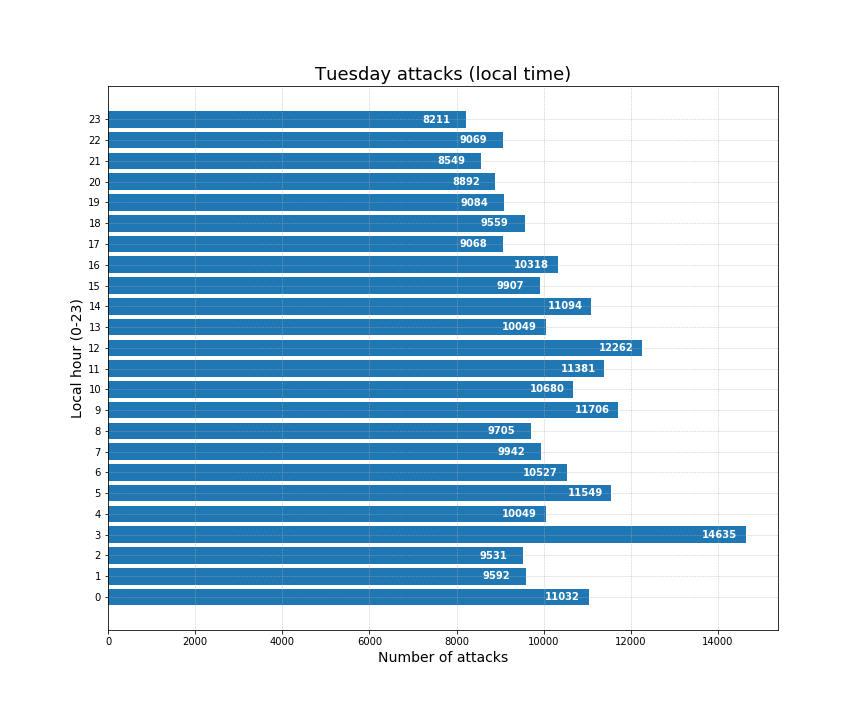
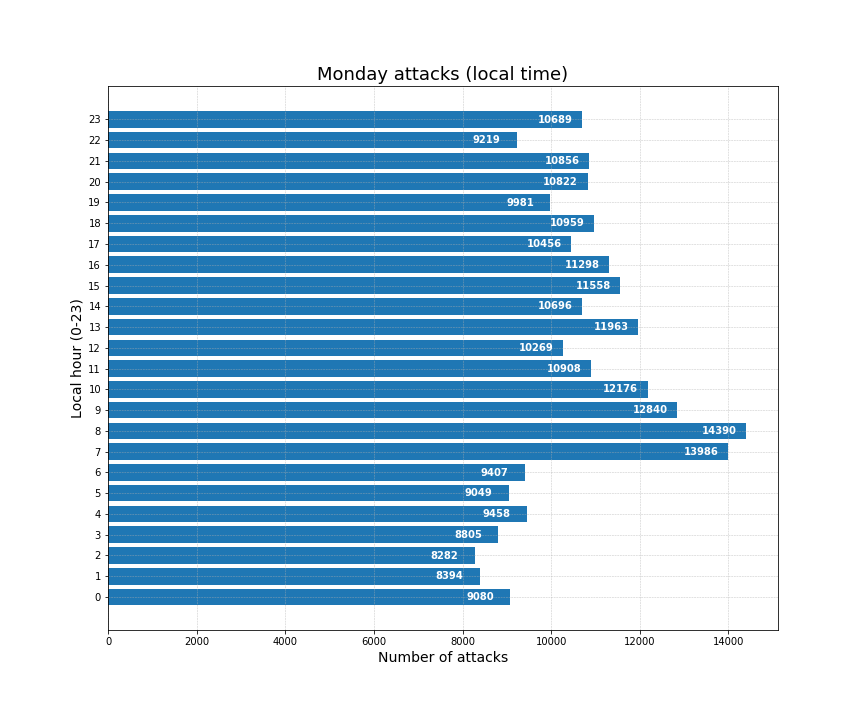


Based on the graph “Attacks per weekday (local time)” which combines all attacks grouping by the day



Sunday has the highest chance of cyberattacks, with Monday and Tuesday to follow. My theory is hackers have more free time to spend on Sundays since it is a holiday for most countries and many hackers will have regular jobs in addition to hacking. The cyberattacks spills over to the next two days but progressively lesser each day until Thursday then it bounces back up. The hackers may have had ideas that they developed on Sunday that they try out early into the week and may run out of ideas midweek, suggesting the lower traffic.

The AllTraffic CSV was used to compile this data. Data points with unresolved geographic location were omitted as it is not possible to determine their local time. However, that amount is small relative to the entire set of data. All data points is considered as regardless whether the attack was successful or not.



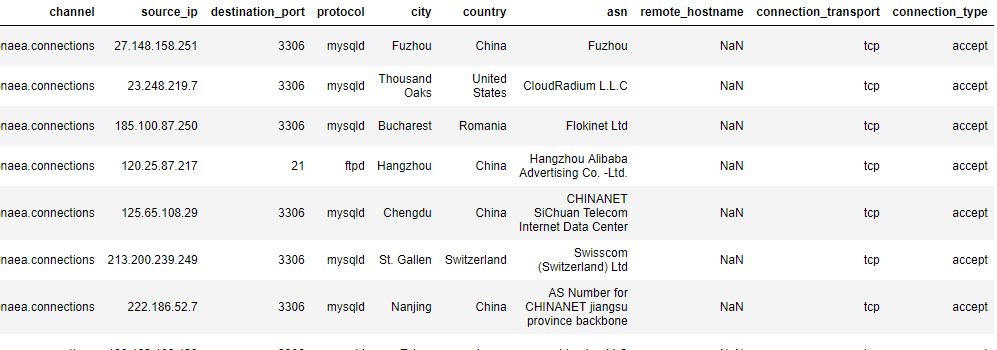
## What attacks were successful?

1. What determines a successful attack? Explain.
2. Support your decision(s).

A successful attack is when an attacker successfully compromises a system, and gain access beyond what a normal user can to do. Each Honeypot is set up for a different purpose. Some Honeypot has a sole purpose of capturing attacker’s requests which has no clear definition of whether successful or not. While some Honeypots has for attackers to break into which can be defined as successful.

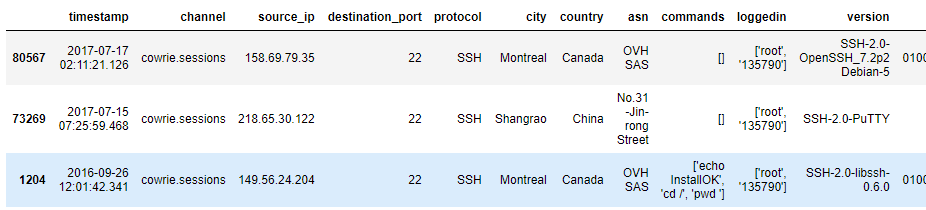
### Dionaea

Any attacks with a connection\_type as accept and repeated attacks of every second (e.g. 164.52.0.x) in attempts to conduct DOS should be considered as successful.



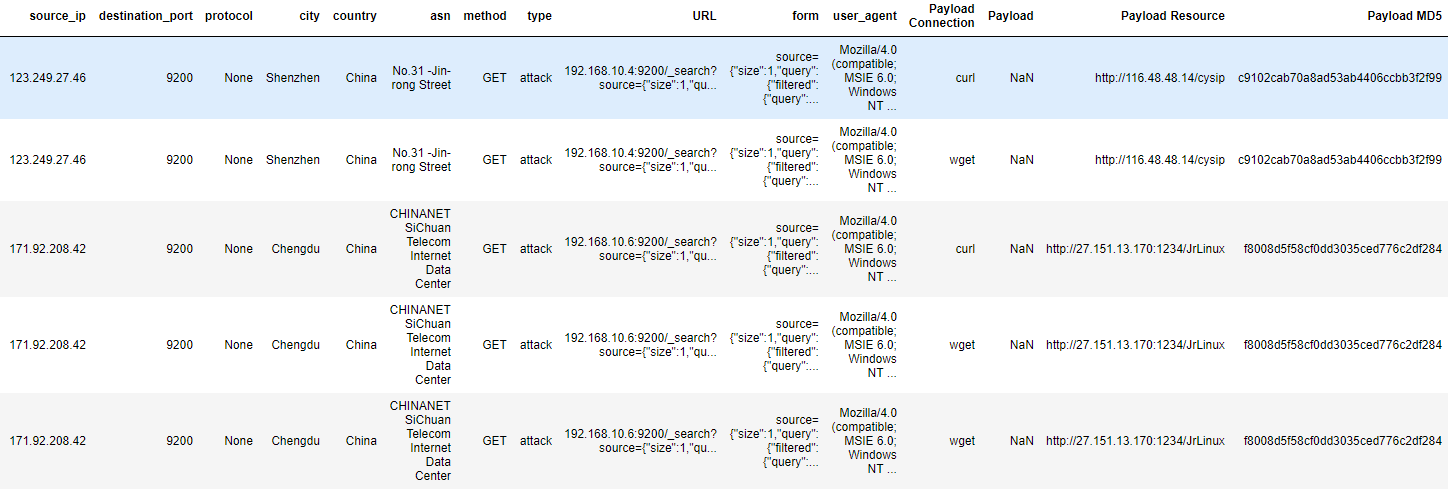
### Cowrie

Any connections logged in with credentials ['root', '135790'] should be considered as successful since it was able to successfully login.



### ElasticHoney

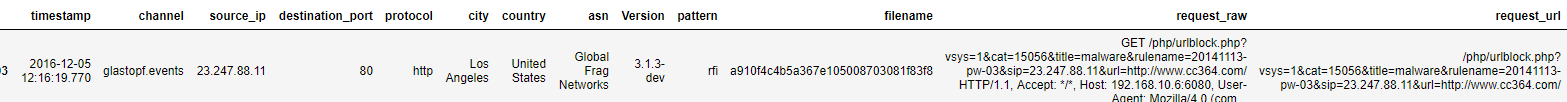
Payload Connection isn’t empty should be considered as attack successful as it was able to make the server transmit a payload that requires MD5 encryption and also provide Payload Resource.



### Glastopf

Request\_raw with keep-alive in its string seemed to suggest something the attacker does is allowing the connection to keep alive, which would otherwise have closed. 

Also there is one particular entry where filename has a value seems to be doing something other attackers aren’t. Therefore can be considered as successful.

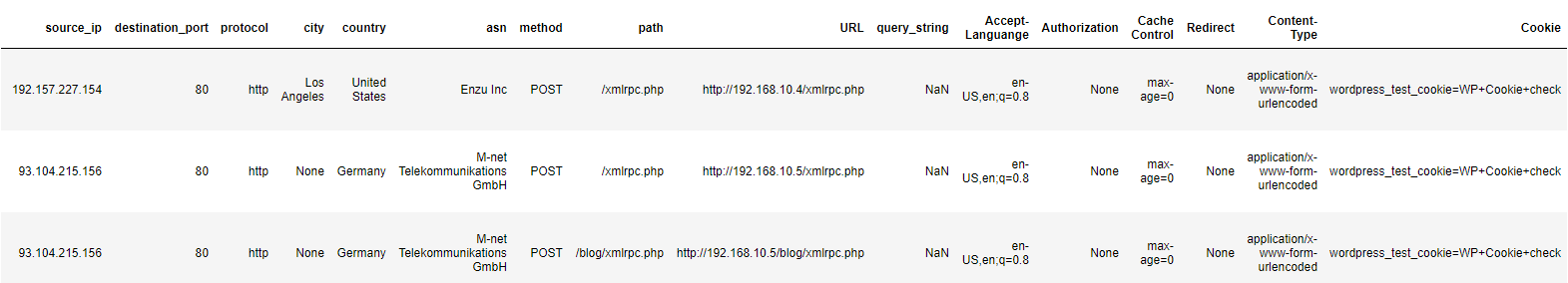


### Wordpot

No attacker seemed to able to fill filename with a value, therefore I suspect none of the attacks were successful in this Honeypot.

### Shockpot

This Honeypot has a column 'is-shellshock' returning only values 'False'. However, those who were able to keep a connection to keep-alive seems to have one foot through the door. Secondly those who were able to write a cookie value seems to be successful in their attack.



## What attacking IPs performed an Internet wide scan/attack?

1. What criteria did you use to identify these IPs? Explain.
2. What attack method(s) did these threat actors use? Explain.

These are the procedures to clean up the data

1. Convert snort entries back to its respective servers. Although snort is an IDS/IPS the attack did occur so we want those attacks be taken into account
2. Resampling all data based on frequency
3. Grouped all IP within a certain range
4. Performed a count on the number of entries, which will tally up the total of attacks performed by a given IP.

#### Sample of 2-day period

2017-02-19 85.17.29.79 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 8612 attacks

2017-02-21 85.17.29.79 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 3373 attacks

2017-05-24 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 2472 attacks

2017-06-01 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1932 attacks

2016-12-23 134.19.176.138 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1508 attacks

2017-05-20 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1359 attacks

2017-06-05 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1342 attacks

2017-06-25 162.209.168.14 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1246 attacks

2017-05-28 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1230 attacks

2017-05-22 23.248.219.51 [cowrie.sessions, dionaea.connections, glastopf.events, shockpot.events, wordpot.events] 1151 attacks

Here is a sample range of 2 days with attackers’ footprints over 5 datafiles. You will notice some recurring IPs but on a different date. All these Honeypots resides in different IPs and we assume the attacker has no inside knowledge that these Honeypots are related and hosted by the same person. Due to the sheer amount of attacks within a short timespan which is humanly impossible we can also assume the attacks were carried automatically through a batch script or bot.

IP 85.17.29.79 (#1) and 23.248.219.51 (#3) has clearly launched an internet attack. The other IPs there were in the top 10 list were 134.19.176.138 and 162.209.168.14

## Identify three IP scan/attack pairs. A scan/attack pair is where one IP is used to scan the honeypots and a second IP or more are used to attack the honeypots.

1. State how you determine the three scan/attack pairs?
2. Support your theory. *Note: The attacking IP may or may not be within the same IP subnet.*

### Set 1

#### Probe 121.18.238.114

2017-04-04 121.18.238.114 11 [cowrie.sessions, snort.cowrie.sessions, snort.shockpot.events]

2017-04-05 121.18.238.114 5 [cowrie.sessions, snort.shockpot.events, snort.wordpot.events]

2017-04-06 121.18.238.114 16 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

#### Probe by 121.18.238.122

2017-04-09 121.18.238.122 21 [cowrie.sessions, snort.cowrie.sessions]

2017-04-16 121.18.238.122 3 [cowrie.sessions]

2017-04-17 121.18.238.122 5 [cowrie.sessions, snort.shockpot.events]

2017-04-18 121.18.238.122 43 [cowrie.sessions, snort.shockpot.events, snort.wordpot.events]

2017-04-19 121.18.238.122 96 [cowrie.sessions, snort.cowrie.sessions, snort.shockpot.events]

2017-04-20 121.18.238.122 8 [cowrie.sessions, snort.glastopf.events]

2017-04-21 121.18.238.122 46 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.wordpot.events]

2017-04-22 121.18.238.122 10 [cowrie.sessions, snort.wordpot.events]

2017-04-23 121.18.238.122 28 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events]

2017-04-24 121.18.238.122 10 [cowrie.sessions, snort.shockpot.events, snort.wordpot.events]

2017-04-25 121.18.238.122 87 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.wordpot.events]

#### Probe/Attack by 121.18.238.(106,119,123,125)

On May till August .106, .119, .123, .125 were used to heavily attack into cowrie, shockpot, wordpot, glastopf on a daily basis.

2017-05-17 121.18.238.106 3 [cowrie.sessions, snort.wordpot.events]

2017-05-17 121.18.238.119 54 [cowrie.sessions, snort.cowrie.sessions, snort.shockpot.events, snort.wordpot.events]

2017-05-17 121.18.238.123 10 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events]

2017-05-17 121.18.238.125 12 [cowrie.sessions, snort.cowrie.sessions, snort.wordpot.events]

2017-05-18 121.18.238.106 13 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

2017-05-18 121.18.238.119 7 [cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

2017-05-18 121.18.238.123 12 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events]

2017-05-18 121.18.238.125 8 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

#### Example of attacks that continued onwards till August.

390 2017-08-07 121.18.238.106 153 [cowrie.sessions, snort.cowrie.sessions]

391 2017-08-07 121.18.238.119 199 [cowrie.sessions, snort.cowrie.sessions]

392 2017-08-07 121.18.238.123 222 [cowrie.sessions, snort.cowrie.sessions]

393 2017-08-07 121.18.238.125 162 [cowrie.sessions, snort.cowrie.sessions]

### Set 2

#### Probe by 164.52.0.130

On July 17 and 24, Aug 14 and 21, .130 and .135 was used to scout into cowrie, glastopf, dionaea, shockpot and wordpot

2017-07-17 164.52.0.130 4 [snort.cowrie.sessions, snort.glastopf.events]

2017-07-19 164.52.0.130 137 [dionaea.connections, snort.dionaea.connections, snort.shockpot.events, snort.wordpot.events]

#### July 24 attack by 164.52.0.x

On July 24 .132, .134, 136, 138, 139, 140 were used to attack glastopf, shockpot, wordpot and mainly dionaea.

2017-07-31 18:50:37.479 164.52.0.136 190 [dionaea.connections]

2017-07-31 18:50:37.479 164.52.0.137 198 [dionaea.connections]

2017-07-31 18:50:37.479 164.52.0.138 102 [dionaea.connections]

2017-07-31 18:50:37.479 164.52.0.139 111 [dionaea.connections]

2017-07-31 18:50:37.479 164.52.0.140 206 [dionaea.connections]

### Set 3

#### Probe by 221.194.44.224 in March and April

From March till April 2017, .224 conducted a series of scouting on all honeypots and stopped on Apr 25.

2017-03-31 221.194.44.224 57 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

2017-04-07 221.194.44.224 53 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events, snort.wordpot.events]

2017-04-21 221.194.44.224 178 [cowrie.sessions, snort.cowrie.sessions, snort.glastopf.events, snort.shockpot.events]

#### Attack by .212 from May till August

Attacks from .212 come in full force after the scout finished. The attacks on cowrie lasted for 3 months.

2017-06-16 221.194.44.212 189 [cowrie.sessions, snort.cowrie.sessions]

2017-06-23 221.194.44.212 609 [cowrie.sessions, snort.cowrie.sessions]

2017-06-30 221.194.44.212 114 [cowrie.sessions, snort.cowrie.sessions]

2017-07-07 221.194.44.212 49 [cowrie.sessions, snort.cowrie.sessions]

2017-07-14 221.194.44.212 1289 [cowrie.sessions, snort.cowrie.sessions]

2017-07-21 221.194.44.212 1345 [cowrie.sessions, snort.cowrie.sessions]

## Do scans and attack pairings follow any trends over time?

1. Are there trends across certain threat actors? Explain.
2. Do any threat actors have trends unique to them? Explain.
3. Are there any trends that imply co-ordination between threat actors? Explain.

There are slight differences in the scan and attack patterns between these attackers.

Based on my filtered method and results, all the attackers possess a range of IP residing in the same subset. The attacker picks a set of IPs to perform attacks sparsely for a period which can span over two months. Near the end of the cycle, once the IP(s) has gained enough exposure, the attacker will then use the IP to scout for other fresh targets that are vulnerable, then seizes all attacks and the IP goes on a hiatus. Another “fresh new set” of IPs will, probably out from hiatus will soon follow and carry on with the attacks on servers that were previously scouted. The whole cycle then repeats and recycle.

Dionaea and Cowrie appears to be particularly popular among attackers, constantly being hammered with brute force attacks. Some attackers will use a set of 4 IPs to rotate the attacks daily. While some attackers choose to use a single IP to attack Cowrie for 3 straight months almost everyday.

In my opinion, there is some sort of co-ordination as it is uncommon for an individual to posses so many IPs addresses. Furthermore, the asn record shows how some of the IPs come from network backbone companies. For example, 121.18.238.x ASN record shows that it is coming from CHINA UNICOM China169 Backbone in China. A quick search on Google reveals that this company has been previously accused of engaging in illegal activities. <https://thehackernews.com/2016/02/china-hacker-malware.html>

## Option E – Attack Method Detection Part 1

This option deals with creating a method to detect the following attack methods: Chart and Graph by date, the following based on source IP and country.

### Search for strings containing the characters which are known to be used in a Shellshock exploit.

According to blog post in cloudflae.com[[3]](#footnote-3) there are three parts of the request that can be susceptible to the Shellshock attack:

1. the request URL
2. the headers that are sent along with the URL
3. what are known as "arguments" (when you enter your name and address on a web site it will typically be sent as arguments in the request).

The code () { :;} is used to conduct the Shellshock exploit but no attackers used it to attack the Shellshock Honeypot after searching through the string through multiple log columns. The column is-shellshock further verifies this by having no True returns.

The following regular expression is used to go through columns on all CSV files to find the shellshock pattern. No matches were found in all files.

mylist = {'\:\;' ,

'/^(\S+) (\S+) (\S+) \[([^:]+):(\d+:\d+:\d+) ([^]]+)\] "(\S+)(?: ((?:[^"]|\\")\*) (\S+))?" (\S+) (\S+) "((?:[^"]|\\")\*)" "((?:[^"]|\\")\*)"$/',

'\(\)\s\*\t\*\{.\*;\s\*\}\s\*;'

}

### Search for attempts where a website/webserver is being used that potentially hosts malware.

wget and curl are common BASH commands used to remotely download scripts. We can fairly assume the destination address where the wget and curl commands are pointing to hosts malware. Usually attacker tries to trick victims into downloading files that are locally executable.

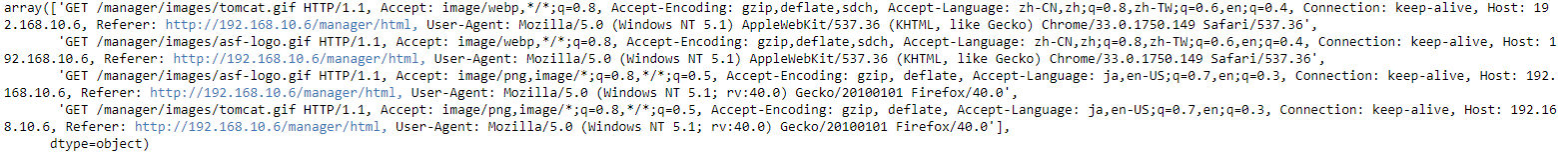
mylist = {'curl','wget','auto\_prepend','\.sh','fetch','dropbox','\.exe[^c]',

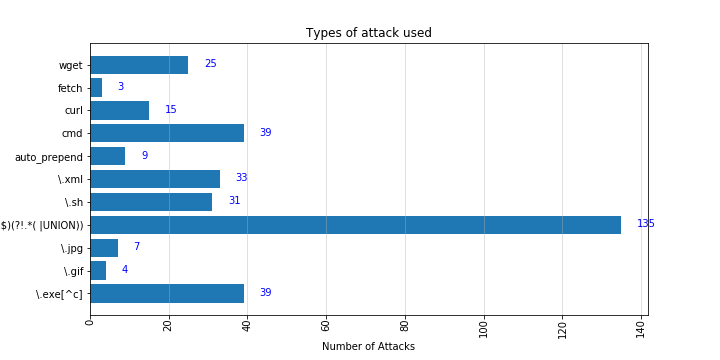
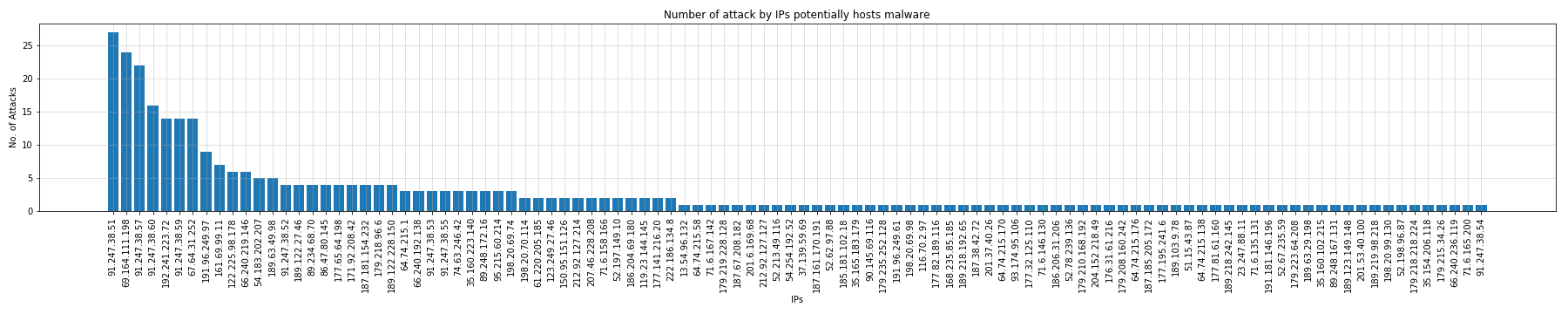
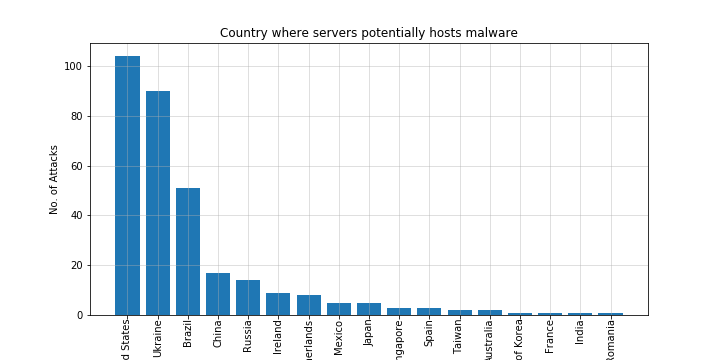
'cmd','cscript','mshta','rundll32','regsv','msbuild','regasm','\.hta','\.vbs','\.xml','\.bat','\.jpg','\.gif','\.png','\.bmp',

'\.php(?!$)(?!.\*( |UNION))',

'\.xlsx','\.doc','\.docx','\.ppt','\.pptx','\.pdf','\.msi','jquery','javascript','\.zip','\.rar','\.7z'}

I have included images extensions because there are reports that image files can contain malicious code[[4]](#footnote-4). Moreover, the request\_raw from the .gif results appears suspicious as the files are .gif yet the Accept: shows other image formats webp and png.



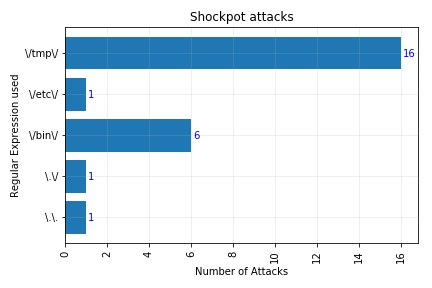
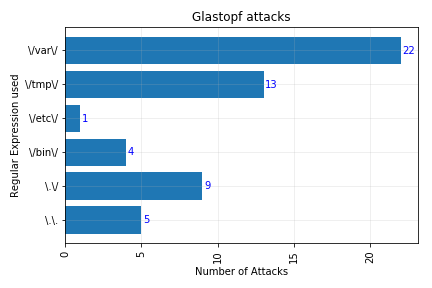
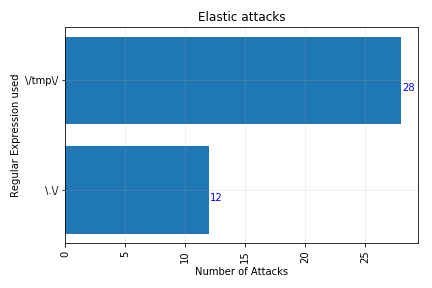
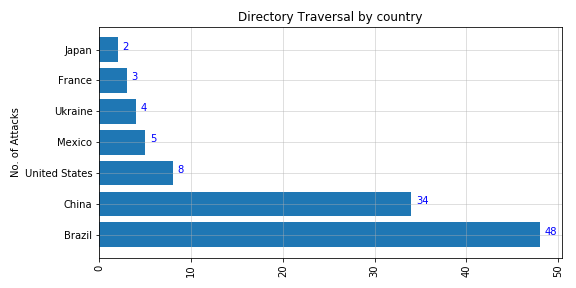


### Search for strings containing the characters which are known to be used in a directory traversal attack

Based on research directory transversal means accessing directories outside of allowed intended area. This attack will make attempts to access common Linux and Windows directories. Either through explicit directory name like /var or /etc or using ../ to access the parent folders.

The following is an example of the results showing both explicit directory traversal and parent directory traversal (with some attempts to execute malicious BASH code).





### Search for cross site scripting (XSS) attacks. What types of threat actors are using the above attacks? Where are these attacks coming from?

Most XSS attacks come from China and United States. Using all sorts of entry points, attackers attempt to try code which would redirect the Honeypot to another site to gain control.

Here’s a list of regular expressions I use to search the attacks

mylist = {'OgnlContext','\.js','\.sh', # potential malicious code that is ready for remote execution

'function \(', '\<script\>', # self executing js

'[Jj]\s\*[Aa]\s\*[Vv]\s\*[Aa]\s\*[Ss]\s\*[Cc]\s\*[Rr]\s\*[Ii]\s\*[Pp]\s\*[Tt]', # javascript

'http\:\/\/(?!192.168|.\*google.com)', # IP address that didnt start with 192.168.x.x

'\<.+\>', #.\

'cookie', # cookie

'&#[\w]{2,7}', # catching &#0000106&#0000097&#0000118&#0000097

'\d{4}\.\d{4}\.\d{4}\.', #Octal encoding

#'[0-9a-fA-F]x[0-9]{1,6}.(?!.\*(UNION))', #Hex encoding

'&[\w]{2,4};', # HTML entities

'(%[0-9a-fA-F]{2}){4}', #URL encoding

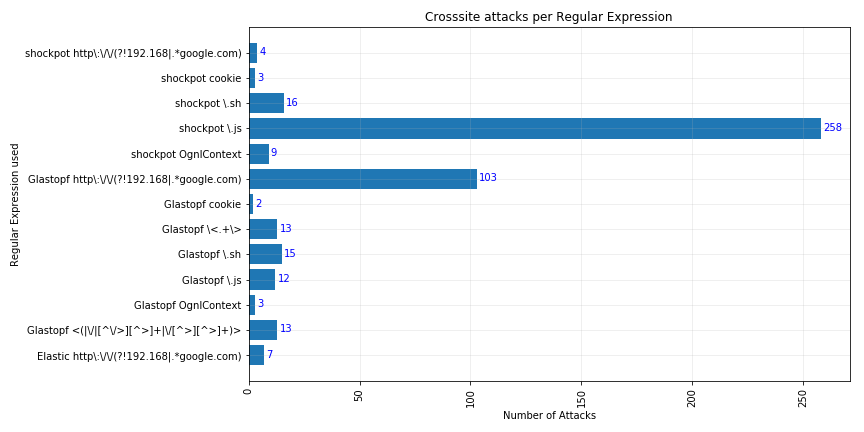
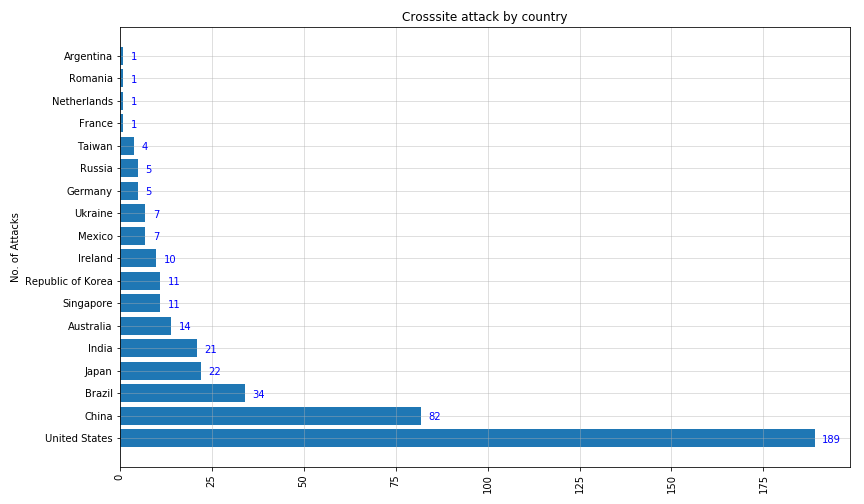
'\<\<','\>\>', #Extraneous open brackets

'<(|\/|[^\/>][^>]+|\/[^>][^>]+)>' # regular HTML tags

}

An example of the result output





### In your conclusions you need to write a report and answer the following questions:

1. From your analysis and the discovered method of attacks, what defence mechanisms would be appropriate?
2. Is the intelligence you gathered useful? Explain why or why not.
3. What intelligence would be useful? Explain.

Based on the discovered methods of attacks, the methods used by attackers are both ingenious and terrifying as they appear to be relentless and patient.

To properly defend against attacks, proper measurements are needed in multiple fronts. Programmers will need to be sure untrusted data not to be dynamically inserted into live code especially in SCRIPT areas and inside HTML tags, which are common areas where attackers insert malicious code to gain further access. All data that can be entered by a user must be validated and escaped before processing, as malicious code can be entered and processed along. There are some libraries designed to help programmers sanitize their code before processing which is highly recommended.

Ever since the dawn of Internet, scripting engines has also been very lenient on programmers. Loose practices allowed web programmers to make many common mistakes while allowing their code to function. This has also inadvertently made it easier for attackers to conduct attacks and harder for everyone else to catch them since loose programming practices create flaws that are unintentional and hard to detect.

Using Honeypots, we can gather intelligence that are useful in many ways. It allows us to know what has already been tried by attackers e.g. what usernames and passwords not to use. It lets us know what are the current trends and attackers attacking patterns so we can effectively divert extra resources to tackle the threat.

However, there are still limitations. Since we are just gathering information from known entry points, we need to constantly evaluate other potential entry points where the logs are not monitoring.

# References

*How a malware can download a remote payload and execute malicious code…in one line?* (2017). Retrieved from https://andreafortuna.org/cybersecurity/how-a-malware-can-download-a-remote-payload-and-execute-malicious-code-in-one-line/.

*How to display the value of the bar on each bar with pyplot.barh()?* (n.d.). Retrieved from https://stackoverflow.com/questions/30228069/how-to-display-the-value-of-the-bar-on-each-bar-with-pyplot-barh/30229062.

*How to display the value of the bar on each bar with pyplot.barh()?* (n.d.). Retrieved from https://stackoverflow.com/questions/30228069/how-to-display-the-value-of-the-bar-on-each-bar-with-pyplot-barh.

*Inside Shellshock: How hackers are using it to exploit systems*. (2014). Retrieved from blog.cloudflare.com/inside-shellshock.

*Is there a short command to test if my server is secure against the shellshock bash bug?* (n.d.). Retrieved from https://security.stackexchange.com/questions/68168/is-there-a-short-command-to-test-if-my-server-is-secure-against-the-shellshock-b.

*Pandas convert datetime with a separate time zone column*. (n.d.). Retrieved from https://stackoverflow.com/questions/39646898/pandas-convert-datetime-with-a-separate-time-zone-column.

*pandas.Series.tz\_localize*. (n.d.). Retrieved from https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.tz\_localize.html.

*XSS Prevention Cheat Sheet*. (n.d.). Retrieved from https://www.owasp.org/index.php/XSS\_(Cross\_Site\_Scripting)\_Prevention\_Cheat\_Sheet.

1. What is happening on 2323/TCP? - https://isc.sans.edu/forums/diary/What+is+happening+on+2323TCP/21563/ [↑](#footnote-ref-1)
2. Timezonefinder library - https://pypi.python.org/pypi/timezonefinder/ [↑](#footnote-ref-2)
3. https://blog.cloudflare.com/inside-shellshock/ [↑](#footnote-ref-3)
4. 'Hacking with Pictures': Stegosploit and How to Stop It - https://www.opswat.com/blog/hacking-pictures-stegosploit-and-how-stop-it [↑](#footnote-ref-4)