POLITECNICO DI TORINO

Master's Degree in Communication and computer network engineering

Master's Degree Thesis

Analysis of application layer attacks on honeypot logs

Supervisors	Candidate
Prof. Marco MELLIA	Chiara DE NOVELLIS

July 2022

Summary

Nowadays, the urge for insights into travelling suspicious traffic on the network is becoming a main issue in the cyber security field. Taking advantage of a monitoring infrastructure already installed comprising passive and active sensors, such as Darknets and honeypots, we focus on a specific one, the Heralding honeypot. Honeypots are becoming a classical tool in cyber security policies since they can be useful to monitor and identify new attacks. More specifically, they are placed on the internet and built ad-hoc to be attacked, usually miming the behaviour of an unsafe system and so interacting with the incoming traffic. A comprehensive analysis of data collected by the Heralding honeypot gives the possibility to observe closer new arising botnets or DDoS attacks. This work aims to provide an insight into the raw information collected by the Heralding, focusing on origin of attacks, targeted services, and frequency of attempts.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Starting from the personal to working sphere, from academic to service industries, everything is planned to further become interconnected with unimaginable amounts of traffic travelling anywhere in the world. This scenario is also supported by the upcoming releases of new cellular network generation: the 5G and the design of 6G that has recently begun. This is merely because these two innovative cellular network generation will help the human kind to be more and more interconnected.

Indeed,according to the Cisco Annual Internet Report, by the year 2023 the number of devices connected to IP networks will be more than three times the global population reaching a number of 29.3 billion networked devices. [1]. Right now the world is facing the fourth industrial revolution scenario which combined with the global pandemic is helping to further

accelerate technological innovations. Thanks to the diffusion of the Internet of Things (IoT) and connected devices, new opportunities and attack vectors comes to black hat community and malware developers.
Moreover, nowadays cyber security is playing a crucial role because every system is built on computer systems and any kind of damage will cost a huge economic loss. Even indirect attacks that does not mean to steal money can have huge economic consequences.

Furthermore, Cyber threats have grown fast enough. As state by Kapersky, only in the last trimester of 2021 there has been seen the arise of several new DDOS. [2] Therefore, is crucial that the countermeasures evolve quickly enough and are being updated as soon as a vulnerability is being disclosed.

Despite big enhancement on the classic security tools, such as antivirus, dynamic Firewalls and Intrusion detection/prevention systems, we need to have more insights in enemies behaviour and monitor if some new botnets has been installed. Some previous literature propose to deploy some monitoring tools, as Dark nets and honeypots, that allows a collection of that traffic in the internet that can be defined as "unsolicited", that could mask some malicious activities (e.g. clients replying to IP spoofed servers which never received an answer).

As already mentioned, all this came in the optic that cyber-security measures has to evolve together with malware evolution and this could be a very hard task to solve if we did not have a full comprehensions and an

updated versions of new malware. To this effort, researchers and the whole white hat community combined their strengths together and still try to develop new strategies.

Therefore, in this work we describe a monitoring infrastructure able to combine "the best of the two worlds" of the two complementary features darknet and honeypots, placed in the internet and configured to collect unsolicited traffic reaching the sensors. In particular, we focus on what a single honeypot at layer 7 has been able to store and if it is possible to use this information to automatically individuate an attack pattern, like botnets.

1.1 Motivation

Under Cisco annual report's heading "security analysis", we can extrapolate how it is important to secure every single devices on the internet that, by 2023, are expected to be 3.6 networked devices per capita thus 29.3 billion. Since connected home, car and smart cities applications are accelerating the Internet of Things growth in our lives, the related internet users experience, now more than ever, is expected to be always available and secure. Together with these technological growing trends, the last several years, have been seen the most diversified and eventful period for security threats : Distributed Denial-of-Service (DDoS), ransomware, Advanced Persistent Threats (APTs), viruses, worms, malware, spyware, botnets, spam, spoofing, phishing are some

examples. No wonder that an higher incidence of cyber attacks have been recorded since this availability of dangerous tools combined with expansion of attack surface. More specifically, DDoS attack occurring when multiple systems flood the bandwidth or resources of a targeted system (one or more wen servers) represent the dominant threat observed by most service providers, with an estimated doubling amount of attacks to 15.4 million by 2023 globally.

It is clear that classical approaches such as firewalls, intrusion detection or prevention systems that can still play a crucial role for businesses and personal devices security, are not enough to support the M2M environment related to IoT contest: here great amount of traffic exchanged between autonomous systems cannot be anymore manually monitored and inspected.

Early in 2003, Spritzner [3] proposed that to understand better malware and its variants it was necessary to capture data to provide a major understanding of malware developers motives and the behavior of the corresponding tools. The specific role of a honeypot is to uncover attack behaviour with longitudinal deployments, capturing large data set for retrospective analysis. Bearing in mind this, in this work we exploit an already installed infrastructure comprising a set of honeypots, the Tpot project [4], placed at application layer to respond unsolicited traffic hitting the adjacent dark net. We select the Heralding honeypot [5] and collect information of its logs.

1.2 Research questions

In this thesis we want to investigate if the senders have malicious intention by observing their behavior from the collected information related to traffic hitting the address space and interacting with Heralding. Due to the nature of *low-interaction* honeypot under analysis, we focus on dependencies and distributions within the observed interaction and to this extent we proposed and investigated different hypothesis. First, the temporal and spatial behavior of the activities against the deployed honeypot is investigated so that we can start from the following two naive hypothesis:

- Are there any significant temporal and/or spatial dependencies in the activity observed? (1)
- Second, since the important role of a protocol playing in information security, are observed any significant dependencies on the protocols ? (2)
- Often the protocol could tell us more about the actions a successful exploitation could lead to, in this case for example through the authentication mechanisms. (3)

Going on, a huge amount of login attempts was captured to be a single honeypot but with a limited number of dictionaries and default credentials,

we answer to this question:

- How many different patterns are involved ? (4)
- For similar reason, senders are efficient with their resources ? (5)

1.3 Methodology

Our approach is based on three different steps:

1. Analysis of existing state-of-the-art honeypots: at this first stage we aim to identify the honeypot with richest collected information, most compliant to our characterization scope.
2. Organizations of chosen honeypot logs: in this section we analyze at high level the information collected in the deployment, starting to investigate the initial hypothesis, such as if there are some dependencies and patterns that could associated to suspicious activities.
3. Analysis of collected traffic: here always at application layer, we try to investigate more in deep the started analysis with a major focus on only flows that includes collected credentials.

After a brief introduction of the honeypots and their state of the art (chapter 2), we present the evaluation of data captured (chapter 3) with

a base characterizations. In chapter 4 are presented the results while the conclusion and future works are described in the chapter 5.

Chapter 2

Related Works

2.1 Honeypot

Honeypots are decoy computer resources whose values lies in being probed, attacked or compromised [6]. Since their first ancestors, honeypots have been changed in order to meet the evolving landscape of cyber threats. Honeypots early versions had low interaction capability: they were simple Internet services detecting the presence of an attacker. However, the developing of new technologies helps with the mid and high interaction honeypots (MiHPs and HiHPs, respectively) implementations allowing more interaction with an attacker.

The honeypot design, configuration and operation require some consideration since regarded as dynamic and real-time analytical tools have certain

limitations: if an attacker is able to discover that he is interacting with a decoy he will compromised the system and this ends up to inadvertently take part in further attacks. Also automated malware employs honeypot detection mechanisms within its code.

For this motivations, some researches are necessary about the importance of development and deployment of a new framework for adaptive and agile honeypots, with the property of responding by what has been learnt during its interactions with the attacks [7]. However, emerging networking technologies have opened up new directions in honeypot deployment. Real time visualization of global attacks is also provided by Deutsche Telekom Honeypot project [7]. This honeypot-development community produced T-Pot used for capturing and visualizing attacks on multiple well-known honeypots. The field of honeypot research consist in two main pillars: the development of honeypot and its deployment and the analysis of the acquired log data in a structured manner. Through the analysis of a specific honeypot log, the Heralding one, we try, here, to define if there are enough collected information to build a pattern recognition algorithm.

2.2 Honeypots usage

Honeypots themselves has no production value or authorized activities.Thus, the traffic reaching them and therefore their connections are suspicious by

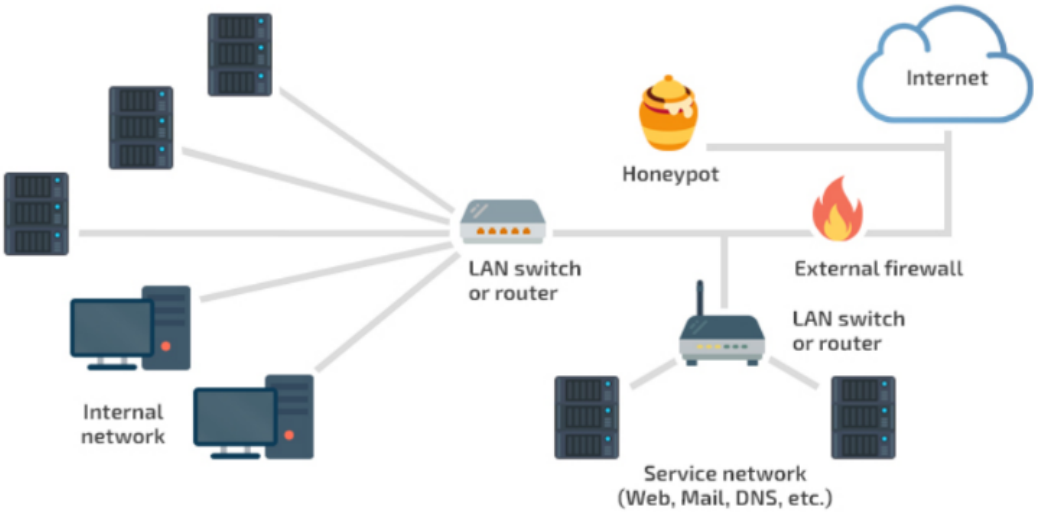

Figure 2.1: Network example infrastructure with a honeypot

nature: they can be identified more likely as probes, scans or attacks. They complement the traditional detection mechanisms since they are able to spot zero days attacks and to give insights on attacker action and motivation. Despite more traditional tools with updated features such as Intrusion Detection Systems, IDS, or Dynamic Firewalls, the traffic collected by an honeypot is not influenced by false positive and negatives : although the collected traffic may be little (compared to a real production system), the majority of traffic may be malicious. More in general, an honeypot objective is either to distract attackers from their actual target or to gather information about the attackers and attack patterns such as the set of popular target hosts and the frequency of requests-responses.

Classified by the areas of operation, security concepts are the following [8]:

- Prevention is any kind of undertaking which discourage intruders and makes breaches in systems unfeasible
- Detection process aims to identify the presence of actions that harm the system confidentiality, integrity and availability.
- Reaction describes the execution of reactive measures after something harmful actions has been detected.

Nowadays, distinction between these concepts is difficult since the current security solutions combine several concepts to minimize the individual trade-off. Talking about prevention, using a honeypot is not a very add-on since it is not featured to protect against security breaches. Firewalls are instead the most suited for prevention : they hide services and block communication on unused ports or from suspicious IP-address-ranges, so that intruders are kept or nor finding any services to communicate with. However, the best prevention is achieved by cyber security standard and practises, such as continuous updating and patch policies. Where the honeypot instead add extensive value is in detection area operations, they can speed up the detection process by simplifying it. By definition, the honeypot have no production activity, meaning that all connections to the honeypot are suspect by nature and detect an unauthorized probe, scan or attack is possible with

almost no false positive, where instead in the case of an Intrusion Detection System isolate suspicious traffic is not feasible without a huge false positive. However, the usage of an honeypot helps in the reaction to attack. A honeypot log analysis can be more accurate since not only data collected by honeypot are not mingled with production activities but also it can be taken off-line at will, to allow a major and full forensic analysis. Having the insight allows to clean the production systems and understand the exploit, being a first step to patch the corresponding vulnerabilities.

A first division of honeypots [6] is based on their field operation: one used for production system and one for research aim. Whereas the in environment production honeypots come with a trade-off between ease of operation and the quantity of collected information, research honeypots provide comprehensive information helping organizations and network forensics scientists to understand the motives, behaviour, tools, and organization of the black hat community. A research organization might be interested in studying the details of the attack, that is which dictionaries were used to guess the passwords to create a recommendation on password strength. In the case of production honeypot the objective is mainly to distract the attackers from their actual target to achieve higher security level within the network of a company, while in case of research honeypot we want to propose general countermeasure against threats founded after the analysis of information collected about attacks. Also, another famous classification of honeypots is

done by considering the level of interaction, going from low to mid up to high level on one hand and on the other hand the direction , if it is server or client.Thus, it is important to notice that despite other security applications,such as Anti Viruses IDS or IPS, honeypots are first being exposed on purpose to attack and secondly, not build with the purpose of addressing a single problem but rather a generic one: depending on how and where they are deployed (e.g. on a router, scripts emulating specific services, virtual or a standard physics system) the collected information can change.

2.3 Honeypot systems

In 2006 Nawrocki et al. [4] present an extensive overview on honeypots software and methodologies to analyse the related data, highlighting that, on the one hand, the most famous honeypot software are Cowrie [9](previously Kippo), Dionea [10] and Honeytrap [11], while, on the other hand, the most famous long- terms honeypot projects are the Honeynet Project (1999) and the T-Pot Project, developed from Telekom-Fruhwarnsystem (2012). These two long-term projects present a multi-honeypot platform where low-interaction honeypot tools cooperate intending to be able to reply to more protocols. The honeypots previously mentioned are nowadays active and under maintenance. A low-interaction honeypot ecosystem able to emulate IoT devices to understand the attack patterns to IoT systems has been proposed

by Tabar et al. [12] in 2020. Indeed, they emphasize that IoT attacks have an increase of 600 % compared to 2016 and thus it is necessary to develop a smarter honeypot able to deal with IoT devices. Since most honeypots are able only to receive passively attacks [4], and they are not able to correctly identify and distinguish the various attack data and scenario,Fan et al.[13] in 2019 propose an efficient honeypot architecture, called HoneyDOC, based on Software Defined Networking (SDN), consisting of three modules, i.e. Decoy, Orchestrator and Captor, to enable all-round design for high-quality attack data capture. In the following, a brief introduction to the already cited honeypots is provided:

- **Cowrie** is a low-interaction honeypot being the evolution of the no longer supported medium-interaction honeypot Kippo. It can handle SSH and Telnet Traffic to observe attacker behaviour. It provides a fake file system, a fake SSH shell and is also able to .In [12] a new extension of Cowrie is exposed: the capacity to deal with IoT traffic, since the exponential increase of IoT devices with many still using telnet and SSH protocols.

- **Dionaea**, is a low-interaction honeypot released in 2013, written in Python that supports up to 14 different protocols (HTTP, MYSQL, SMB, MSSQL, FTP, MQTT). It emulates various vulnerable protocols commonly found in a Windows system. The main goal of Dionaea is to

capture malware and worms [12].

- **Honeytrap**, is an open-source framework for running, monitoring and managing honeypots. It can be used to deploy complex honeypot architectures or only to deploy a single server. Furthermore, it is a flexible architecture: it is possible to listen to all ports for detecting threats and collecting information, or listen to a specific port and give predefined answers.

- **Honeynet**, according to [14] it is a security-research organization committed to learning the techniques, strategies, and motivations of the black-hat community and then disseminating any lessons discovered. The group consists of professionals in international security who donate their time and money to set up honeynets that are intended to be attacked. The team then examines the data gathered as a result of these attacks.

- **Tpot**, is a platform that combines all of the honeypots into one bundle along with some informative reporting. Dockerized versions of 19 honeypots, such as Cowrie, Dionaea, Heralding, and Honey-trap, are included. The user is given a front-end view of all attacks against each service by the T-Pot framework, which collects all the logs from each honeypot and centralizes them into an elastic stack. Malware samples are also recorded, allowing for future analysis of the attacks. Fig. 4.10

shows a schematic representation of the T-Pot[15]. Additionally, T-Pot offers a straightforward dashboard user interface based on Kibana [16], a data visualization dashboard for Elasticsearch[17] and a Java-based distributed search engine with an HTTP web interface.

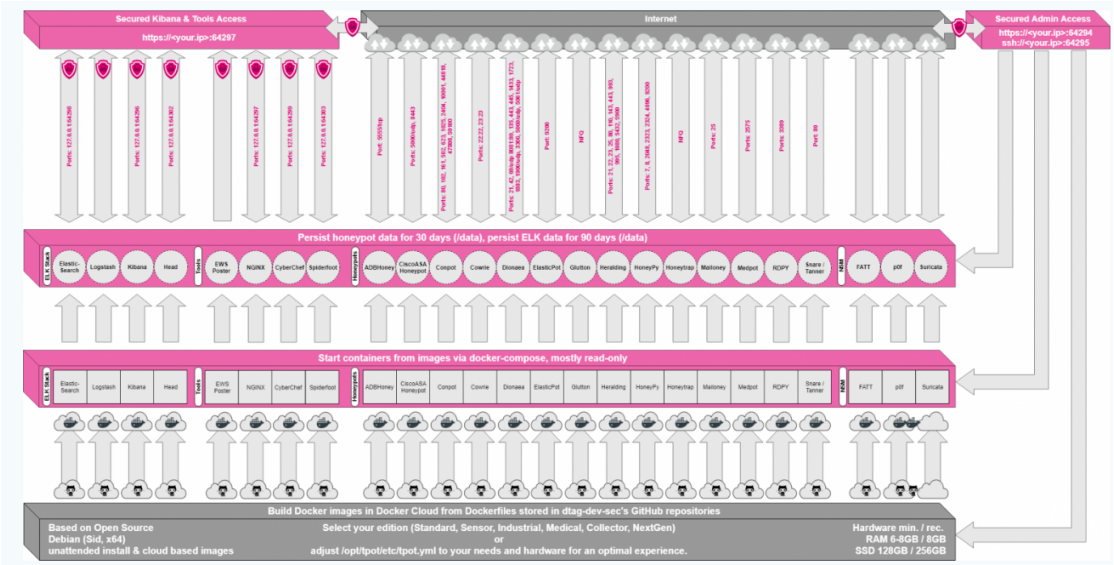

Figure 2.2: T-pot general Framework

Chapter 3

Methodology

3.1 Infrastructure

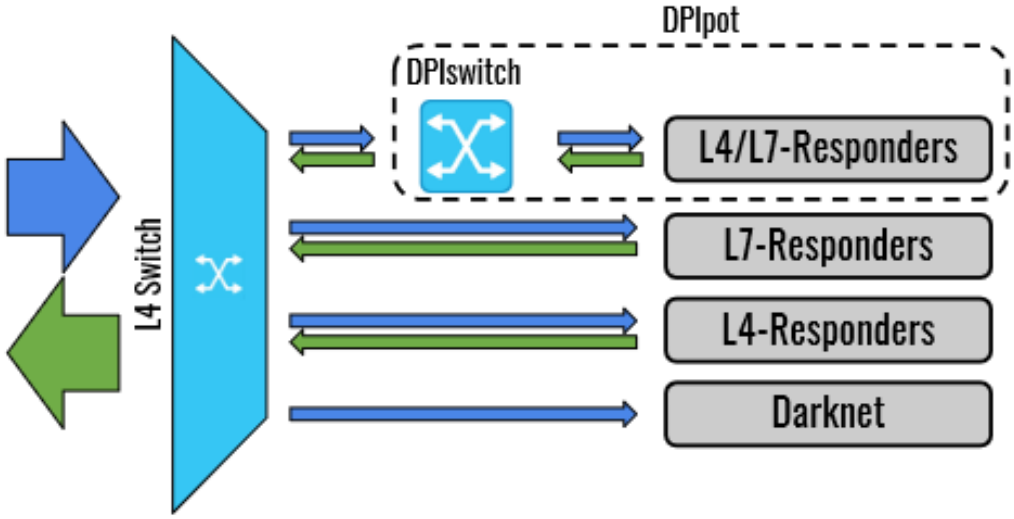

Figure 3.1: Infrastructure overview

Our reference infrastructure is the augmented monitoring one which

combine darknets and active responders at different layers and it is described in detail in the paperwork [18] (fig. 3.1) As unsolicited traffic reaches the dedicated address space is reroute towards one of the four deployments corresponding to different interactivity levels: Darknet, L4-resp, L7-resp and DPIpot. What we are going to study in deep is the outcome of what is inside the traffic reaching one of the L7 responders. Their deployment has been settle relying on honeypots distributed and organized by the Tpot project acting as backend, the honeypots settlement has been activated in way to handle a range of popular application protocols. Tpot offers a collection of third-party low-interaction honeypots, i.e., programs crafted to simulate a vulnerable service communicating over an layer-7 protocol. As the chosen Heralding Honeypot, most of the L7-responders offer login interfaces [Heralding] registering the brute force attempts against services (e.g., VNC, PostgreSQL, POP, and IMAP). Others L7-responders rely on more sophisticated systems but we will not be going further in deep. They are deployed on standard TCP ports of the given services, e.g., the HTTPs honeypot deployed on port TCP/443 whereas Virtual Network Computing, VNC, the open-source remote controller, on default port TCP/5900.

Then, the honeypot interact with the traffic reaching it and let the potential attacker insert username and password in a login session, to access the specific application layer services . So, what the heralding honeypot do is to simply collect credentials inserted by potential attackers doing brute

force attempts. After the credentials has been inserted the communication is broken down by Heralding.

3.2 Heralding Honeypot

Since there are some honeypot inside the campus deployment that haven't already been inspected we select the Heralding honeypot, and start our high-level analysis. Among the others, **Heralding** is a low interaction honeypot that allows us to emulate several protocols with a credentials user interface. Once the attacker will attempt to login to the system all credentials will be captured in a log file. Among the supported protocols we have: ftp, telnet, ssh, http,https, pop3, pop3s, imap imaps, smtp, vnc, postgreSQL and socks5. This huge number would be a good advantage to our extent since can allow to have more insight not only on who attacks at application layer but also how they distributed their attacks among application protocols within the same honeypot. The considered honeypot, operate on TCP well-known ports, so the application protocol service is easily deducible. A brief introduction to the most popular protocol provided by our Honeypot login interface, is proposed to the previously mentioned honeypots:

- VNC is, Virtual Network Computing (VNC) is a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer. It transmits the keyboard and

mouse input from one computer to another, relaying the graphical-screen updates, over a network. The authentication login can or not being configured by insertion of username and password (whose length can be no more than eight characters).

Going more in details, heralding logs relevant data are organized in three main files, one in a json format and two in a csv files:

log_session.json: here we can find any available information for a given activity, including timestamp, authentication attempts and protocol specific information (on an auxiliary data record). The log entry for a specific session will appear in the log file only after the session has ended;

log_auth.csv:contains entries for all authentication attempts to the honeypot. The attempts where it is possible to log a username and password. Also here, it is important to bearing in mind that the log entries will appear in this file as soon as the password has been transmitted

log_session.csv: contains entries for all connections to the honeypot.
These three kind of format are collected as the activity happen in the docker environment dedicated to our honeypot day-by-day, so per each day in our testing period we have three different log types. Then, we retrieve the complete activities and put it on single huge database in Python, keeping only the useful columns for our analysis. We work on a full join of the three main different logs files. The columns we select to do our further evaluation are

the following : *authentication_id, timestamp, date, session_id, source_ip, source_port, destination_ip, destination_port, protocol, username* and *password.* . The high-level application layer analysis is done by unique sessions opened: each record is an activities identified by a unique timestamp, authentication id and a session id. From a logical stand point it is worth to mentioned that an "interaction" can be identify by the tuple (*source_ip, source_port, destination_ip, destination_port,protocol*) and that to each interaction can corresponds several opened sessions. This aspect will be dissected more in detail in the chapter of results.

3.2.1 Dataset

We merge all the logs together so that we have complete information grouped into a unique huge data frame, organized in the following fields : ip source, that give us information about the source generating the traffic towards our honeypot, the ip destination that is our installed machine, the contacted protocol, the session id, unique identifier of the session, the timestamp that allows us to trace time series of the suspicious login attempts and two fields for passwords and usernames used.

These log files are collected every day for a period that goes from the 26th of october 2021 to the 25th of January 2022, almost three months of collected data. Also as mentioned in table (3.1) in total we observe 24 051 660 different

opened sessions.Also, among the total application layer protocols available, in table are mentioned the 5-most contacted ones during our tracing period.

Attribute	Value
Protocol	VNC,pop3,pop3s,imap,postgreSQL
Number of HP	1
Duration	[26-Oct-2021] - [25-Jan-2022], 92 days
Overall number of interactions	24 051 660

Table 3.1: Overview of the conducted experiment: zoom on tracing period [Nov 2021-Jan 2022]: IP activities in the honeypot split in contacted services.

With an high application level analysis to have a base characterization of the information collected, we passed to a more in deep analysis to either reject or not the initial hypothesis.

In fact, first high level application analysis is conducted by means of the hypothesis already presented in section 1. As the previous similar conducted studies on this matter [19] [20] we started from the dependencies and distributions of the login attempts Temporal behavior of the activity against the deployed honeypot is investigated.

- H1: there are no significant temporal dependencies in the observed activity

As is affirmed in [21] knowing where hosts and users are in the network can be a powerful tool for identifying reconnaissance behaviors of bots and for tying them to specific machines or users. So with the help of Max Mind we extrapolate this information for the most active senders class and we

want to analyze:

- H2: there are no significant spatial dependencies in the observed activity

The targeted protocol is a clue for the intentions of a sender after an authentication mechanism is successful.

- H3: there are no significant protocol dependencies in the observed activity

Then, after collecting the origins of each attack, we were interested in the behaviour after a successful three-way handshake formalize three different phases of logins: the zero phase is the one in which the IPs interact with our honeypot without doing any login attempt, whereas in the first phase at least one attempts is done but without the credential insertion and finally, in the second phase are collected also usernames and passwords. To see the general interaction of a source we look deeper to the first phase's flows while to further investigation, e.g. which are the preferred used password and who send them we look at the phase two flows.

A vast amount of login attempts,phase 2 flows was captured but with a limited amount of dictionaries and default credentials since we can assume that

- H4: Only few different activities pattern are employed.

In previous works the authors found duplicates in significant amounts of attack sessions, in this work the efficiency of the attackers is examined and evaluated. A working hypothesis is :

- H5: senders are efficient with their resources.

Chapter 4

Results

4.1 Initial data dissection

In this section we examine in detail the captured data.

4.1.1 Senders and interactions

We refer to IP addresses as senders, and they are observed during our experiment period. Notice that with the scope of enriching our final statistics analysis, we consider the single IP addresses, even if some of them may belong to the same sub-net. For a period of three months, we see 5,355 different senders interacting with our honeypot. In figure 4.1 is shown the overview on the activities observed inside the honeypot during the tracing

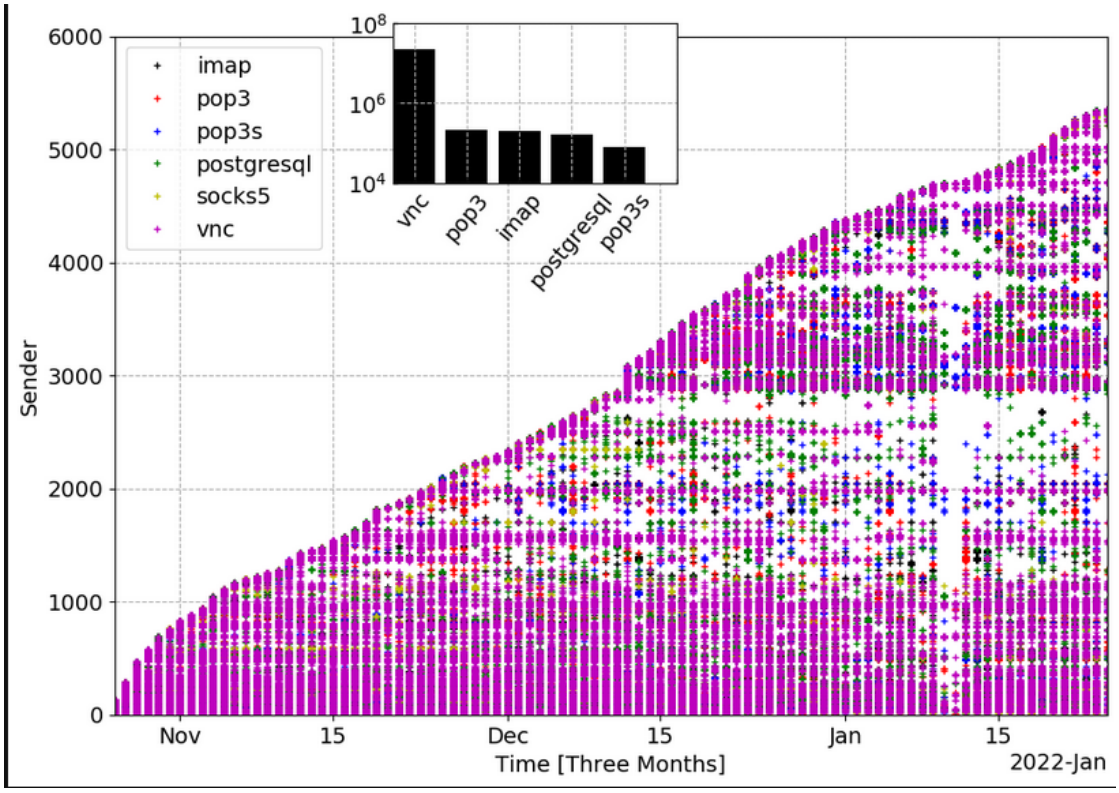

Figure 4.1: Heralding overview: senders' activity pattern split by contacted protocols

period: the y-axis lists the different sender IP address sorted by the timestamp of first appearance in the trace, while the x-axis represents time. A dot is a session opened from a sender at given time, the color refers to the protocol chosen. At this point, the term interaction is employed for sessions opened to a specific protocol and identified from the same unique source and destination IP-addresses pair within the same date (we assumed that login attempts sessions have a duration of at most 24 hour). As the total number of interactions is equal to 44,620 , we have that on average unique opened sessions per interaction is 538 and in figure 4.2 there is the representation on

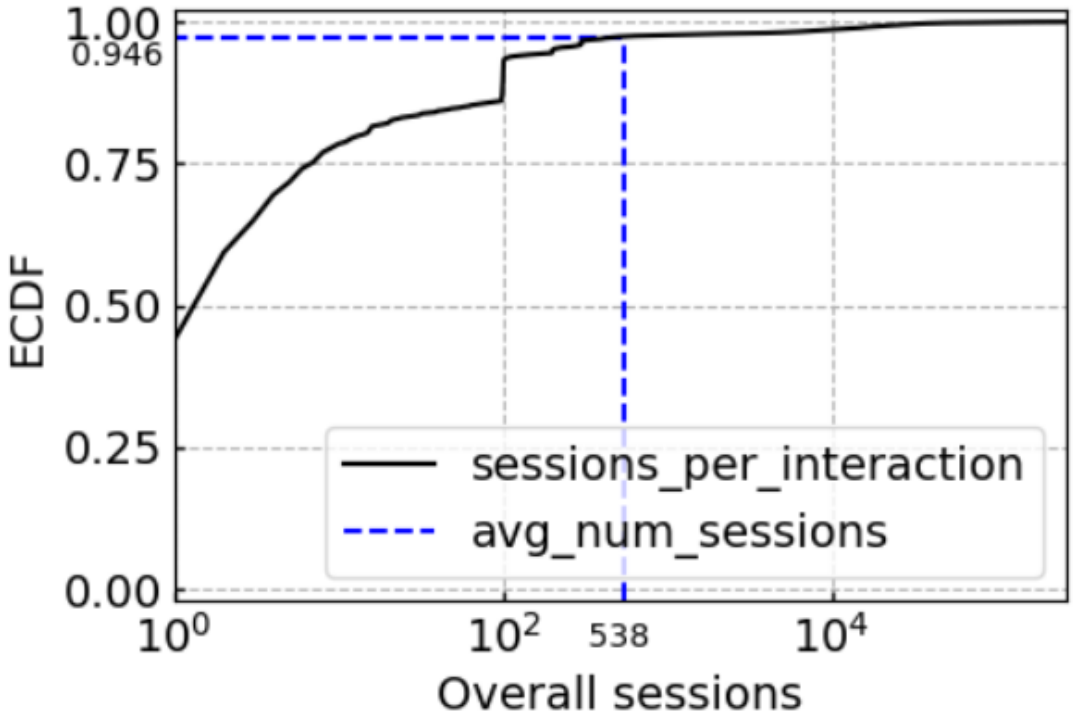

Figure 4.2: Distribution of number of sessions per interactions

how is distributed this average. More specifically, we have that almost 50 % of interactions has single opened session whereas only 5% opened number of sessions going from the average value, 538 sessions, to the max number of sessions per interaction, being 267,913.

4.1.2 Distribution of senders activities

For further evaluation of the hypothesis proposed, we consider more in deep the IPs distribution by their total number of activities as shown in figure (4.3) Considering the number of sessions opened by each actor in play, it

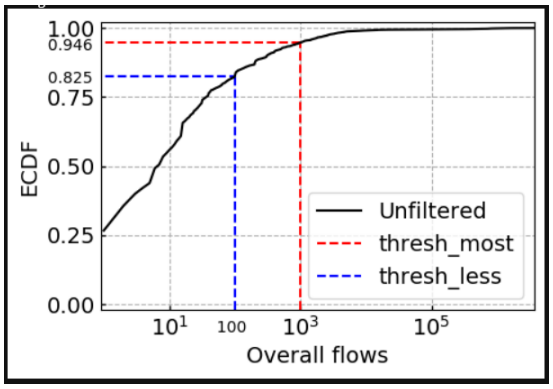

Figure 4.3: cumulative distribution of number of flows per IP

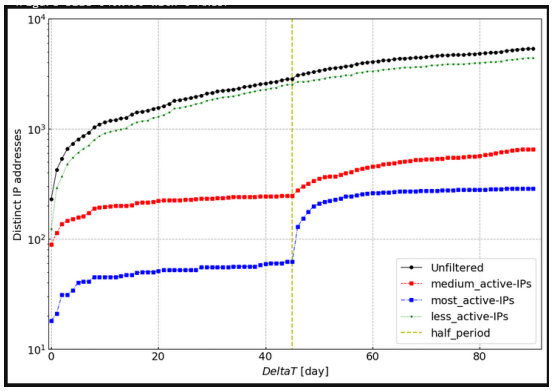

Figure 4.4: Cumulative number of senders over time

comes out that 82.5% of observed IPs generate less than hundred interactions in three months, and that the 94.6% of senders contributes to the overall traffic with only about 1% of the total opened sessions. This means that the slice of most active senders, being the 5.54% meaning 286 different IPs, are the one we are interested in. They, in deed, opened at least 1,000 sessions each, and with their activities produce in total the 98.9% of the overall traffic. After the filtering of IP classes we illustrate in figure 4.4 the cumulative sum of distinct IPs seen per day over the 92 days divided by class of activities: the most active class of IPs, do more than 1,000 flows each (blue dot line), class of medium active ones(red line)generates from hundred to 1,000 flows and finally the less active do less than hundred interactions(green line). Also, looking the overall curve (in dark) in 4.4 and also in figure 4.1 is underlined and confirmed a continuous growth of the number of senders over time. At last, this trend is more evident considering the blue curve, the most active

sender classes, for which there is a growth peak of hundred new IPs day to day, and this correspond to the half of the tracing period. It is possible to attribute this phenomenon to the juxtaposition of a darknet to the Tpot deployment that included our honeypot. [18]

4.1.3 Temporal distributions analysis

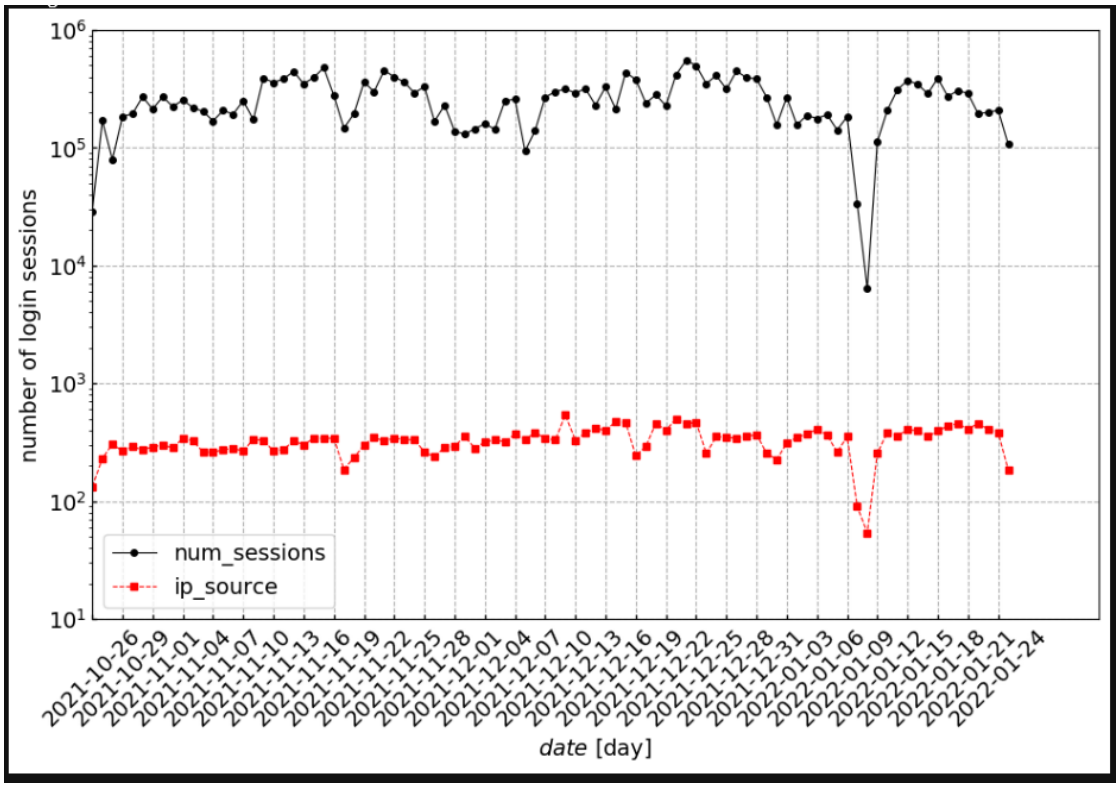

Figure 4.5: Temporal distribution: attempts (black curve) and unique IPs (red curve)

Feature	**Attempts**	**IP**	**C(A,IP)**	**C(A,A/IP)**	**C(IP,A/IP)**
Sum	24,051,660	30.230	0.487	0.86	0.039
Maximum	552,664	540	-	-	-
Minimum	6,383	54	-	-	-
Median	256327.5	332	-	-	-
Average	261067.24	328.58	-	-	-
SD	111389.64	82.23	-	-	-

Table 4.1: Explanation of attempts frequency peaks

Total opened sessions in our honeypot are of 24,051,660 and they regard the general activities observed. On a daily time scale, on average 261,431 sessions are opened with a standard deviation of 110,822. In figure 4.5 is shown not only the temporal distribution of opened sessions but also the number of unique IP addresses from which the interaction originated and since several peaks and troughs can be seen, further investigations are needed to evaluate them.

So, we consider the number of attempts, unique senders by day and number of sessions per sender. To better investigate if there is temporal dependency within captured data we computed the correlation indices of the two curves. According to table 4.1 values, in which have been reported the general statistics of distribution of login attempts and IPs and their correlation, it results that the total number of opened sessions by day results to be medium correlated to the number of IP (0.487), while results very strongly correlated to the number of opened sessions per IP (0.86).

Interestingly, the number of unique IPs is not correlated to the number of opened sessions meaning that they are independent (0.039).

Since the numbers of attacks and the number of IPs, and the number of attacks per IP are respectively medium and strongly correlated , it is not possible to say that the distribution of attacks is completely independent over time. However, it remains that IPs are independent of the number of attempts they make on two different days. This means that if we choose two distinct days but having similar subset of senders will end up having similar behavior in attempts they make. This can be a valid proof to reject the first hypothesis.

Other possible attributes to go further into this kind of analysis could be the increases in attempts from certain countries or with certain protocols.

Furthermore, to understand the general trend of busy time for our honeypot once a session is installed, we compute the cumulative distribution function of their duration in seconds. In figure 4.6 we computed a threshold on weighted average corresponding to 4,5 seconds and from the chart we can see that almost 75% of sessions are under this threshold. The 98% of sessions last at most 20 seconds. Finally, almost 45% of the overall sessions last less than 1 second. In the same way we compute the weighted average of overall sessions duration, we consider the mean duration of sessions

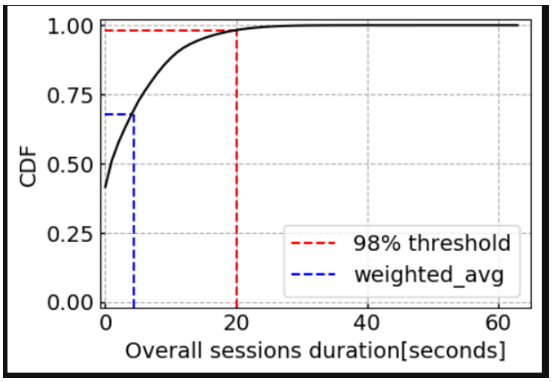

Figure 4.6: Cdf of overall duration sessions with weighted average and top threshold

4.1.4 Spatial distributions

We exploit the MaxMind [22] database to collect which are the geographical provenience areas of the most active senders to evaluate the countries of origin for each attempt. Over 44 different countries detected, 82% of sessions are opened from the same ten countries.

Most of the captured sessions are traced back to the Netherlands. However, to investigate reasons for these observations the number of sessions is normalized by several features. The top 10 countries by number of flows received are listed in table 4.7 . The ratio R in respect to the total number of attempts for the Top 10 list also is shown in table as R normalized by the countries GDP, population, area and number of IP producing it. [23]

The normalization reveals that the Netherlands is the country with more attempts in absolute values, this is confirmed also when you consider the

GDP based normalization. Considering the area, instead, we observed that Russia has a major impact, while for the population and IPs ratio, the country with more influence is the USA.

ALL THE FLOWS	country_code	attempts_byCC	IP_src_num	R	R_by_GDP	R_by_area	R_by_pop	R_by_IP
0	NL	8342497	20	0,351143	0,045789	0,000336	0,000231	0,025353
1	RU	6391721	12	0,269033	0,005143	0,125132	0,048163	0,011655
2	US	3545122	119	0,149217	0,040255	0,041674	0,060800	0,064104
3	EE	400602	3	0,016862	0,000129	0,000022	0,000027	0,000183
4	UA	385465	1	0,016225	0,000122	0,000267	0,000861	0,000059
5	GB	268098	13	0,011284	0,005456	0,000067	0,000929	0,000530
6	HK	134065	11	0,005643	0,000380	0,001218	0,000053	0,000224
7	PL	81415	8	0,003427	0,000026	0,000030	0,000159	0,000099
8	BD	70438	1	0,002965	0,000012	0,000013	0,000614	0,000011
9	SC	52620	1	0,002215	0,000000	0,000000	0,000000	0,000008
10	CZ	24104	11	0,001015	0,000003	0,000002	0,000013	0,000040

Figure 4.7: Investigation of different normalization features to investigate the spatial distribution of login attempts

4.1.5 Brief inspection of contacted protocols

Feature	Protocols	AVGduration	TOTflows	AVGdup
0	imap	4.220210	0.008415	36.1
1	imaps	4.350653	0.002502	16.12
2	pop3	4.202182	0.008643	37.65
3	pop3s	4.639269	0.003275	21
4	postgresql	4.620865	0.006971	22.35
5	smtps	11.266169	0.000580	4.9
6	socks5	1.161763	0.000750	15.04
7	vnc	3.898453	0.968862	1601.97

Table 4.2: Statistics by protocol: average session duration and average duplicated messages

As already mentioned the protocol can be eloquent about the final intentions of a sender, so through the Heralding has been exposed different

eight interfaces and they are listed in table 4.2. In absolute values the contribution of a single protocol depends on how strong the choice of VNC is, but considering also the number of active IPs by each protocol we have the result that the number of senders is not so much lower than the one active on the single VNC whereas this latter is the 96% of the total requests. Further investigation can be done by clustering the IPs targeting either a single or multiple protocols. If we consider the average session duration by protocol the major has a value very similar to the weighted average except for the vnc that is slightly lower than average and equals to 3.9 seconds, with smtps and socks5 having respectively 11.3 seconds and 1.16 seconds, but with a negligible contribution. At last, it has been considered the amount of "duplicates " similar sessions, having the same source and destination opened for each protocol on average, they are very high if considered VNC flows.

4.2 Traffic characterization

Before going into any further considerations, we already said that three different groups of sessions emerged : attempts that only opened sessions, attempts that tried after opening the session to trespass the authentication phase without inserting credential, the one that try to gain access by complete insertion of credentials. More than half traffic (51%) belong to the

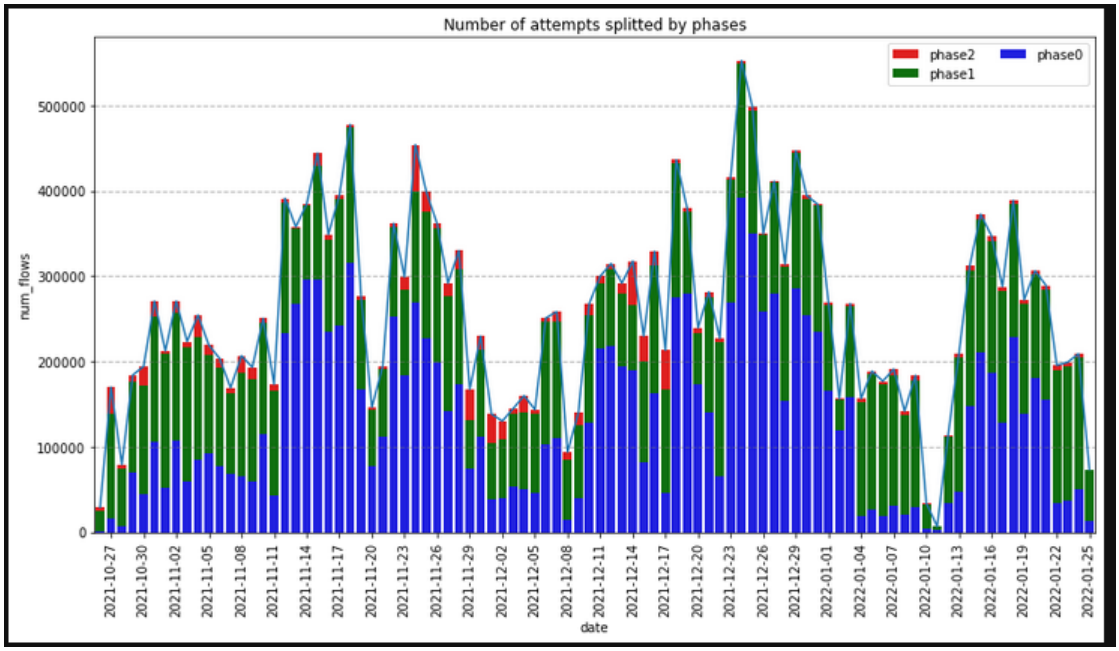

Figure 4.8: Number of attempts split by phases

first category whose 48% requested VNC protocol, while the 45,5% traffic is attempting to trespass the login phase without inserting credentials and it is entirely for VNC protocol. The limited amount of total traffic that try to login inserting credentials, is about 4% and also in this case VNC is the protocol prevalent involved.

The figure 4.8 shows the three different temporal distributions confirming the above mentioned trends and the splitting by phase portions. More than that, the chart in figure 4.9 shows what can be discussed about the rate at which each IP transverse from phase 0 to phase1 and then goes to phase 1 to finally reach phase 2, on a daily time scale. In general, we can state that

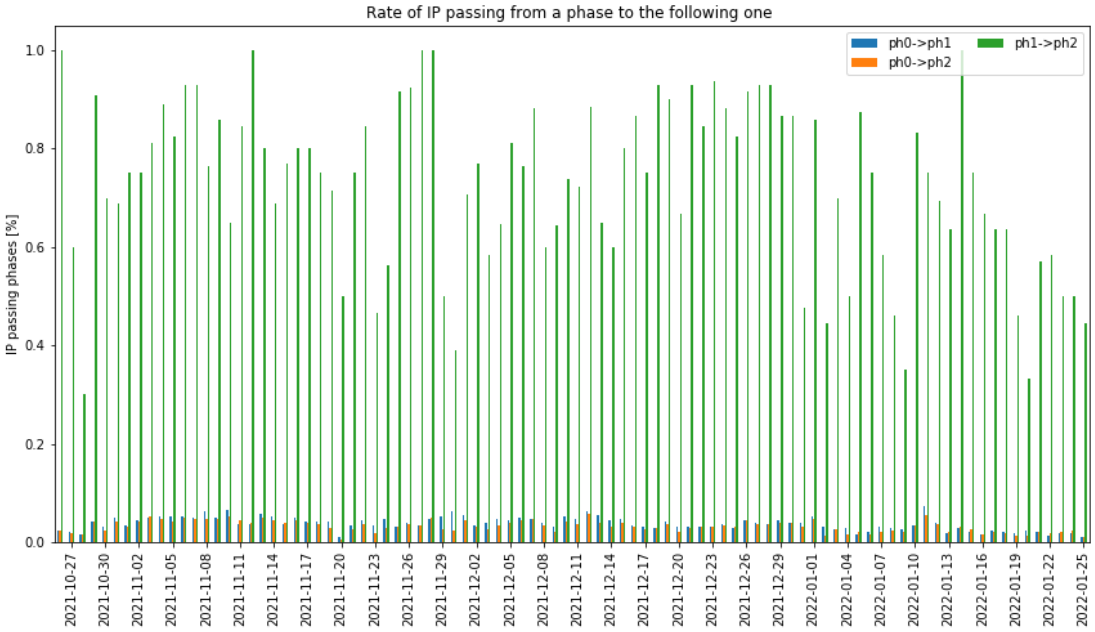

Figure 4.9: Senders daily switching rates {0,1} from phase0 to phase1, from phase1 to phase2 and from phase0 to phase2

for most ips when the phase 1 also the phase 2 modality is used with a rate always greater than 0.25, with some days in which the rate is total (all the phase 1 switch to phase 2). Different is for the number of ips passing from ph0 to ph1 (and consequently ph0 to ph2) which underline that the portion of IP acting prevalent on phase 0 and passing through the phase 1 or 2 is very poor and low.

In order to evaluate the fourth hypothesis we consider phase 1 and phase 2 login attempts that contributes for an amount of 50% of the total collected traffic since we finally inspected the dictionary used for passwords and

usernames.

4.2.1 Temporal and analysis on phase 1 and 2

Temporal and spatial dependencies on phase 1 and phase 2 Total login attempts for phase 1 and phase 2 are respectively equals to 10,828,193 and 913,249.

Feature	Attempts	IP	C(A,IP)	C(A,A/IP)	C(IP,A/IP)
Sum	10,828,193	125	0.207	0.628	-0.479
Maximum	159239	35	-	-	-
Minimum	4054	3	-	-	-
Average	117697.75	13.75	-	-	-
SD	36540.15	4.32	-	-	-

Table 4.3: Explanation of attempts frequency peaks for phase 1 sessions

Feature	Attempts	IP	C(A,IP)	C(A,A/IP)	C(IP,A/IP)
Sum	913,249	92	0.135	0.89	-0.142
Maximum	9926.6	33	-	-	-
Minimum	6,383	54	-	-	-
Average	9926.6	11.20	-	-	-
SD	10820	4.25	-	-	-

Table 4.4: Explanation of attempts frequency peaks for phase 2 sessions

In figure 4.8 are illustrated the two distributions, distinct by phase. In tables 4.3 and 4.4 are described the login attempts frequency peaks respectively for the phase 1 and phase 2, apart for the number of total login attempts and total unique senders are other attributes, such as the total number of

senders, the number attempts by number of unique senders. Also this time, we compute the correlation indices of the two curves per each phase and we conclude the followings: For phase 1 the total number of login attempts per day results to be not correlated(0.207) to the number of unique IP, while results correlated(0.628) with the number of opened sessions per unique IP. For phase 2 we have similar results for phase 1 case : a correlation index for number of login attempts and the number of unique IP being 0.135, while concerning the index for the number of attempts per sender we have a high correlation being 0.899. Moreover, if we consider both cases, phase 1 and phase 2, the correlation indices for number of senders and number of attempts per sender are respectively -0.479 and -0.15. Thus, we can conclude for the two distinguished phases that senders are strongly independent to the number of attempts observed, that confirms also the previous frequency analysis results achieved by considering all dataset. However, what really differ in this more granular analysis, lying on the fact that in this case there is no more a dependency between the number of unique senders and the number of attempts.

4.2.2 Password analysis

What we are going to study here is the phase 2 interactions with the complete set of credentials employed by senders. The total number of sessions opened is 913,249 and in figure 4.5 is reported the temporal distribution. When

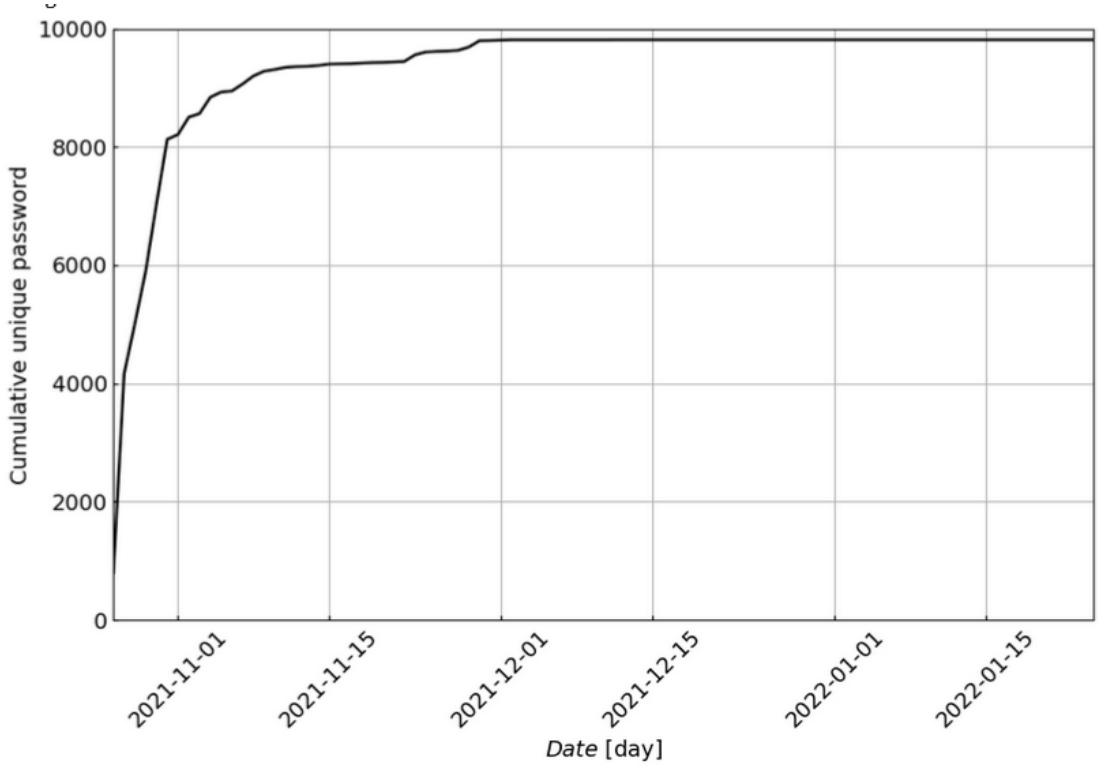

Figure 4.10: Time-series of different number of password observed during the tracing period

talking of credentials we want to discover if there are any significant strings injected that could remind us of default passwords used by famous botnets, Mirai or Mirai-like [23]. The work [21] in fact aims to demonstrate that there is a strict correlation between the senders and the credentials used in the login phase.

We observe that most of the attempts are done without inserting usernames, so we can neglect this attribute for the credential analysis and focus on password values.

At high level analysis, the number of unique passwords inserted in three

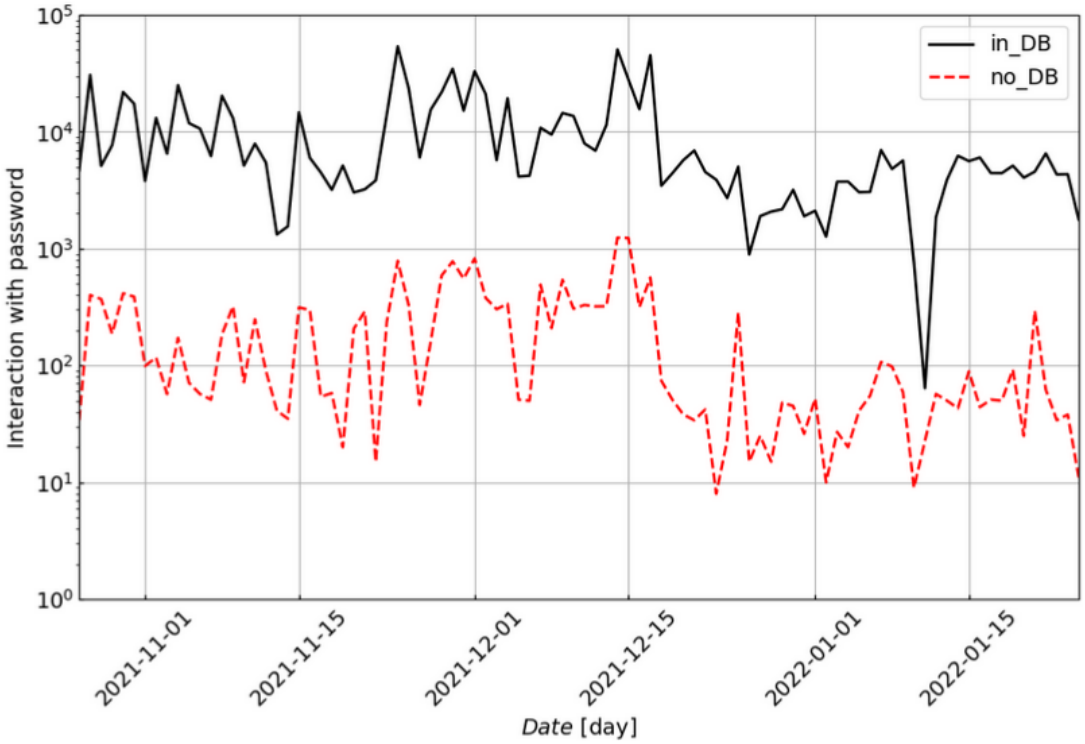

Figure 4.11: Time series on the two distinct kind of password interactions: interactions with password found in database (black curve), password did not find in database (red dashed curve)

months for the phase two flows, so not considering the blank space when a login request is forwarded, is 9,811. If it is considered the number of new password values observed by time, depicted in figure 4.10 , we can see that the curve saturates very soon in the considered tracing period. A first consideration can be that the sender has a limited dictionary in respect to the number of attempts they make and this could be an evidence that the senders are not efficient with their resources. Consequently, we plot a cumulative distribution function of passwords, in figure 4.12, to see how they

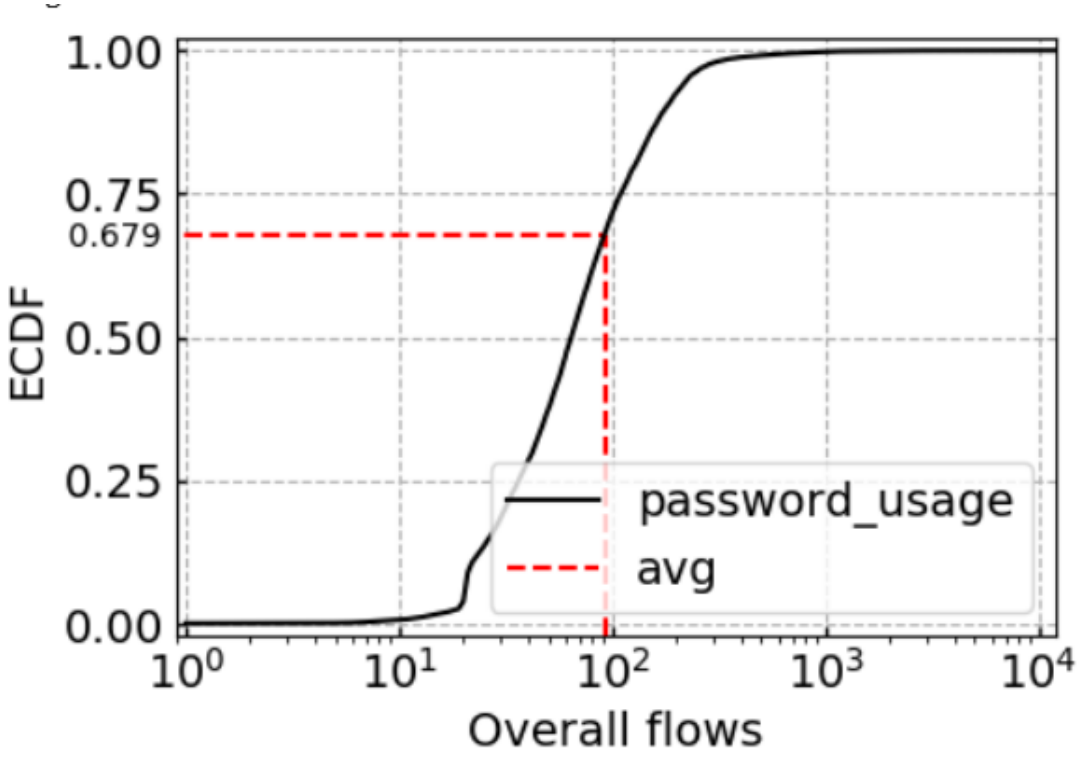

Figure 4.12: Cumulative Distribution Function on password average usage

are distributed by number of attempts and the 32,1 % of unique passwords are used more than 92 times, that is the average value of attempts per password.

Further, we provide the list of top 10 passwords used in the table 4.5.

At this point, we inspected the values of the password and in general, they all come from well known databases used in case of attacks requiring authentications, as for example in botnets. But more in detail, we examined if some strings come from a smaller database that is the common password list (rockyou.txt) files containing 4,341,564 unique passwords, used in

	Password	Occurrences
0	password	11874
1	Password	10462
2	12345678	5692
3	1q2w3e4r	2917
4	Passw0rd	2239
5	1qaz2wsx	1618
6	personal	1618
7	iloveyou	1604
8	Samantha	1561
9	Jennifer	1330

Table 4.5: Most 10 used passwords

32,603,388 accounts and it is a dictionary provided by Kali Linux OS as part of its standard installation. We compare them against our collected passwords with the results that only 344 seems to not belong to it. Then, in figure 4.11 it has been inspected the temporal distribution of attempts for password found in the dictionary of "rockyou.it" and the one do not found, to see if there are any particular dates in which can be observed a trend reversal in usage of the two curves.

4.2.3 Password by region

As we did for the evaluation on overall dataset we inspect the geographical origin of the password inserted. Then we compute how many occurrences of the same password there are for each origin country. Also, to be more exhaustive we compare the overall dataset scores, by most used country and

by most used password, with the phase 2 subset of login attempts, flows that we recall to be the one for which actually has been inserted a string for passwords. Consequently, the two heatmaps in figure 4.13 (overall dataset) and figure 4.14 (phase 2 subset) show the occurrences of a single password by countries : on the y axis the countries are listed in descendant order by number of the total open sessions while on the x-axis there are the most used password by their occurrences. On the color bar is shown the intensity of the flows. In the first heatmap 4.13, we can see that the majority of countries prefer the non insertion ("empty" stands for blank field) while for the top 6 countries the usage of passwords is more or less distributed over all the password set. This is also confirmed by what we observed in the heatmap 4.14 that is a zoom on only the login attempts including passwords, i.e. the phase 2 sessions: the number of active countries is reduced and most of them used more or less all the passwords, except for Japan and Singapore that only used with more intensity the same top 6 passwords. Obviously, this is coherent with the general spatial analysis of before, whose objective was to discover the spatial dependencies of the attempts. Here, we can say in addiction that the Netherlands and Russian operate all the three category of flows we individuate, since they are on the top of both heatmaps. If considering only the subset of password 4.14 we see that Ucraina contribution is more relevant than the one of the US (that scores at the third place for the intensity in the first heatmap).

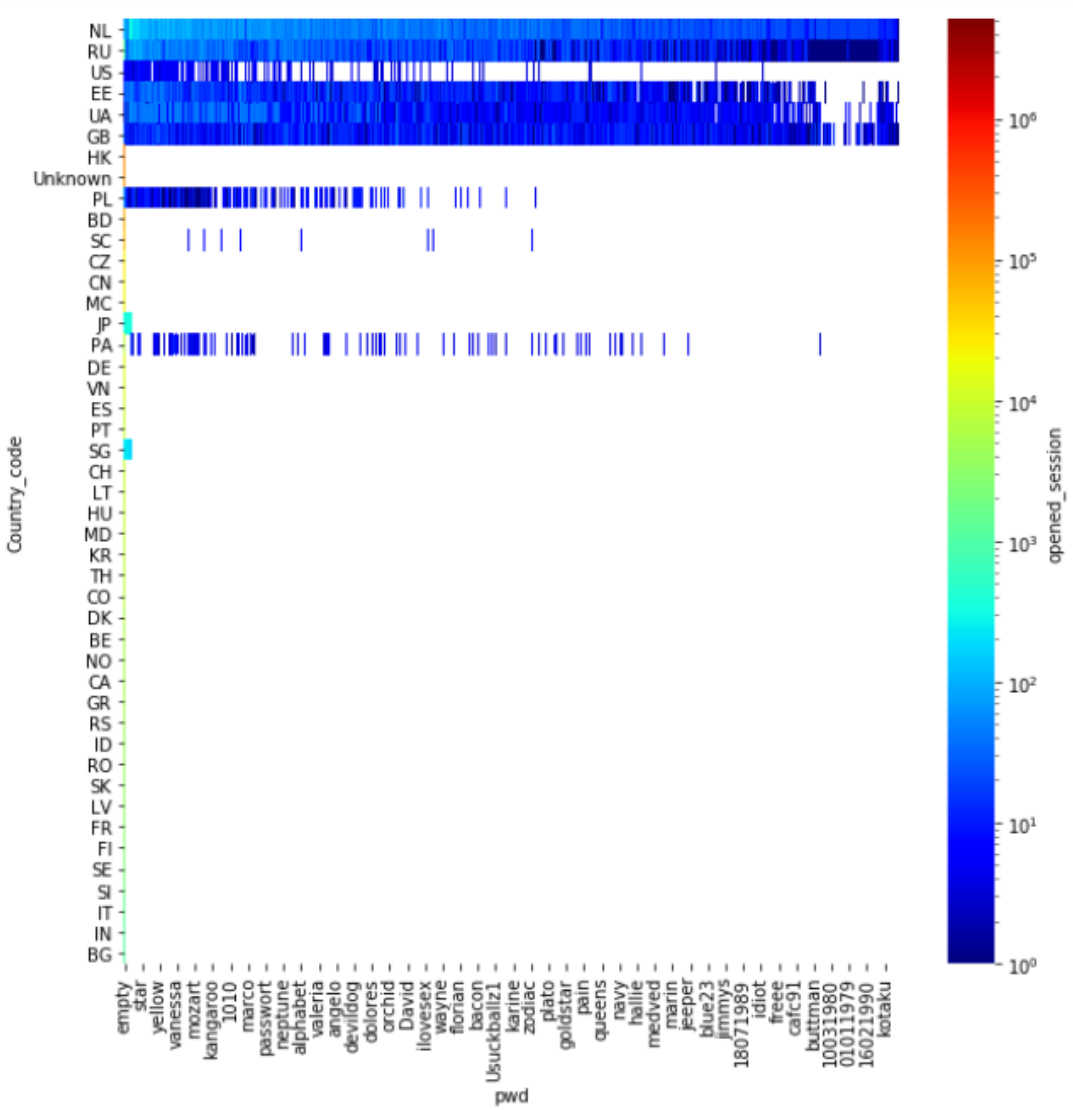

Figure 4.13: Heatmap of password occurrences by country to investigate the spatial distribution of login attempts, on all dataset

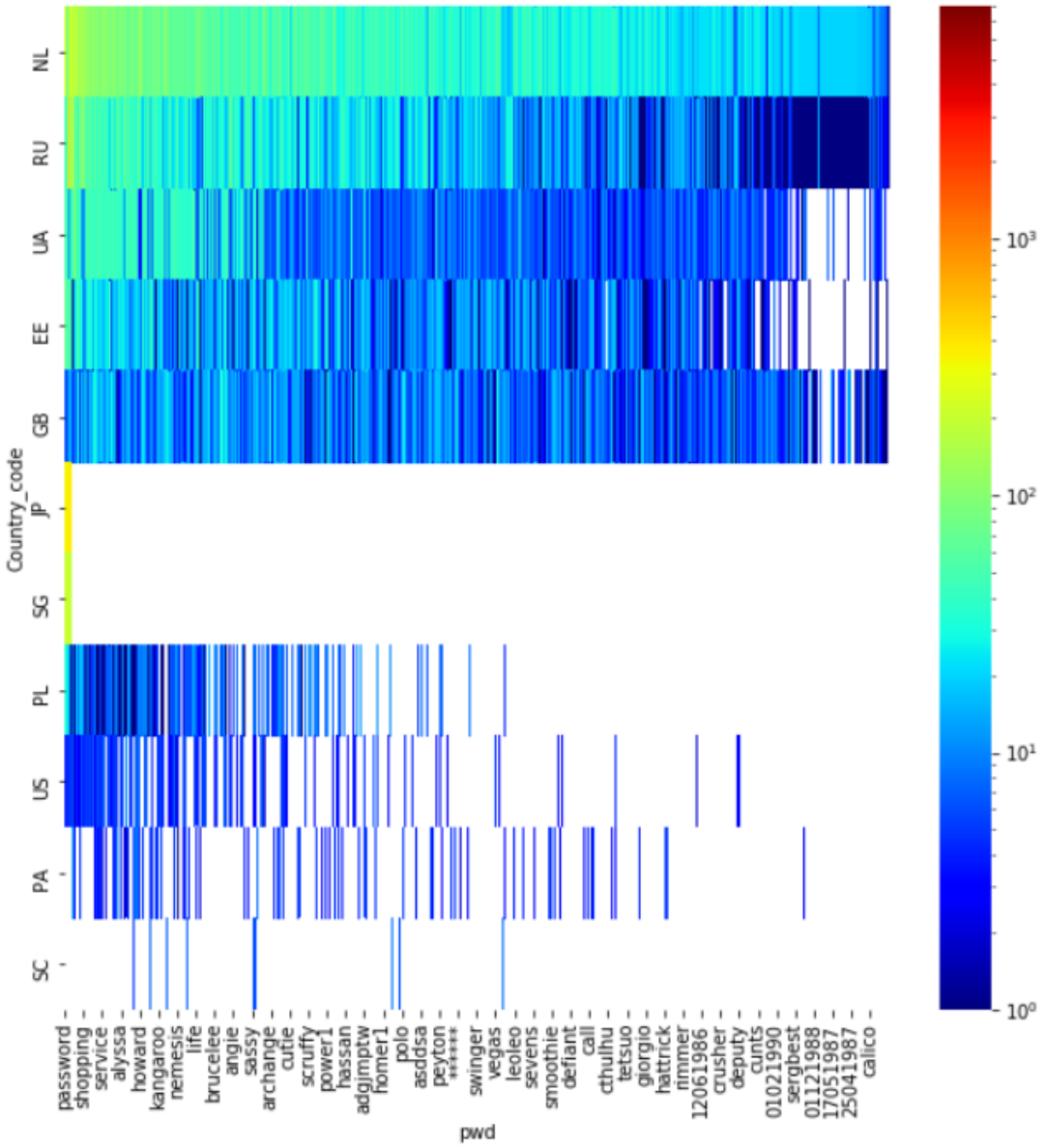

Figure 4.14: Heatmap of password occurrences by country to investigate the spatial distribution of login attempts, on phase 2

Chapter 5

Conclusions and future works

5.1 Conclusions

In this thesis, firstly we evaluate among the available honeypots in the Tpot framework deployed in Polytechnic of Turin the one to analyze more in deep, at application layer. Among them we choose the Heralding honeypot that is a low-interaction honeypot able to simulate a login interface of several protocols. Furthermore, it has been observed than in respect to others deployed honeypot within the same framework, the under analysis honeypot was simple but it attracts during our tracing three months period a vast variety of collected credentials information. At this point we proceed with our

first high level dissection of collected dataset. What comes out by this initial data dissection it is that among the overall number of protocols provided by the honeypot the most requested protocol is the VNC one, underlining a final intentions by senders to access remotely to devices. Another additional information we derived is that the senders observed growth as the time goes by, whereas a poor percentage, about 4% of total, it contributes to the overall activities observed. Afterwords, we enrich our collected information with the Maxmind database through which we derived the origin country of the attempts of the most active IPs. At this stage, we conduct a temporal and spatial distribution analysis to understand the dependencies of the login attempts. The attempts may vary a lot during the experiment but it comes out that even if there is a temporal dependencies between the number of attacks per day and the number of IP, the IPs are independent from the number of attempts they make. Whereas, from a spatial point of view, we observed that 72 % of the attempts, of the most active senders, come from only three country the Netherlands, USA and Russia. For all the analysis done up to now we consider the whole data set. By the way, we individuate three different phase of interactions differentiated by the kind of action the sender made, being the last two phases interactions more interesting since they go further on the login phase with respect to the first phase that only opened the sessions and never ask for authentication. This is why for more consistent analysis we consider a subset including only 50% of total

attempts, that includes the last two phases flows in which the senders ask for authentication. Thus, we proceed with similar temporal analysis on what we called phases 1 and 2 flows that collected actual login attempts(with or without insertion of passwords). Also in this case, even if less strict the temporal dependency cannot be rejected at all suggesting us that could be behind some orchestrated activities, so automated typical of botnets activities. Finally, after figuring out the behaviour of senders reaching the honeypot, we study the set of credentials have been collected. It comes out a very poor dictionary of default disclosed passwords with respect to the attempts done, suggesting that there is a repetition of few patterns and so that the most active senders are very inefficient with their resources. We can therefore conclude that among all the observed activities and senders, there are few small actors (making less than 100 attempts) who are attracted to the honeypot perhaps by adjacent structures and few actors who generate most of the traffic. The latter attack mainly from only three countries (with medium-high GDP) with a waste of resources because many interactions show many sessions directed to the same target (VNC) with the same attack method.

5.1.1 Future works with a clustering algorithm

In real-life applications and networks, it is crucial to be able to classify any attack as fast as possible so that is possible to react it, in almost zero

time. A key point to improve the management of this kind of events is the a-priory knowledge of collected attacks. So, to achieve a more comprehensive acknowledgement about attacks a future works to be done on the collected and dissected features of the honeypots can be the clustering of senders. They can be grouped by number of attempts, origin country, credential information and the different dictionaries employed.

Bibliography

[1] Cisco Report. «Cisco Annual Internet Report (2018–2023) White Paper». In: *Cisco* (March 9 2020). URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (cit. on p. 1).

[2] Kapersky. «DDoS attacks in Q4 2021». In: *Kapersky Journal* (February 2022). URL: https://securelist.com/ddos-attacks-in-q4-2021/105784/ (cit. on p. 2).

[3] Lance Spitzner. «Honeypots: Catching the insider threat». In: *19th Annual Computer Security Applications Conference, 2003. Proceedings.* IEEE. 2003, pp. 170–179 (cit. on p. 4).

[4] Marcin Nawrocki, Matthias Wählisch, Thomas C Schmidt, Christian Keil, and Jochen Schönfelder. «A survey on honeypot software and data analysis». In: *arXiv preprint arXiv:1608.06249* (2016) (cit. on pp. 4, 13, 14).

[5] URL: https://github.com/johnnykv/Heralding (cit. on p. 4).

[6] Iyatiti Mokube and Michele Adams. «Honeypots: concepts, approaches, and challenges». In: *Proceedings of the 45th annual southeast regional conference*. 2007, pp. 321–326 (cit. on pp. 8, 12).

[7] Seamus Dowling, Michael Schukat, and Enda Barrett. «New framework for adaptive and agile honeypots». In: *ETRI Journal* 42.6 (), pp. 965–975. DOI: https://doi.org/10.4218/etrij.2019-0155. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.4218/etrij.2019-0155. URL: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2019-0155 (cit. on p. 9).

[8] Daniel Fraunholz, Marc Zimmermann, and Hans D. Schotten. «An adaptive honeypot configuration, deployment and maintenance strategy». In: *2017 19th International Conference on Advanced Communication Technology (ICACT)*. 2017, pp. 53–57. DOI: 10.23919/ICACT.2017.7890056 (cit. on p. 11).

[9] URL: https://github.com/cowrie/cowrie (cit. on p. 13).

[10] URL: https://github.com/DinoTools/dionaea (cit. on p. 13).

[11] URL: https://github.com/honeytrap/honeytrap (cit. on p. 13).

[12] Armin Ziaie Tabari and Xinming Ou. «A first step towards understanding real-world attacks on IoT devices». In: *arXiv preprint arXiv:2003.01218* (2020) (cit. on pp. 14, 15).

[13] Wenjun Fan, Zhihui Du, Max Smith-Creasey, and David Fernandez. «Honeydoc: an efficient honeypot architecture enabling all-round design». In: *IEEE Journal on Selected Areas in Communications* 37.3 (2019), pp. 683–697 (cit. on p. 14).

[14] Lance Spitzner. «The honeynet project: Trapping the hackers». In: *IEEE Security & Privacy* 1.2 (2003), pp. 15–23 (cit. on p. 15).

[15] URL: `https://github.security.telekom.com/2015/03/honeypot-tpot-concept.html` (cit. on p. 16).

[16] URL: `https://github.com/elastic/kibana` (cit. on p. 16).

[17] URL: `https://github.com/elastic/elasticsearch` (cit. on p. 16).

[18] M. Mellia et al. *Enlighting the Darknets: augmenting darknet visibility with active probes*. ? to be published. 2021 (cit. on pp. 18, 29).

[19] Daniel Fraunholz, Marc Zimmermann, Alexander Hafner, and Hans D Schotten. «Data mining in long-term honeypot data». In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2017, pp. 649–656 (cit. on p. 22).

[20] Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. «Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot». In: *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*. IEEE. 2017, pp. 1–7 (cit. on p. 22).

[21] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, and Manish Karir. «A Survey of Botnet Technology and Defenses». In: *2009 Cybersecurity Applications Technology Conference for Homeland Security*. 2009, pp. 299–304. DOI: 10.1109/CATCH.2009.40 (cit. on pp. 22, 39).

[22] URL: https://www.maxmind.com/en/geoip-demo (cit. on p. 32).

[23] Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. «Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot». In: *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*. 2017, pp. 1–7. DOI: 10.1109/CyberSecPODS.2017.8074855 (cit. on pp. 32, 39).