CKDF150 - Final Report - Natalia Shevcun.docx

by Natalia Shevcun

Submission date: 12-Apr-2019 11:59PM (UTC-0400)
Submission ID: 1111567953
File name: CKDF150_-_Final_Report_-_Natalia_Shevcun.docx (592.9K)
Word count: 16792
Character count: 86859

G. Raymond Chang School of Continuing Education
Computer Security and Digital Forensics • CKDF 150

Honeypot Capstone Project
Topic #4-O - Threat Actor Analysis IV

March 18, 2019

Studying Honeypot Data – Threat Actor Analysis

Natalia Shevcun
Ryerson University • G. Raymond Chang School

*This course is part of Ryerson University-Chang School **Computer Security and Digital Forensics Certificate Program.**
For more information about this program, please visit https://ce-online.ryerson.ca/ce/default.aspx?id=3357

TABLE OF CONTENTS

INTRODUCTION

Eyes are the windows to one's soul, according to the timeless proverb. However, oftentimes in the modern world, one's electronic device may provide such a window. The global digitization of information has become the hallmark of the twenty-first century; and as people's dependence on technology for the management of their daily activities increases, so does the need for adequate protection of their digital data vaults from prying eyes.

Much like living beings, electronic devices require energy and 'good health' in order to properly function. However, a computer's health is not guaranteed simply by the state of its physical components. The gadget's 'mind', which is its software, is prone to a plenitude of virtual vulnerabilities and ailments. Miscreants of all calibers, eager to sneak a peek into the souls and secrets of others, have created a myriad of cyber-parasites to exploit these vulnerabilities and get their hands onto data they were never meant to possess. Thus, just like the realm of the living, the digital world relies on teams of doctors and police to keep it safe; as well as scientists, whose research aids in treatment and defence of the modern human's new best friends.

Studying the tactics, strategies and techniques employed by malevolent computer users is essential in learning how to protect against them. However, analyzing millions of logs generated on a daily basis by busy servers of important institutions or large corporations is an arduous task. Distinguishing malicious interactions from legitimate ones in such cases is even more tricky. Thus, in order to successfully observe the behaviour of attackers with as few distractions as possible, the best solution is to lure the predators into a place free of all other cyberactivity. Since black-hatters are drawn to software vulnerabilities like bees to honey, deliberately exposing these vulnerabilities can serve as perfect bait. Hence, 'honeypots' is a suitable name for such decoy hosts.

HONEYPOTS

Honeypots are systems created for the sole task of observing black-hat hackers in their natural habitat and tracking their actions for the purpose of research. Thus, no other productive activity normally happens on these hosts. Honeypots can be placed anywhere on the network, including outside of the firewall and the DMZ (demilitarized zone); inside the DMZ, and on an internal network. In fact, they may also be set up on virtual machines as sandbox environments and run services that only emulate real ones. [1]

Unfortunately, due to honeypot source codes being freely available via a plethora of online resources, attackers have devised ways of recognizing fake systems for what they are. Some honeypots, such as the ones running on virtual machines with limited space or an especially restricted system, are more obvious than others as being simple bait. Low-interaction honeypots

like *Glastopf*, which are the systems that expose certain ports and lie in wait of a malicious entity's interaction with them, are also among those considered to be rather detectable. [1]

Medium-interaction honeypots such as *Cowrie* have fewer restrictions in the system for the illegitimate user, although the services run on them are still emulated rather than real. They may therefore produce responses to certain actions or commands that a real program would not generate, thus alerting an intruder that he or she is navigating a phoney system. [1]

Finally, high-interaction honeypots permit access to real systems and services, and feel the most convincing as ordinary fully-functional environments to attackers. However, with honeypots such as these, there is always a risk that an intruder may find a way of escalating privileges and access other, legitimate systems that are part of an essential working network, damaging them in the process. [1]

As it follows, there is a variety of honeypot systems available for every purpose and use on any scale, ranging from large government-managed institutions to computer networks set up by ordinary users in the privacy of their homes. It is thus up to the research team to weigh the risks and benefits of using one honeypot type over another, and pick the optimal option for their particular purposes.

INFORMATION AND INTELLIGENCE

Once the data about malicious interactions with the honeypots is collected, the next task is to extract certain information from this data. This information by itself, however, is simply an assortment of facts about the events that took place. In order to become useful, these facts will then have to be interpreted and turned into intelligence via analysis of information for various trends. These trends could include the attackers' preferred operational methods and tactics, as well as their goals, origins, or any other patterns that may in turn 'fingerprint' the intruders and aid in building effective defences against them.

PURPOSE

This project makes use of real-world data that has been collected by six honeypots placed on five different hosts scattered at participating locations in five different countries. The aim of this project is to study how specific data is extracted from a large honeypot raw log file and processed from information into strategic, operational and tactical intelligence that could be used for further trend analysis and cybersecurity architecture development.

PROJECT SPECIFICATIONS

Listed in the table below are the honeypots used for this project, with their corresponding IP addresses. The honeypots are **Dionaea, Cowrie, Elastichoney, Glastopf, Shockpot** and **Wordpot**. The systems are additionally monitored by the Intrusion Detection System **Snort**, which has also generated logs for the studied log file. The honeypots ran over a period from **September 14, 2016** to **September 30, 2017** and collected a total of **1,839,858** logs.

Table A. Honeypots and Their System IPs.

IP	Honeypot Name
192.168.10.2	Dionaea + Cowrie
192.168.10.3	Wordpot
192.168.10.4	Elastichoney + Shockpot Sinkhole
192.168.10.5	Cowrie + Shockpot Sinkhole
192.168.10.6	Elastichoney + Glastopf

The main log file that will be examined in this paper is titled **sorted-AllTraffic.csv**. It is **555,634 KB** large, and encompasses a total of **1,839,858** entries. Log files for separate honeypots are included as well. The table below lists the specifics for each.

Table B. Additional Files, Their Sizes and Amount of Logged Entries

File Name	File Size	Number of Logged Entries
sorted-cowrie.csv	126,981 KB	234,744
sorted-dionaea.csv	456,903 KB	1,445,628
sorted-elastichoney.csv	60 KB	64
sorted-glastopf.csv	8,429 KB	9,526
sorted-shockpot.csv	3,232 KB	6,181
sorted-snort.csv	58,231 KB	141,686
sorted-wordpot.csv	785 KB	2,029

The research package also contains **commonports.csv** (a dictionary of common ports used by system services) and **Tor_IPs.csv**, a file listing all the IPs that were known to be TOR exit nodes at certain dates falling within the observed time period, along with these dates. **Tor_IPs.csv** is **19,336 KB** large and contains a total of **547,985** entries, an IP with its corresponding date counting as one entry.

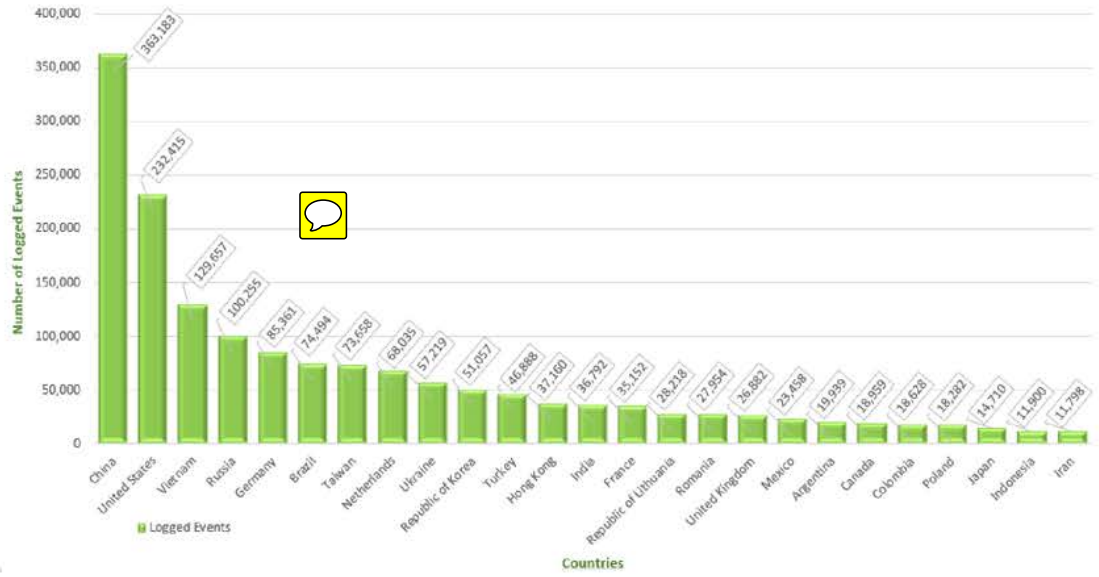
The scope of the project is to answer a series of eleven questions and perform a number of tasks specified in those questions, basing the answers on the files mentioned above. These files were examined with codes created using Python 3 and Anaconda Spyder as the scripting platform. The scripts used to extract data for every question are available as separate files. The breakdowns of mined information in the format of Excel spreadsheets are included as well. Each question is discussed in its own section subsequent to this paragraph.

SECTION 1. TOP 25 ATTACKING COUNTRIES

For the purpose of this study, the honeypots have been configured to record the geolocation of an attack's origins. Thus, the scope of the first section is to determine the top 25 attacking countries by the number of events originating from each country, as recorded by the logging system.

There were two codes written to accomplish this task. The language used was Python 3, and Anaconda's Spyder was the compiler utilized to test and run the scripts. The file '**sorted-AllTraffic.csv**' was analyzed with each of the two scripts consecutively. One of the codes determined the set of all countries encountered in the file, including the entries listed as empty spaces (or 'None') or "not available" ("n/a"). The second code asked the user to manually input ten nation names, and counted all the entries that displayed the corresponding country as the source of the attack. The results for the top 25 attacking countries are presented in the chart below.

FIGURE 1.1: TOP 25 COUNTRIES BASED ON TOTAL NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)



The table in **Figure 1.2** includes some additional information about the countries. The percentage that each country's attacks make up of the total number of events logged over the specified time period is shown in the column to the left of the rank.

As can be observed from the table and the chart, China is the country with the top number of originating events – its entries make up almost 20 percent, or one fifth, of the total records, which include 1,839,858 logs for 206 different countries (excluding the entries that returned 'None' or 'n/a' for the country). United States follows China closely with about 12.6% of total logged events. Vietnam, Russia and Germany are the other three nations that top the list, coming in third, fourth and fifth respectively. Canada ranks twentieth (20) on this list, with only about 1.03% of the overall logged events.

The interesting detail is that the top 5 attacking countries constitute **48.91%** (almost half) of all the recorded events, and the top 25 countries make up **87.01%** of the log, the other 181 countries composing a negligible 13% of all attacks in total.

Figure 1.2: TABLE OF THE TOP 25 ATTACKING COUNTRIES

Country	# of Events Logged	% of Total Events Logged	Rank
China	363,183	19.14%	1
United States	232,415	12.63%	2
Vietnam	129,657	7.05%	3
Russia	100,255	5.45%	4
Germany	85,361	4.64%	5
Brazil	74,494	4.05%	6
Taiwan	73,658	4.00%	7
Netherlands	68,035	3.70%	8
Ukraine	57,219	3.11%	9
Republic of Korea	51,057	2.78%	10
Turkey	46,888	2.55%	11
Hong Kong	37,160	2.02%	12
India	36,792	2.00%	13
France	35,152	1.91%	14
Republic of Lithuania	28,218	1.53%	15
Romania	27,954	1.52%	16
United Kingdom	26,882	1.46%	17
Mexico	23,458	1.27%	18
Argentina	19,939	1.08%	19
Canada	18,959	1.03%	20
Colombia	18,628	1.01%	21
Poland	18,282	0.99%	22
Japan	14,710	0.80%	23
Indonesia	11,900	0.65%	24
Iran	11,798	0.64%	25

While more information and context are needed to achieve results that would be truly accurate and reflective of the real situation, this information may be used to draw some general conclusions about certain trends in the cyberspace.

From the data gathered, it appears that people living in various countries of Europe, Asia and the American continent have the most opportunities, skillsets and technology in order to perform malicious cyber interactions, while Africa and the Australian continent seem to be falling behind in this race.

It also looks like the top five nations taking the most interest in the internal digital workings of a Canadian institution such as Ryerson University include the current global superpowers or aspiring superpowers (USA, China, Russia and Germany). It is unclear from this information alone whether these actions were attempts at intelligence gathering by the governments or the work of private citizens and organizations.

Whatever the case, cybersecurity teams should keep a close eye on the technologies and software innovations developed in these countries. They should also pay close attention to the operational systems and platforms preferred in these countries, as well as taking note which vulnerabilities they prefer to exploit – this may help cybersecurity specialists conduct focused research to single out and fix the weak links in the existing defence technologies and strategies.

Please refer to ***Question_01_Code.py*** for scripts that were used to extract the information necessary for this section. This file may be found in the zipped ***Supplementary Materials*** folder, inside the folder titled ***Scripts***, along with the files containing the codes for all of the sections below.

SECTION 2. ATTACKING COUNTRIES AND THE TOR NETWORK

While the previous section examined information that could provide a general idea about the origins of most threats to Ryerson's cybersecurity, there is a number of factors that may have skewed the acquired results. One of such factors is the TOR network.

TOR stands for 'The Onion Router', which is the name of an open-source software project utilized for anonymous online communication. [2] It is used to protect a user's privacy and identity, or bypass censorship in cases where restrictive governments may block access to certain information or online resources (however, it should be noted that TOR is illegal in some countries with dictator regimes, such as China). [2]

TOR works by routing the online traffic of a user through a path of three proxies. [2] After obtaining a list of TOR nodes from a special directory server, the TOR client guides the traffic into an entry node from this list. Then, packets flow through a middle relay; and finally, from the exit node, the user is able to access the coveted destination server. [2] All three nodes are randomly-picked, with the entry and exit nodes being chosen out of those available in the list.

Over time, however, some of the IP addresses listed as TOR entry or exit nodes may cease to serve this purpose. Or to the contrary, new ones may be added to the network. Thus, it is not enough to only look at an IP address when trying to distinguish a regular user from the one who had accessed a server through TOR – the date on which the activity in question took place is of utmost importance in this case, as it helps reveal whether a particular IP had been used as a TOR exit node at the time of the interaction or not.

The scope of this section is to examine the ***AllTraffic*** file with the aid of the information provided in the spreadsheet ***Tor_IPs.csv***, and determine which of the recorded attacking countries may have hosted IP addresses that were used as TOR exit nodes at the time of attacks.

To extract the needed information, Python's Pandas library was used to analyze the ***Tor_IPs.csv*** file and create a list of **547,985** unique entries where the IP address in each row would be concatenated with its corresponding date.

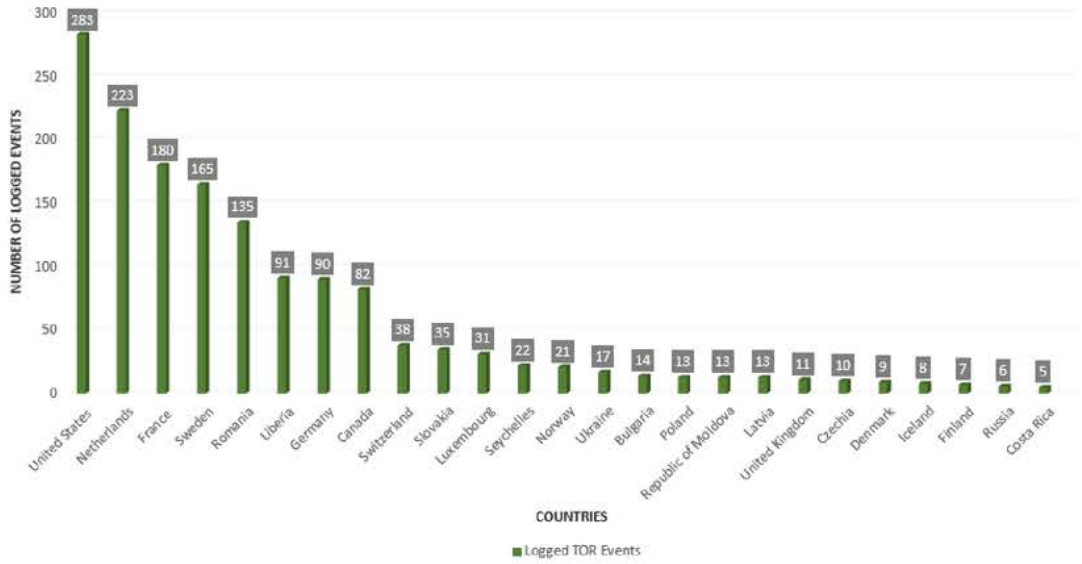
The code then added a new column of Boolean values titled '*Tor_Flag*' to the ***AllTraffic*** file and checked whether the source IP and the attack date of each entry in the ***AllTraffic*** file matched any items in the aforementioned list. If any such items were detected among the ***AllTraffic*** logs, the '*Tor_Flag*' was marked as True for this item; otherwise, it was noted as False.

Then, each nation from the set of attacking countries established in the previous section was checked for entries with a True '*Tor_Flag*', and the total number of entries matching these parameters for each country was stated at the end of each script execution.

The resulting top 25 TOR countries are listed in the chart below, with their corresponding number of logged TOR events. The full script used to mine this information may be found in the following file: ***Question_02_Code.py***.

The chart in **Figure 2.2** contains further details which may help flesh out the TOR user picture a little bit more.

FIGURE 2.1: TOP 25 COUNTRIES BY NUMBER OF LOGGED EVENTS ORIGINATING FROM TOR IPS
(September 14, 2016 - September 30, 2017)



The table in **Figure 2.2** contains some additional information about the discovered TOR countries. Their non-TOR rank is listed alongside their TOR rank, for the purpose of comparing the findings from the first section with the results of the second study. The percentage that TOR entries make up of the total logged entries for each country has also been calculated. This information may add some extra brushstrokes to the overall picture as well – for example, it is interesting to note that 100% of the entries originating from Liberia (ranked 6th in the TOR table and 133rd in the overall list) were, in fact, generated by the TOR traffic. This was also the case for almost 30% of the logs generated by Luxembourg (ranked 11th in the TOR table, and 127th in the overall list). This looks unusual compared to the rest of the numbers, where most other TOR entries make up anywhere from 0.01% to 1.7% of their respective country's traffic, with Slovakia being the only slight outlier, with 6.19% of its traffic being TOR-originated.

FIGURE 2.2: TABLE OF THE TOP 25 TOR COUNTRIES, BY NUMBER OF LOGGED EVENTS

Country	# of Times Flagged as TOR	% of Country's Total Logged Events	TOR Rank	Non-TOR Rank
United States	283	0.12%	1	2
Netherlands	223	0.33%	2	8
France	180	0.51%	3	14
Sweden	165	2.56%	4	34
Romania	135	0.48%	5	16
Liberia	91	100%	6	133

Germany	90	0.11%	7	5
Canada	82	0.43%	8	20
Switzerland	38	1.26%	9	50
Slovakia	35	6.19%	10	95
Luxembourg	31	29.52%	11	127
Seychelles	22	0.20%	12	27
Norway	21	1.70%	13	75
Ukraine	17	0.03%	14	9
Bulgaria	14	0.19%	15	31
Poland	13	0.07%	16	22
Moldova	13	0.42%	17	49
Latvia	13	0.63%	18	59
United Kingdom	11	0.04%	19	17
Czechia	10	0.24%	20	42
Denmark	9	0.25%	21	45
Iceland	8	0.88%	22	85
Finland	7	0.74%	23	82
Russia	6	0.01%	24	4
Costa Rica	5	0.69%	25	39

It is also worth noting that the top five countries from **Figure 2.2** (United States, Netherlands, France, Sweden and Romania) make up **63.98%** of all TOR traffic, and the top twenty-five (25) countries as per **Figure 2.2** make up **98.77%** of all TOR traffic. Out of the remaining 181 countries, only 10 more are registered as having hosted TOR exit addresses in addition to the top 25, and the amount of TOR entries for each of these ten countries ranges between 1 and 4 entries. Together, these ten countries make up about **1.23%** of all the logged TOR traffic. Not included in the list below is "None", which returned 2 entries, and was listed as 30th among TOR countries.

FIGURE 2.3: TABLE OF THE BOTTOM 10 TOR COUNTRIES, BY NUMBER OF LOGGED EVENTS

Country	# of Times Flagged as TOR	% of Country's Total Logged Events	TOR Rank	Non-TOR Rank
St Kitts and Nevis	4	7.69%	26	147
Austria	3	1.12%	27	110
Italy	2	0.02%	28	28
Hungary	2	0.07%	29	53
Taiwan	1	0.001%	31	7
Turkey	1	0.002%	32	11
Indonesia	1	0.008%	33	24
Panama	1	0.04%	34	54
Ireland	1	0.05%	35	63
Croatia	1	0.06%	36	68

Another interesting observation to be made is that Asian countries become entirely absent from the top 25 list when only TOR entries are counted. Only three of them are present at all in the entire TOR list – the odd TOR exit nodes in Taiwan, Turkey and Indonesia. That is to be expected, as some of these countries, such as China, have outlawed TOR entirely, and other countries, such as Russia, Iran and Saudi Arabia, are beginning to crack down on the use of the network. [2] Hence, Russia slid on the list to the 24th position from its overall 4th place, and Iran with Saudi Arabia are missing from the TOR list altogether.

However, it is some smaller countries (Luxembourg, Costa Rica), African countries (Liberia) and various islands (St Kitts and Nevis, Seychelles) that seem to have found their way to the top 26 in this list. It is highly possible that some of these, such as Liberia with its 100% TOR traffic rate, have been used as TOR exit nodes by users residing elsewhere on the globe, rather than native Liberians themselves. And while the use of TOR, not being quite as widespread among online surfers yet as regular browsing, did not significantly skew the Top 25 results from Section 1, it could have an impact on the results of smaller countries, making it seem as though their nationals may have been complicit in cyberattacks which they may or may not have had anything to do with.

There are several other factors that skew results when it comes to determining the geographical origins of an attack. These may include the following:

1) The use of Virtual Private Networks, which also help obfuscate a user's identity with the help of proxy servers and encryption. [3]

2) IP spoofing, which involves modifying the source IP address in order to, once again, hide the identity of an attacker and make it seem as though another client is conducting the attack. [4]

3) An experienced hacker will rarely, if ever, attempt to penetrate the target server directly from his own host. Instead, he may hack a server located in, for example, South Korea, through which he may take over a computer in Russia, through which he may gain access to a host in Seychelles, from which he will then perform the attack on the final target. A honeypot will only register the location of the host from which the penetration was conducted, which will not reveal the true identity or location of the perpetrator.

4) When it comes to TOR traffic, one single attacker may have used various TOR exit nodes at different times, which means that various TOR logs registered at different times and geolocations may, in fact, have been just one user, who had nothing to do with said countries at all. Likewise, one TOR exit node may have been used by different people accessing it within a short timespan, while it was still listed as such – this would have distorted the acquired geographical data as well.

5) It is also possible that scanning software performing a scan from a single IP address would have registered as many separate attacks – this would be misleading, as based on logged events per country alone, it may look like the country with the most events had the most attackers, whereas in reality it could have been one or a few IP addresses that have generated all of the logs.

6) The geolocation systems used to pinpoint the location of a certain IP are not always very accurate. [5] In addition to this, these systems may be hacked and information they provide about an IP may skew as a result.

Another factor that might affect the quality and accuracy of data registered by a honeypot is the detection of the decoy server as such by the intruder. [2] This may not affect the registered geolocation as much as other aspects of the data, but it is still an aspect worth noting.

The information revealed in this section adds a little more context to the trends uncovered in the previous part. Here, the reader has learned that attackers may use an anonymous network in order to conduct attacks, and this makes their real location more difficult to determine. However, the data collected shows that the use of TOR traffic is not that widespread, making up about **0.08%** of the overall traffic – this may mean that only a small portion of the world's web-surfing population is well-versed enough in information technology to know about TOR and use it.

The use of TOR is so far limited to mostly the Americas and Europe, its use in Asian, African and Australian locations remaining rather sporadic. In terms of intelligence, this information is valuable because it manages to draw a rough map of the regions that not only have the largest number of skilled and knowledgeable people with technology advanced enough to conduct cyberattacks, but also have the most people who have the necessary resources to disguise their real identities.

In addition to this, higher rates of TOR network usage on the American continent and Europe mean that data coming out of these countries, especially concerning geolocation, has a bigger chance of being inaccurate. Thus, when examining traces left by attackers from these locations, additional testing and analysis should be conducted to establish other trends that will help identify the miscreants.

SECTION 3. TOP 25 ATTACKING TOR IPS

Having found that a number of countries hosts IP addresses that serve as TOR exit nodes, the next step is to determine the number of events generated by each distinct TOR IP. This may provide a

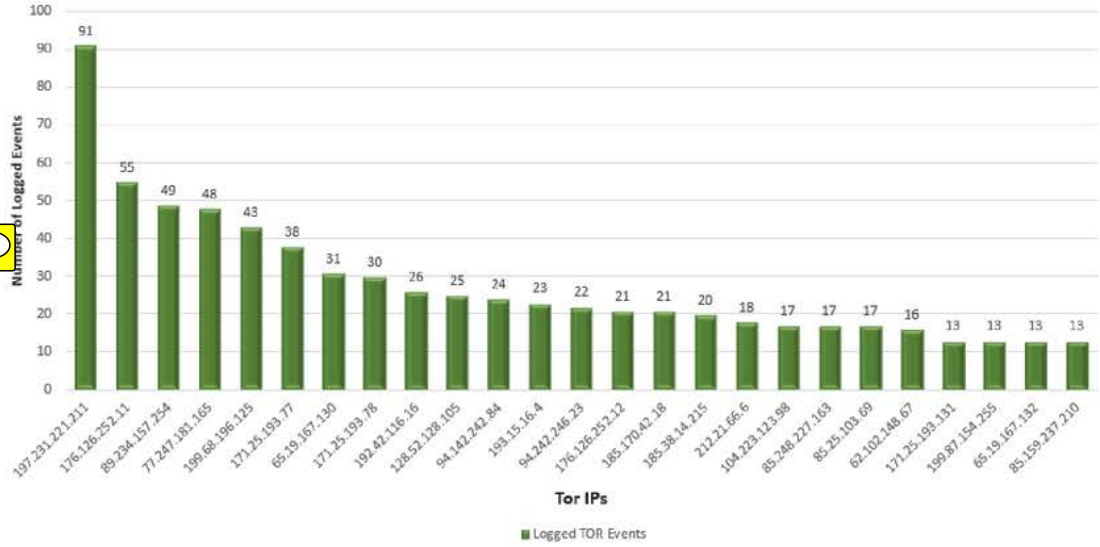
little more context for the attacks. It might also reveal which countries' IP addresses were the most popular TOR exit nodes.

The Python script for the extraction of needed information was based on the codes used for previous sections. By means of the method discussed in Section 2, a file listing only the "*Tor_Flag – True*" entries was created and exported as a separate CSV file. Then, a set of all unique IPs was determined, and the number of entries was counted for each IP from the set. The last script for this session determined which country was hosting each IP from the set. Please refer to ***Question_03_Code.py*** for the complete script.

A total of **285** unique TOR IPs was registered by the honeynet over the course of the examined time period. The resulting top 25 TOR IPs, ranked by their corresponding number of logged events, are presented in the chart in **Figure 3.1**. The chart displays each IP and the count of its logs, and the table in **Figure 3.2** reveals the country associated with each of the top 25 TOR IPs.

The TOR IP with the highest event count is located in Liberia – it appears that all the Liberian-generated traffic in this data sample comes from a single IP. The other countries that made it into the top 25 are all in Europe or the Americas. They include Romania, France, the Netherlands, Sweden, Luxembourg, Germany, Slovakia, the United States and Canada. Following **Liberia**, the top 5 countries on this list are **Romania**, **France**, the **Netherlands** and the **United States**.

FIGURE 3.1: TOP 25 ATTACKING TOR IPS BY TOTAL NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)



Sweden and USA both boast the largest number of TOR IPs (5 each) that made it into the top 25 list, according to the number of logged events generated by each IP. Following close behind are the Netherlands and Germany, with four of the 25 top IPs belonging to one, and four to the other.

FIGURE 3.2: TOP 25 TOR IPS BY TOTAL NUMBER OF LOGGED EVENTS, WITH CORRESPONDING COUNTRIES

Tor IP	Country	# of Logged Event	Rank
197.231.221.211	Liberia	91	1
176.126.252.11	Romania	55	2
89.234.157.254	France	49	3
77.247.181.165	Netherlands	48	4
199.68.196.125	United States	43	5
171.25.193.77	Sweden	38	6
65.19.167.130	United States	31	7
171.25.193.78	Sweden	30	8
192.42.116.16	Netherlands	26	9
128.52.128.105	United States	25	10
94.142.242.84	Netherlands	24	11
193.15.16.4	Sweden	23	12
94.242.246.23	Luxembourg	22	13
176.126.252.12	Romania	21	14
185.170.42.18	Germany	21	15
185.38.14.215	Netherlands	20	16
212.21.66.6	Germany	18	17
104.223.123.98	United States	17	18
85.248.227.163	Slovakia	17	19
85.25.103.69	Germany	17	20
62.102.148.67	Sweden	16	21
171.25.193.131	Sweden	13	22
199.87.154.255	Canada	13	23
65.19.167.132	United States	13	24
85.159.237.210	Germany	13	25

Thus, the data seems to indicate that the most popular TOR exit points are located in USA and northern Europe: Sweden, Germany and the Netherlands.

Nonetheless, with a total of 91 events originating from a single IP, this list is topped by Liberia. One of the possibilities in this case is that the Liberian TOR exit node has been used to conduct a

scan or a series of attacks by a single user, as all the registered entries originate from the same IP address. It is also likely, however, that this could simply be a popular exit node for different users; or that this is the only TOR server existing in Liberia in general. The data mined so far does not provide enough context for definitive conclusions to be drawn regarding this particular IP. An interested investigator would have to use other details recorded by the log in order to piece this puzzle together.

The intelligence gathered in this section outlines some of the most popular TOR exit points used in the recent years among those users preferring anonymity online. Cybersecurity specialists may use this information to pay closer attention to traffic coming from these countries. They may also want to conduct more detailed reconnaissance on digital traffic originating in these nations, as their servers have the biggest potential to be used by seasoned hackers who know how to cover their tracks.

SECTION 4. COMMONLY ATTACKED PROTOCOLS

Studying the protocols preferred by malicious users may prove useful in educating cybersecurity specialists about the digital criminals' favourite way of obtaining access to networks and individual devices. This may help identify the most vulnerable or targeted services in need of increased protection.

For the extraction of data for this segment, two scripts were used. The first script determined the set of all protocol events registered by the honeypots. The second script counted how many times each protocol was targeted, according to the number of corresponding events registered in the log file. Please see ***Question_04_Code.py*** to test the aforementioned codes.

It should be noted that certain honeypots expose a particular range of specific services. For example, Dionaea opens services such as MSSQL, SIP, FTP, HTTP, TFTP, SSH and SMB, while Cowrie is designed to emulate the SSH service and log brute force attacks. Thus, it is expected that the variety of protocols attacked may be quite limited in comparison to a real-life scenario in a big corporation or a governmental institution.

The results have returned a set of 14 unique protocol events (excluding "None", this leaves only 13 unique items registered by the honeypots as protocol events). These include SSH, TCP, UDP, HTTP and ICMP; as well as items such as *pcap, SipSession, SipCall, mysqld, ftpd, RtpUdpStream, ftpdatalisten* and *smbd*. For a fuller picture, both the protocols and the services have been presented in two separate tables, with the count of logged events shown for each.

Please refer to **Figure 4.1** and **Figure 4.2** for detailed breakdowns.

FIGURE 4.1: TABLE OF THE MOST COMMONLY ATTACKED PROTOCOLS, BY NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)

Protocol	Total # of Times Logged	Rank
SSH	234,744	1
TCP	102,906	2
UDP	29,882	3
HTTP	17,736	4
ICMP	8,898	5

FIGURE 4.2: TABLE OF THE MOST COMMONLY ATTACKED SERVICES AND OTHER EVENTS, BY NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)

Service	Total # of Times Logged	Rank
pcap	1,353,108	1
SipSession	55,349	2
SipCall	26,013	3
mysqld	9,025	4
ftpd	1,272	5
RtpUdpStream	839	6
"None"	64	7
ftpdatalisten	20	8
smbd	2	9

It is worth mentioning that the number of total ICMP attacks for the time period in question was returned as **8,898** logged events (please refer to ***Question_04_ICMP_Proof.py***). Despite the fact that the Python script returns this as the total number, it is **not** reflected by the number of ICMP events generated by the top 25 non-TOR countries**,** as demonstrated in **Section 5** of this report (even though these countries account for **87.01%** of all non-TOR attacks, according to **Section 1**). TOR countries have not generated any ICMP events as well; however, this is because TOR does not support the use of the ICM Protocol (see **Section 5**).

The reason for this discrepancy is that **8,300** of these ICMP events were generated by digital entities that returned no result in the "country" section of the log. When run in Spyder 3.3.2, the script returns the results for the countries responsible for these logs as "**n/a**". A more detailed breakdown of total ICMP events by country may be found below, in **Figure 4.3**. It illustrates the drastic difference between the ICMP numbers presented in **Figure 4.1** and **Figure 5.1.**

FIGURE 4.3: TABLE OF ALL COUNTRIES THAT USED THE ICMP PROTOCOL
(September 14, 2016 - September 30, 2017)

Country	# of Times ICMP Logged	Percentage of Total # of Times ICMP was Logged
n/a	8,321	93.52%
China	510	5.73%
United States	19	0.21%
Canada	18	0.20%
Guatemala	15	0.17%
Germany	9	0.10%
United Kingdom	5	0.06%
Czechia	1	0.01%

The following conclusions may be reached upon interpreting all of the information presented in this section. According to the data collected over the specified time period, the most targeted service was **SSH**, with close to 235,000 entries. The most commonly used protocol was **TCP**, with 102,906 events logged. However, close to **75%** of the entire data sample is taken up by **pcap**, a packet sniffing protocol.

SipSession and *SipCall* are also quite popular, with just over 80,000 entries to their names in total. This is understandable, as SIP is the protocol used for initiation of voice and video calls over the Internet. The speed and quality of such online communication is improving on a daily basis, and Internet calls are often far less expensive for those living in different countries and on different continents than their mobile phone-provided equivalent. Thus, with communication migrating online at an ever-growing speed, it is little wonder that attempts to eavesdrop on such sessions are becoming a growing trend.

SecureShell, or **SSH,** permits a user to log in to any system on any server remotely if he or she has the correct credentials. Thus, a successful brute force attack onto a privileged account such as "root" or "admin" on a real server would grant a cybercriminal unchecked access to the entire system directly from the comfort of his or her home; as well as the ability to alter any settings or credentials, or to upload or download any file to or from the system. As the data shows, there is no shortage of malicious users trying to gain illegitimate admittance to an exposed SSH service.

While the **ICMP** protocol was not at the top of any lists in this case, it has been used in a fair share of interactions with the honeynet (just under 8,900 times). Over **93%** of ICMP events, however, were generated by users or entities whose geolocation could not be determined. Although further information would be needed to establish precise details, it may very well be the case that most of such events were, in fact, generated by bots or other software rather than human users directly.

Packet sniffing software using the **pcap** protocol may reveal much about the traffic travelling to and from a server, and is a very common and convenient way for hackers to collect useful information about the target system. Thus, it is unsurprising that the bulk of the entries appear to have been such recon scans by threat agents.

In terms of intelligence, this means that companies, institutions and private individuals should take extra care to make sure their SSH service (by default located at TCP port 22) is filtered by a firewall and does not show up as open or visible. It may be an even better idea to switch the ports for SSH, and configure any port between 1024 and 32767 for this purpose instead. [6]

Moreover, it is advisable to ensure that any default accounts having access to SSH are either disabled, or have a strong password in place. Furthermore, any ports using the SIP protocol must be well-protected, as the number of eavesdroppers on people's Skype or Viber video chats is only going to increase with time.

Companies and individuals should also be careful with cookies they leave on pages while browsing the web, as data shows that packet sniffing software is a very popular way of collecting credentials and fishing for other important information in the depths of cyberspace.

SECTION 5. ATTACKING COUNTRIES AND ATTACKED PROTOCOLS

This section will take a look at which protocols are the most favoured by users attacking from various TOR and non-TOR countries. This will help add some context to the nature of attacks preferred by miscreants from various nations, and shed some light onto the goals and motivation behind their intrusion attempts.

The script written for this section asks for the user to input the desired country (one of the top 25 TOR or non-TOR), and analyzes how many entries are registered for this country for each of the following protocols: ICMP, HTTP, SSH, TCP and UDP. The results for the two categories of countries are presented in **Figure 5.1** and **Figure 5.2**. The full original script may be found in the file ***Question_05_Code.py***.

FIGURE 5.1: PROTOCOLS MOST COMMONLY USED BY TOP 25 NON-TOR COUNTRIES, BY NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)

Non-TOR Country	ICMP	HTTP	SSH	TCP	UDP
China	510	793	184,719	60,469	284
United States	19	2,752	8,374	10,418	6,537

Vietnam	0	206	1,044	389	25
Russia	0	2,341	9,704	1,724	1,630
Germany	9	115	1,279	1,966	5,345
Brazil	0	909	2,001	377	16
Taiwan	0	585	454	861	10
Netherlands	0	135	938	6,242	1,930
Ukraine	0	5,319	672	654	6
Republic of Korea	0	98	858	1,247	180
Turkey	0	85	104	189	24
Hong Kong	0	37	6,875	1,834	55
India	0	198	2,217	533	87
France	0	887	2,215	1,658	5,869
Republic of Lithuania	0	51	51	113	3,592
Romania	0	65	87	442	28
United Kingdom	5	153	720	696	424
Mexico	0	420	392	34	1
Argentina	0	50	1,533	194	1
Canada	18	104	1,745	341	2,725
Colombia	0	29	314	59	0
Poland	0	16	202	140	24
Japan	0	261	303	1,877	0
Indonesia	0	69	331	63	10
Iran	0	29	811	247	10

According to the data for the non-TOR countries, the interest in various protocols varies from country to country.

Such countries as **China**, **Vietnam, Russia**, **Brazil**, **India**, **Hong Kong**, **Argentina**, **Colombia**, **Indonesia** and **Iran** had focused most of their efforts on attacking the **SSH** protocol.

Germany, France, Lithuania and **Canada** seem to prefer the **UDP** protocol to the others.

United States, Netherlands, Korea, Romania and **Japan** favoured the **TCP** protocol in the case of the observed honeynet. **Ukraine,** on the other hand, has targeted **HTTP** by far the most, whereas other countries have shown significantly less interest in it.

The **ICMP** protocol went unused by most of the top 25 countries, with the exception of **China, United States, Canada, Germany** and the **United Kingdom**, which have only utilized it a minuscule amount of times compared to the number of their respective total logs. The majority of the **ICMP** logs were generated by users or entities whose geolocation was not established by the honeynet (see **Figure 4.3** in **Section 4**).

FIGURE 5.2: PROTOCOLS MOST COMMONLY USED BY TOP 25 TOR COUNTRIES, BY NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)

TOR Country	ICMP	HTTP	SSH	TCP	UDP
United States	0	0	96	144	0
Netherlands	0	6	30	139	0
France	0	0	49	113	0
Sweden	0	0	25	120	0
Romania	0	0	11	109	0
Liberia	0	0	5	61	0
Germany	0	2	19	46	0
Canada	0	0	29	44	0
Switzerland	0	9	8	21	0
Slovakia	0	0	2	33	0
Luxembourg	0	0	6	25	0
Seychelles	0	0	6	13	0
Norway	0	0	3	18	0
Ukraine	0	0	6	5	0
Bulgaria	0	0	6	5	0
Poland	0	0	11	0	0
Republic of Moldova	0	0	1	6	0
Latvia	0	0	2	7	0
United Kingdom	0	0	7	4	0
Czechia	0	7	1	2	0
Denmark	0	0	2	7	0
Iceland	0	0	2	6	0
Finland	0	0	0	7	0
Russia	0	0	2	1	0
Costa Rica	0	0	1	4	0

The TOR list analysis shows slightly different trends. ICMP and UDP are not targeted at all by TOR countries. The reason for this is that TOR simply does not allow users to send UDP packets or ICMP echo requests. [7]

HTTP is only attacked a few times, by the **Netherlands**, **Germany**, **Switzerland** and **Czechia**. The main focus is on the **SSH** and **TCP** protocols, **TCP** being preferred by most of the countries except for **Poland**, which in this case opted for **SSH**.

According to the information uncovered in this section, the following general trends can be distinguished: Russia, the Asian nations and many Latin American countries focus on **SSH** attacks, most likely trying to brute-force their way into vulnerable servers to gain unauthorized control,

while Canada and some of the European nations prefer to send **UDP** packets, which are commonly associated with DDoS attacks.

United States has split its efforts among the **TCP**, **UDP** and **SSH** protocols almost equally, with a slight preference for the **TCP** protocol; while Ukraine has overwhelmingly favoured the **HTTP** protocol, perhaps attempting to exploit its various vulnerabilities or collect cookies.

From this, an analyst can see that Asia, Latin America and Russia are far more interested in attacks that will result in some sort of tangible gain, such as access to restricted, personal or financial information. Another possible aim these attackers may be pursuing is admission to servers with the goal of installing malware on them and turning them into zombie-hosts in a botnet.

Canada and some of the European countries (France, Germany and Lithuania) seem to prefer DDoS-type attacks, which are more commonly associated with take-down of certain websites, usually for activist purposes or out of mischief.

Ukraine's tactic places a heavy emphasis on vulnerability exploits, most likely with the intent of collecting personal information or gaining unauthorized access to servers; while USA is almost equally focused on all types of attacks, with more preference for potential scans, SSH attacks and UDP packets over HTTP.

Knowing these trends may help cybersecurity forces to better understand and predict the motivation of various nations behind their attacks, and help devise defence strategies that would be more "personalized" and effective against attackers from each individual country. For example, if the task of a specific team of security specialists involves protecting important classified information from Russian, Ukrainian or Chinese spies, intelligence such as this will let them know that they should greatly increase the protection of their Secure Shells and HTTP ports.

SECTION 6. TOP ATTACK TIMES

This section takes a look at the times of attacks in order to determine the threat actors' most and least favourite times and week days for their operations. This information may single out the safest hours for the conduction of online activities; as well as the busiest and potentially most dangerous times, during which network security teams may need to double down on server defense.

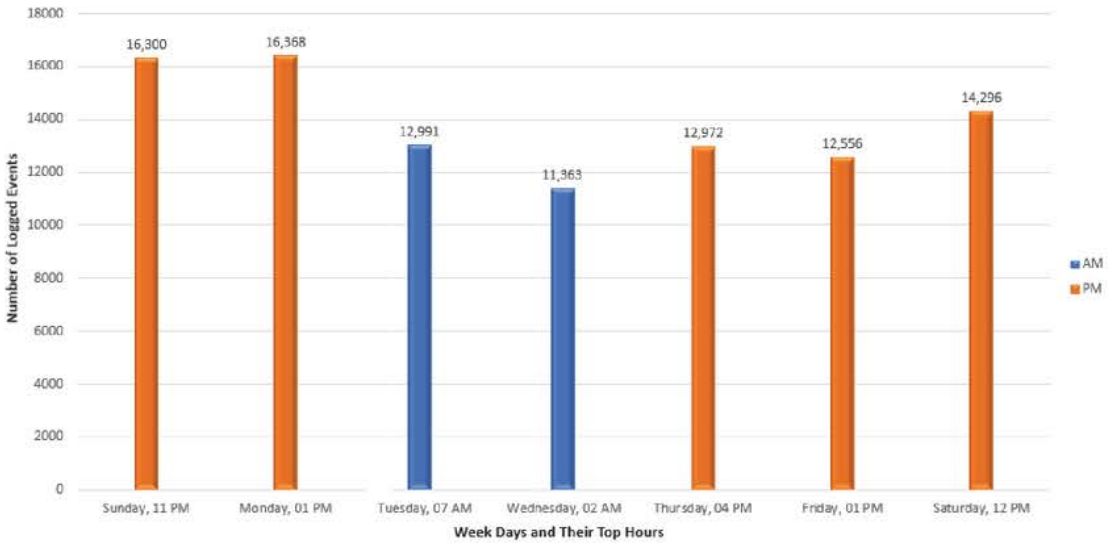
Several Python scripts were used to extract the needed information. They can be found in the file ***Question_06_Code.py***.

The first script obtained a set of week days and hours used in the log file, in order to find out the exact notations of week days and hours generated by the honeypots. This was important, because

the honeypots use the 24-hour notation system rather than AM and PM, and abbreviate the weekday names, so trying to filter the results for "Monday" or "4 PM" would not have worked.

Once all the proper notations were established, the next script would ask for the user to input a specific day of the week, count the entries for each hour made on these days, and then display the results in an AM/PM format presented in **Figure 6.1** and **Figure 6.2**.

FIGURE 6.1: BUSIEST HOURS FOR EACH WEEKDAY, BY NUMBER OF LOGGED EVENTS
(September 14, 2016 - September 30, 2017)



Data shows that the busiest hour of the week is between **1 P.M. and 2 P.M.** on **Monday**, with the greatest number of events having occurred at this hour. The least busy hour of the top-ranked ones is between **2 A.M. and 3 A.M.** on **Wednesday**. Most of the peak hours for malicious activity are in the afternoon, with early morning being the busiest only during Tuesdays and Wednesdays.

The table in **Figure 6.2** shows a detailed breakdown of attacks for each week day, by the hour.

FIGURE 6.2: BREAKDOWN OF ALL LOGGED EVENTS BY HOUR AND WEEKDAY
(September 14, 2016 - September 30, 2017)

Hours	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour Totals
12 AM	9,513	14,237	11,634	10,796	9,415	11,908	8,382	**75,885**
1 AM	10,901	11,897	11,882	8,758	9,880	11,640	9,384	**74,342**

2 AM	9,978	11,249	11,772	**11,363**	9,964	11,640	9,685	**75,651**
3 AM	11,714	11,708	11,090	10,076	11,423	11,683	10,510	**78,204**
4 AM	9,401	10,090	12,231	11,335	11,797	10,258	9,172	**74,284**
5 AM	9,508	10,131	10,211	9,011	11,974	11,983	9,302	**72,120**
6 AM	**8,500**	**9,974**	11,095	8,798	9,531	10,932	9,990	**68,820**
7 AM	9,304	11,093	**12,991**	**7,266**	9,595	9,585	9,504	**69,338**
8 AM	8,859	10,599	12,499	9,214	**8,958**	10,946	10,684	**71,759**
9 AM	15,402	11,682	11,965	8,093	9,080	11,535	9,625	**77,382**
10 AM	14,159	12,442	11,626	8,567	10,259	12,392	10,425	**79,870**
11 AM	13,559	11,183	11,983	7,933	9,874	11,941	9,585	**76,058**
12 PM	14,518	15,196	10,281	8,431	12,660	10,462	**14,296**	**85,844**
1 PM	14,608	**16,368**	9,716	10,987	12,273	**12,556**	10,909	**87,417**
2 PM	13,197	12,488	10,302	10,684	12,944	11,429	10,425	**81,469**
3 PM	13,333	11,129	9,719	9,149	12,567	8,608	11,176	**75,681**
4 PM	13,158	11,925	**9,021**	8,971	**12,972**	9,265	10,998	**76,310**
5 PM	15,196	12,414	11,346	8,183	11,470	**8,395**	9,221	**76,225**
6 PM	13,284	11,368	11,070	8,711	12,718	8,569	10,607	**76,327**
7 PM	13,162	16,348	9,379	9,268	10,629	9,416	10,054	**78,256**
8 PM	13,951	11,170	12,777	8,933	9,132	9,307	10,821	**76,091**
9 PM	14,596	12,664	9,106	7,790	10,787	10,087	11,098	**76,128**
10 PM	13,522	12,989	10,168	8,617	10,602	9,059	9,103	**74,060**
11 PM	**16,300**	12,929	10,785	8,832	10,645	11,957	10,889	**82,337**
Weekday Totals:	**299,623**	**293,273**	**264,649**	**219,766**	**261,149**	**255,553**	**245,845**	**All Entries: 1,839,858**

The numbers for the hours during which **the least** attacks took place on each weekday are highlighted in **green**. The numbers for hours during which **the most** attacks took place on each weekday are highlighted in **red**.

The day when the **least** attacks were recorded for the time period between September 2016 and September 2017 was **Wednesday.** The **most** attacks were logged on **Sunday**.

Statistically, the **safest** hour for online interaction over that period was determined to be between **6 A.M. and 7 A.M:** this hour had the fewest recorded attacks in total, and also the fewest recorded attacks on Sundays and Mondays.

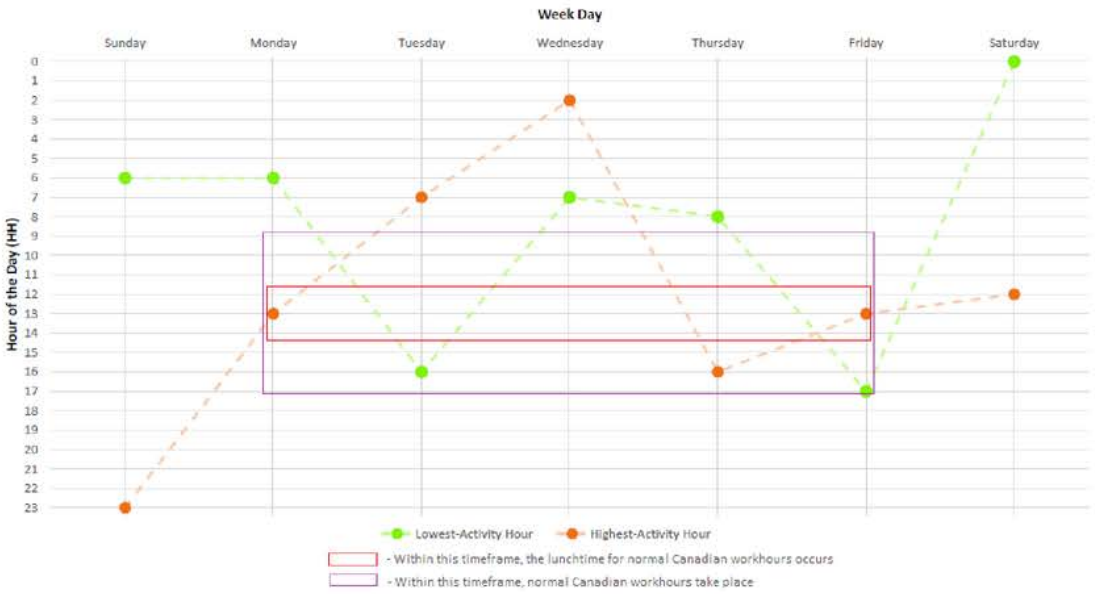
The time period between 7 A.M. and 8 A.M. had the second fewest total entries, and had the fewest attacks recorded on Wednesday – this number was also the smallest out of any hour for any week. However, on Tuesday, the **most** attacks were recorded at this hour.

The hour with **the most** total attacks registered was between **1 P.M. and 2 P.M** – right after lunch time. On Monday and Friday, the most attacks for the day happened between these two hours. In fact, the number of Monday attacks between these two hours was the greatest out of any hour for any day of the week over the studied time period. The standard lunchtime, **12 P.M. to 1 P.M.**, saw

the second largest number of total attacks, and the most attacks for any hour on Saturday happened within this time period.

FIGURE 6.3: HOURS OF HIGHEST AND LOWEST ACTIVITY FOR EACH WEEKDAY, BASED ON NUMBER OF LOGGED EVENTS

(September 14, 2016 - September 30, 2017)



Thus, according to this study, the safest time for online transactions in Toronto may be in the morning, between **6 A.M. and 8 A.M**. (except on Tuesdays), while the time to watch out for would be at lunch, between **12 P.M. and 2 P.M**.

SECTION 7. ATTACKS ON COWRIE

This section examines the honeypot known as Cowrie. Cowrie logs brute-force penetration attempts made on the SSH service. It records the username/password combinations tried by the threat actors, and registers the failed and the successful ones. By examining Cowrie logs, it becomes evident whether a penetration attempt was carried out by a bot or a human, which dictionary was used, how long each brute-force attempt took until success, and what tactics a threat agent may have used to access the target server (e.g. used one host to perform a scan and a brute-force attack, and another one at a later time to log into the server with the uncovered credentials). Cowrie also logs and reveals all commands a malicious user executes while inside the system.

In the case of Ryerson University's honeynet, Cowrie is configured to register successful or unsuccessful login attempts under the column header '*loggedin*'. A failed login returns "None" (or empty square brackets) as a result, while a successful attempt returns the credentials used to log in. Thus, the script for the extraction of this data prompts the command line to print an item's country, source IP, timestamp and credentials whenever the *loggedin* parameter does not return 'None'. As a result, a total of eight successful attacks on Ryerson University's Cowrie has been registered in the specified time period. Their details are displayed in **Figure 7.1**.

FIGURE 7.1: TABLE OF SUCCESSFUL COWRIE ATTACKS
(September 14, 2016 - September 30, 2017)

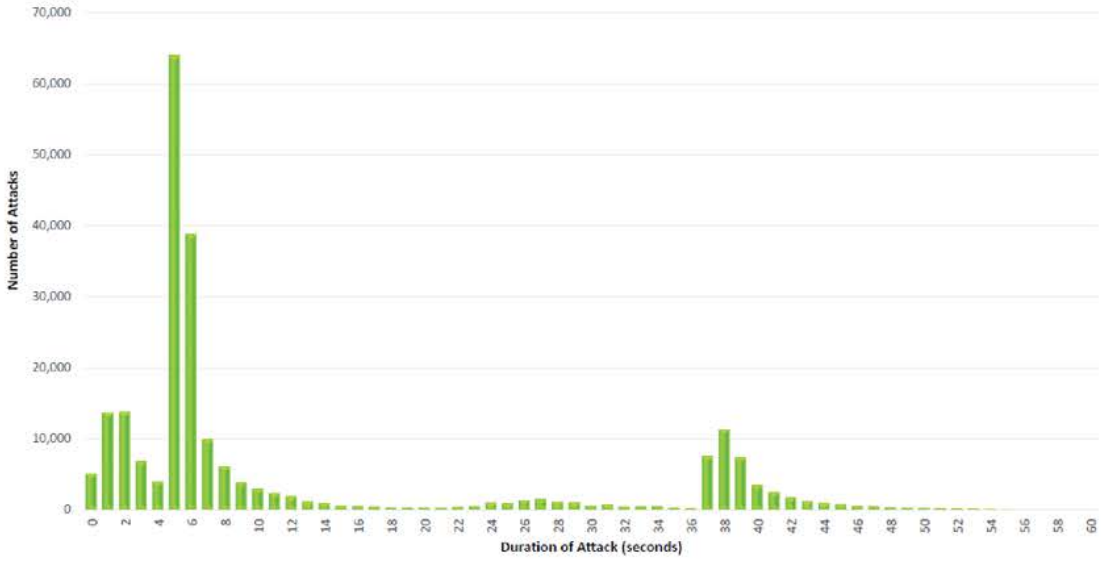
Source IP	Source Country	Date	Time	Username	Password	Duration of Attack Until Success (sec)
218.87.109.248	China	2016-09-26	00:38:21.000	root	135790	2,472.10
149.56.24.204	Canada	2016-09-26	12:01:42.341	root	135790	3.9
218.65.30.122	China	2017-07-15	07:25:59.468	root	135790	14,202.30
61.177.172.19	China	2017-07-15	16:06:28.292	root	135790	6,893.60
59.63.166.80	China	2017-07-16	06.18.07.221	root	135790	20,455.10
158.69.79.38	Canada	2017-07-16	12:15:10.930	root	135790	1.8
158.69.79.35	Canada	2017-07-17	02:11:21.126	root	135790	1.6
158.69.79.38	Canada	2017-07-17	02:30:16.862	root	135790	1.5

The successful penetrations were divided evenly between China and Canada, four by each country, all having uncovered the correct password for the **root** directory: **135790**. It appears as though China's 'breakthroughs' happened during scans spanning several hours each. These interactions consisted of multiple consecutive attacks most likely indicating software rather than human guesses. Meanwhile, all Canadian perpetrators managed to log in on their first attempt.

Another interesting detail is that the attacks seem to come in pairs. Two of them occur on **September 26, 2016**, two more on **July 15, 2017**, then two others on **July 16, 2017**, and the final two on **July 17, 2017**. Moreover, the last three Canadian attacks seem to have been conducted from the same IP subnet. There seems to be a pattern emerging that indicates a potential relationship between some or all of these attacks, which will be explored in greater depth in Section 10.

The next aim of this section is to examine and uncover any trends related to the duration of Cowrie attacks. In Python scripting, this can be achieved using the *datetime* module. The span of each event recorded by Cowrie was calculated, and the attacks were then grouped into three categories according to their length. Category A, which covers attacks 1 minute in duration or shorter, is depicted in **Figure 7.2** and **Figure 7.3**.

FIGURE 7.2: ATTACK DURATION FOR COWRIE: 60 SECONDS OR LESS (BASED ON FREQUENCY OF EVENTS) - CATEGORY A
(September 14, 2016 - September 30, 2017)



In Category A, the largest number of attacks spans between 5 and 6 seconds in length and includes **63,931** entries. The timespan of 6-to-7 seconds in length counts over half as many entries (**38,843**). Entries spanning between 1 and 2 seconds in duration and between 2 and 3 seconds in duration have scored just over **13,700** events each.

A detailed event count for each timespan beginning with 0 seconds and ending at the 60 second cut-off is displayed in **Figure 7.3**.

FIGURE 7.3: DURATION OF ATTACKS FOR COWRIE, BASED ON FREQUENCY OF EVENTS
1 MINUTE OR LESS: CATEGORY A
(September 14, 2016 - September 30, 2017)

Duration of Attack (Seconds)	# of Events Logged	% of Total Events Logged
0	5,102	2.21%
1	13,723	5.93%
2	13,796	5.97%
3	6,899	2.98%
4	4,034	1.74%
5	63,931	27.65%
6	38,843	16.80%

7	9,944	4.30%
8	6,136	2.65%
9	3,910	1.69%
10	3,004	1.30%
11	2,393	1.03%
12	1,934	0.84%
13	1,205	0.52%
14	962	0.42%
15	665	0.29%
16	553	0.24%
17	474	0.20%
18	367	0.16%
19	332	0.14%
20	342	0.15%
21	321	0.14%
22	436	0.19%
23	490	0.21%
24	1,071	0.46%
25	975	0.42%
26	1,388	0.60%
27	1,610	0.70%
28	1,148	0.50%
29	1,109	0.48%
30	654	0.28%
31	763	0.33%
32	464	0.20%
33	486	0.21%
34	513	0.22%
35	348	0.15%
36	234	0.10%
37	7,631	3.30%
38	11,408	4.93%
39	7,429	3.21%
40	3,548	1.53%
41	2,526	1.09%
42	1,776	0.77%
43	1,242	0.54%
44	997	0.43%
45	801	0.35%
46	584	0.25%
47	505	0.22%
48	400	0.17%
49	322	0.14%
50	291	0.13%
51	249	0.11%
52	218	0.09%

53	171	0.07%
54	149	0.06%
55	105	0.05%
56	77	0.03%
57	68	0.03%
58	74	0.03%
59	56	0.02%
60	59	0.03%

Category B includes attacks 61 to 90 seconds in length and Category C encompasses events longer than 91 seconds. These two groups are presented by **Figure 7.4**, **Figure 7.5** and **Figure 7.6.**

The reason for the split of the 60+ second group into two categories is the sharp spike in the number of events spanning 120 seconds (2 minutes) in length. Two-minute attacks count a total of over **2,700** events. Compared to the amount of events registered for other timeframes (which range between **1** and **56** events), this is an immense number. It dwarfs all other entries on the chart, making it difficult to spot any trends that may have otherwise been revealed through a visual representation of the data. Thus, attacks with a span of 61 seconds to 1.5 minutes are demonstrated in a separate graph.

FIGURE 7.4: ATTACK DURATION FOR COWRIE: 61 - 90 SECONDS (BASED ON FREQUENCY OF EVENTS) - CATEGORY B
(September 14, 2016 - September 30, 2017)

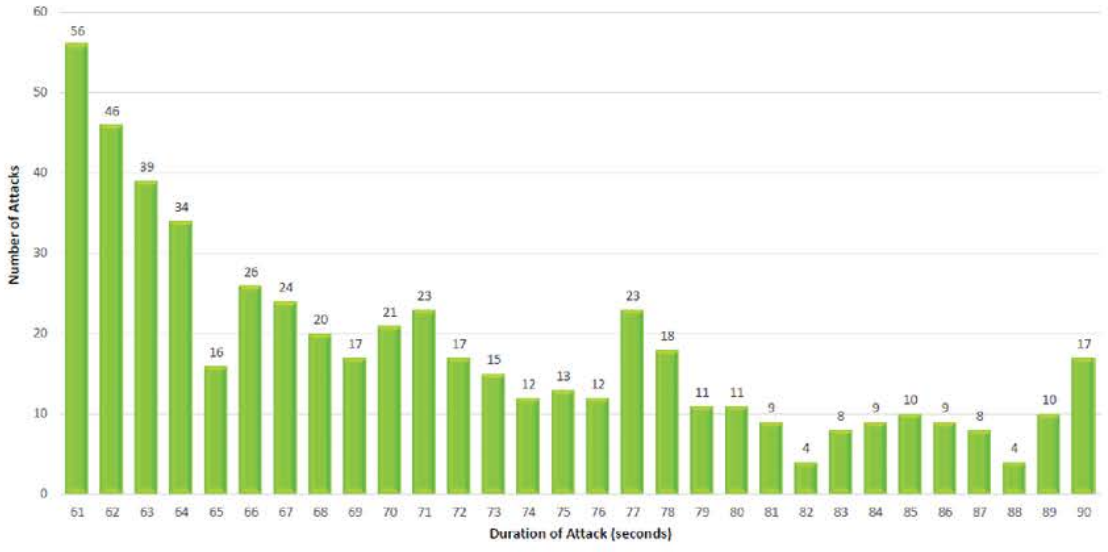
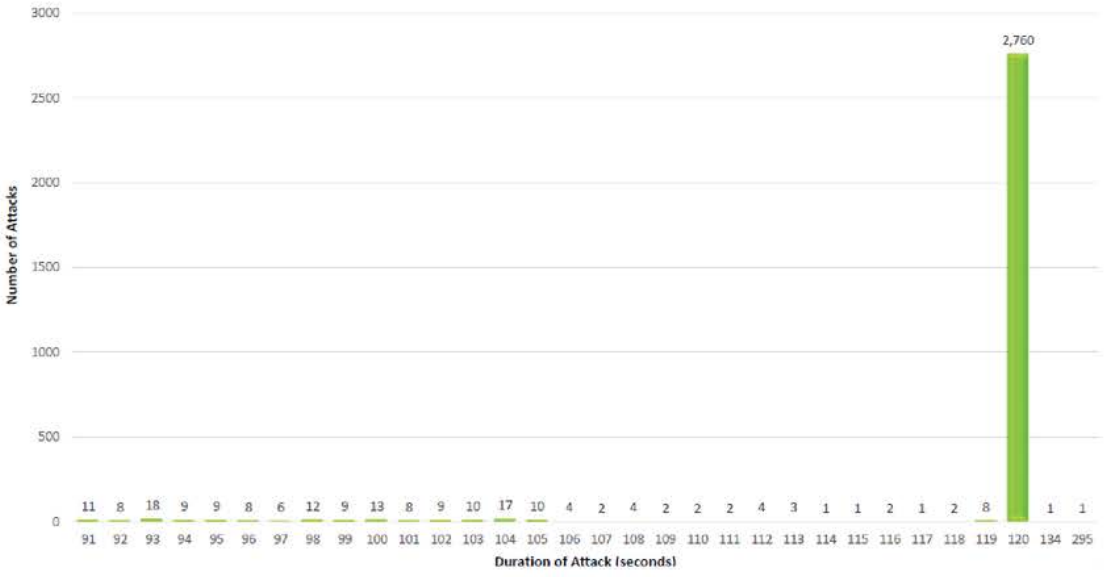


Figure 7.4 demonstrates a general decrease in the amount of attack events for each duration after the 1-minute mark, with the allowance of a few small spikes. The only big exception occurs at the 2-minute mark, where the number of events skyrockets to **2,760** attacks.

It is possible that the cause for such a sharp rise lies in the fact that many of the attacks were performed by brute-force software. The even divide of break-in attempts into 2-minute password-guessing sessions could have either been hard-coded into the program or configured as a setting by a user. It may be speculated that throughout each session, such software would try a specific number of username/password combinations, and would keep going regardless of success or failure until the dictionary supplied to the program is iterated to the end. Since threat agents with more experience tend to use software to automate and speed up otherwise time-consuming processes rather than performing repetitive tasks manually, this may be one likely explanation for the 2,700+ 2-minute events.

FIGURE 7.5: ATTACK DURATION FOR COWRIE: 91 SECONDS OR MORE (BASED ON FREQUENCY OF EVENTS) - CATEGORY C
(September 14, 2016 - September 30, 2017)



The table below provides a numeral breakdown of the visual graphs presented in **Figure 7.4** and **Figure 7.5.** The percentages of total events logged by Cowrie for each of the specified durations have also been calculated and presented in **Figure 7.6**.

FIGURE 7.6: DURATION OF ATTACKS FOR COWRIE, BASED ON FREQUENCY OF EVENTS
MORE THAN 1 MINUTE: CATEGORIES B AND C
(September 14, 2016 - September 30, 2017)

Duration of Attack (Seconds)	# of Events Logged	% of Total Events Logged
61	56	1.60%
62	46	1.31%
63	39	1.11%
64	34	0.97%
65	16	0.46%
66	26	0.74%
67	24	0.69%
68	20	0.57%
69	17	0.49%
70	21	0.60%
71	23	0.66%
72	17	0.49%
73	15	0.43%
74	12	0.34%
75	13	0.37%
76	12	0.34%
77	23	0.66%
78	18	0.51%
79	11	0.31%
80	11	0.31%
81	9	0.26%
82	4	0.11%
83	8	0.23%
84	9	0.26%
85	10	0.29%
86	9	0.26%
87	8	0.23%
88	4	0.11%
89	10	0.29%
90	17	0.49%
91	11	0.31%
92	8	0.23%
93	18	0.51%
94	9	0.26%
95	9	0.26%
96	8	0.23%
97	6	0.17%
98	12	0.34%
99	9	0.26%

100	13	0.37%
101	8	0.23%
102	9	0.26%
103	10	0.29%
104	17	0.49%
105	10	0.29%
106	4	0.11%
107	2	0.06%
108	4	0.11%
109	2	0.06%
110	2	0.06%
111	2	0.06%
112	4	0.11%
113	3	0.09%
114	1	0.03%
115	1	0.03%
116	2	0.06%
117	1	0.03%
118	2	0.06%
119	8	0.23%
120	**2,760**	**78.88%**
134	1	0.03%
295	1	0.03%

The information discovered in this section provides more details for various threat actors and their attempts to exploit an exposed SSH service. It shows, for example, that the only countries whose apparent representatives were successful in cracking the password in this case were China and Canada, meaning that the dictionaries or methods used by the other 138 countries that attempted to tackle Ryerson's Cowrie were not good enough, or did not last long enough in order to crack a six-digit password.

From the data extracted, the time it took to crack the password in each case can be calculated. Another observation that can be made is that the successful threat actors did not spend too much time in the system they managed to access. It is probable that they were seasoned hackers who quickly realized that they were inside a honeypot, so they chose not to waste their time trying to explore it. In contrast, a script kiddie with limited knowledge of honeypot specifics may have been excited at the prospect of successfully breaking into a server, and would have likely spent longer inside the honeypot, performing various commands.

The vast majority of attacks on Cowrie numbered between 0 seconds and 1 minute in frequency, with another sharp spike of events registered at the 2-minute mark. Most attack events on Cowrie, however, were just 5-to-7 seconds or 1-to-3 seconds long. Only two logged attacks were longer than 2 minutes in duration. One lasted **134 seconds** (2.2 minutes), and the other spanned **295**

seconds (4.9 minutes). This demonstrates that whether the entity attacking an SSH service is a human being or a bot, short attacks or those split into bite-size sessions are preferred to those lasting between 1 and 2 minutes, or over 2 minutes, the latter being the rarest.

There might also be a connection between the Canadian and Chinese attacks in this case, which may indicate a trend in how certain attackers operate to get the results they are seeking. In addition, this research could shed some more light onto the threat agents' methods of covering up their tracks.

This information can provide useful intelligence for cybersecurity teams looking to fortify their systems against threat actors targeting SSH. They may take note of all the passwords already known to brute-force software, and the length it takes for hackers to crack a password of varying difficulties. This would then be taken into account when the security administrators will be setting future safety policies for the entity they are trying to protect. Researchers may also study the scan/attack patterns to better familiarize themselves with the tactics used in achievement of successful penetrations, and devise better defence strategies to counter such threats. Another suggestion could be to configure any defence mechanisms protecting the SSH to pay attention to series of repeated interactions 5-6 seconds or 1-2 seconds in duration, and alert security administrators to investigate such occurrences more closely if they are unusual for the network in question.

All of the scripts used to extract the data for this section may be found and tested in the file ***Question_07_Code.py***.

SECTION 8. DESTINATION IP SCANS

The scope of this section specifies the type of scan to be researched as the IP scan that aims to quickly determine which IP addresses are accessible in order for the perpetrator to focus attack efforts more precisely. A scan that may be well-suited for such a situation is a regular ICMP echo request. Thus, ICMP scans will be at the centre of the discussion in this section.

Since not every series of events generated in quick succession by the same IP will necessarily indicate a quick accessibility scan, one has to be careful with the criteria chosen to extract the information representative of such a scan.

The assumption is that this type of scan will be using the ICMP protocol; thus, it will be one of the filter criteria in this scenario. The second specification is that such a scan will eventually involve all or most of the honeypot IPs.

However, the catch in this situation is that a series of events displaying a succession of quick interactions one right after another may be indicative of a script kiddie rather than a more serious threat actor. The reason for this is that an experienced hacker will know that such a scan may set off an intrusion detection alarm, and will be far more careful and selective when trying to scan a network. Unlike an impatient script kiddie, such a threat actor may wait for several hours, days or weeks between host scans, as to not raise any alarms. So, while the time span of attacks is a noteworthy aspect in this study, it should not be used as the main criterium for dismissal of a potential IP scan candidate.

The script used to extract the necessary information in this scenario involved separating ICMP logs into five lists, based on the targeted destination IP (please refer to ***Question_08_Code.py)***. Next, these lists were searched for intersections which returned only the IPs that scanned three or more honeynet destination IPs. More details about the resulting perpetrator IPs, such as their city and country of origin, as well as the number of honeypots that logged the attack in addition to Snort were then uncovered. The results of the query are displayed in **Figure 8.1** below.

FIGURE 8.1: DESTINATION IP SCANS USING THE ICMP PROTOCOL
(September 14, 2016 - September 30, 2017)

Source IP	City	Country	Number of Honeypots That Logged Attack	Number of IPs Scanned	Protocols Used
188.110.229.198	Braunschweig	Germany	0 (it was a snort alert)	5	ICMP
141.117.210.27	Toronto	Canada	0 (it was a snort alert)	5	ICMP
10.112.8.26	n/a	n/a	0 (it was a snort alert)	5	ICMP
180.153.224.25	Shanghai	China	0 (it was a snort alert)	4	ICMP
141.117.240.125	Toronto	Canada	1 (Cowrie, SSH)	5	ICMP, TCP, SSH
171.35.195.36	Nanchang	China	0 (it was a snort alert)	5	ICMP
60.6.210.131	Hebei	China	0 (it was a snort alert)	5	ICMP
60.12.5.27	Hangzhou	China	0 (it was a snort alert)	4	ICMP
59.46.193.114	Dalian	China	1 (Dionaea, pcap)	4	ICMP, pcap
186.151.197.254	Guatemala City	Guatemala	0 (it was a snort alert)	4	ICMP
59.63.1.196	Nanchang	China	0 (it was a snort alert)	3	ICMP

Based on the attack times, the first five attackers in the list (Braunschweig, two Torontonians, no specified country and Shanghai) were most likely script kiddies, as they have scanned all of the

destination IPs within the timespan of a few seconds. In fact, in the case of the attacker with an unspecified location, the scan persisted over the course of several hours. It is thus possible that this particular case may have been a DDoS attempt rather than an accessibility scan.

However, it is likely that the attackers working from Guatemala City, Nanchang, Hebei, Hangzhou and Dalian were from the more advanced crowd. In their scenario, the IP address would systematically generate only four "ping" packets to one of the honeynet destination IPs a day, scanning every day until all or almost all of the destination IPs were covered. This seems like a far more careful tactic that would be likely to let the intruder know which of the systems were up with a far smaller chance of alerting the intrusion detection system that something odd was going on.

SECTION 9. COUNTRIES AND DISTINCT ATTACKING IPS

This section compares countries in terms of their number of unique TOR and non-TOR IPs (i.e. individual clients performing the attacks). It makes use of modified versions of Python scripts previously developed for this project in order to count the unique IPs for every TOR and non-TOR country. The results are displayed in **Figure 9.1** and **Figure 9.2**. The scripts may be found in the file ***Question_09_Code.py***.

FIGURE 9.1: TOP 25 COUNTRIES BASED ON NUMBER OF UNIQUE NON-TOR IPS
(September 14, 2016 - September 30, 2017)

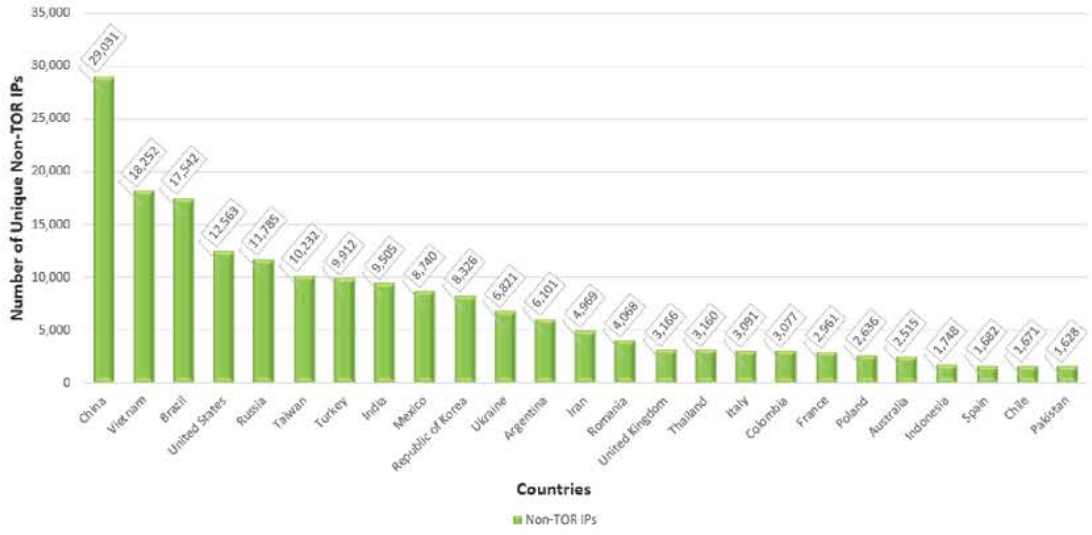
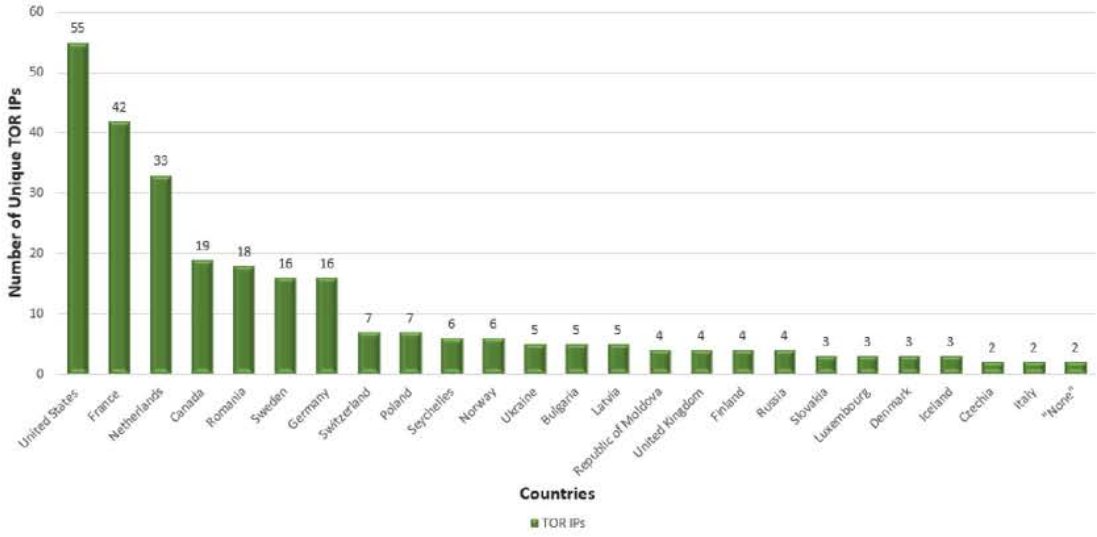


FIGURE 9.2: TOP 25 COUNTRIES BASED ON NUMBER OF UNIQUE TOR IPS
(September 14, 2016 - September 30, 2017)



Changes in ranks for each country can be observed in both lists. Germany vanishes from the list of leaders for the non-TOR countries altogether. It looks like the top five countries with the greatest number of unique IP addresses are China, Vietnam, Brazil, United States and Russia, with several other countries moving up the ranks into the top 25.

With TOR nations, there is also a slight shuffle in ranks. Sweden now ranks 6th (as opposed to its 4th place in **Section 2**), and Liberia is gone from the top 25 entirely.

In terms of intelligence, this information may add perspective to the results discovered previously (in the sections that focus on rating countries purely by number of logged events). Filtering the results thusly may help identify the nations that might have simply been used as TOR exit nodes for a single threat actor; or which host IPs that had been used to only conduct scans. As a result, one may discover that such countries may have shown an artificially increased number of logged events. Consequently, the rank of these and other countries on the cybersecurity threat list may be affected; and decisions might be made to allocate resources to different, more imminent threats.

SECTION 10. COWRIE IP SCAN/ATTACK PAIRS

It has been mentioned in Section 7 that some of the successful attack patterns on Cowrie's SSH service suggest that several of them may (or may not) be related. There is a possibility that these are not eight different attacks by unrelated users, but four scan/attack pairs performed by four or fewer threat actors from different hosts. However, more context is needed to determine whether this is truly the case or not. This section will analyze these events in greater detail in order to clarify the situation.

The first clue suggesting a relationship between the attacks is that in four instances, the intruders were able to log into Cowrie on their first try. It is highly unlikely that these were lucky guesses, thus there must have been successful scans performed at an earlier time that have found the correct combination and provided the attackers with the needed credentials.

In this scenario, a scan is a series of attempts by an entity to guess the correct credentials required for admittance into the victim system. Unlike a human, a bot would be able to try multiple username/password combinations within seconds, and keep going for a long time - several hours or more. Even if a correct combination is found at a point, the attacking malware will likely not stop there. Instead, it will keep going until it iterates through the entire dictionary without logging into the system or performing any actions other than scanning.

The threat actor would then log in on the first try, some time after the scan was performed but not much later. If the attacker waits too long to try the discovered combination, the assumption is that the legitimate user might have time to change the password. The threat actor does not want this to happen, so he or she must act fast enough after the correct credentials are discovered, while taking due care to avoid detection. Thus, in order to determine which IP could have possibly been the attacker's scanning "partner in crime", the first step an investigator should take would be to look at how many combinations, how fast and over what timespan any IP has attempted some time shortly preceding the day and hour of the attack that was successful on its first try.

Since there can be multiple scans taking place at the same time, a good clue that would help an analyst find the scan "pair" to the attack would be the correct username/password combination registered in the "*loggedin*" section of Cowrie by an IP that did not log into the system immediately upon successful discovery of the required credentials. Of course, there is always the possibility that some of the unsuccessful scans were performed by the same person from other hosts and using other dictionaries as well. However, since there will be no direct links between such failed scans and the first-try attacker, these cannot be considered as part of the "team". Thus, only the number of events generated by the IP that can be reliably tied to the successful attacker (through the fact that it had uncovered the correct credentials as part of a longer scan shortly before the intrusion) should be considered as the reasonable number of potential scan events per attack.

Overall, only eight "log-in" attempts recorded by Cowrie over the timeline studied were successful. Two of them took place on September 26, 2016, and the rest in mid-July 2017.

Thus, in order to find the scan/attack pair for the September 26 attack, all logs for September 26, as well as the two days preceding it, were mined and examined. It was confirmed that for the log-in successful on the first try there was only one successful neighboring scan. It appears that on September 26, 2016, a host in China ran a seven-hour brute-force scan of Cowrie starting shortly before midnight. Then the credentials uncovered over the course of 578 attack events were utilized to log into the victim system from a host in Canada at around noon that same day. The attacker then navigated through several directories, although the logger recorded just a few commands. Perhaps the perpetrator realized that this was a decoy system, and quickly left.

FIGURE 10.1: SCAN/ATTACK PAIR TABLE, A COMPARISON OF NUMBERS OF LOGGED EVENTS

Source IP	Source Country	Attack Duration	User Name	Password	# of Events by IP within Attack Timespan
218.87.109.248	China	2016 Sep. 25, 23:57:50.694 to 2016 Sep. 26, 07:24:36.543	root	135790	578
149.56.24.204	Canada	2016 Sep. 26, 12:01:14.3 to 12:01:18.2	root	135790	1
218.65.30.122	China	2017 July 15, 03:29:54.478 to 08:57:05.834	root	135790	518
61.177.172.19	China	2017 July 15, 14:12:12.935 to 2017 July 16, 08:44:55.589	root	135790	944
59.63.166.80	China	2017 July 16, 00:37:40.016 to 2017 July 16, 16:20:40:965	root	135790	1,017
158.69.79.38	Canada	2017 July 16, 12:15:10.930 to 12:15:44.3	root	135790	1
158.69.79.35	Canada	2017 July 17, 02:11:21.126 to 02:11:54.2	root	135790	1
158.69.79.38	Canada	2017 July 17, 02:30:18.862 to 02:30:50.9	root	135790	1

The other three scan/attack pairs are even more interesting. Three different hosts in China were used to perform scans of various length. The first one took place on July 15, 2017 and lasted from 3:30 AM to 8:57 AM. The second scan was started on the afternoon of the same day, July 15 (at 2:12 PM, to be precise) and completed on 8:45 AM of July 16. The third scan ran simultaneously with the second, beginning at about half past midnight on July 16 and ending at a quarter past noon of the same day. All three scans have cracked the password for the "root" directory of Cowrie, this being **135790**.

Then, over the span of July 16 to July 17, three separate log-ins to Cowrie via SSH happened from two stations located on the same subnet in Canada, these computers having the IP addresses of **158.69.79.35** and **158.69.79.38**. Since the timing and the same subnet are unlikely a coincidence, it may be safe to assume that this triple scan/attack pair was orchestrated by the same threat actor or team of threat actors. No further commands were performed after the threat actor logged in – perhaps he or she also realized that this was a trap, and decided not to spend any more time on this "compromised" system.

A quick check in the list of all TOR IP addresses created for this time period earlier showed that these two Canadian entries were never listed as TOR exit nodes within the observed timespan. This means that these stations were unlikely to be used by random, unrelated people that just happened to all be scanning and attacking the same server over the course of three days.

Based on this data alone, it may be difficult to determine whether the originators of these attacks resided in China and hacked into Canadian computers or vice versa; or if they have assailed Ryerson's Cowrie remotely from a different country altogether. Additional information would have to be analyzed in order to pinpoint their exact origins. There is also a possibility that all four attacks were conducted by the same person – the first one was done when he or she was less experienced; and the second time, he or she came back with a better skillset and more resources, to conduct a second, slightly more complex attack, perhaps for training or testing purposes.

In terms of intelligence, the uncovered information holds many useful clues about the strategy picked by the attackers. For example, it was interesting to learn which hosts and in what country were used to conduct the scans; as well as at what times and for how long each of these scans ran. Other useful details include the time it took the brute-force software to crack a password of a certain composition and length; the time the attacker then waited before trying to log into the compromised system; the actions taken by the intruder once inside the host; and the total time spent by the perpetrator in the system. This intelligence may also hint at the skill level of the attacker. As such, it may give the cybersecurity teams an idea how many script kiddies and seasoned attackers are taking an interest in a certain object, and aid in developing tactics and rules that would help detect and prevent relatively complex attacks.

This section required the use of scripts located in the following files: ***Question_07_Code.py*** and ***Question_10_Code.py***

SECTION 11. THREAT ACTOR ANALYSIS IV

The scope of this section is to identify the top 5 countries for each honeypot at each honeynet IP; and for every one of these top five countries in every case, determine the following parameters: most commonly used client, operating system, as well as the most popular password, username

and password/username pair, if credentials were recorded by the honeypot. The top attack week day and top attack hour for every country have to be identified as well. The findings comprise a table listing five countries for each honeypot, and a total of top five of the aforementioned parameters, one for each of the top countries.

To spare the reader the necessity of returning to the beginning of this report, here is a refresher of the set-up of the studied honeynet.

FIGURE 11.1: Honeypots and Their System IPs.

IP	Honeypot Name
192.168.10.2	Dionaea + Cowrie
192.168.10.3	Wordpot
192.168.10.4	Elastichoney + Shockpot Sinkhole
192.168.10.5	Cowrie + Shockpot Sinkhole
192.168.10.6	Elastichoney + Glastopf

Most of the log files corresponding to its respective honeypot contain some extra information that has not been included into the *AllTraffic.csv*. Thus, each of these files had to be examined separately in order for the necessary data to be extracted. The scripts that were used to analyze each log file have been compiled into the following two files: ***Option_O_Single_IP.py*** and ***Option_O_Double_IP.py***. Since some honeypots were twinned to two IPs (such as Cowrie, Elastichoney and Shockpot), the codes require some extra input from the user, and were thus separated into a different file for clarity's sake.

The results of the query are presented in the series of tables below. The fields that were not applicable to each honeypot (due to the information not being recorded or required for the interaction with the service in question) were not included into these tables. For every **one** country, **one** top OS, Client, credential pair (if applicable), day and hour are listed, resulting in **top five** entries indicated for each parameter in total.

ELASTICHONEY

Despite spanning over two hosts, this honeypot has only recorded a total of 64 entries. Thus, one of the IPs (192.168.10.4) has only registered interactions with three countries; and the other IP (192.168.10.6) with two. Elastichoney logs attempts to exploit remote execution code vulnerabilities in a service called Elasticsearch, and it does not collect any credentials. Hence, credentials were dropped from the table in **Figure 11.2**.

FIGURE 11.2: TOP RESULTS FOR ELASTICHONEY'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 OS	Top 5 Clients	Top 5 Days	Top 5 Hours
192.168.10.4	United States	Windows NT 6.1	Mozilla 5.0	Fri	11 PM, 12 AM
	China	Windows NT 5.0	Mozilla 4.0	Fri	3 PM
	Seychelles	Windows NT 6.1	Mozilla 5.0	Mon, Sat	10 PM, 12 AM
192.168.10.6	China	Windows NT 5.0	Mozilla 4.0	Tue	8 AM
	United States	Windows NT 6.1	Mozilla 5.0	Mon, Thu	10 PM, 11 PM

It looks like the only users interested in attacking RCE vulnerabilities operated out of China, Seychelles and the United States. The top pick in terms of OS for every country was either Windows NT 6.1 (United States and Seychelles), or Windows NT 5.0 (China). The favourite client was Mozilla, either version 4.0 (for China) or 5.0 (for Seychelles or the U.S.).

Friday and Monday were the days when most attacks occurred in two cases (for each day). Most of the attacks happened between 10 PM and midnight for USA and Seychelles. China, however, was the outlier, attacking 192.168.10.4 mostly at 3 PM, and 192.168.10.6 at 8 AM.

GLASTOPF

Glastopf is a Python-based honeypot application that has the ability to emulate a plethora of Web vulnerabilities. Located at 192.168.10.6, it also does not collect or log credentials; hence, they are absent from this table as well.

FIGURE 11.3: TOP RESULTS FOR GLASTOPF'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 OS	Top 5 Clients	Top 5 Days	Top 5 Hours
192.168.10.6	Ukraine	**Linux x86_64***	Mozilla 5.0	Fri	1 PM
	United States	Linux x86_64	Mozilla 5.0	Wed	6 AM
	Brazil	Windows NT 6.1	Mozilla 5.0	Fri	11 PM
	China	**Mac OS S 10.11***	Mozilla 5.0	Tue	1 PM
	France	Windows NT 5.0	Mozilla 5.0	Tue	11 AM

* Windows (of all versions) was the most commonly used OS for Ukraine and China. However, if specific versions are to be taken into consideration, then the OS and versions listed in the starred cells for these countries counted the most entries out of each individual uncovered user agent (a detailed breakdown may be found in the Excel spreadsheet ***Option_O_Entries.xlsx***, under the ***Glastopf*** tab.

In the case of Glastopf, more variety can be seen in the operating systems preferred by each country. China has opted for Mac OS X 10.11 in this case, France and Brazil used Windows NT, USA showed an overwhelming preference for Linux x86_64 (Windows remaining the second flavour of choice), while Ukraine's OS choices were almost evenly divided between Linux and Windows of assorted versions. However, Linux x86_64 and Linux i686 had the most entries for the individual user agent.

Mozilla 5.0 remained the favourite client for all of the top 5 countries. On an interesting side note, Kali Linux, a platform occasionally used by hackers, has Firefox or Iceweasel as browsers, both of which are based on Mozilla software.

Glastopf was most attacked on Fridays, Tuesdays and Wednesdays, most of the attacks taking place at 1 PM or 11 PM. It is curious to note that in this case as well as in the Elastichoney scenario, most attacks originating from China occurred on Tuesdays. Whereas other countries will show top activity at various days of the week, varying between IPs and honeypots, China will remain rather consistent in this respect.

WORDPOT

Wordpot is a honeypot designed to emulate WordPress, a PHP & MySQL-based tool intended for website and blog creation and content management.[8] Most of the attacks against this system, located at 192.168.10.3, were performed by Russia, and made up 2,013 entries out of the total of 2,029. All of the other countries, including an entity whose geolocation could not be identified, only registered under 5 interactions each. Wordpot does record credentials attempted to attain access to the system; thus, the table for this honeypot looks as follows:

FIGURE 11.4: TOP RESULTS FOR WORDPOT'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 Usernames	Top 5 Pwds	Top 5 User/Pwd Pairs	Top 5 OS	Top 5 Clients	Top 5 Days	Top 5 Hours
192.168.10.3	Russia	admin	[blank]	[blank/ blank]	Python-urllib/2.6	Python-urllib/2.6	Mon	7 AM
	"none"	[blank]	[blank]	[blank/ blank]	zgrab/0.x	Mozilla 5.0	Sun	3 AM
	Seychelles	[blank]	[blank]	[blank/ blank]	Windows NT 5.1	Mozilla 5.0	Sun	2 AM, 10 PM, 11 PM
	Netherlands	[blank]	[blank]	[blank/ blank]	[blank]	[blank]	Mon, Tue	4 PM, 9 PM
	United Kingdom	[blank]	[blank]	[blank/ blank]	[blank]	[blank]	Wed	1 PM, 10 PM

The Wordpot in this case allows its entry with both credential fields empty. The passwords attempted by a bot operating from Russia included 2012 unique entries, with the blank being the successful entry. At a later time, there was a single log-in to the host from Russia using blank entries as credentials. Presumably, this was done by the same user who had performed the 2012-event-long scan attack earlier.

The entities attempting to log into the host from other nations were able to get in on their first try. However, it is unclear whether they were at all related to the scan that originated from Russia.

In this case, it looks like software was used to conduct most of the attacks. The records show that the Russian scan used *Python-urllib/2.6*, a Python module for fetching Internet resources such as URLs or cookies. [9] Other agents included *zgrab/0.x* and *Ruby* (please refer to ***Option_O_Entries.xlsx*** for details).

Sundays and Mondays were the most popular days for attacking this honeypot. The top interaction times were not confined to a specific time period, and instead were scattered throughout the day. All Russian attacks, including the scan and the following log-in, happened at 7 AM (Toronto time) on a Monday.

SHOCKPOT

This honeypot is another Web application emulation. It exposes CVE-2014-6271 on IPs 192.168.10.4 and 192.168.10.5, which is a vulnerability that can allow miscreants to use susceptible Unix Bash shell versions to execute arbitrary commands. [10] This vulnerability is also known as ***Shellshock*** or ***Bashdoor***. [10] However, Shockpot does not record any credentials, hence they were excluded from Shockpot's table in **Figure 11.5**.

FIGURE 11.5: TOP RESULTS FOR SHOCKPOT'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 OS	Top 5 Clients	Top 5 Days	Top 5 Hours
192.168.10.4	United States	Windows NT 6.1	Mozilla 4.0	Wed	11 PM
	France	Windows NT 5.0	Mozilla 5.0	Sat	1 AM
	China	Mac OS X 10.11	Mozilla 5.0	Tue	6 AM
	Czechia	Windows NT 5.1	Mozilla 5.0	Sat	7 PM
	Brazil	Linux	Wget	Tue	8 PM
192.168.10.5	United States	Windows NT 6.1	Mozilla 4.0	Tue	6 AM
	France	Windows NT 5.0	Mozilla 5.0	Tue	2 AM
	China	Mac OS X 10.11	Mozilla 5.0	Tue, Sat	4 PM
	Seychelles	Windows NT 6.1	ZmEu	Fri	5 PM
	Taiwan	Googlebot 2.1	Mozilla 5.0	Wed	12 AM

In this case, Windows NT 5.0, 5.1 and 6.1 were preferred overwhelmingly by the U.S., France, Czechia and Seychelles, while China stuck to its guns – the guns in this case being Mac OS X 10.11.

In terms of clients, Mozilla 4.0 and 5.0 were once again among the top favourites. Brazil, however, opted for the Wget program available via Linux. The most entries originating from Seychelles were created by *ZmEu*, a vulnerability scanner and backdoor-generator of Romanian origin, whose name translates to "serpent". [11] Whereas the Googlebot, a web-scraping bot owned by Google, was responsible for most logs of Taiwanese origin.

The most attacks or scans of this honeypot, at both IPs, took place on Tuesday, with Wednesday and Saturday sharing the second place. The attack hours vary – the only one that shows any consistency in this situation is France, having interacted with 192.168.10.4 mostly at 1 AM, and with 192.168.10.5 at 2 AM.

DIONAEA

Dionaea is a honeypot that reveals such services as FTP, SIP, TFTP, SFTP, SSH, HTTP, MSSQL and SMB. However, it does not record the operating systems and clients of the perpetrators; nor does it log any credentials the threat agents may have tried. Thus, the only information gathered by this decoy host in this case includes the top five countries, and their respective preferred day and hour of attack.

FIGURE 11.6: TOP RESULTS FOR DIONAEA'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 Days	Top 5 Hours
192.168.10.2	United States	Sunday	12 PM
	Vietnam	Monday	10 PM
	China	Saturday	11 PM
	Russia	Monday	1 PM
	Germany	Thursday	12 AM

COWRIE

Usernames and Passwords

Cowrie is an SSH honeypot that records brute force attacks; hence, it has recorded both the credential and user agent information. The first section dedicated to Cowrie covers the top five countries and the most popular credentials for these top nations.

It should be noted that in this case, Usernames and Passwords indicated **represent the overall top 5 for all the five top countries combined,** as they were for the most part very similar for all of the top countries.

The table examining these credentials, as well as the analysis of popular password and username trends uncovered by Cowrie are presented below.

FIGURE 11.7: TOP CREDENTIALS FOR COWRIE'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 Usernames	Top 5 Passwords	Top 5 User/Pass Pairs
192.168.10.2	China	root	**welcome***	**root/password***
	Hong Kong	admin	**password***	**admin/admin***
	United States	test	**123456***	**root/000000***
	France	user	**000000***	**root/123456***
	Russia	support	**admin***	**root/welcome***
192.168.10.5	China	root	**password***	**root/password***
	Russia	admin	**000000***	**root/000000***
	United States	support	**123456***	**root/123456***
	India	user	**toor***	**root/toor***
	Canada	test	**centos***	**root/centos***

* The author of this paper realizes that simply finding the top five usernames and passwords may not be informative enough; whereas observing and outlining the general trends in passwords and username/password pairs may be more helpful in terms of research. For example, a "top five" search may return the word "**password**" as the most popular password choice; but the trend shows that other combinations of characters used to make up the same word (**p455w0rd**, **p@$$w0rd**, **Pasw0rd**, **passw0rd123**, etc.) are equally unsafe.

Thus, the entries were sorted, and all the entries generated by the top five countries (which made up the majority of the entries for Cowrie) were manually examined for the purpose of finding and exposing these popular trends.

There were several tendencies that were common to all five observed nations; and certain patterns that were only encountered in interactions with some of them.

The most often-attacked usernames are **root** and **admin.** Some other common choices include **test, guest, user, support, debian, ubuntu, ubnt** and **usuario.** The username **pi** is also encountered quite frequently, usually in combination with the password "**raspberry**" or "**banana**". Common English words or proper nouns, such as Alex, George, Emily, Alice, Chris, etc. are often tried as well, usually in all lowercase. Hence, it is a good idea for system administrators to generate usernames following the formula of initial of first name + full last name, such as "**nshevcun**" as opposed to "**natalia**" – in such a case, they would be far more difficult for a threat agent to guess.

Some of the most recurring passwords include **welcome, password, admin, toor, centos** and various numeric sequences, usually in increasing order, such as **12345** or **123456**; or repeating numerals, such as **000000**, **111111**, **666666** or **777777**. As mentioned above, these could be combined, to create such variations of the common words as "**welc0me", "welcomewelcome",**

"mypassword", "notmypassword", "p455w0rd123", "admin1", "admin123" or **"centos123"**. **Qwerty, changeme** and **motorola** were frequently seen as well. Some variations of "**qwerty**" included options like "**qwert123**" or "**qweqwe123**". Another recurring letter combination was "**QAZ**", often with "**123**" attached to it. Interestingly, it was far more common to see the numbers in increasing order starting from "1" and at the end of the password rather than at the beginning or breaking up the word in the middle. A variation of **123456** was a sequence that results when one types these numbers while pressing "Shift" on a regular keyboard: "**!@#$%^**". The special characters most often encountered in passwords seem to be "**!**" and "**@**".

The most common length for passwords tried was between 4 and 8 characters; however, some entries were much longer than that. While Chinese attackers most often tried password lengths that were within range, many of the passwords tried by attackers from Russia and USA were far longer and more complex, often involving random combinations of capital and lowercase letters, numbers and special characters.

There was also a peculiar series of passwords that came up in the scans of all top countries for both Cowrie IPs. These include "**anko", "waldo", "openlec", "xmhdipc", "dreambox", "ubnt", "seiko2005", "uClinux", "nosoup4u",** and "**7ujMko0admin", "wubao", "jiamima"** and **"raspberrypi".**

Another interesting trend, favoured by those who clearly belong to a very "mature" part of the computer-user crowd, is using English curse words, genitalia names or other such words of suggestive nature as passwords. Since hackers know all about anatomy and what can be done with all the body parts, one must not be deluded that one's information will be secure with a lewd word for a passkey.

One more noteworthy pattern was the use of season names: **Spring, Summer, Autumn** and **Winter**, often with a year numeral attached to the word at the end, as follows: **Spring2012** or **Summer13**. This somewhat makes sense, considering that the system attacked belongs to a university – university tends to be seasonally (Fall Semester, Winter Semester), and perhaps default passwords for some accounts could very well look like that, especially for any guest user receiving credentials at a particular point of the year.

It was interesting to note that among the entries for one country, it was possible to encounter a dictionary that had clearly been generated in another. For example, among the entries for France, a sequence of the following passwords appeared that were of Russian origin: **kilyakov, baklanov, nabiyullin, dima.k, andrey, zlygostev, denis, ablaimov, slava, tarelov, regentov, kharitonova, golubev, roman, homenko**. Another humorous example was "**KurVozil?NaVokzal!**" (translating to "Did you drive the chickens? Yes, to the train station!"). Some other typically

Russian entries included **"dima", "stalin", "anton", "mazafaka", "kikimora", "vysotski", "yurist", "zernova".**

At other times, some quite clearly Chinese dictionaries were used by other countries. These were characterized by a plenitude of passwords including the words "**china**", "**ilovechina**", "**beijing", "shanghai", "huawei**", "**nihao**" (meaning "hello" in Chinese), or various Chinese names in the combination. This could be an indicator of hackers operating remotely through a host located in another country.

An interesting trend among Romanians was to use phrases with entire words spelled out, such as **floiubestepeina** (meaning "Flo loves Ina" – this one was seen several times, often with a long alphanumeric sequence at the end), **NU.E.BUN** ("not good"), **numergeniciodata** ("never goes"), and even an entire love story: **teiubescdartunumaiubestiasacahaidesaterminam** (translating literally to "I love you, but you don't love me, so let's break up").

Some popular username/password combinations included **root/toor, support/support, usuario/usuario, admin/admin, admin/admin1,** and **admin/admin123**. These were common for all of the countries. In addition to that, **root** or **admin** were often found in combination with any of the passwords in the aforementioned trends. The most popular username, especially among the threat agents attacking from China, was **root**.

Another country-specific trend involved the amount of passwords tried per attack. Threat agents operating from China, for example, would attempt several passwords per session, while those from Hong Kong would often only venture one username/password pair per attack.

This information could be very useful to anybody looking for a password to protect their sensitive data. Knowing these trends, people should become more aware about the sort of words often encountered in typical hacker dictionaries, and will likely avoid them when picking an important alphanumeric key.

Popular User Agents and Dates

The second part dedicated to the Cowrie honeypot deals with the user agents most commonly registered for the top five attacking countries at each IP, and their preferred attack times. In this case, one top Client, Day and Hour are listed **for every one top country,** resulting in top 5 parameters in total.

These are listed in **Figure 11.8** below.

FIGURE 11.8: TOP USER AGENTS AND TIMES FOR COWRIE'S TOP 5 COUNTRIES

IP	Top 5 Countries	Top 5 OS	Top 5 Clients	Top 5 Days	Top 5 Hours
192.168.10.2	China	n/a	SSH-2.0-PUTTY	Fri	2 PM
	Hong Kong	n/a	SSH-2.0-Go	Thu	4 PM
	United States	n/a	SSH-2.0-libssh2_1.4.3	Wed	2 PM
	France	n/a	SSH-2.0-libssh-0.1	Wed	8 PM
	Russia	n/a	SSH-2.0-sshlib-0.1	Sat	3 AM
192.168.10.5	China	n/a	SSH-2.0-PUTTY	Sat	12 PM
	Russia	n/a	SSH-2.0-Go	Tue	8 AM
	United States	n/a	SSH-2.0-libssh2_1.4.3	Sun	2 AM
	India	n/a	SSH-2.0-MEDUSA_1.0	Sat	11 AM
	Canada	n/a	SSH-2.0-Go	Wed	12 AM

Since Cowrie only records SSH interactions, the OS for the threat agents were unavailable in this case, and the top versions of SSH used were considered as clients in this case.

Overview of Other Interesting General Trends for All Honeypots

These trends mostly revolve around the top hours and weekdays for every one of the top five countries. The reason why the top hours and weekdays for this segment are best to be studied by country rather than by honeypot is because various nations and cultures' unique traits may find themselves reflected in the peak (or their slowest) hours and days of operation.

For example, let us consider the data collected by Dionaea, as this honeypot has collected the most abundant amounts of data out of all the honeypots studied (over 1,000,000 logs). Let us consider attack days alone, as hours would need to be adjusted for every country, since the honeynet recorded hours in Toronto time. In addition, hackers or software could be operating around the clock from any country; thus, the focus for this part of the analysis will be just on the attack days.

It shows that the top day of operation for the malicious users of the United States falls on the Sundays, with the total attack numbers nearing 80,000 in stark contrast to circa 36,500 attacks on Mondays and roughly around 17,000 attacks falling on all other days of the week. The reason why this is happening may lie in the normal working hours for an average IT worker in the United States. Unlike Canada, USA does not have a Labour Code that would regulate the working day and restrict the employers from firing their subordinates for refusing to work overtime. Thus, many companies in USA make their employees work large shifts throughout the work week, leaving

them little time for leisure hacking. However, since on Sunday most of these people are off work and well-rested, they have the opportunity to freely engage in a little bit of hacking "on the side".

In contrast to USA, China's attacks on the same Dionaea honeypot are distributed rather evenly throughout the week, with around 16,000 – 17,000 of attacks per day. This could mean one of the following things: that the Chinese workers are busy all week long, and only have as much time to dedicate to hacking every day as Americans do at the end of a workday; or that these Chinese threat actors are working for the government and only conducting these attacks in their work time. Whatever the case, it is interesting to see just how evenly distributed the numbers are throughout the week for China as opposed to the US. Vietnam has a similar attack distribution to China, with slightly more attacks falling on Sunday and Monday rather than other weekdays (circa 20,000 – 21,000).

With Russia, another pattern of attacks emerges. The most attacks have been registered on Monday: about 23,500 attacks. On Tuesday, this number drops almost by half, showing about 14,000 attacks. Wednesday, Thursday and the weekend are the least busy, with around 8,600 – 9,600 attacks taking place per day, and spiking up to slightly over 11,300 attacks on Friday. Based on observation, the author happens to know the Russian working ethic quite well, and, based on that, the following picture may be true.

Russians consider Monday a "difficult day", so on this day they are far more likely to slack off at work, and do things that have little to do with their immediate task at the workplace, take longer lunch breaks, skip class, and turn in earlier for the night (and the number of hack attacks is up).

Tuesday is considered "the second most difficult" day of the week (as the day after Monday); and while some may still do things unrelated to their direct responsibilities at work, others are getting more into their working headspace (and the amount of attacks drops drastically).

Wednesday and Thursday are "good work days", so during these days, an average Russian is far more likely to be honestly working on the task he or she is actually paid to do. Incidentally, these are also the days with the least numbers of malicious interactions with Dionaea.

Friday is "the day before Saturday", so many again start caring less about work, and prefer to distract themselves with other things in anticipation of the weekend (and during these days, the number of attacks spikes up again). And, as Russians are far more social than many of the Western nationals (such as Americans, Canadians or the British), many prefer to spend the weekend with friends rather than alone in front of a computer screen – and the numbers of Dionaea interaction fall again to Tuesday and Wednesday levels. This may also be an indication of the fact that many of such attacks are committed by these individuals of their own accord rather than as direct work for the government or any other organization. However, there is also a possibility that the

government attacks are happening through hosts in other countries rather than the Russian ones directly, of course.

This may seem like an amusing interpretation, and of course many other factors must be taken into consideration as well. However, since behavioural psychology is also often taken into consideration when analyzing the attack patterns of threat actors, this hypothesis may provide a very probable explanation for such rises and drops in attacks.

As for most popular user agents in other countries, it was interesting to note that Windows NT was among the most popular OS, while Mozilla was by far the most preferred client.

Knowing all of this information and taking all this analysis into consideration may greatly help cybersecurity agents when determining who is attacking them, which nation the perpetrator may belong to, whether he or she is doing this as part of their work or not; and, most importantly, what steps should be taken in order to protect the data from such threat agents.

CONCLUSION

The aim of this paper is to demonstrate the effectiveness of honeypots in collecting data generated by malicious cyberinteractions with various virtual network services for the purposes of research. To provide a practical illustration of the application of honeypots, the author of this paper used Python to analyze and interpret a data sample of close to two million entries gathered by the university honeynet with hosts located in five different countries over the course of a time period spanning from September 14, 2016 to September 30, 2017. Intelligence gathered in the process of such studies helps uncover trends in the relationship between the origins of threat agents and their preferred *modus operandi,* as well as possible motivations behind their attacks. Knowing these trends and properly interpreting this information turns raw data into tactical, strategic and operational intelligence that will aid the cybersecurity specialists in developing and maintaining proper defence methods against spies and assorted cybercriminals around the globe.

REFERENCES

[1] https://apps.dtic.mil/dtic/tr/fulltext/u2/1046884.pdf

[2] https://www.csoonline.com/article/3287653/what-is-the-tor-browser-how-it-works-and-how-it-can-help-you-protect-your-identity-online.html

[3] https://www.howtogeek.com/133680/htg-explains-what-is-a-vpn/

[4] https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/

[5] https://www.iplocation.net/index.php

[6] https://www.juniper.net/documentation/en_US/nsm2012.2/topics/concept/security-service-port-number-ssh-telnet-connection-nsm-overview.html

[7] https://www.torproject.org/docs/faq-abuse.html.en

[8] https://kinsta.com/knowledgebase/what-is-wordpress/

[9] https://docs.python.org/3.2/howto/urllib2.html

[10] https://en.wikipedia.org/wiki/Shellshock_(software_bug)

[11] https://en.wikipedia.org/wiki/ZmEu_(vulnerability_scanner)

CKDF150 - Final Report - Natalia Shevcun.docx

ORIGINALITY REPORT

3%	2%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Ryerson University Student Paper	1%
2	Submitted to Centennial College Student Paper	<1%
3	www.asmokoskinen.net Internet Source	<1%
4	Submitted to Brigham Young University Student Paper	<1%
5	ce-online.ryerson.ca Internet Source	<1%
6	Submitted to University of Greenwich Student Paper	<1%
7	detector.dmolsen.com Internet Source	<1%
8	J C Gunn. "An objective evaluation of geriatric ward meetings.", Journal of Neurology, Neurosurgery & Psychiatry, 1968 Publication	<1%

9	Submitted to Institute of Technology Blanchardstown Student Paper	<1%
10	lib.dr.iastate.edu Internet Source	<1%
11	Tobias Ihde. "Dynamic Alliance Auctions", Springer Nature, 2004 Publication	<1%
12	www.icrealtime.com Internet Source	<1%
13	Submitted to Higher Education Commission Pakistan Student Paper	<1%
14	Submitted to Jakarta International School Student Paper	<1%
15	anime-candy.ru Internet Source	<1%
16	Submitted to University of St. Gallen Student Paper	<1%
17	www.kanowaard.nl Internet Source	<1%
18	www.purc.com.gh Internet Source	<1%
19	Submitted to Arab Open University	

	Student Paper	<1%
20	www.southbayhearing.com Internet Source	<1%
21	www.clearlycontacts.com.au Internet Source	<1%
22	getd.libs.uga.edu Internet Source	<1%
23	gctv.asia Internet Source	<1%
24	www.evergreenfair.org Internet Source	<1%
25	instagloss.com Internet Source	<1%
26	www.bikestation.org Internet Source	<1%
27	www.computer-geek.net Internet Source	<1%
28	www.delaat.net Internet Source	<1%
29	www.prospect.sa.gov.au Internet Source	<1%
30	Ron Houtrouw, Pete Reiland. "Rotating Shifts	<1%

Satisfy Workers and Management", Opflow, 1991
Publication

31 Zakas, Nicholas C.. "Client Detection", Professional Javascript® for Web Developers, 2015. <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		