applied sciences

MDPI

Article

Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection

Gabriel Arquelau Pimenta Rodrigues [1,†], Robson de Oliveira Albuquerque [1,†], Flávio Elias Gomes de Deus [1,†], Rafael Timóteo de Sousa Jr. [1,†], Gildásio Antônio de Oliveira Júnior [1,†], Luis Javier García Villalba [2,*,†] and Tai-Hoon Kim [3,†]

1 Cybersecurity INCT Unit 6, Decision Technologies Laboratory-LATITUDE, Electrical Engineering Department (ENE), Technology College, University of Brasilia (UnB), Brasília-DF 70910-900, Brazil; gabriel.arquelau@redes.unb.br (G.A.P.R.); robson@redes.unb.br (R.d.O.A.); flavioelias@unb.br (F.E.G.d.D.); desousa@unb.br (R.T.d.S.J.); jrgildasio@gmail.com (G.A.d.O.J.)

2 Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain

3 Department of Convergence Security, Sungshin Women's University, 249-1 Dongseon-Dong 3-ga, Seoul 136-742, Korea; taihoonn@daum.net

* Correspondence: javiergv@fdi.ucm.es; Tel.: +34-91-394-7638

† These authors contributed equally to this work.

Received: 20 September 2017; Accepted: 13 October 2017; Published: 18 October 2017

Abstract: Any network connected to the Internet is subject to cyber attacks. Strong security measures, forensic tools, and investigators contribute together to detect and mitigate those attacks, reducing the damages and enabling reestablishing the network to its normal operation, thus increasing the cybersecurity of the networked environment. This paper addresses the use of a forensic approach with Deep Packet Inspection to detect anomalies in the network traffic. As cyber attacks may occur on any layer of the TCP/IP networking model, Deep Packet Inspection is an effective way to reveal suspicious content in the headers or the payloads in any packet processing layer, excepting of course situations where the payload is encrypted. Although being efficient, this technique still faces big challenges. The contributions of this paper rely on the association of Deep Packet Inspection with forensics analysis to evaluate different attacks towards a Honeynet operating in a network laboratory at the University of Brasilia. In this perspective, this work could identify and map the content and behavior of attacks such as the Mirai botnet and brute-force attacks targeting various different network services. Obtained results demonstrate the behavior of automated attacks (such as worms and bots) and non-automated attacks (brute-force conducted with different tools). The data collected and analyzed is then used to generate statistics of used usernames and passwords, IP and services distribution, among other elements. This paper also discusses the importance of network forensics and Chain of Custody procedures to conduct investigations and shows the effectiveness of the mentioned techniques in evaluating different attacks in networks.

Keywords: cybersecurity; network security; traffic analysis; deep packet inspection; intrusion detection; network forensics

1. Introduction

It is no longer possible to consider business and government services without the use of the Internet. The impressive growth rate of the Internet and its technologies allows many companies and government agencies to provide their services online, bringing more practicality to users. This growth,

however, also results in a more frequent occurrence of cyberattacks that impact users and business, such as to shut down victims' servers or steal critical information from attacked networks.

For most applications, especially those considered critical (e.g., Internet Banking), having its assets compromised or being forcibly off for a period is inadmissible because it represents revenue loss. It has become clear that, not only when attacked, one must have early detection and mitigation techniques to minimize damages and recover normal operation as fast as possible. This early detection can be achieved with the use of advanced network tools and analysis in conjunction with skilled people within the cybersecurity domain.

One approach to investigate network attacks is the use of Deep Packet Inspection (DPI). Basically, this technique consists of a thorough examination of the fields contained in the packets that flow within the investigated network. It allows for detecting anomalies into the network flow, along with the other important information that is useful when dealing with incident response, such as the IP address involved, type, time and duration of the attack, along with other data that helps security professionals to mitigate incidents.

Network forensics is used to identify and generate evidence against attacks that can be related to the commitment of crime such as the stealing of information that is not publicly available. Although it may not always be the case, it is important to gather and examine evidence as if they were to be used in court. Having such assumptions in mind, it is important to follow Chain of Custody (CoC) procedures when dealing with incident response and cyberattack mitigation.

This paper provides DPI and analysis of network traffic data obtained from a Honeynet maintained for research purposes in a network laboratory at the University of Brasilia. This data was analyzed to evaluate the behavior of attackers, the content and other characteristics of some attacks directed to the environment.

The main contribution of this paper is to provide a characterization of real attacks using DPI to evaluate the modus operandi of some bots (i.e., W32.IRC), botnets (i.e., Mirai), evaluation of brute-force attacks' behavior conducted either by humans or bots providing distribution statistics of the source of the attacks, protocols used and link analysis of the data.

The remainder of this paper is organized as follows. Section 2 provides a review of the DPI technique and researches within the traffic analysis and network forensics domain. Section 3 describes techniques used to detect network anomalies using DPI. Section 4 shows the results obtained against real attacks traffic data using DPI and network forensics procedures. Section 5 concludes this paper.

2. Review of the State of the Art about Packet Analysis and Related Work

Inspection of packets in a network flow can be performed in various depths for different purposes. It is considered a common practice in network security analysis and is widely used by cybersecurity experts. The following subsections present some review of the state of the art and a brief related work.

2.1. Deep Packet Inspection

Although it faces many challenges, application of DPI for security purposes has proven to be effective and it is a technique whose utilization is expanding [1]. Packet inspection refers to the ability to analyze network traffic for a given purpose, be it in a real-time scenario or in an offline analysis. This kind of inspection may vary depending on its depth.

As stated by Parsons [2], Shallow Packet Inspection (SPI) is the examination of the headers of every packet, but not the payload, focusing on layers 2, 3 or 4 regarding the TCP/IP stack. This kind of inspection is often used in firewalls that compare the source or destination addresses against a blacklist, for the sake of deciding to let a packet pass or drop it.

Conversely, Medium Packet Inspection (MPI) examines the headers of the packet and the presentation layer of its payload that can be compared against a list. This inspection is commonly used by application proxies and provide a more thorough analysis, being able to determine, for instance, some file formats and thus filter packets based not only on their IP addresses.

On the contrary, when using DPI, all of the headers of all layers and the payload are prone to be checked. With DPI, it is possible to detect well-known malware signatures and network anomalies that may be considered attacks. DPI allows for identifying the IP source of an attack, anomalies in the payload, intentions and goal. In conjunction with the network flow, it is a powerful tool to understand the sequence of an attack, steps the attacker may have taken and other techniques that may have been used. With DPI, it is possible to evaluate new exploitation techniques in networks; thus, it allows researchers to create new defense mechanisms and signatures.

Table 1 shows the different depths of the packet inspection (SPI, MPI and DPI) and the layers up to which they reach. From levels one to four, the inspection consists of an analysis of the headers of the packets, whilst from levels five to seven, the analysis is performed in the payload [2].

Table 1. Different depths of packet inspection and the OSI and TCP/IP layer they encompass.

Level	SPI	MPI	DPI	OSI Model	TCP/IP Model
7			✓	Application Layer	
6		✓	✓	Presentation Layer	Application Layer
5		✓	✓	Session Layer	
4		✓	✓	Transport Layer	Transport Layer
3	✓	✓	✓	Network Layer	Network Layer
2	✓	✓	✓	Data Link Layer	Data Link Layer
1	✓	✓	✓	Physical Layer	Physical Layer

Although DPI is a more informative inspection, being more useful for the detection of anomalies, it also produces a greater amount of data, requiring more processing and storage capacities. Nowadays, in modern networks, a good approach to analyze the amount of information DPI creates is the use of Big Data tools, such as Hadoop [3] and Kafka [4]. In addition, as investigation, mitigation and incident response may take place in real time, a deeper analysis of the contents of each packet requires better performance of the device performing the inspection. Thus, monitoring data in real time require pipeline techniques, queues, visualization and processing power bigger than it was necessary before.

DPI can be used in many applications. It is useful for network management, by maintaining a Quality of Service (QoS) with content optimization (treating different traffic types accordingly to provide required QoS), application distribution and load balancing; it also contributes to network security and forensics, by detecting and dropping potential harmful packets trying to enter or leave the network [5]. Other applications of DPI include network visibility, user profiling, copyright policing, censorship or content regulation [6].

2.2. DPI Common Challenges and Requirements

Deep packet inspection is not only a powerful tool regarding network anomalies detection, but it may also be used in bandwidth management, advertisement and copyright content filtering [7]. However, it may be used with bad purposes, such as interfering within net neutrality [8] or allowing governmental surveillance [7] and privacy invasion [9]. Due to the nature of DPI, some limitations in its implementation arise some challenges, requiring, therefore, a study of the viability of its deployment.

Regarding some of the most common challenges Deep Packet Inspection faces, as presented in Table 2, its implementation, in order to be effective, should consider the attributes of the hardware assigned to perform the inspection. Slow hardware or software may become the bottleneck in packet processing, compromising the results that could be achieved. To allow the implementation of these devices in high speed networks, existing studies focused on improving their performance with algorithms that integrates CPU and GPU [10]; hardware acceleration [11]; and new architecture for DPI devices [12].

Table 2. Resume of DPI common challenges.

Challenges	Description	Supporting References
Performance	When monitoring a network in real time, the device assigned to perform the DPI should be able to process the packets information in the least time possible, for it to be capable of analyzing every packet of the flow, without accumulated delay. Optimizing the use of the memory of the device, for instance, improves the performance. As internet traffic generates a huge amount of data, its processing may require tools specialized for Big Data, such as Hadoop and Kafka.	[3,4,13]
Encryption	As DPI consists of the analysis of the payload and headers of each packet, encrypting it hinders the detection with DPI, it being possible to analyze only the metadata. In such cases, solutions include the use of proxies which can decrypt the traffic, in cases where it performs Man-In-The-Middle (MITM) transactions on behalf of users, or analyzing packets after they are decrypted.	[14–16]
Anonymity	Cases in which the attacker uses network anonymizers, such as The Onion Router (TOR), will lead the DPI investigation to an inaccurate conclusion regarding the source of the attack. Spoofing packets will also masquerade its source.	[17–19]
Number of attacks	With the increasing number of different attacks, thorougher inspections are necessary to detect their patterns and signatures, and keep the database updated. In addition, as zero-day vulnerabilities may occur at any time, a more detailed investigation is necessary in order to identify and follow the attacker's trail. To circumvent this, Machine Learning techniques may be applied, to detect known and emerging malware.	[20–22]

The implementation of DPI devices may vary according to requirements specific to the technology in use. Bouet et al. [23] described the implementation of Deep Packet Inspection in a Software-Defined Network (SDN) and proposed a method for reducing the network load and improve the efficiency of the devices, significantly reducing the implementation cost.

When deployed for network forensics, the inspection depends on a good localization of the network sensors. In turn, it depends on what information from the network is wanted and may vary from inbound or outbound interfaces, pre-Network Address Translation (NAT) or post-NAT traffic, behind or in front of a firewall, and so on, in relation to the nature of the data wished to be checked [24]. Encrypted traffic also poses a challenge to DPI and should be considered prior to its implementation. Deep packet inspection and encryption are discussed in more detail in Section 2.4.

2.3. Network Forensics and the Cybersecurity Market

Internet traffic grows rapidly, with an expectation of 278 Exabytes of data being transmitted per month by 2021 (see Figure 1), as stated in [25]. As DPI is correlated to network traffic, this growth also implies a higher amount of DPI devices used in the several different types of its applications and segments. Major factors for the DPI market expansion are evolution of cyberattacks, which influences the need for modern and good performance network analysis devices and advancements in the communication technology.

As a consequence of this growth, there is an increased occurrence of cyberattacks and more money being invested by companies to secure their assets, as a data breach may cost millions of dollars and result in a loss of market [26]. Figure 2 shows that the security alerts has risen from 2013 to 2016. It is possible to see a spike of alerts in 2015, evidencing an explosion of attacks in that year [27], which can be verified by the Symantec Report [28], which shows an increasing appearance of, among other threats, new mobile malware variants and a higher rate of e-mail malware in recent years.

As reported by a Cisco Report [27], with the advent of new technologies, such as Internet of Things (IoT), the attack surface will only expand, giving malicious users more space to operate. As a consequence, the use of DPI is expected to grow, in order to prevent targeted networks from being exploited [29]. One of the biggest reasons for this rise is the growing trend of ransomware, which are dominating the malware market, partially due to the use of bitcoins as the preferred method of payment, as it grants anonymity to its users. This lucrative threat became more widespread and powerful in the first half of 2016, targeting individual and enterprise users with a high success rate [29]. The recent worldwide infection of WannaCry [30] corroborates the ascension of this kind of malware. As this type of malware depends on a private key to conclude the encryption, DPI can be used to detect and drop suspected egress traffic, matching ransomware's communication protocols.

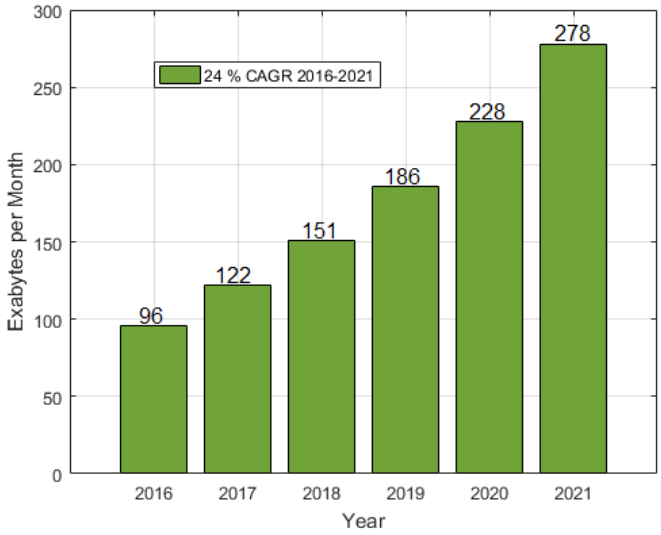

Figure 1. Global IP traffic forecast for up to 2021—Compound Annual Growth Rate (CAGR) [25].

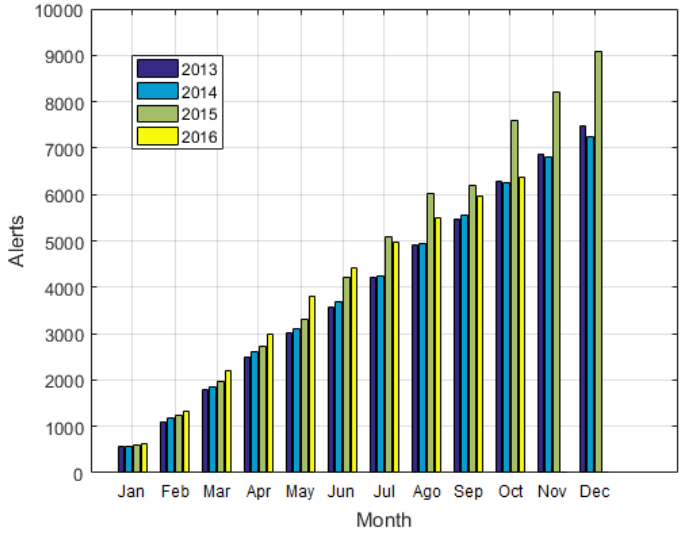

Figure 2. Cumulative annual security alert totals [27].

As claimed by Morgan [31], the worldwide investment in cybersecurity will increase from $75 billion, registered in 2015, to $170 billion in 2020. The expectation for the DPI market is to grow at a CAGR of 21.6% from 2016, $7.01 billion to 2021, and forecast of $18.60 billion [1].

2.4. Related Research about Deep Packet Inspection

DPI may present a low detection rate for some attack categories. In order to try to increase this performance and decrease the false positive rate, Guo et al. [14] combined DPI and Deep Flow Inspection, which is considered a macroscopic analysis of the network traffic. As this technique does not read the packet's content nor headers, it is more effective than DPI when detecting encrypted packets or unknown protocols, increasing significantly the detection rate of probing and Denial-of-Service (DoS) attacks.

Encrypted data represents a challenge for DPI because the contents can not be identified, impeding the pattern matching. Sherry et al. [15] proposed a solution named by the authors as BlindBox, which consists of a rule generator. This technique generates malicious signatures according to the encryption key and a middlebox, which compares the rules against the packets' payload. With this, approach attacks using Secure Sockets Layer (SSL) could be identified.

Despite the BlindBox capacity to perform DPI in encrypted traffic, Yuan et al. [16] say it is currently unpractical due to to its expensive connection setup and propose a different system architecture to implement DPI in encrypted packets, preserving the privacy of the clients and increasing the amount of processed packets per second. To achieve this, they parse and encrypt packet payloads into tokens, which are fed to an encrypted rule filter and, using searchable encryption techniques, detect and act in response and in accordance with the suspicious traffic.

Besides the encrypted traffic, anonymizers also interfere within an investigation, impeding the determination of the source of given events. An example of such service is The Onion Router (TOR), which contributes to malicious people spreading malware and committing crimes. Saputra et al. [17] developed a system, consisting of DPI, to detect TOR's characteristics in packets, to drop it and avoid potential malicious traffic.

Such devices are commonly deployed in networks with dedicated resources, each requiring a copy of the data from network [32], and, as this data rapidly grows in high speed networks, processing and storing multiple copies of it may require expensive software and hardware resources. Ashraf et al. [32] introduced a Service-Oriented framework for heterogeneous Deep Packet Inspection and Analysis, which is capable of providing DPI services with a single copy of data, reducing significantly the CPU and memory utilization.

As DPI systems inspect all the contents of a packet, from any protocol, it is necessary to have a variety of rich and specific dissectors, which implies burdensome work for developers and may result in software vulnerabilities, such as the one identified as CVE-2014-4174, in Wireshark (1.10.x before 1.10.4), which allows attackers to execute arbitrary code [33]. Addressing this problem, Sommer et al. [33] proposed a framework named Spicy, which is a set comprised of a format specification language, a compiler toolchain and an Application Programming Interface (API) for DPI applications. Spicy has proven to be an efficient and robust dissector tool, useful for DPI. As a result, their research has also provided a new capability for developing powerful and reusable dissectors for DPI applications.

Huang et al. [34] studied different problems in DPI-enabled SDNs to minimize the delay of DPI processing and trip latency, proposing algorithms that can be divided into two phases. The first phase aims to solve the problem of a DPI proxy selection based on the ingress switch through which the packet entered the SDN. The second phase finds the shortest path to the selected proxy using a modified Dijkstra's algorithm. These algorithms achieved a better performance than the compared ones.

Nevertheless, malicious users are constantly improving their techniques as well and, thus, DPI systems need to frequently update their definitions of suspicious network activity. For this purpose, Trivedi and Patel [20] have developed a DPI system whose signatures and rules are updated with machine learning techniques, in conformity with the traffic it analyzes. Their tests showed that this implementation of DPI detects signatures in the shortest possible time and reduces man hours of work in creating rules and new signatures.

2.5. Network Forensics

When considering cybersecurity, the main goal of conducting network forensics is to acquire evidence to be to able to reconstruct and gather information about an incident, crime or cyberattack. In this case, evidence can be obtained from inspection of the network flow, log files from firewalls or hosts, Intrusion Detection Systems (IDS), Intrusion Prevent Systems (IPS) and so on.

Nonetheless, there are some challenges to forensics procedures when dealing with network security evidence [35]. Due to the high transmission rate in networks, data is generated very fast, enforcing the necessity to have large storage capacity devices. Another challenging consequence of this high rate of data generation is that evidence of an investigated event will correspond to smaller percentages of the dataset, thus forcing cybersecurity professionals to dig and mine large amount of data searching for strings, files or patterns that are not easily found.

Khan et al. [35] also cite data integrity as a network forensics challenge. Digital evidence may be easily manipulated, either accidentally or not, compromising the validity of the evidence. It can be avoided by copying the data and examining it as if it was the first obtained evidence, reducing the risks of data being corrupted or lost. In addition, it is easy to identify corrupted data by comparing its hash with the original's. Another feature of digital evidence that cooperate with an investigation is the fact that, depending on the type of supporting media being used, even if evidence is deleted, in some cases they can be recovered using appropriate tools.

When investigating a network event, many sources of data must be inspected (e.g., firewall, IDS, IPS, raw packets), each generating new data at a high rate. Analyzing this voluminous and heterogeneous data can be a cumbersome task and delay the investigation. To support investigators in this scenario, Vallentin et al. [36] proposed a platform named Visibility Across Space and Time (VAST), which provides both continuous ingestion of voluminous event streams and interactive query performance. It aims to achieve interactivity, scalability and expressiveness with its distributed architecture and powerful indexing technology, which allows the capture and analysis of the entire activity of the network. Their tests showed that VAST supports interactive investigation more efficiently than other current systems.

Different network technologies require different forensics procedures. As an example, Khan et al. [37] studies the emerging SDN technology and its forensics. The authors compare the implementation of SDN forensics with traditional network forensics, discussing the differences; highlighting potential locations for evidence acquisition in each layer of the technology architecture; and describing the challenges in SDN forensics.

Investigation Procedures

Forensics techniques are used to help to answer when, how, why and who is responsible for the investigated act. These answers can be obtained with some procedures after the incident is noticed. However, prior to the opening of an investigation, it is important to determine whether the event is worth the investigation or not. As resources, including the investigators, are limited, it may not be worth it to investigate small events that do not jeopardize important assets. Once it has been decided the event should be investigated, all data that might be useful to the investigation should be gathered and preserved. Considering an investigation taking place regarding cybersecurity incident, evidence may include firewall logs, packet flow, temporary files, proxy logs.

Different types of evidence require different analysis. In the case of a network event, it could be investigated with DPI, searching for evidence and information about the event. Reports should be created about the investigation, containing a detailed explanation of each step taken, so any independent investigator can repeat the process and try to achieve the same results in order to support or validate conclusions about an incident.

To guarantee the credibility of the evidence, it is important to maintain its integrity. Only people with direct relation to the investigation must have access to it, keeping original copies safe and never modifying them because of the risk of destroying it.

Another important action is to document all the people who had the custody of each evidence and what analysis each person carried out. This document is known as the Chain of Custody, where each person is a link of the chain, and provides a chronological trail of the evidence from when it was first discovered until its presentation in the court [38].

Remember that every person in the chain has responsibility for the evidence and must, likely, testify in court. It is preferable that the chain is no longer than it must be and no person not mentioned in the CoC should have access to the evidence. Another important part of the CoC is to prove the integrity and authenticity of the evidence [39] and, therefore, it is essential to grant reliability and admissibility to the evidence.

Flores and Jhumka [40] cite four principles the CoC should be in accordance with:

1. No action taken by any insider should change the evidence.
2. In circumstances where accessing original data is required, an explanation of the relevance and implications of such actions must be provided.
3. An audit trail, or similar record, of all events should be generated, collected and preserved. An independent third party should be able to examine those events and achieve the same conclusion.
4. The person in charge of the investigation must ensure the application of these principles.

To establish a Chain of Custody, it is necessary to have the names, signatures and dates associated to every event related to the evidence, along with its description. Any evidence whose CoC is not well established may be inadmissible in court [41], thus important work may be misinterpreted and not considered as evidence due to incorrect steps taken in investigation procedures.

According to Prayudi and Sn [38], the main challenge of digital CoC is the ease with which digital evidence may be accessed remotely, to perform an investigation anywhere and anytime, thus requiring an accurate and complete documentation. The authors also cite other challenges, such as the storage of digital evidence, with secure infrastructure, that fulfills the investigation's needs and meets the criteria set by the law.

3. Network Anomaly Detection

There are several different techniques applied on the flow of Honeynets [42], with the purpose of studying attackers' behavior, allowing a thorough analysis of the tools used by them, along with their motivation and vulnerabilities exploited. Such techniques can be applied on corporate networks to detect, prevent and react to attacks in different situations using criteria such as anomalies in a packet flow.

This section reviews string matching and header analysis, which are based on an analysis of the content of each packet using DPI.

3.1. String Matching

When inspecting a packet, useful information may be contained in its payload, and one effective way to detect the presence of signatures in it is using regular expression (RegEx), which provides a concise manner to identify strings, substrings and even patterns of characters.

Snort [43], one of the most used open-source IDS, for example, uses signature matching to generate an alert when an anomaly is found. Considering there is a unique expression in a packet that identifies an anomaly, a known attack can be detected with Snort. For instance, there is the buffer overflow in the Samba server (CVE-1999-0811) where the byte pattern 0xEB2F5FEB4A5E89FB893E89F2 [44] is detected, and it characterizes an attack exploiting that vulnerability.

As regular expressions allow the use of wildcards and ranges, it can also be used to search for packets addressed or originated from a specific subnetwork (when matching against the *destination* or *source* fields, respectively, of the datagram), with the purpose of investigating only the events related to that subnetwork. This also applies to the other fields in any layer of the packets, for example, filtering during an investigation for packets transmitted in a time range, from a specific Media Access Control

(MAC) address or packets of a specific protocol. Table 3 shows some other scenarios in which RegEx are useful.

Table 3. Possible scenarios where DPI is applicable.

Scenario	Description	Supporting References
Web application attack detection	Application layer attacks, such as Structured Query Language (SQL) injection, cross-site scripting and cross-site request forgery, are based on commands sent to the server by the attackers through forms and, therefore, regular expression may be used to detect characters that may identify these attacks.	[45–47]
Worms detection	Many worms, such as SQL Slammer and Nimda, have well-known signatures and, therefore, using regular expression to detect their occurrence in the network is effective.	[6,48,49]
Web inspection	Inspecting Hypertext Transfer Protocol (HTTP) packets for specific web contents or web pages accessed by specific users.	[15,50,51]
File inspection	Search for specific contents in files transmitted in a File Transfer Protocol (FTP) session.	[52–54]
DNS inspection	Inspecting Domain Name System (DNS) packets for specific accessed domains and possible malicious payload.	[54–56]
E-mail inspection	Search for specific words, in the Multipurpose Internet Mail Extensions (MIME) standard, contained in e-mails, along with attachments.	[32,53,57]

This string matching approach provides some advantages to an investigation, as it is simple and easy to understand what the detection tool is looking for and, thus, when a match is found for some signatures, it may be interpreted that an attack has been detected discarding false positives matching. It is also easy to add and share many signatures, creating a library of known malicious signatures and/or other string patterns to be searched for in a packet's payload.

Using RegEx in DPI, however, is not that simple and results in difficulties that make it unpractical in some cases. Firstly, because the string matching process must be, at least, as fast as the line speed to detect threats in real time. Secondly, the detection of a known malicious string may not always imply in the occurrence of an attack, as there are many signatures that also occur in a normal packet flow, resulting in too many false positive alerts overloading dashboards with incorrect information.

A stateful analysis of the payload string decreases the rate of false positive detections. Thereby, not only the payload of each packet is compared against the known malicious strings, but also the state of the conversation between the two devices. For instance, if supposedly malicious content is expected, then an alert should not be emitted because of its occurrence.

Sommer and Paxson [58] cite as an example the CVE-2000-0778, whose signature is :'*Translate: F*' and it is common in normal flows. The authors also describe the use of regular expression with context in DPI to reduce the false positive occurrence.

For the string matching process to be fast enough, it is necessary dedicated hardware and software. However, RegEx requires large bandwidth, many algorithms and other string matching implementations, which have been studied to solve this issue—among them, bloom filters [59] that accelerate the string matching process with a controllable false positive rate and finite state machines [60–62], such as Deterministic Finite Automaton and Nondeterministic Finite Automaton.

3.2. Header and Payload Analysis

Many network attacks cannot be detected with string matching, as they do not present a signature nor a pattern in the payload. It means that, for every attack, different payloads can be used so the signature matching process fail. In such cases, to detect these attacks, another approach has to be taken. One of them is to analyze packet headers content that may present anomalies, providing evidence of either an attack or a probe is taking place.

The following subsections present some review on how packet fields can be used in a network attack. Although attacks in the physical layer do exist, they are not in the scope of this review because its detection is not achieved with the inspection of the packets' contents.

3.2.1. Data Link Layer

Source and destination MAC addresses can be spoofed so actual addresses are changed to forged ones. This can be used, for instance, to perform MITM attacks. Detection of MAC spoofing can be achieved with an analysis of the *sequence number* field, in the MAC header of IEEE 802.11. As a rule, this field is incremented by one (limit 4096) for every transmitted frame. As reported by [63], the existence of a gap in this field for a sequence of packets of the same MAC address evidence the attack.

Carnut and Gondim [64] proposed a method for detecting MAC spoofed packets using Simple Network Management Protocol (SNMP) protocol and the second version of Management Information Base (MIB-II) specification. Using the management protocol, it is possible to count the bytes entering and leaving a given port, along with the amount of unicast and non-unicast packets entering and leaving the port. With these counters, the imbalance between the ARP requests and replies may be calculated and depending on the result, the ARP spoof attack is detected.

Kolias et al. [65] studied different attacks in IEEE 802.11 networks, classifying them as key retrieving, keystream retrieving, MITM and availability attacks. They also published a dataset containing the second layer fields that provide evidence of such attacks among normal traffic.

3.2.2. Network Layer

Time-To-Live (TTL) limits its lifespan to prevent a packet from circulating indefinitely in a network, but it can also be used maliciously. For example, an attacker can send a benign packet with a TTL large enough to reach the victim's network monitor, but not the victim's host. After that, the attacker sends another packet, with the same sequence number, but this time with malicious content and enough TTL to reach the victim. If the monitor ignores the "retransmitted" packet, it will not detect the attack, whilst checking every retransmitted packet may overload the monitor.

Source and destination IP addresses are easily spoofed hiding the source of an attack. One possible way to detect spoofed packets is analyzing the TTL field, which is an indication of how many hops there are between the two hosts. As TTL should not change drastically with time, recording this field for each host helps in the detection of IP spoofed packets. Templeton and Levitt [66] describe this techniques, among others, to detect spoofed packets that, when combined, achieve a high detection rate.

3.2.3. Transport Layer

The same IP address accessing several ports in a short period of time indicates a probe where an attacker is seeking for open ports and possible vulnerabilities to be exploited.

TCP flags can be used with an illegal combination to determine whether a port is open or not. For instance, packets that are used to start and finish a connection (SYN and FIN) should never be set at the same time. FIN should also not be the only flag set in TCP packet. Packets with these combinations or no TCP flags set indicate an anomaly and should be dropped. In addition, SYN flood is a common DoS attack that consists of the transmission of many TCP packets with the SYN flag set, opening many connections with the victim's machine to overload it. Table 4 reviews some of the other TCP flags' anomalies used in port scanning. Such TCP packets can be transmitted by bots using tools like Nmap.

Table 4. TCP flags used to scan for open ports and their respective response interpretation.

Flags Set	Response Interpretation
SYN	If a RST is received, the port is closed.
ACK	If the target does not respond, the port is filtered by a firewall. If a RST is received, it is not filtered.
FIN, URG & PUSH (Xmas Scan)	If the target does not respond, the port is open. If a RST is received, it is closed.
FIN	
No flags set	

3.2.4. Application Layer

The field analysis in the application layer is specific to the protocol being used and there is a big variety of application layer protocols. Table 5 summarizes some possible analysis to be performed on the protocols.

Table 5. Application layer protocols and their respective possible headers' analysis.

Protocol	Description
Telnet	The Data field, that holds the actual sent message, may be searched for brute-force attacks, when an individual tries several different and unsuccessful credentials; and for botnets such as Mirai, when detecting the presence of keywords.
FTP	The Request Command field shows, among others, the username and password used to connect to the server, allowing the detection of brute-force attacks.
SMB	The Path and File fields shows, respectively, the path and name of the file the client wants to fetch from the Server Message Block (SMB) server. Specific content in these fields may provide evidence of malicious activity, such as W32.IRCBot.
HTTP	The Request Uniform Resource Identifier (URI) field shows the URI the client wants to get, and the presence of some characters and/or keywords may provide evidence of the use of probe tools, like Nmap, or other malicious activities.
NTP	The Request Code field specifies the operation requested by the client. Specific values in this field may be used in a DoS attack.
HTTPS & SSH	As these protocols are encrypted, their fields cannot be analyzed, but only their metadata, such as the packet's source, time, etc.

4. DPI Applied to Honeynet Traffic and Attacks

In this section, DPI techniques are applied in the traffic destined to the high interaction Honeynet maintained for research purposes in the Network Laboratory at University of Brasilia ([67]). This Honeynet was used to generate statistics and different types of attacks were analyzed, detailed in the following subsections.

4.1. Description of the Architecture

The traffic used to show the results corresponds to the months from August 2016 to December of 2016. Figure 3 depicts the architecture of the Honeynet and its active honeypots.

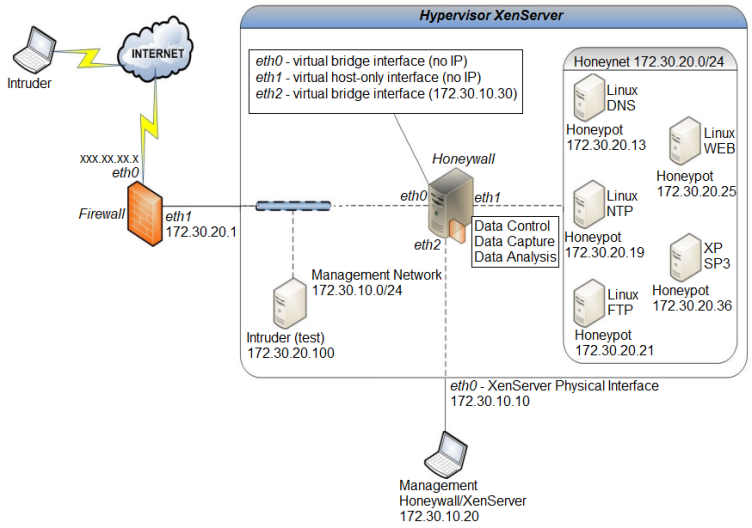

Figure 3. Honeynet architecture. [67].

Host IP 172.30.20.100 is set for test purposes. Basically, it is used to verify if the packets are being correctly captured and logged. Host IP 172.30.10.20 is the machine from where data is collected and pcap files are exported.

Using TShark tool [68], pcap files are converted to csv format. Each line in the csv file represents a packet received or transmitted at the Honeynet and columns represents the fields of the analyzed protocols and layers (see Table 1 for details). After the conversion, the csv files are processed, indexed and visualized using the Elastic Stack [69], through which the analysis is performed and statistics generated. It is important to state that the CoC procedures and its principles (described in Section 2.5) were followed, creating copies of the evidence and documenting every investigation carried out.

As noted in Figure 4, which presents sample lines of the csv file, the character '}' is used as a separator, because some fields, such as *telnet.data* in the first line of the figure, contains the character ',' (default separator) and, for the purpose of keeping constant the number of columns in every line, the separator must be a character not commonly present in the analyzed fields. The first line also shows that packets originated in the Honeynet are not georeferenced, as the IP addresses of the honeypots are private.

```
1477958972}eth:ip:tcp:telnet}172.30.20.36}}}8x.21x.36.216}}}23}45058}0x0018}252}65}\x0a,\x0dlogin: }...}
1477959442}eth:ip:tcp:ssh}21x.8x.139.56}France}Paris}172.30.20.21}}}59183}22}0x0018}}}}}}}}}}}}}}}}}...}
1477959932}eth:ip:tcp}9x.4x.130.174}Belarus}Baranavichy}172.30.20.36}}}48701}23}0x0002}}}}}}}}}}}}}...}
```

Figure 4. Sample lines of the exported csv file. Reticence is used to suppress the repetition of empty fields.

The fields analyzed in this research, in the order presented in Figure 4, are *frame.time_epoch*, *frame.protocols*, *ip.src*, *ip.geoip.src_country*, *ip.geoip.src_city*, *ip.dst*, *udp.srcport*, *udp.dstport*, *tcp.srcport*, *tcp.dstport*, *tcp.flags*, followed by all fields supported by TShark of the application layer protocols Telnet (in the first line, respectively, *telnet.cmd*, *telnet.subcmd* and *telnet.data*), hl Server Message Block (SMB), Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Network Time Protocol (NTP).

As the Honeynet traffic comes from the Internet, analysis on layer 2 was not very informative, thus the only inspected field of the data link layer was the *arrival time*, which was analyzed and considered to understand the temporal behavior of the attackers. Detail about the timeline of the collected information in this case is shown in Figure 5.

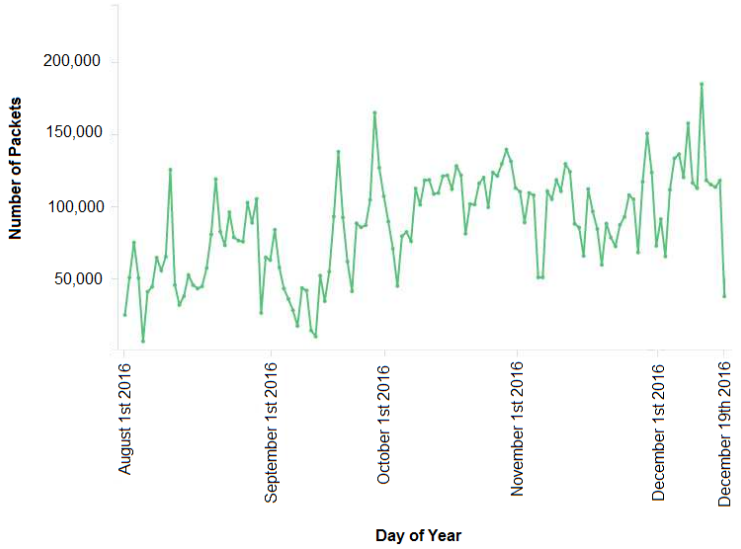

Figure 5. Numbers of packets sent to the Honeynet per day.

During 141 days of data collected, the average number of packets per day was 81,052 packets per day. Considering this data with previous one collected months before ([70]), it was possible to note that the average number of packets transmitted to the Honeynet has increased significantly. This is a strong indicator that the Honeynet was receiving many more attacks.

In the following subsections, more results using DPI regarding data captured and its correspondence with layers of TCP/IP stack are described.

4.2. Layer 3 Header Analysis

As a start, inspection on the network layer consisted of analyzing the source and destination IP addresses on each packet; thus, using this information georeferencing the origin of the attack and determining which honeypots were the most targeted. There was a total of 119.620 unique IP addresses that connected to the Honeynet and some of them may have been spoofed. Therefore, the expected number of IP addresses of the attacks are possibly lower, which basically does not interfere in the following analysis. For anonymity and privacy purposes all of the last digits of the first two octets of valid IP addresses were substituted by an 'X', for them not to be fully identified by the public.

4.2.1. Classification of Traffic by Its Geoinformation

Analysis of the header of the third layer of the packets shows that China is the most representative source of the attacks to the Honeynet. The fifteen IP addresses that most attacked the network are all from this country and originated from the cities Shenzhen, Nanchang and Nanjing.

Figure 6 shows the relative contribution of the most frequent countries and IP address for the total traffic, along with the ports to where each IP sent its packets. Such analysis provided information about the service attack type and, although the majority of the analyzed packets where directed to port 22, the most targeted port was 23, which suggests that a greater amount of traffic to port 23 is distributed in a higher number of less frequent IP addresses, while traffic to port 22 is concentrated in the most frequent IP addresses (more details about analysis of these ports are provided in Section 4.3.1).

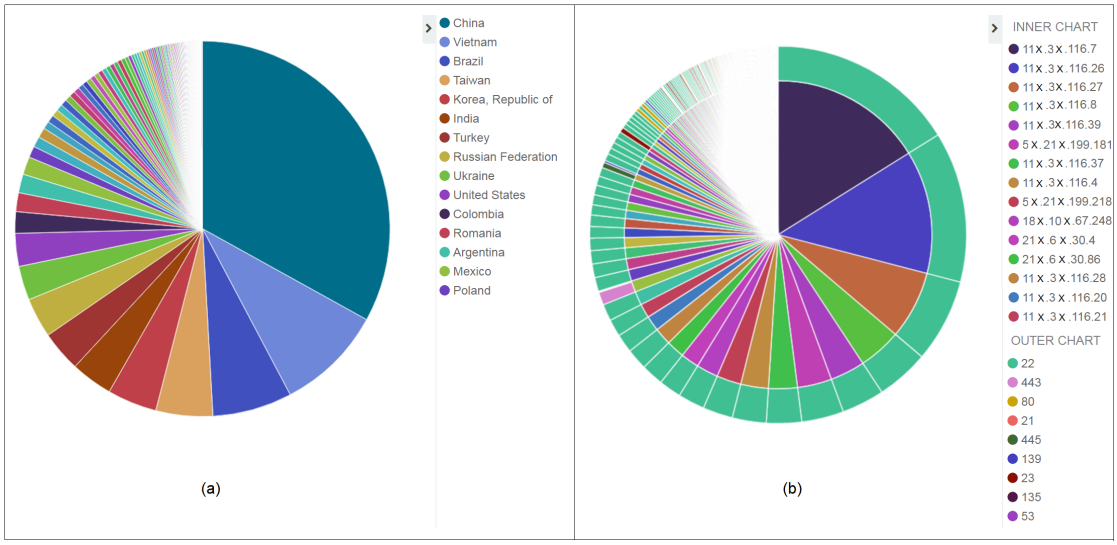

Figure 6. Classification of traffic according to its source, from August 2016 to December 2016. (**a**) indicates source country of the attacks; (**b**) indicates IP source, in the inner chart, and their respective service attacked type represented in the outer chart.

During the analysis of layer 3, it became evident the similarity of the IP addresses from the city Shenzhen. This is a strong indication that the hosts using those IP addresses probably belong to the same subnetwork and, therefore, it either corresponds to a network with several infected devices (i.e., as part of a botnet) or to a network setup for malicious purposes. Table 6 shows these findings in the collected data.

Table 6. The fifteen countries and IP addresses that most attacked the Honeynet.

Source Country	Number of Packets	IP Address (City)	Number of Packets
China	4,027,465	11x.3x.116.7 (*Shenzhen*)	590,553
Vietnam	1,122,658	11x.3x.116.26 (*Shenzhen*)	467,034
Brazil	836,715	11x.3x.116.27 (*Shenzhen*)	261,392
Taiwan	596,842	11x.3x.116.8 (*Shenzhen*)	165 551
Republic of Korea	521,197	11x.3x.116.39 (*Shenzhen*)	133,976
India	439,838	5x.21x.199.181 (*Nanjing*)	132,058
Turkey	428,563	11x.3x.116.37 (*Shenzhen*)	110,694
Russia	417,440	11x.3x.116.4 (*Shenzhen*)	106,413
Ukraine	358,706	5x.21x.199.218 (*Nanjing*)	94,049
United States	345,360	18x.10x.67.248 (*Nanchang*)	70,285
Colombia	225,551	21x.6x.30.4 (*Nanchang*)	70,285
Romania	197,977	21x.6x.30.86 (*Nanchang*)	66,535
Argentina	196,200	11x.3x.116.28 (*Shenzhen*)	58,361
Mexico	187,786	11x.3x.116.20 (*Shenzhen*)	54,130
Poland	118,792	11x.3x.116.21 (*Shenzhen*)	52,454

Like in the previous months ([70]), China still corresponds to most of the traffic, although the percentage of packets from this country has lowered. On the other hand, countries such as Taiwan, Turkey and Colombia had attacks arising and became responsible for a greater percentage, while the Netherlands and Germany no longer appeared as the top 15 countries based on the historical data. A total of 163 different source countries appeared as the origin of the attacks directed to the Honeynet.

Regarding information provided in Table 6 and, in accordance with Figure 6b, the 15 IP addresses that most attacked the Honeynet correspond to the source country being China from three different cities. Regarding the historical analysis of the Honeynet, it shows opposed information when compared to the previous months ([70]), where several countries appeared as the most frequent IP source countries

with little difference among the number of packets. Based on the newer data collected and analyzed, this indicates that the Honeynet now has more attention from countries such as China, Vietnam and Brazil, which together correspond to almost 50% of the traffic.

Although the majority of the most frequent IP addresses were also georeferenced to the city of Shenzhen, in the previous dataset ([70]), host 11x.3x.116.21 was the source corresponding to the greatest percentage of the attacks, and is now in the 15th position, giving the first place to 11x.3x.116.7, which was not even in the top 13.

4.2.2. Classification of Traffic towards Its Destination in the Honeynet

Besides the different services provided by each honeypot, the most important difference between the most targeted honeypot and the others is its operating system. While honeypot host 172.30.20.36 was running a Windows XP system, the rest of the honeypots run Linux based OS. This analysis suggests a preference of the attackers for the XP system, which is, of course, an outdated system; thus, it is considered easily exploitable because its patch life cycle has ended. Although the analysis considers this system outdated, there are still many real networks around the world where XP is widely deployed, as recent WannaCry attacks showed [71].

Among the Linux honeypots, the host running FTP was the most attacked, followed by the ones running DNS, WEB and NTP services, respectively, as in Figure 7.

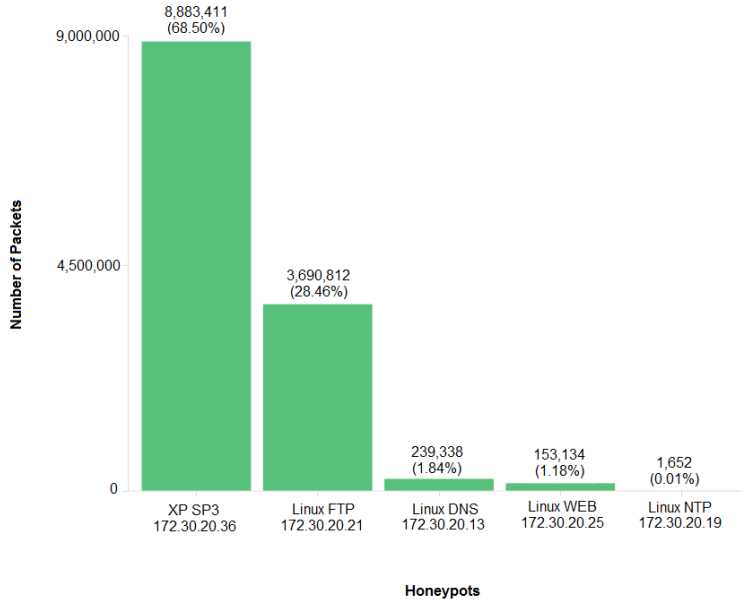

Figure 7. Most attacked honeypots.

4.3. Layer 4 Header Analysis

Among other information, an investigation that the transport layer will return is the port each packet is directed to. Associating well-known ports to their correspondent protocols, it is possible to determine which were the application layer protocols that the attackers most tried to exploit. Besides this analysis, investigation of layer 4 may also provide evidence of the presence of anomalies in either a TCP connection or bad User Datagram Protocol (UDP) packets.

4.3.1. Classification of Traffic by Its Destination Port

The attacks to the Honeynet were organized in relation to the port trying to be exploited in all of the honeypots. Figure 8 shows the distribution of ports destination of the attacks. In the data collected and analyzed, there is a significant preference for ports 23 and 22.

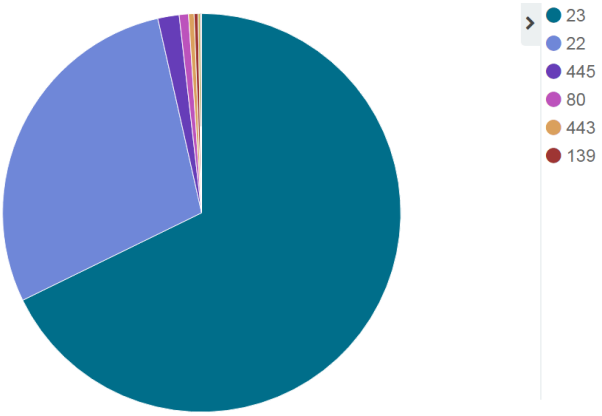

Figure 8. Classification of traffic according to its destination port.

Table 7 presents the absolute number of packets directed to each port of the honeypots during the time of the collected data. From the data analyzed and based on the architecture of the Honeynet, it may be inferred that ports 23 and 22 together correspond to 90.1% of the attacks, respectively Telnet and SSH services and protocols.

Table 7. The six most attacked ports.

Port Number	Number of Packets
23	8,209,474
22	3,474,610
445	209,689
80	91,459
443	53,950
139	39,934

Detailing the analysis performed on those ports and protocols, the attacks on port 22 corresponded mainly to SSH brute force. Once the payload of SSH packets are encrypted, the log analysis of the attackers had to be done. Thus, it was possible to collect information about the users and passwords trying to login into the system. The same consideration applies to the traffic on port 443 (HTTPS protocol) and the payload cannot be understandable without log analysis.

On the other hand, the attacks directed to port 23 (Telnet), 445 (microsoft-ds), 80 (HTTP) and 139 (netbios-ss), are not encrypted and, therefore, they can be analyzed with DPI. Although not present in the top six destination ports, malicious users tried to exploit port 21 (FTP) and also tried use port 123 (NTP) to perform a DoS attack on hosts outside the Honeynet. These protocols exchange the messages in plain text as well, allowing the study of the attackers' behavior.

4.3.2. TCP Flag Anomalies

An analysis of the behavior of the less frequent attackers may also return useful information, as they might have noted they were targeting a controlled environment or a host that did not have what they were expecting. On the contrary of what was observed in the collected data, each IP address shown in Figure 9 transmitted only four packets to the Honeynet and this was considered a strange behavior based on all other data analyzed.

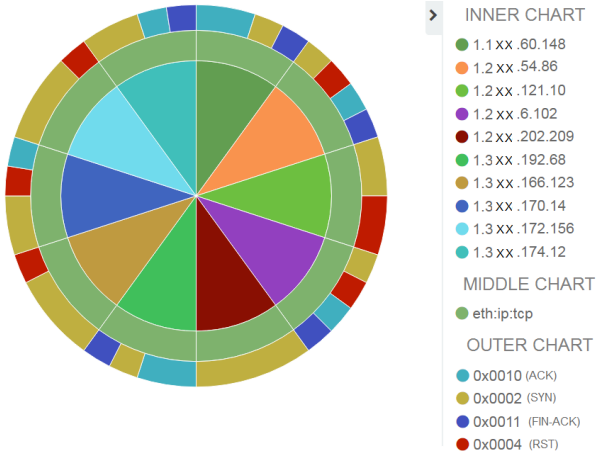

Figure 9. Some of the IP addresses that sent four packets to the Honeynet (inner chart); protocol used (middle chart) and their corresponding TCP flags (outer chart).

Although none of the IP addresses reached layer 5, some of them presented anomalies in the TCP flags such as 1.2xx.202.209, that transmitted only SYN packets. This behavior may be associated to a port scan or a TCP SYN flood, but these activities usually have a much greater number of packets transmitted.

Other IPs transmitted a significant amount of RST packets, which is not common in normal TCP connections. The performed analysis could not identify what they were trying to achieve due to the short amount of packets and none of them sent information using layer 5. Furthermore, the first octet of all IP addresses is identical, whereas the second seems to be increasing. All of the IP addresses were trying to connect to port 23 and are georeferenced to countries in Southeast Asia. The fact that the IPs are georeferenced to different countries indicates that maybe they are somewhat not part of the same source, but the similarities in the first two octets of the IPs and its behavior may provide evidence that they are somehow related, possibly belonging to the same attacker, who spoofed the source IP addresses.

4.4. Layer 5 to 7 Payload Analysis

Different application layer protocols have different fields and information, and can be used to exploit different vulnerabilities. Thus, each protocol requires a specialized investigation on its fields. In this section, inspection and analysis of payload is performed on some of the protocols the attackers tried to exploit.

4.4.1. Traffic Analysis of Port 23

Recently, the code of Mirai botnet was made available to the public. Mirai can be considered a kind of malware that infects IoT devices running BusyBox. Regarding its capabilities as a botnet, information about it emerged in September 2016 when it was used in a Distributed Denial of Service (DDoS) attack against a security blog [72] as one of the first targets. One of the main characteristics of Mirai is that it works by brute forcing, over Telnet, weak and default credentials on devices. Once it gains controls of the device, it reports the infection to a command and control server and the device is now part of a botnet. Once many owners of devices directly and indirectly connected to the Internet never change the default user and passwords, Mirai could infect over 380 million devices and then it was used in one of the largest DDos attacks known so far, generating at least 1.1 Tbps on OVH (Roubaix, France), a French cloud computing company [72].

The attack commands usually follow the sequence below:

`{username}` and `{password}`, then `enable` or `system` or `shell`, or `sh`, then

`/bin/busybox MIRAI`,
where *{username}* and *{password}* are those present in the Mirai dictionary, and the following commands are used to detect if the target is not a router or common honeypot, like Cowrie.

If the login succeeds, it runs the following commands in order to download the malware's payloads and then scan for new targets:

```
'busybox tftp' -r [MalwareFileName] -g [IPsource]
'busybox tftp' -g -l 'dvrHelper' -r [MalwareFileName] [IPsource].
```

After execution, the malware is self-deleted to avoid early detection and leaves the process running in memory of the compromised device.

Exploring the Telnet protocol payload, it was possible to read and analyze data to determine which were the most guessed credentials that appear in the *data* field of the protocol. The most guessed usernames and passwords are shown in Figure 10a, from which can be noted a prevalence of the username *root*. Together with 14 other guessed usernames, analysis shows they correspond to 98.86% of usernames, thus implying a concentration and a little variety of usernames guessed. This analysis may be used to create signatures used by perimeter defense mechanisms.

Regarding the prevalence of a password, there is a bigger variety. Fifteen passwords correspond to 76.06% of the data and the distribution of the guesses is closer to a uniform view as seen in Figure 10b. Table 8 contains more detail about the number of times each username and password has been guessed by Mirai infected devices towards the Honeynet.

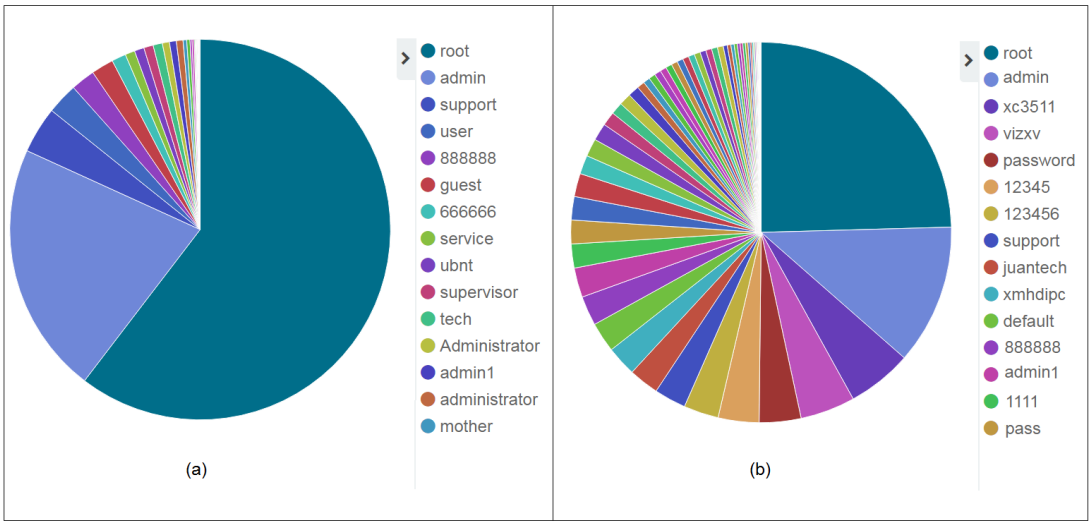

Figure 10. Users (**a**) and passwords (**b**) guessed by the Mirai botnet.

The usernames and passwords displayed in Figure 10 are hardcoded in Mirai source code and are either commonly used credentials in general devices, such as `root` and `123456`, or are default credentials defined by popular IoT devices manufacturers, as `xc3511` and `xmhdipc`.

4.4.2. Evaluating Mirai Details

To analyze and check the modus operandi of the data regarding Mirai botnet towards the Honeynet, a social analysis network as a graph visualization was created as seen in Figure 11.

In the corresponding graph, the green nodes represent IP addresses while the orange nodes represent the contents of Telnet *data* field. The relationship between two nodes indicates which data each IP transmitted and the thicker the edge, the more packets with this data value were transmitted by this IP.

Although not all IP addresses and Telnet *data* appear in Figure 11 due to the limited number of nodes and relationships, it may be inferred that some IP addresses transmitted many times *enable*, *system*, *shell* and *sh* commands and this kind of data corresponds to the behavior of Mirai, thus providing evidence that this malware is widely available on the Internet with compromised devices.

Table 8. The fifteen most guessed usernames and passwords by the Mirai botnet.

Username Guessed	Number of Packets	Password Guessed	Number of Packets
root	257,044	root	104,522
admin	91,141	admin	50,649
support	16,825	xc3511	32,434
user	11,272	vizv	19,825
888888	8761	password	15,087
guest	8076	12345	14,762
666666	5160	123456	12,620
service	3513	support	11,493
ubnt	3501	juantech	10,897
supervisor	3402	xmhdipc	10,880
tech	3360	default	10,827
Administrator	2562	888888	10,689
admin1	2488	admin1	10,678
administrator	2403	1111	8673
mother	1320	pass	8578

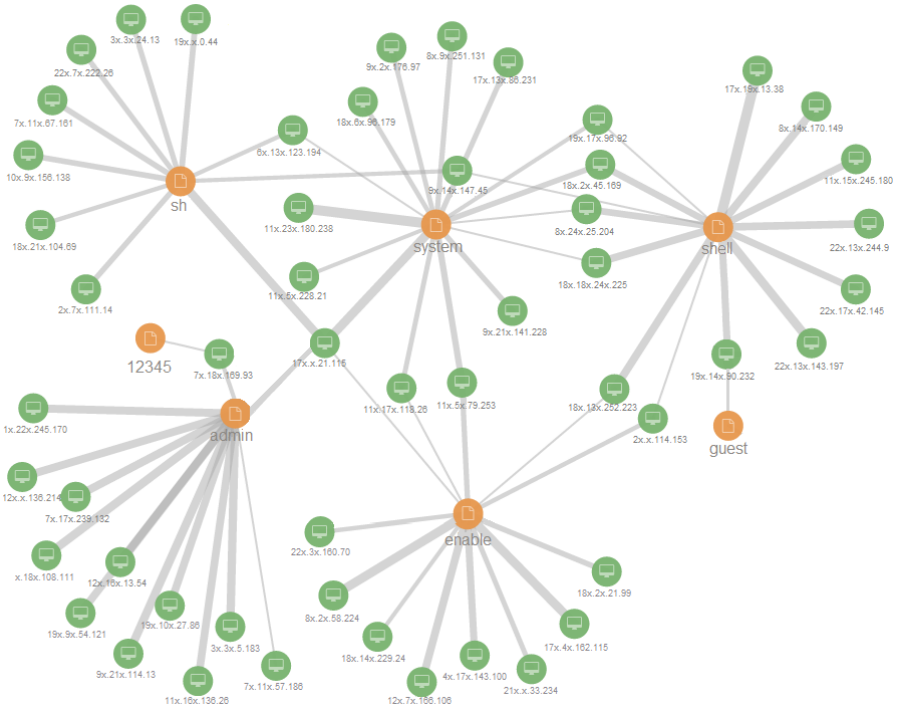

Figure 11. Evidenced *modus operandi* of Mirai botnet.

Though botnets are not easily detected, as they continuously evolve and adapt to detection techniques [73], Mirai transmits specific and identifiable commands, such as `/bin/busybox MIRAI`. Therefore, DPI, and other devices deployed to improve the network security, may be configured to drop packets containing these suspicious strings. In addition, as Mirai has hardcoded credentials to be guessed, the appearance of many packets transmitted with these usernames and passwords provide

evidence of malicious activity. This analysis raises awareness regarding the importance of changing default credentials and having strong passwords.

4.4.3. Traffic Analysis of Ports 445 and 139

W32.IRCBot-TO is a malware that creates another kind of botnet that opens a backdoor exploiting a buffer overflow vulnerability for Windows systems (CVE-2006-3439), allowing attackers to remotely execute code via a crafted Remote Procedure Call (RPC) message. Most of the traffic from these hosts infected with the malware of this botnet towards the Honeynet were destined to ports 445 and 139. The worm spreads itself using the Internet Relay Chat (IRC) protocol as a backdoor, which grants the botmaster access to the infected machines, and can be used, for instance, to collect information from the infected machine, perform DDoS attacks and terminate and run arbitrary processes [74].

As modus operandi, it downloads and executes a binary file named *netadp.exe* and then scans for other vulnerable victims. For these protocols, the analyzed fields were *path* and *file*, which are usually assigned in the presence of this malware to, respectively, `\\<target-ip>\IPC\$`, `\\browser` or `\\srvsvc`.

In the following, there is the sequence of packets transmitted by an attacker (SEQ 1), whose IP was 9x.4x.53.209 (georeferenced to Ireland) towards the XP honeypot. The first number in each line is the transmitted packet number, and the arrows represent its direction.

```
SEQ 1

523379 <-> 9x.4x.53.209 TCP 1963 - 172.30.20.36 TCP 139 [SYN, SYN,ACK]
523380 -> SMB Negotiate Protocol Request
523385 <- SMB Negotiate Protocol Response
523387 -> SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
523388 <- SMB Session Setup AndX Response, NTLMSSP_CHALLENGE, Error:  STATUS_MORE_PROCESSING_REQUIRED
523389 -> SMB Session Setup AndX Request, NTLMSSP_AUTH, User:  \
523390 <- SMB Session Setup AndX Response
523391 -> SMB Tree Connect AndX Request, Path:  \\<honeypot-public-Ip>\IPC$
523392 <- SMB Tree Connect AndX Response
523393 -> SMB NT Create AndX Request, Path:  \srvsvc
523394 <- SMB NT Create AndX Response, FID: 0x0000, Error:  STATUS_ACCESS_DENIED
523394 <- SMB NT Create AndX Response, FID: 0x0000, Error:  STATUS_ACCESS_DENIED
523397 <- SMB NT Create AndX Response, FID: 0x800e
523398 -> DCERPC Bind:  call_id:  1, UUID: SRVSVC
523399 <- SMB Write AndX Response, FID: 0x800e, 116 bytes
523401 -> SMB Read AndX Request, FID: 0x800e, 1024 bytes at offset 0
523402 <- SMB Bind_ack:  call_id:  1, result:  Provider rejection
```

After the TCP handshake, the attacker tried to connect to the victim and then download and execute *netadp.exe*, using the IPC share to connect to the SRVSVC pipe. Host 9x.4x.53.209 did not succeed in its attack because it had its access denied by the honeypot. As reference, there is another sequence of transmitted packets (SEQ 2) obtained from the work of Gu [75]. This sequence represents a successful attack by W32.IRCBot-TO.

```
SEQ 2

6 <-> <infector-ip> TCP 2971 - <honey-ip> 445 [SYN, SYN,ACK]
13 -> SMB Negotiate Protocol Request
14 <- SMB Negotiate Protocol Response
17 -> SMB Session Setup AndX Request, NTLMSSP_AUTH, User:  \
18 <- SMB Session Setup AndX Response
19 -> SMB Tree Connect AndX Request, Path:  \\<honey-ip>\IPC\$
20 <- SMB Tree Connect AndX Response
21 -> SMB NT Create AndX Request, Path:  \browser
22 <- SMB NT Create AndX Response, FID: 0x4000
23 -> DCERPC Bind:  call_id:  0 UUID: SRVSVC
24 <- SMB Write AndX Response, FID: 0x4000, 72 bytes
25 -> SMB Read AndX Request, FID: 0x4000, 4292 bytes at offset 0
26 <- DCERPC Bind_ack
27 -> SRVSVC NetrpPathCanonicalize request
28 <- SMB Write AndX Response, FID: 0x4000, 1152 bytes
```

```
29 -> SMB Read AndX Request, FID: 0x4000, 4292 bytes at offset 0

Initiating Egg download
30 <-> <honey-ip> TCP 1028 - <infector-ip> 8295 [SYN, SYNACK]
34-170 114572 byte egg download ...

Connecting to IRC server on port 8080
174 <-> <honey-ip> TCP 1030 - 66.25.XXX.XXX 8080 [SYN, SYNACK]
176 <- NICK [2K|USA|P|00|eOpOgkIc]\r\nUSER 2K-USA
177 -> :server016.z3nnet.net NOTICE AUTH
         :*** Looking up your hostname...\r\n'' ...
179 -> ...  PING :B203CFB7
180 <- PONG :B203CFB7
182 -> Welcome to the z3net IRC network ...

Joining channels and setting mode to hidden
183 -> MODE [2K|USA|P|00|eOpOgkIc] +x\r\nJOIN ##RWN irt3hrwn\r\n

Start scanning 203.0.0.0/8
185 -> ....scan.stop -s; .scan.start NETAPI 40 -b -s;
       .scan.start NETAPI 203.x.x.x 20 -s;
       .scan.start NETAPI 20 -a -s;.scan.start SYM 40 -b -s;
       .scan.start MSSQL 40 -b -s\r\n...
191 -> 203.7.223.231 TCP 1072 > 139 [SYN]
192 -> 203.199.174.117 TCP 1073 > 139 [SYN] scan,scan...
```

Although in SEQ 1 of packets the attack was not successful, the similarity in both sets of traffic is evident and corresponds to the W32.IRCBot modus operandi (SEQ 2). If infection is successful after the malware is installed, it connects to the IRC server and scans for other potential targets. It also creates the following registry entry to execute *netadp.exe* on startup of the infected host:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Network Bridge
<System>\netadp.exe
```

The presence of this entry in the registry is an Indicator Of Compromise (IOC), and implies the system has been infected by W32.IRCBot-TO.

4.4.4. Traffic Analysis of Port 80

For this kind of payload, the inspected fields analyzed were *HTTP request method*. These fields inform the HTTP verb used in the packet and *HTTP request URI* contains the URI to which the client intends to connect.

Within this investigation, automated malicious requests sent to the HTTP honeypot trying to gather useful information for a potential further exploitation were detected. Table 9 shows some of these requests and the tools with which they are associated. Most of the data analyzed in this case correspond to bots that also scan for vulnerabilities in port 443 and, therefore, part of the encrypted traffic to this port may correspond to them.

Besides scan requests, a worm known as *The Moon* was also caught in the traffic towards the Honeynet. This worm infects some Linksys routers by exploiting an authentication bypass vulnerability.

The Moon sent a `GET /HNAP1 HTTP/1.1` to the honeypot. This is used to identify the router's model and firmware's version. Then, it transmitted *tmUnblock.cgi* and *hndUnblock.cgi* scripts that allows execution of commands with no authentication required. After the infection, the worm uses the router to spread itself to other vulnerable devices.

The HTTP requests associated with the Moon worm and the ones displayed in Table 9 can not correspond to legit traffic, and their presence indicates suspicious activity. Therefore, the Deep Packet Inspection device should block these and similar requests. Addressing other exploitations in this port, DPI can also be implemented, for instance, in application layer firewalls, to detect and respond to malicious packets, such as strings that may evidence code injection.

Table 9. Examples of malicious HTTP requests received and their description.

HTTP Request	Description
`HEAD http://18X.16X.113.82/check_proxy HTTP/1.1`	Scanning what kind of proxy it could be in use
`HEAD /robots.txt HTTP/1.0`	Scanning what bots are blocked or allowed by the server
`GET /muieblackcat HTTP/1.1` `GET //pma/scripts/setup.php HTTP/1.1`	*muieblackcat* is a bot that scans for PHP vulnerabilities

`GET /w00tw00t.at.blackhats.romanian.anti-sec:) HTTP/1.1`
`GET /phpmyadmin/scripts/setup.php HTTP/1.0`
`GET /dbadmin/scripts/setup.php HTTP/1.1`
`GET /mysqladmin/scripts/setup.php HTTP/1.1`
`GET /admin/phpmyadmin/scripts/setup.php HTTP/1.1`
`GET /admin/pma/scripts/setup.php HTTP/1.10`
`GET /MyAdmin/scripts/setup.php HTTP/1.1934`
	ZmEu is a bot that scans for phpMyAdmin vulnerabilities. It also performs SSH brute-force.

`GET /nmaplowercheck1487075443 HTTP/1.1` `GET /nice%20ports%2C/Tri%6Eity.txt%2ebak HTTP/1.0`	*Nmap* probes for information about the server.

4.4.5. Traffic Analysis of Port 21

The *FTP request argument* field contains the password when *FTP request command* is equal to *PASS* and the username when *FTP request command* is equal to *USER*. After analyzing these fields, it was possible to generate statistics about the credentials with which malicious users tried to connect to the FTP server.

When brute-forcing FTP service, most attackers tried to connect as anonymous using standard credentials. Traditionally, anonymous accounts accept any string as a password, it also being common to request for an e-mail as authentication. This FTP behavior explains the appearance of some standard password and e-mail addresses in the password fields shown in Table 10.

One interesting detail in this analysis is that one attacker, with IP address georeferenced to the city Brasilia, tried to connect to the anonymous FTP, but using the actual personal and corporate e-mail addresses of the local network administrator and later the credentials of one professor from the department where the Honeynet is installed. This attacker also guessed nonexistent e-mail addresses in the University's domain and most other common credentials. The main guess in this case is that it probably corresponds to an attacker with insider's knowledge of the network, but he was not aware of the Honeynet itself.

Figure 12 shows the distribution of usernames and passwords guessed. As in the Mirai botnet analysis (Section 4.4.1), a few usernames concentrate a high percentage of the total of guesses, while the passwords guessed are more equally distributed in a higher amount of different strings. However, as the distribution shows, there still is a preference for some kinds of usernames and passwords.

Anonymous access is by default enabled in FTP servers and should be disabled, as it allows users to upload, for instance, illegal files and copyright protected material without authentication. The DPI analysis of traffic to this protocol showed that many users tried to connect to this service using anonymous access, probably with malicious intentions.

To secure FTP servers, authentication and strong password requirements should be used, as attackers may easily and rapidly guess many weak credentials, as shown in Figure 12. Therefore, security devices should block many sequential and unsuccessful password guesses coming from the same source. Furthermore, attackers may use social engineering to obtain insider's knowledge and guess related credentials. Thus, legit users of access to limited services should be oriented regarding the disclosure of personal information, in order to prevent such exploitation.

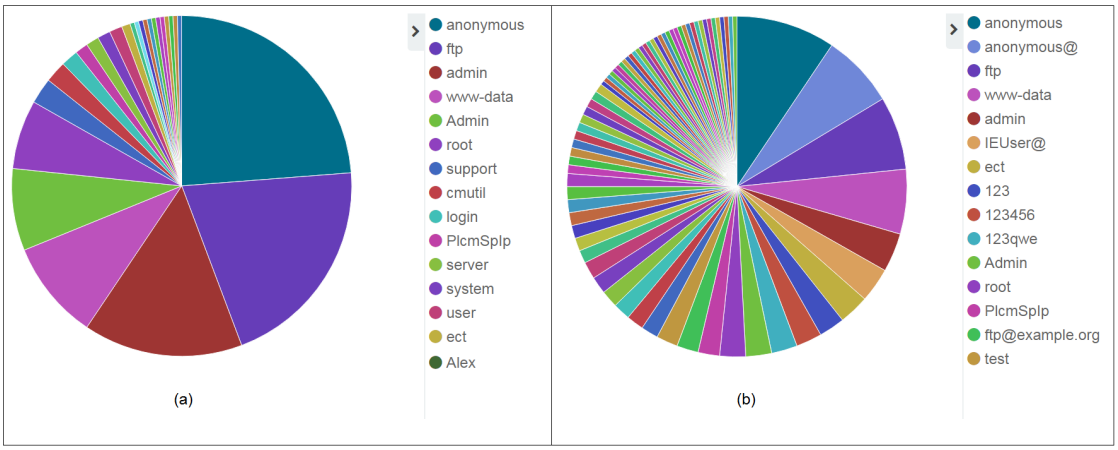

Figure 12. Users (**a**) and passwords (**b**) guessed in the FTP brute force.

Table 10. The fifteen most guessed usernames and passwords in the FTP brute-force.

Username Guessed	Number of Packets	Password Guessed	Number of Packets
anonymous	58	anonymous	23
ftp	50	anonymous@	17
admin	37	ftp	17
www-data	23	www-data	15
Admin	19	admin	9
root	16	IEUser@	9
support	6	ect	7
login	4	123456	6
PlcmSpIp	3	123qwe	6
server	3	Admin	6
system	3	root	6
user	3	PlcmSpIp	5
ect	2	ftp@example.org	5
Alex	1	test	5

4.4.6. Traffic Analysis of Port 123

Monlist is a remote command that requests from a NTP server a list of the most recent machines the server has interacted with, up to a maximum of 100 UDP datagrams with 440 bytes of payload each [56]. The response provides statistics useful for debugging, but may also be used by attackers as a reconnaissance or reflective DDoS tool.

Attackers can send small request packets to an NTP server with the IP source address spoofed with the value of the victim's, who receives larger response packets. By these means, the attacker amplifies the consumed bandwidth by a factor that may vary from 556.9 up to 4670.0 [56]. This vulnerability is identified as CVE-2013-5211.

Analyzing the *NTP private request code*, as in Figure 13a, two different requests were made to the NTP honeypot. Most of them were `MON_GETLIST_1` (code 42), associated with the vulnerability of the possibility of amplification, and just two occurrences of `REQUEST_KEY` (code 32).

Filtering out the packets with request code equal to 32, it was noted that several different IP addresses sent the command to the NTP honeypot, of which most had the *NTP transmit timestamp* field equal to *24 November 2004 10:12:11.444111000*, as presented in Figure 13b's inner chart.

The appearance of the same outdated timestamp in packets with different sources indicates they were actually transmitted by the same attacker, who was believed to spoof the source IP address, as seen in Figure 13b, who are supposed to be the victims of NTP attacks. Analysis in this case showed

that the similarity between some of the IP addresses implied that the attacker had targeted hosts in the same subnetwork.

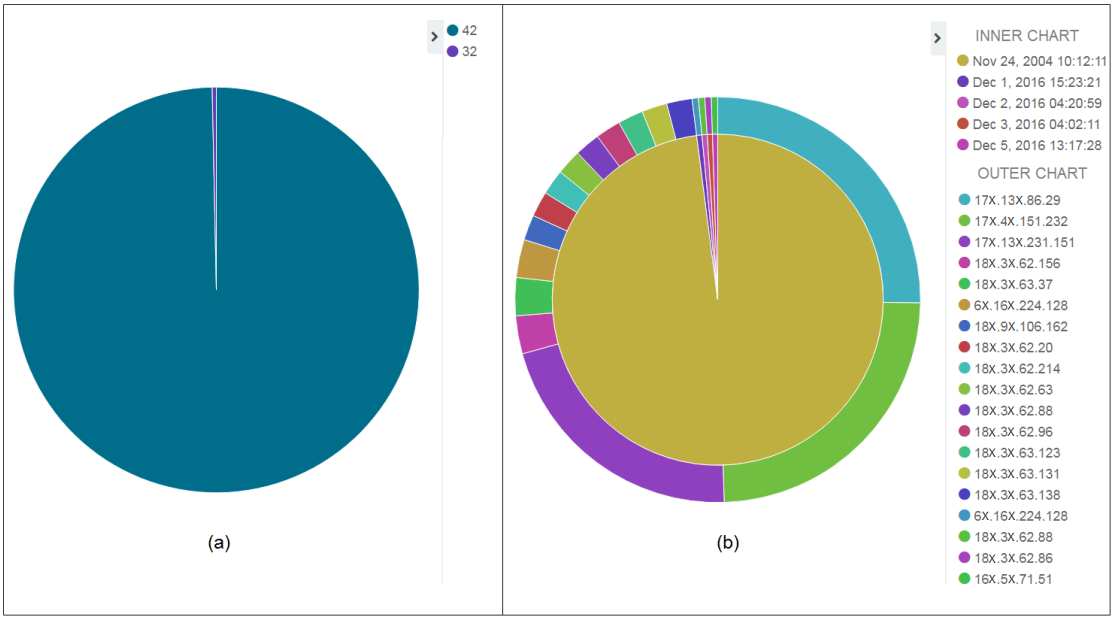

Figure 13. NTP Analysis. Most frequent NTP request codes (**a**); and NTP transmit timestamp (inner chart) and the respective IP source believed to be spoofed (outer chart) (**b**).

Furfaro et al. [76] cite this exploitation as a common means to perform reflective DDoS attack, with great bandwidth exhaustion power. To avoid this exploitation, *Monlist* command should not be permitted to public hosts, but only internally, for management purposes, and DPI devices should drop packets with NTP request code equal to 42. Other defense mechanisms include upgrading NTP servers to version 4.2.7 p26 or later, removing the *Monlist* function, and implementing BCP 38 (RFC 2827) on the network [77]. This analysis reinforces, therefore, the importance of having updated systems to prevent malicious activity.

5. Conclusions

Checking the contents of attacks directed to a Honeynet is considered a good approach for evaluating threats. DPI has proven to be effective in the detection of the anomalies shown in this article, also providing information about the attackers' behavior and, although it faces challenges and possible misuses depending on the source or the analysis, its use can help investigators to detect anomalies in any layer in their network.

Contributions of this research rely on layer-by-layer analysis of malicious traffic, studying anomalies present in fields specific to each analyzed protocol and generating statistics regarding the behavior of the attackers. With this kind of analysis, it is possible to learn new techniques to prevent, detect and respond to similar activities, as it provides evidence of the modus operandi of the exploitation of common vulnerabilities that may be present in any network. In addition, the use of such technologies helps to detect new kinds of attacks, allowing security experts to warn about new forms of anomalous behavior.

The architecture proposed and developed during this research work is effective in terms of providing offline analysis of network data to provide evidence of suspicious traffic. There are still some limitations, such as requiring a specific process to convert data from pcap to csv so it can be indexed and studied. Another limitation at the moment is the nonexistence of a data pipeline, which does not allow a real-time analysis of the payload of the packets.

As future work, this research aims to improve the architecture so that a real-time payload analysis can be performed, with automatic data conversion and a pipeline between the data source in the Honeynet and Logstash, for indexing data as it is created. This is expected to be established with big data technologies such as Kafka, Hadoop, Solr and other tools.

Additionally, future works also aim to use and compare the results reported in this paper with different technologies and implementation of DPI, including Machine Learning, to detect real-time similar vulnerabilities, based on the information acquired from this work. Likewise, the implementation of different depths of packet inspection devices, focusing the analysis on lower layers of the TCP/IP model, may return interesting information, useful for comparing and correlating the results presented here and identifying more malicious patterns.

This work intends to expand the research for different application protocols, such as DNS, RTP, RSTP, etc. This can be achieved by including these protocols' fields in the TShark exportation command and in the Logstash configuration file. The Honeynet environment also received attacks such as the ransomware WannaCry (May 2017) and NotPetya (June 2017). Therefore, future works also include the analysis of malware and its behavior considering the evaluation of the malware payload with dynamic analysis.

Acknowledgments: This research work was supported by Sungshin W. University. In addition, this research work has the support of the Brazilian research and innovation Agencies CAPES–Coordination for the Improvement of Higher Education Personnel (Grant 23038.007604/2014-69 FORTE–Tempestive Forensics Project), FINEP–Funding Authority for Studies and Projects (Grant 01.12.0555.00 RENASIC/PROTO–Secure Protocols Laboratory of the National Information Security and Cryptography Network) and FAPDF–Research Support Foundation of the Federal District (Grants 0193.001366/2016 UIoT–Universal Internet of Things and 0193.001365/2016–Secure Software Defined Data Center (SSDDC)), as well as the Ministry of Planning, Development and Management (Grants 005/2016 DIPLA–Planning and Management Directorate and 11/2016 SEST–Secretariat of State-owned Federal Companies) and the DPGU–Brazilian Union Public Defender (Grant 066/2016).

Author Contributions: G.A.P.R., R.d.O.A. and G.A.d.O.J. conceived the general architecture of the DPI suite and the Honeynet services and developed the validation prototype; F.E.G.d.D. and R.T.d.S.J. modeled the formats for data collection and aggregation and specified the related visualization formats and methods; L.J.G.V and T.H.K. conceived and designed the experiments. All authors contributed equally to performing the experiments, analyzing data resulting from the experiments and writing the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CAGR	Compound Annual Growth Rate
CoC	Chain of Custody
DDoS	Distributed Denial of Service
DoS	Denial-of-Service
DPI	Deep Packet Inspection
IDS	Intrusion Detection System
IOC	Indicator Of Compromise
API	Application Programming Interface
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IPS	Intrusion Prevention System
IRC	Internet Relay Chat
MAC	Media Access Control

MIB-II	Management Information Base Version Two
MITM	Man-In-The-Middle
MPI	Medium Packet Inspection
NAT	Network Address Translation
NTP	Network Time Protocol
QoS	Quality of Service
RegEx	Regular Expression
RPC	Remote Procedure Call
SDN	Software-Defined Network
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SPI	Shallow Packet Inspection
SSL	Secure Socket Layer
TOR	The Onion Router
TTL	Time To Live
UDP	User Datagram Protocol
VAST	Visibility Across Space and Time

References

1. Markets and Markets. Deep Packet Inspection and Processing Market by Application (IDS and IPS, Network Performance Management, and Data Loss/Leak Prevention and Management), by Service, by Organization Size, by Vertical, by End User, & by Region—Global Forecast to 2021. Available online: http://www.marketsandmarkets.com/Market-Reports/deep-packet-inspection-processing-market-252816977.html (accessed on 20 September 2017).
2. Parsons, C. *Deep Packet Inspection in Perspective: Tracing Its Lineage and Surveillance Potentials*; Queen's University, Surveillance Studies Centre: Kingston, ON, Canada, 2008.
3. White, T. *Hadoop: The Definitive Guide*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2012.
4. Narkhede, N.; Shapira, G.; Palino, T. *Kafka: The Definitive Guide*; O'Reilly Media: Sebastopol, CA, USA, 2016.
5. Parvat, T.J.; Chandra, P. A Novel approach to deep packet inspection for intrusion detection. *Procedia Comput. Sci.* **2015**, *45*, 506–513.
6. Xu, C.; Chen, S.; Su, J.; Yiu, S.; Hui, L. A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2991–3029.
7. Bendrath, R. Global technology trends and national regulation: Explaining Variation in the Governance of Deep Packet Inspection. In Proceedings of the International Studies Annual Convention, New York, NY, USA, 15–18 February 2009; Volume 15.
8. Bendrath, R.; Mueller, M. The end of the net as we know it? Deep packet inspection and internet governance. *New Media Soc.* **2011**, *13*, 1142–1160.
9. Fuchs, C. *Implications of Deep Packet Inspection (DPI) Internet Surveillance for Society*; The Privacy & Security Research Paper Series 1; PACT: Uppsala, Sweden, 2012.
10. Lin, Y.S.; Lee, C.L.; Chen, Y.C. Length-bounded hybrid CPU/GPU pattern matching algorithm for deep packet inspection. *Algorithms* **2017**, *10*, 16.
11. Shankar, S.S.; Lin, P.; Herkersdorf, A.; Wild, T. Hardware acceleration of signature matching through multi-layer transition bit masking. In Proceedings of the 26th IEEE International Telecommunication Networks and Applications Conference, Dunedin, New Zealand, 7–9 December 2016; pp. 217–224.
12. Su, J.; Chen, S.; Han, B.; Xu, C.; Wang, X. A 60Gbps DPI Prototype based on Memory-Centric FPGA. In Proceedings of the ACM SIGCOMM Conference, Florianópolis, Santa Catarina, Brazil, 22–26 August 2016; pp. 627–628.
13. Piyachon, P.; Luo, Y. Efficient memory utilization on network processors for deep packet inspection. In Proceedings of the ACM/IEEE Symposium on Architecture for networking and communications systems, San Jose, CA, USA, 3–5 December 2006; pp. 71–80.
14. Guo, Y.; Gao, Y.; Wang, Y.; Qin, M.; Pu, Y.; Wang, Z.; Liu, D.; Chen, X.; Gao, T.; Lv, T.; Fu, Z. DPI & DFI: A Malicious Behavior Detection Method Combining Deep Packet Inspection and Deep Flow Inspection. *Procedia Eng.* **2017**, *174*, 1309–1314.

15. Sherry, J.; Lan, C.; Popa, R.A.; Ratnasamy, S. Blindbox: Deep packet inspection over encrypted traffic. In Proceedings of the ACM SIGCOMM Computer Communication Review, New York, NY, USA, 22 April 2015; Volume 45, pp. 213–226.
16. Yuan, X.; Wang, X.; Lin, J.; Wang, C. Privacy-preserving deep packet inspection in outsourced middleboxes. In Proceedings of the International Conference on Computer Communications, San Francisco, CA, USA, 10–15 April 2016; pp. 1–9.
17. Saputra, F.A.; Nadhori, I.U.; Barry, B.F. Detecting and blocking onion router traffic using deep packet inspection. In Proceedings of the International Electronics Symposium, Bali, Indonesia, 29–30 September 2016; pp. 283–288.
18. Abe, K.; Goto, S. Fingerprinting Attack on Tor Anonymity using Deep Learning. *Proc. Asia Pac. Adv. Netw.* **2016**, *42*, 15–20.
19. Hubballi, N.; Tripathi, N. An event based technique for detecting spoofed IP packets. *J. Inf. Secur. Appl.* **2017**, *35*, 32–43.
20. Trivedi, U.; Patel, M. A fully automated deep packet inspection verification system with machine learning. In Proceedings of the Advanced Networks and Telecommunications Systems (ANTS), Bangalore, KA, India, 6–9 November 2016; pp. 1–6.
21. Comar, P.M.; Liu, L.; Saha, S.; Tan, P.N.; Nucci, A. Combining supervised and unsupervised learning for zero-day malware detection. In Proceedings of the 32nd IEEE International Conference on Computer Communications, Turin, Italy, 14–19 April 2013; pp. 2022–2030.
22. CISCO. 2016 Midyear Cybersecurity Report. Available online: http://www.cisco.com/c/dam/m/en_ca/never-better/assets/files/midyear-security-report-2016.pdf (accessed on 11 July 2017).
23. Bouet, M.; Leguay, J.; Conan, V. Cost-based placement of virtualized deep packet inspection functions in sdn. In Proceedings of the Military Communications Conference, San Diego, CA, USA, 18–20 November 2013; pp. 992–997.
24. Lillard, T.V. *Digital Forensics for Network, Internet, and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data*; Syngress Publishing: Amsterdam, The Netherlands, 2010; pp. 56–58.
25. Cisco Systems. The Zettabyte Era: Trends and Analysis. Available online: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html (accessed on 10 July 2017).
26. IBM. Cost of Data Breach Study. Available online: https://www-01.ibm.com/marketing/iwm/dre/signup?source=urx-15763&S_PKG=ov58441 (accessed on 10 September 2017).
27. Cisco Systems. 2017 Annual Cybersecurity Report. Available online: http://www.cisco.com/c/dam/m/digital/1198689/Cisco_2017_ACR_PDF.pdf (accessed on 10 July 2017).
28. Symantec. Internet Security Threat Report Government. 2017. Available online: https://www.symantec.com/content/dam/symantec/docs/reports/gistr22-government-report.pdf (accessed on 30 September 2017).
29. Symantec. Internet Security Threat Report. 2017. Available online: https://s1.q4cdn.com/585930769/files/doc_downloads/lifelock/ISTR22_Main-FINAL-APR24.pdf (accessed on 11 July 2017).
30. Berry, A.; Homan, J.; Eitzman, R. WannaCry Malware Profile. Available online: https://www.fireeye.com/blog/threat-research/2017/05/wannacry-malware-profile.html (accessed on 21 September 2017).
31. Morgan, S. Worldwide Cybersecurity Spending Increasing To $170 Billion By 2020. Available online: https://www.forbes.com/sites/stevemorgan/2016/03/09/worldwide-cybersecurity-spending-increasing-to-170-billion-by-2020/#7d8106b06832 (accessed on 11 July 2017).
32. Ashraf, M.A.; Jamal, H.; Khan, S.A.; Ahmed, Z.; Baig, M.I. A Heterogeneous Service-Oriented Deep Packet Inspection and Analysis Framework for Traffic-Aware Network Management and Security Systems. *IEEE Access* **2016**, *4*, 5918–5936.
33. Sommer, R.; Amann, J.; Hall, S. Spicy: A unified deep packet inspection framework for safely dissecting all your data. In Proceedings of the 32nd Annual Conference on Computer Security Applications, Los Angeles, CA, USA, 5–9 December 2016; pp. 558–569.
34. Huang, H.; Li, P.; Guo, S. Traffic scheduling for deep packet inspection in software-defined networks. *Concurr. Comput. Pract. Exp.* **2016**, *29*.
35. Khan, S.; Gani, A.; Wahab, A.W.A.; Shiraz, M.; Ahmad, I. Network forensics: review, taxonomy, and open challenges. *J. Netw. Comput. Appl.* **2016**, *66*, 214–235.

36. Vallentin, M.; Paxson, V.; Sommer, R. VAST: A Unified Platform for Interactive Network Forensics. In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, Boston, MA, USA, 27–29 March 2016; pp. 345–362.
37. Khan, S.; Gani, A.; Wahab, A.W.A.; Abdelaziz, A.; Ko, K.; Khan, M.K.; Guizani, M. Software-defined network forensics: Motivation, potential locations, requirements, and challenges. *IEEE Netw.* **2016**, *30*, 6–13.
38. Prayudi, Y.; Sn, A. Digital chain of custody: State of the art. *Int. J. Comput. Appl.* **2015**, *114*, 1–9.
39. Ćosić, J.; Ćosić, Z.; Baća, M. An ontological approach to study and manage digital chain of custody of digital evidence. *J. Inf. Organ. Sci.* **2011**, *35*, 1–13.
40. Flores Armas, D.; Jhumka, A. Implementing chain of custody requirements in database audit records for forensic purposes. In Proceedings of the 16th International Conference on Trust, Security and Privacy in Computing and Communications, Sydney, NSW, Australia, 1–4 August 2017.
41. Dutelle, A.W. *An Introduction to Crime Scene Investigation*; Jones & Bartlett Publishers: Burlington, MA, USA, 2016.
42. Spitzner, L. The Honeynet project: Trapping the hackers. *IEEE Secur. Privacy* **2003**, *99*, 15–23.
43. Roesch, M. Snort: Lightweight intrusion detection for networks. In Proceedings of the 13th USENIX Conference on System Administration, Seattle, WA, USA, 7–12 November 1999; Volume 99, pp. 229–238.
44. Paxson, V. Detecting Attacks. Available online: https://inst.eecs.berkeley.edu/~cs161/sp11/slides/4.14.intrusion2.pdf (accessed on 11 July 2017).
45. Gupta, S.; Gupta, B.B. Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art. *Int. J. Syst. Assur. Eng. Manag.* **2017**, *8*, 512–530.
46. Sudhodanan, A.; Carbone, R.; Compagna, L.; Dolgin, N.; Armando, A.; Morelli, U. Large-Scale Analysis & Detection of Authentication Cross-Site Request Forgeries. In Proceedings of the 2nd IEEE European Symposium on Security and Privacy, Paris, France, 26–28 April 2017; pp. 350–365.
47. Prokhorenko, V.; Choo, K.K.R.; Ashman, H. Web application protection techniques: A taxonomy. *J. Netw. Comput. Appl.* **2016**, *60*, 95–112.
48. Kim, H.A.; Karp, B. Autograph: Toward Automated, Distributed Worm Signature Detection. In Proceedings of the 12th USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004; Voume 286.
49. Ellis, D.R.; Aiken, J.G.; Attwood, K.S.; Tenaglia, S.D. A behavioral approach to worm detection. In Proceedings of the ACM workshop on Rapid malcode, Washington, DC, USA, 25–29 October 2004; pp. 43–53.
50. Li, Z.; Pan, H.; Liu, W.; Xu, F.; Cao, Z.; Xiong, G. A network attack forensic platform against HTTP evasive behavior. *J. Supercomput.* **2017**, *73*, 3053–3064.
51. Sun, X.; Hou, K.; Li, H.; Hu, C. Towards a fast packet inspection over compressed HTTP traffic. In Proceedings of the 25th International Symposium on Quality of Service, Vilanova i la Geltrú, Spain, 14–16 June 2017; pp. 1–5.
52. Oberholzer-Gee, F.; Strumpf, K. File sharing and copyright. *Innov. Policy Economy* **2010**, *10*, 19–55.
53. Dreger, H.; Feldmann, A.; Mai, M.; Paxson, V.; Sommer, R. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In Proceedings of the 15th USENIX Security Symposium, Vancouver, BC, Canada, 31 July–4 August 2006; pp. 257–272.
54. Bujlow, T.; Carela-Español, V.; Barlet-Ros, P. *Comparison of Deep Packet Inspection (DPI) Tools for Traffic Classification*; Technical report; Universitat Politècnica de Catalunya: Barcelona, Spain, 2013.
55. Kara, A.M.; Binsalleeh, H.; Mannan, M.; Youssef, A.; Debbabi, M. Detection of malicious payload distribution channels in DNS. In Proceedings of the Communication and Information Systems Security Symposium, Sydney, NSW, Australia, 10–14 June 2014; pp. 853–858.
56. Rossow, C. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, USA, 23–26 February 2014.
57. Lotfollahi, M.; Shirali, R.; Siavoshani, M.J.; Saberian, M. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning. *arXiv* **2017**, arXiv:1709.02656.
58. Sommer, R.; Paxson, V. Enhancing byte-level network intrusion detection signatures with context. In Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington, DC, USA, 27–30 October 2003; pp. 262–271.
59. Dharmapurikar, S.; Krishnamurthy, P.; Sproull, T.; Lockwood, J. Deep packet inspection using parallel bloom filters. In Proceedings of the 11th Symposium on High performance interconnects, Stanfords, CA, USA, 20–22 August 2003; pp. 44–51.

60. Yu, F.; Chen, Z.; Diao, Y.; Lakshman, T.; Katz, R.H. Fast and memory-efficient regular expression matching for deep packet inspection. In Proceedings of the ACM/IEEE Symposium on Architecture for networking and communications systems, San Jose, CA, USA, 3–5 December 2006; pp. 93–102.
61. Kumar, S.; Dharmapurikar, S.; Yu, F.; Crowley, P.; Turner, J. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. In Proceedings of the ACM SIGCOMM Computer Communication Review, Pisa, Italy, 12–15 September 2006; Volume 36, pp. 339–350.
62. Becchi, M.; Crowley, P. A hybrid finite automaton for practical deep packet inspection. In Proceedings of the 2007 ACM CoNEXT conference, New York, NY, USA, 11–13 December 2007; pp. 1–12.
63. Benzaïd, C.; Boulgheraif, A.; Dahmane, F.Z.; Al-Nemrat, A.; Zeraoulia, K. Intelligent detection of mac spoofing attack in 802.11 network. In Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, 4–7 January 2016; p. 47.
64. Carnut, M.; Gondim, J. ARP spoofing detection on switched Ethernet networks: A feasibility study. In Proceedings of the 5th Simposio Seguranca em Informatica, São José dos Campos, SP, Brazil, 4–6 November 2003.
65. Kolias, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 184–208.
66. Templeton, S.J.; Levitt, K.E. Detecting spoofed packets. In Proceedings of the DARPA Information Survivability Conference and Exposition, Washington, DC, USA, 22–24 April 2003; Volume 1, pp. 164–175.
67. Oliveira Júnior, G.A.d.; Sousa Júnior, R.T.d.; Tenório, D.F. Desenvolvimento de um Ambiente Honeynet Virtual para Aplicação Governamental. In Proceedings of the 9th International Conference on Forensic Computer Science, Brasilia, DF, Brazil, 23–25 June 2015; pp. 70–78.
68. Combs, G. TShark—Dump and Analyze Network Traffic. Available online: https://www.wireshark.org/docs/man-pages/ (accessed on 7 September 2017).
69. Elastic. The Open Source Elastic Stack. Available online: https://www.elastic.co/products (accessed on 7 September 2017).
70. Oliveira Júnior, G.A.d. Honeyselk: Um Ambiente Para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real, 2016. xvi, 62 f., il. Dissertation (Masters in Electrical Engineering), University of Brasilia, Brasília, Brazil, 2016. [Google Scholar].
71. US-CERT. Alert (TA17-132A) Indicators Associated With WannaCry Ransomware. Available online: https://www.us-cert.gov/ncas/alerts/TA17-132A (accessed on 10 September 2017).
72. US-CERT. Alert (TA16-288A): Heightened DDoS Threat Posed by Mirai and Other Botnets. Available online: https://www.us-cert.gov/ncas/alerts/TA16-288A (accessed on 10 September 2017).
73. Jerkins, J.A. Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In Proceedings of the 7th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 9–11 January 2017; pp. 1–5.
74. Nazaryan, G. W32.IRCBot. Available online: https://www.symantec.com/security_response/writeup.jsp?docid=2002-070818-0630-99 (accessed on 30 September 2017).
75. Gu, G. *Correlation-Based Botnet Detection in Enterprise Networks*; Georgia Institute of Technology: Atlanta, GA, USA, 2008.
76. Furfaro, A.; Malena, G.; Molina, L. A simulation model for the analysis of DDOS amplification attacks. In Proceedings of the 17th International Conference on Computer Modelling and Simulation, Cambridge, UK, 25–27 March 2015; pp. 267–272.
77. Arukonda, S.; Sinha, S. The innocent perpetrators: reflectors and reflection attacks. *Adv. Comput. Sci.* **2015**, *4*, 94–98.