CIP-PSP-325188

Advanced Cyber Defence Centre

D4.1
Documentation of botnet metrics methodology and development

Editor(s)	Giovane C. M. Moura, Michel van Eeten
Responsible Partner	TU Delft
Affiliations	[1] Delft University of Technology (TU Delft),
Status-Version:	Draft-1.0
Date:	Jan. 27th, 2015
EC Distribution:	Consortium
Project Number:	325188
Project Title:	Advanced Cyber Defence Centre

Title of Deliverable:	Documentation of botnet metrics methodology and development
Date of delivery to the EC:	01/31/2014

Workpackage responsible for the Deliverable	WP4
Editor(s):	editors
Contributor(s):	contributors
Reviewer(s):	reviewers
Approved by:	All Partners

Abstract	Summary on botnet metrics state-of-the-art, issues, and proposals.
Keyword List:	botnet, metrics, CCH, WP4

Document Revision History

Version	Date	Description	Author
First draft	Jan 18, 2015	Initial Version	Giovane C. M. Moura
1.0	Jan 30, 2015	Added outline and new sessions	Giovane C. M. Moura, Jan Kolhrausch, Qasim Lone, Michel van Eeten

Table of Contents

List of Figures

List of Tables

Executive Summary

Work Package 4 (WP4) of the ACDC Competitiveness and Innovation Framework (CIP) project has committed itself to develop comparative metrics that capture the number of bots, their command & control structures as well as related botnet infrastructure. Such metrics serve a dual purpose: evaluating the Pilot as well as incentivizing the actors in the relevant markets to actually undertake mitigation – which can include using the tools and services provided by the Pilot.

This report – "Documentation of botnet metrics methodology and development" – starts by addressing the challenges related to these tasks. We begin by discussing how data obtained from the central clearing house (CCH) and WP3 should be submitted to a rigorous quality control process and then aggregated to make them amenable to statistical analysis, while at the same time complying with privacy requirements. The raw data used for operational abuse handling and botnet mitigation typically lacks in quality for reliable comparative metrics. It contains gaps, double counting and other problems that have to be controlled for. The outcome of this task is a statically robust aggregated data set that can support the development of comparative metrics for evaluating and incentivizing botnet mitigation.

After that, we present a survey of the state-of-the-art of botnet metrics. We produce an analysis on comparative metrics across various ISPs – which can be ultimately used to determine and assess the botnet presence across ISPs and countries. We identify the requirements for such metrics and identify how current botnet metrics fail to meet then. Then, we present an methodology and approach to deal with one of these main shortcomings – the effects of dynamic IP address allocation on reliably measuring the number of infected machines. We present a measurement method to estimate the degree of churn caused by Dynamic Host Configuration Protocol (DHCP) and Network Address Translation (NAT) technologies in ISPs and subnetworks. In a pilot study, we collect the relevant measurement data and carry out a preliminary test of our approach. Finally, we introduce the improved specifications of the metrics, beyond the state-of-the-art, to be used in our evaluation, and present a country level evaluation using current datasets.

This report is the final version of Deliverable 4.1, which covers the documentation of the botnet metrics developed within ACDC. The work package is now entering the second phase, where the focus shifts to the assessment of the performance of ISPs and the impact of countermeasures, such as those provided by ACDC.

1 Introduction

The Internet is currently so important for the functioning of modern societies that it is actually considered part of the *critical infrastructure* of many countries [1]. All kinds of critical services, such as banking, energy, and transportation, heavily rely upon the Internet to perform.

Such dependence, however, has made the Internet very attractive for criminal organizations, nation states, and activists as a medium in which crimes, cyber war, and protests can be conducted. One well-known example of malicious activity on the Internet is spam, the abuse of electronic email. It is estimated that between 84% and 90% of all e-mail messages are spam nowadays [2, 3], and behind it, cyber gangs run lucrative operations by selling pharmaceuticals [4], distributing malicious software (malware), carrying out distributed denial-of-service (DDoS) attacks, among other illegal activities [5, 6]. The impact does not stop on the Internet: it is estimated that worldwide spam causes losses from $10 billion to $87 billion yearly [7].

Behind many types of attacks, we typically find a large amount of IP addresses, part of the so-called botnets, which are essentially a large number of *distributed* compromised machines (called bots or zombies) under control of a botmaster [8, 9]. The zombies can be seen as "hijacked" computers, located at homes, schools, and businesses, controlled by the botmaster to carry out malicious activities.

Recent economic research has found that the infected machines of end users (zombies) are a key source of security externalities, most notably home users and small and medium-size enterprise (SME) users[1]. In contrast to larger corporate users, these groups often fail to achieve desirable levels of protection.

WP4 has committed to develop comparative metrics that capture the number of bots, their command & control structures as well as related botnet infrastructure, across networks. The effectiveness of the Pilot, or any other mitigation measure for that matter, cannot be established without accurate and reliable reputation metrics across countries, markets and actors. Without such metrics, there is only anecdotal evidence that cannot be reliably interpreted. Many factors impact the size and the rise and fall of botnets. These have to be disentangled in order to develop evidence-based mitigation strategies, including the strategy developed in the Pilot.

Using the data collected in WP3, WP4 aims to extract comparative metrics at the levels of countries and the relevant actors, focusing on, e.g., the number of bots per user in access networks, persistence of those bots, C&C infrastructure density within hosting provider networks, and other metrics. All of this needs to take into account relevant control variables, such as the size of the population or user base.

[1] See US GAO (2007). Cybercrime: Public and Private Entities Face Challenges in Addressing Cyber Threats. United States Government Accountability Office. Available online at `http://www.gao.gov/new.items/d07705.pdf`.

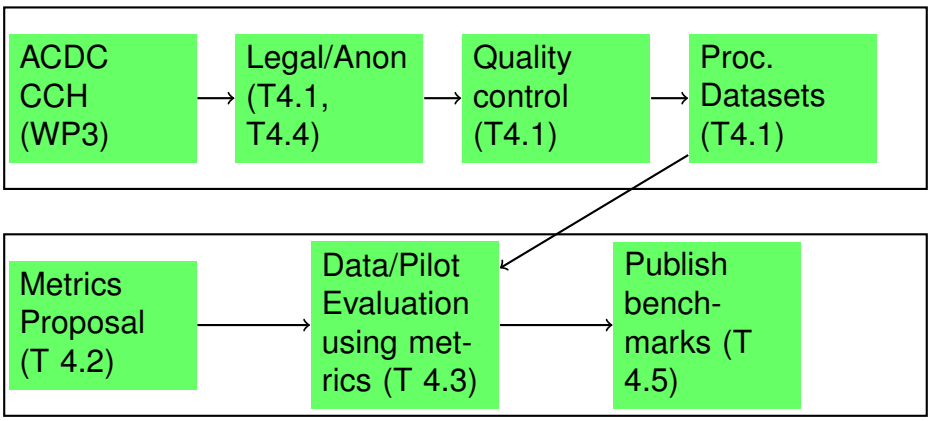


Figure 1: WP4 Tasks Workflow

1.1 WP4 Tasks Workflow

In order to accomplish its goal, we have divided WP4 into 5 major tasks, as can be seen in Figure 1. First, we obtain botnet data from the Central Clearing House (CCH), and make it sure it conforms with the legal requirements (Tasks 4.1, 4.4). This data is to be produced by WP3 and other partners, in a series of experiments and data collection initiatives. Please refer to WP3 for more details on this matter.

Then, this data is subjected to a quality control process, in which a series of steps are taken in order to guarantee that bogus data is removed and the quality of the data is assessed (T 4.1). The output of T4.1 is robust datasets that can be then used in the evaluation of the infection rates of various ISPs (T 4.3), which, in turn, have as input the specification of the metrics developed in T 4.2. Finally, the benchmarks for this ISPs will be published in T4.5. This interim report focus on Task 4.1 and 4.2.

1.2 Botnet Data and WP3 – Input for WP4

In WP4, the data we will use to assess the evaluate the impact of the pilot will be obtained from the CCH (Figure 1). The CCH provides a centralized point in which botnet-related data can be shared among all parters, taking into account the necessary privacy and law requirements.

Data is fed into the CCH based on a series of experiments described in WP3, and by all other sources the project partners might obtain. Regardless the type of experiment and the source of data, there is currently no authoritative data source to identify the overall population of infected machines around the world [10] or within the EU. Commercial security providers typically use proprietary data and shield their measurement methods from public scrutiny. This makes it all but impossible to correctly interpret the figures they report and to assess their validity.

The publicly accessible research in this area relies on two types of data sources:

- Data collected external to botnets. This data identifies infected machines by their telltale behavior, such as sending spam or participating in distributed denial of service attacks;

- Data collected internal to botnets. Here, infected machines are identified by intercepting communications within the botnet itself, for example by infiltrating the command and control infrastructure through which the infected machines get their instructions.

Each type of source has its own strengths and weaknesses. The first type typically uses techniques such as honey pots, intrusion detection systems and spam traps. It has the advantage that it is not limited to machines in a single botnet, but can identify machines across a wide range of botnets that all participate in the same behavior, such as the distribution of spam. The drawback is that there are potentially issues with false positives. The second type typically intercepts botnet communications by techniques such as redirecting traffic or infiltrating IRC channel communication. The advantage of this approach is accuracy, in the sense of very low rates of false positive. The machines that connect to the command and control server are really infected with the specific type of malware that underlies that specific botnet. The downside is that measurement only captures infected machines within a single botnet. Given the fact that the number of botnets is estimated to be in the hundreds [11], such data is probably not representative of the overall population of infected machines.

Neither type of data sources sees all infected machines, they only see certain subsets, depending on the specific data source. In general, one could summarize the difference between the first and the second source as a trade-off between representativeness versus accuracy. The first type captures a more representative slice of the problem, but will also include false positives. The second type accurately identifies infected machines, but only for a specific botnet, which implies that it cannot paint a representative picture.

Botnets are typically design to attack one or more different applications: they can be employed to send spam, carry out DDoS attacks, perform port-scanning, host illegal files, identity theft, bitcoin mining, cracking passwords, among others. As a consequence, data observed from the attacks is highly dependent on the exploited application.

In WP4, we will obtain data from the CCH and consider all these differences in the data sources whenever evaluating the performance of ISPs.

1.3 Document Outline

Chapter 2 contains a detailed description of the initial research on data processing and quality control. It takes into account the difference among the sensors used to detect network incidents and filters out data based on duplicate entries, false entries (e.g., non-valid/routable IP addresses, incomplete data). Then, it includes an analysis of each data source obtained from the CCH to assess its properties: data stability, similarity, and correlation using different models for that. After that, in Section 2.3, we cover how data can be aggregated and enriched in such a way that ultimately produces a robust dataset that can be used as input to asses the ISPs performance with regards with botnet metrics.

Chapter 3 survey the current existing botnet comparison metrics for botnets, and produce an analysis on comparative metrics across various ISPs – which can be ultimately used to determine and assess the botnet presence in various ISPs. We show a list of requirements the metrics must fulfill and the problems with the current botnet metrics.

In Chapter 4 presents an approach, we presents a novel approach to deal with two main issues related to botnet metrics: DHCP churn and NAT effects. IP address counts are typically used as a surrogate metric for unique identifiers. However, due to effects of DHCP, the actual number of infected customers (in botnets) in an ISP network is inflated by how often the ISP changes the customer's addresses. In , we present an active measurement methodology and apply it to measure entire ISPs, and validate the precision of our measurements using ground truth from a 1 million IP addresses, showing we were able to capture up to 78% of all sessions and 65% of their duration. We then apply our method to four major ISPs and show how their session duration varies. We finally present a statistical model to estimate DHCP churn rates and validate against the ground truth from a mid-size ISP.

Chapter 5 finally contains a proposal for new metrics that we will employ to evaluate the performance of European ISPs and the impact of the ACDC project in mitigating botnets. Differently from Chapter 3, in this chapter we present a hands-on and detailed description that shows how measurement data should be parsed, enriched, anonymized, and exported to the CCH, so we can carry our our evaluations. In addition, we go a step further in the scope of this deliverable, by illustrating how the proposed metrics can be used, by assessing the performance of countries as a bulk over the last years for the data sources we currently have.

Finally, Chapter 6 we present the conclusions and a summary of the fundamental aspects of the work.

2 Data processing and Quality Control

ACDC Task 4.1 strives to produce data that enables to measure and to assess the exposure of botnet activity. Critical point is to deliver reliable and trustworthy data that allow to derive the required information of the employed metrics. This requires to pre-process the data in such a way that it is statistically stable by ensuring strict quality control.

This section begins with an overview of the aims and challenges that have to be addressed. After that, we present approaches that can be used to measure and to ensure the statistical stability of the data. Especially mathematical methods to analyse time series are assessed for their applicability. The next part provides an overview of the data aggregation and further processing. This also comprises the workflows to retrieve data from the Central Clearinghouse (CCH) and to pass on the data for the application of metrics. We then detail the specific implementation of workflows covering formats for the data exchange and collaborative workflows for the specification and implementation of metrics.

2.1 Aims and Challenges

The data is gathered by different sources and methods[2]. For example, different technical sensors exists to detect compromised systems and attacks which include, for example, signature based Intrusion detection systems (IDS), anomaly detection systems, and honeypots. Each sensor has its own individual advantages and disadvantages that differ, for instance, with the rate of false positives and the level of recorded attack details. On the one hand a honeypot provides a very accurate and detailed detection but on the other hand cannot detect any system that does not directly connect to the honeypot. In contrast, a network based IDS or netflow collector can be deployed at the edge of a network and theoretically allows to detect all compromised systems that communicate with the Internet. However, IDS are prone to false-positives and can be avoided by malware. This results in different characteristics of each sensor or data source that has to be considered by the pre-processing of the data. These are in detail:

Characteristics of the sensor This includes:

Accuracy The accuracy of the sensor is closely related to the rate of false-positives and false-negatives. In addition, the sensors differ in the details of the results. Although an infected system might be correctly detected by a sensor, it may be erroneously assigned to a wrong malware family.

Method of Detection Multiple different methods for attack and malware detection exist which include behavioral analysis, anomaly detection, and signature based attack or malware detection. It is important to note that the gathered data differ significantly in its properties. Because each sensor might

[2]For examples see also the experiments in Task 3 of the ACDC Project.

contribute different details of an attack, it is not reasonable or at least difficult to assess or compare the trustworthiness of data originating from different classes (e.g. anomalies versus honeypot data). However, correlation of the data might lead to a more versatile perspective on botnet monitoring.

Level of Detail The level of detail varies with the method of detection. Usually, a honeypot provides very detailed attack data whereas a method for anomaly detection in netflow data lacks all details. As previously stated, aggregation or correlation of the data might add value.

Characteristics of the source This includes:

DHCP Churn A lot of metrics for counting compromised systems rely on source IP addresses. However, this address might change dynamically over time resulting in counting unique systems twice or even more times.

NAT NAT gateways hide a private network. All connections from this network have to use the public IP address of the NAT gateway. Therefore, it may look like a single host is infected multiple times. In analogy to DHCP the source IP address does not generally allow to identify or recognize a unique computer. However, botnets exist that assign a unique identifier to each compromised system allowing such identification. For example, according to [12] the communication of Torpig bots comprises such an unique identifier.

Reliability of the data source and transport To produce statistically stable data, it is important to monitor the reliability of the data source. For example, a failure of the data submission has to be distinguished from a decrease in the number of compromised systems.

In addition, some other challenges exist:

- Sensors might fail to correctly classify the attack. Even if an attack is correctly detected the sensor might erroneously label the malware or type of attack (e.g. by assigning a wrong malware family).
- The notation of malware is often ambiguous. For example, the Conficker worm is commonly denoted as W32.Conficker, but other AV companies refer to it as Downup, Downadup, and kido.

Assessment of the Data Quality and Stability

Pre-processing and aggregation aim at achieving a statistically stable data set. This requires a precise definition of stability for which different mathematical models have been introduced (for example in [13, 14]. These models apply to time series and other stochastic data.

The first step of the assessment of the data quality is to determine the rate of bogus data, false-positives, and false-negatives. Bogus data comprises all data that is syntactically or semantically invalid. For example, this applies to malformed reports or

reports that contain private or unallocated IP addresses. Unfortunately, it is very hard to reliably measure the rates of false-positives and false-negatives.

It is unfeasible to exactly count the number of infected systems and attacks as previously outlined in Section 2.1. Instead, we propose to estimate this number and to quantify the uncertainty that comes with the limitations. Although number and distribution of infected systems over networks and ISPs are subject to variations, it is reasonable to assume that at large scale the statistical properties are stable unless a change point occurs. Change points include, for example, a significant change of the botnet's behavior or a failure of a sensor which leads to a major change in the number of detected systems. We propose to focus in the work package on the following statistical properties of:

Number of Infected Systems and Attacks. Objective is to determine the stability of the number of countable infected systems and attacks. The details are presented in Section 2.2.

Similarity of the Distribution of Infected Systems. Objective is to determine the stability of the distribution of infected systems and attacks (e.g. their IP addresses). The details are presented in Section 2.2.3.

Temporal Correlations in the Data. It is reasonable to look for correlations in the data. For example, this could reveal a relationship between infected systems and attacks originating from different data providers. We refer to Section 2.2.3 for the details.

Ideally, the results give a good indication which data sets are reasonably stable.

Botnet Reconnaissance

Botnets exhibit in many aspects dynamic behavior. Almost all botnet related malware are capable of updating the bot-software to impede detection and react to countermeasures. Furthermore, techniques are deployed to avoid pinpointing either the central C&C server or the botnet structure. For example, peer-to-peer network structures are deployed to make it more difficult to enumerate the number of hosts that are part of the botnet. For that reason, researchers and Anti-Virus vendors spend efforts into the reconnaissance of botnets and bots that contributes important information pertaining the analysis and interpretation of data. This is valuable in many aspects:

- It is likely that there are variations in the accuracy of measuring botnet population depending on the type and structure of the botnet.
- A change of botnet related software or behavior might result in a different nomenclature.
- Gaps in the data set may result either from a change in the botnet's behavior or from a successful countermeasure. Botnet intelligence could distinguish between both causes.

- Botnet intelligence may improve the quality control of the data, e.g. by providing information about unique identifiers.
- Pre-processing of data might take the botnet structure into account. Botnet intelligence contributes to grouping the data correctly.

2.2 Methods for Quality Control

Design and evaluation of comparative metrics require statistically stable data to analyze the effectiveness of initiatives reacting to the threat arising from botnets. These metrics could, for example, reveal trends or change points in the number of infected systems. Such conclusions cannot be derived without understanding the mathematical properties of the data. This section presents mathematical models to measure the stability, similarity, and correlation of the data.

2.2.1 Detection of Bogus Data, Duplicates, and False-Positives

A first step is to exclude all data that is obviously bogus such as private IP addresses[3]. Furthermore, duplicates should be excluded in this step. It is important to note, that the definition of duplicates depends on the context in which the data is analyzed. For example, two subsequent connection attempts from a unique source can be condensed to a single event in cases in which only the number of attacking sources counts. However, all metrics that count the exact number of attacks might rely on the exact number of connection attempts which therefore prohibits excluding duplicates.

A next step is to assess the rate of false-positives and false-negatives. Unless the ground truth is known, it is practically unfeasible to exactly measure these rates. False-positives can have the following reasons:

- Bogus IP addresses (e.g. private or unassigned IP addresses)
- Inaccuracies of detection methods. Legitimate connections are erroneously classified as attacks.
- Security systems such as honeypots that, for example, download malware for research purposes may be detected as being infected.

False-negatives can have the following reasons:

- Fragmentary coverage of the monitored networks leads to incomplete data.
- Inaccuracies of detection methods. Attacks are not detected.
- Technical problems of the sensors. For example, network based IDS are limited concerning their data processing capabilities and might not be able to monitor a network cable at full load.

[3] An overview of theses addresses can be found in [15]

There are at least two methods to address the problem of false-positives:

- A honeynet provides an environment in which selected malware (e.g. IRC-bots) is executed under monitoring. This could be used to determine the rate of false-positives as well as false-negatives for this specific but comparable environment. However, it is hard to reproduce all sensor environments used in the ACDC project to produce the resulting ground truth.
- The rate of false-positives can be determined by requesting additional data from the data provider. For example, a feedback channel could be established through which false-positives are reported.

As previously mentioned, it is very hard if not virtually unfeasible to determine the rate of false-negatives. However, an estimation can be based on comparing data from a variety of sources. This data includes attacks on the one hand and infected systems on the other. Since botnets are often abused to send spam emails or to conduct DDoS attacks, it is reasonable to assume that all systems involved in such an attack are infected by a specific malware. Therefore, the resulting data sets should be similar if all attacks and infected systems are detected by all sensors. If both data sets differ significantly this might indicate a high rate of false-negatives. We refer to Section 2.2.3 and Section 2.2.3 how such correlations can be determined.

2.2.2 Mean and Variance and Trends

There are several reasons why an exact measurement of the number of compromised systems, attacks, and other malicious activity is unfeasible. First, the sensors such as IDS and honeypots are prone to false-positives and false-negatives. Second, it is unfeasible to monitor the entire Internet leading to incomplete data. Attacks are often conducted during initiatives. For example, botnets are used to send spam, conduct distributed denial of services, or to scan for other vulnerable systems. All this results in variations in the data which are difficult to predict.

Although the exact variations cannot be predicted, it is reasonable to assume stable statistical properties on a large scale (e.g. the number of gathered compromised systems) if the behavior of botnets and the capabilities of detection do not change during this interval. Thus, our aim is to analyze the data in order to find such stable conditions.

We consider a simple model of a time series

$$x_t = \mu + w_t \tag{1}$$

where x_t is, for example, the number of infected systems, μ is a constant value, and w_t are the previously introduces random fluctuations. Usually, the variations are modeled by a white noise process[4]. Thus, we assume that the variations are distributed

[4] In detail, this is stationary process consisting of identical and temporally independent random variables that are normally distributed $N(0, \sigma)$ (see e.g. [13, 14] for further details)

around 0 having a constant variance σ. It is important to note, that we omitted a trend in (1).

Since we assume randomly distributed variations well-known statistical methods can be used to characterize the data. The mean value $\bar{x}$ is

$$\bar{x} = \frac{x_{t_1} + x_{t_2} + ... + x_{t_n}}{n} \tag{2}$$

is a good estimation of μ which is in our example the average number of compromised systems. In (2) x_{t_i} is the number of compromised system at time slice t_i. The sample standard deviation S is

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{x} - x_{t_i})^2} \tag{3}$$

estimating the variance of the number of compromised systems. The mean and standard deviation are important properties that allow to distinguish between random variations and systematic changes which, for example, include a successful countermeasure against a botnet. The difference between the observed change and the standard deviation is a measure for the confidence with which the change is, for instance, be caused by an anti botnet measure. A common method is to apply the Student's t-test or Chi-Square test (e.g. [16]) to test this hypothesis.

We propose to compute the statistical properties of:

- Overall number of reported IP addresses for each data source. To avoid DHCP churn, we propose to use time intervals of one day.
- Number of reported IP addresses that are related to prominent malware. This list could take into account the work on Task 1.1.8 (Malware Prevalence Analysis).
- Number of specific attacks (e.g. number of spam mails or SSH scans)

It is important to note that the statistical properties may vary with the data contributor and malware type.

Estimating Trends in the Data

As previously mentioned, our model in (1) does not regard any trend in the data which is, for example, be caused by a linear increase or decrease of infected systems. Although this simplifies the mathematical approach, applying (2) and (3) provides erroneous results estimating the statistical properties of data containing a trend. This is compensated by fitting the model:

$$x_t = \beta_1 + \beta_2 * t + w_t \tag{4}$$

In (4), β_1 and β_2 are parameters that are usually estimated by applying the method of linear regression (see e.g. [13, 14] for further details). After subtracting the trend $\beta_1 + \beta_2 * t$ in (4) we get

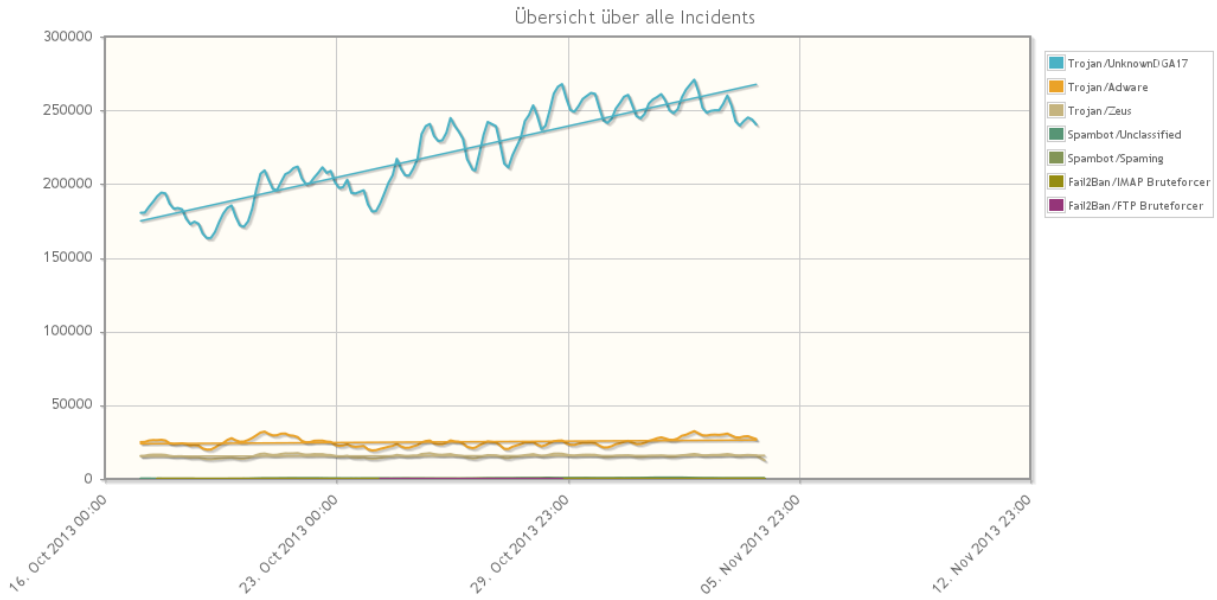


Figure 2: Statistical properties of infected systems indicating stability.

$$x_t = w_t \tag{5}$$

which is the detrended data.

Since w_t is assumed to be a white noise process, its statistical properties, including the mean (2) and variance (3) can be computed after detrending.

A sample plot of the number of infected systems referring to some malware families is presented in Fig.2. The curves show the number of infected IP addresses related to the botnet “UnknownDGA17” (up-most curve), the Trojan family “Adware” (middle curve), and the Trojan family “Zeus” (lowest curve). The botnet “UnknownDGA17” was discovered on 8 October 2013 (see [17] for further information) whereas the infected systems were recorded by a sinkhole to which botnet connections were redirected. Although the exact number of gathered systems vary, the curve exhibits a stable linear increase which fit our model in (4). The other curves related to “Adware” and “Zeus” also provide stable statistical properties.

2.2.3 Similarity of the Distribution of Compromised Systems

Another aspect of stability is the similarity of the distribution of compromised or infected systems. The distribution is given by the set of reported IP addresses which can be, for example, grouped by data providers or malware types. In analogy to the statistical properties we expect a quite stable distribution unless significant changes in the botnet behavior or detection occur. It is important to note that statistical stability does not exclude any changes in the data which, for example, could be caused by a countermeasure. However, the statistical properties of the changes are expected to be stable in order to draw conclusions out of the reported data (e.g. a decreasing trend).

An indicator for stability of the data is the similarity of the distribution of reported IP addresses over time. Let X_t be a vector of IP addresses[5] $(1_{|IP_1}, ..., 1_{|IP_n})$ that are reported in a specific time interval t (e.g. one day). Then as outlined in [18] the similarity with the corresponding vector X_{t-1} at an earlier time $t-1$ can be expressed by

$$\frac{X_t X_{t-1}^T}{\|X_t\|\|X_{t-1}\|} \tag{6}$$

where XY^t is the vector product of X and Y and $\|X\| = \sqrt{x_1^2 + ... + x_n^2}$ is the Euclidean norm of the vector. Since each component of X is either 0 (if the IP is not reported as being compromised) or 1 (the IP is reported as being compromised) (6) can be rewritten as

$$\frac{\|S_t \cap S_{t-1}\|}{\sqrt{\|S_t\|\|S_{t-1}\|}} \tag{7}$$

where S_t and S_{t-1} are sets containing the infected IP addresses and $\|S\|$ is the cardinality of S. If both sets S_t and S_{t-1} are disjoint (6) results in 0. If S_t and S_{t-1} are equal the resulting value is 1.

Alternatively, we introduce the vector N_t for networks. In contrast to X_t whose vector components consist of the values 0 or 1 representing the occurrence of unique IP addresses, the components of N_t contain the number of reported IP addresses within a unique network[6]. Thus, the components represent networks and contain the number of reported systems within that network:

$$N(t) = (n_1(t), n_2(t), ..., n_m(t)) \tag{8}$$

where $n_m(t)$ is the number of unique IP addresses that are part of the network n_m and that are reported in the time slice t as being infected. Eq. (6) changes to

$$\frac{N_t N_{t-1}^T}{\|N_t\|\|N_{t-1}\|} \tag{9}$$

For example, it is reasonable to define the vector $N(t)$ to comprise all class-c networks of a ISP or other data contributor.

It is important to note that (9) may compute to 1 even if absolute values of N_t and N_{t-1} are different (for example $N_t = (1, 0)$ and $N_{t-1} = (5, 0)$). Thus (9) could result in a perfect similarity even if the absolute number of reported systems increases or decreases which is counter-intuitive. This can be addressed by a modification of (9)

$$\frac{N_t N_{t-1}^T}{\|N_{max}\|^2} \tag{10}$$

[5]For example, X_t would contain 256 components for the class-c network 192.168.0.X ordered from 192.168.0.0 until 192.168.0.255. Each component is either 0 or 1 dependent on the occurrence of the corresponding address

[6]For example N_t could be $(\sum(192.168.0.X), \sum(192.168.1.X), ..., \sum(192.168.255.X)$ where $\sum(192.168.0.X)$ is the number of reported systems in that network. In this example the IP addresses are aggregated at a level of class-c networks.

where $\|N_{max}\|$ is $max(\|N_t\|, \|N_{t-1}\|)$. Thus, the similarity decreases with increasing difference between X_t and X_{t-1}. We propose to use this measure if there is a large difference between X_t and X_{t-1}.

Similarity and Correlation between different Data Sources

The similarity measures in (6) and (9) can also be computed between different sources of data. Thus, the intention is to analyze if different sources report similar data. This can be used to estimate the security tools precision and the trustworthiness of the data contributors. Ideally, both sources gather similar data. If not, it is important to understand why the data sets differ among the sources. In addition, a large similarity between systems that are reported as being infected and sources of attacks likely indicate that this botnet is conducting these attacks.

The ACDC project provides long-term data of compromised systems on the Internet. The data covers different networks and sources. The aims of correlation is to find meaningful relationships within this data. This includes:

- It can be expected that a change in the data series is simultaneously observed by multiple data sources and networks. This can be confirmed by computing the cross correlation between the time series representing the number of compromised systems.
- It is likely that there is some form of relationships between different types of data. For example, a growth of a specific botnet might lead to an increase in attacks or spam emails.

The number of infected systems or attacks can be represented as a time series $X(t)$ whereas x_t is the number of infected systems or attacks at time t. Let $X(t)$ and $Y(t)$ be time series, then their correlation is given by

$$r_{XY} = \frac{\sum_i^n (x_{t_i} - \bar{x})(y_{t_i} - \bar{y})}{\sqrt{\sum_i^n (x_{t_i} - \bar{x})^2}\sqrt{\sum_i^n (y_{t_i} - \bar{y})^2}} \tag{11}$$

where $\bar{x}$ and $\bar{y}$ are the mean of X and Y, respectively (see [14] for further details). It is important to note, that we assume in (11) a lag of 0 between both series. The correlation coefficient r varies from 1 (perfect correlation) to -1 (perfect inverse correlation). If X and Y are uncorrelated the value is 0. For example, if both time series consist of random values[7], they are uncorrelated.

Our aim is to find specific correlations within the data:

Correlations between Malware X and Y represent the number of infected systems by different botnets or malware.

Correlation between botnets and attacks Alternatively, X could be the number of compromised systems related to a botnet, and Y could be the number of attacks (e.g. spam emails).

[7] For example, this is true for two jointly stationary time series consisting of Gaussian noise

Correlation of different networks and data sources If two data sources are trustworthy, they should share the same temporal behavior.

2.2.4 Autoregressive Moving Average Model (ARMA)

Although the models in Section 2.2 allow to efficiently assess important properties of the data they suffer from drawbacks: Correlations between the fluctuations are not considered. For example, it is reasonable to assume that an attack initiative of a botnet lasts for a couple of days and a growth of attacks will be observable for more than one day. For that reason, the number of attacks of two adjacent days are likely correlated, which is not considered by the model in (1). Considering the correlations of the fluctuations is especially important to reliably predict future measurements.

ARMA (Autoregressive Moving Average) models linear dependencies between the current value and subsequent values of a time series. It comprises a family of models that can be separated into the autoregressive (AR) and moving average (MA) part differing in their mathematical properties. These models are explained in the remaining part of this subsection. A critical task is the identification of an appropriate model and the estimation of its parameters. Unfortunately, there is no simple and straightforward approach and the choice of the model is sometimes neither easy nor obvious.

Mathematical Properties

The ARMA model is composed of the autoregressive (AR) and moving average (MA) part. Common to both models is that they consider linear dependencies in the data. Since both models differ in their properties we discuss their properties in two separated parts and start with the AR models.

AR Models

An AR model of order p is defined as

$$X_t = \phi_1 X_{t-1} + \phi_2 x_{t-2} + .. + \phi_p X_{t-p} + w_t \tag{12}$$

In (12) w_t is a Gaussian noise process $WN(0, \sigma^2)$ and $\phi_1 ... \phi_p$ are constants $\phi \in \mathbb{R}$. The simplest model of the AR family is the AR(1) model:

$$X_t = \phi_1 X_{t-1} + w_t \tag{13}$$

It is similar to the previously introduced model in (1). However, in contrast to (1) the AR(1) model considers a linear dependency between the lags t and $t+1$. Thus, if we have a large random fluctuation at t it affects directly the future value at $t+1$.

It is important to note, that even the model in (1) does not consider any direct correlation of the values at t and $t+n, n \geq 2$, there is an indirect correlation which can be seen in the autocorrelation function (ACF). The ACF computes the correlation of different lags in a time series and is defined by

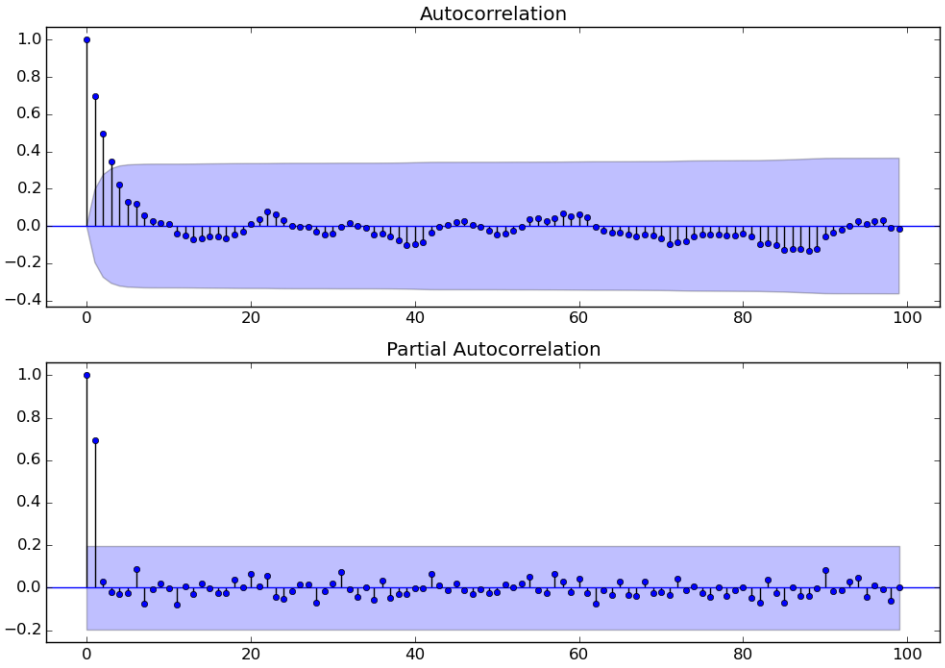


Figure 3: ACF (above) and PACF (below) of an AR(1) model.

$$R(\tau) = \frac{E[(X_t - \mu)(E_{t+\tau} - \mu)]}{\sigma^2} \tag{14}$$

where μ and σ are the mean value and the standard deviation of the time series and τ is the time-lag. It is important to note, that we assume a stationary behavior from which a constant mean μ follows. The previously mentioned indirect correlation are eliminated by the partial autocorrelation function (PACF). Thus, the PACF is similar to the ACF. However, in contrast to the ACF only the direct linear dependence between X_t and $X_{t-\tau}$ are considered. For example, an AR(1) model has only a direct linear dependence between X_t and X_{t-1}.

A time series that correspond to an AR(p) model has a characteristic pattern regarding its autocorrelations (ACF) and partial autocorrelations (PACF). An example for an AR(1) model is presented in Fig. 3. The exponential decay in the ACF is characteristic for this model. Thus, a random distortion at time t has an exponential decreasing influence on the lag $t + \tau$. As previously mentioned, the indirect dependence between X_t and X_{t-2} which can be seen in Fig. 3 in the ACF is eliminated in the PACF below.

MA Models

The MA model of order p is defined as

$$X_t = w_t + \phi_1 w_{t-1} + \phi_2 w_{t-2} + .. + \phi_p w_{t-p} \tag{15}$$

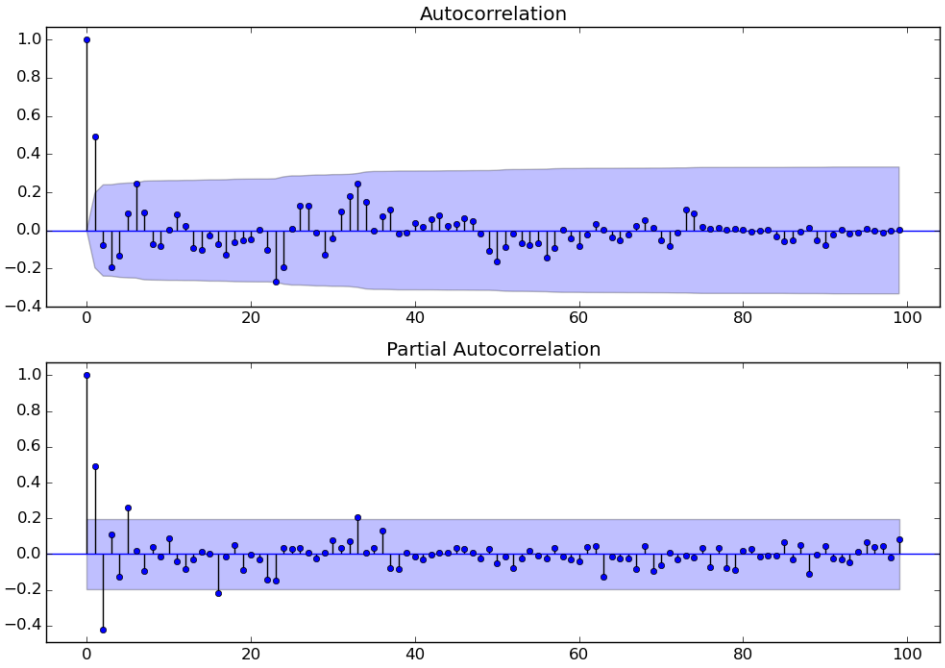


Figure 4: ACF (above) and PACF (below) of a MA(1) model.

In (15) w_t are Gaussian noise processes $WN(0, \sigma^2)$ and $\phi_1 ... \phi_p$ are constants $\phi \in \mathbb{R}$. The simplest model of the MA family is the MA(1) model:

$$X_t = \phi_1 w_{t-1} + w_t \tag{16}$$

The characteristic ACF and PACF of an MA(1) are shown in Fig. 4. The ACF shows a correlation between X_t and X_{t-1} whereas the other lags are unrelated. The PACF of the MA(1) model exhibits a characteristic exponential decay. It is important to note, the the characteristic pattern of AR and MA models are converse.

Model Identification and Parameter Fitting

As mentioned above there is no simple and straightforward approach and the choice of the model is sometimes neither easy nor obvious. Instead, multiple different approaches including the Akaike Information Criteria (AIC) and the Box-Jenkins method have been proposed. AIC is a measure for the relative quality of the model that maximizes a trade-off between the likelihood of the model fitting and the complexity of the model (we refer e.g. to [14] for more information). Although the accuracy of a model may benefit from more parameters, it may lead to an over fitting of the model. That is why AIC prefer a less complex model.

The Box-Jenkins method takes the ACF and PACF of the time series into account. As previously shown each MA and RA model has a characteristic pattern regarding

the ACF and PACF. For example, the number of related lags in the PACF indicates the order p of a RA(p) model. Both approaches are, for example, supported by Python "statsmodels" which also provides effective methods for the parameter fitting of a ARMA model and the prediction of future values.

An criterion for the appropriability of a selected ARMA model and its parameters is given by the ACF of the residuals. The residuals are the differences between the predicted values of the chosen ARMA model and the current time series. If the model is appropriate all values of the ACF of the residuals are uncorrelated.

Trend Models

A time series is weakly stationary if the mean does not change over time

$$E[X_t] = \mu \; \forall \; t \tag{17}$$

and the relations between the lags do only depend on the lag h and not the time t

$$Cov(X_t, X_{t-h}) = Cov(X_0, X_h) \tag{18}$$

ARMA models are weakly stationary if the AR component is weakly stationary. For example, this is true for the AR(1) model if and only if $\phi < 1$. Analogous constraints can be derived for higher order AR models. It is important to note that all time series that exhibit a trend or a seasonal component are not stationary. For that reason, it is important to assess a trend and seasonal component before applying an ARMA model.

A linear deterministic trend is given by

$$Y_t = \alpha + \beta t + X_t \tag{19}$$

where X_t is a stationary ARMA process. In this case, the ARMA time series results from subtracting the deterministic trend $\alpha + \beta t$ from (19). A time series that exhibits a linear trend is presented in Fig. 5.

As previously mentioned, the AR(1) model $X_t = \phi X_{t-1} + w_t$ is weakly stationary for $\phi < 1$. A special case is the random walk process which is given by $\phi = 1$:

$$X_t = X_{t-1} + w_t \tag{20}$$

Although the expected mean is $E(X_t) = 0$ the variance growth over time violating the second constraint in (18). A sample random walk process is presented in Fig. 6. It is important to note, that subtracting enables to transform the process in (20) into a stationary process:

$$X_t - X_{t-1} = w_t \tag{21}$$

For a given time series, the augmented Dickey-Fuller test (e.g. [14]) allows to determine stochastic trends.

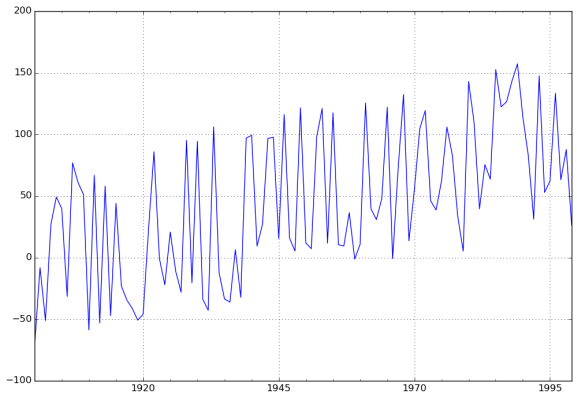


Figure 5: Sample time series exhibiting a linear trend.

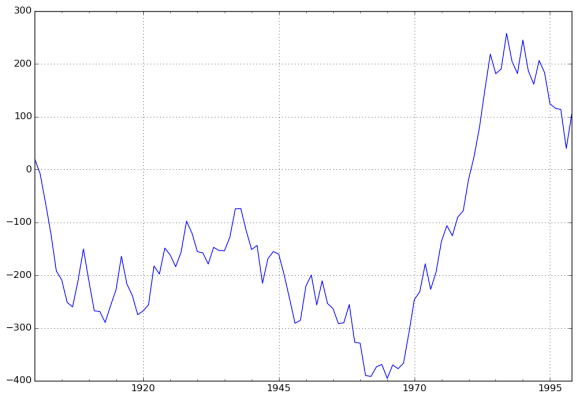


Figure 6: Sample time series exhibiting a stochastic trend.

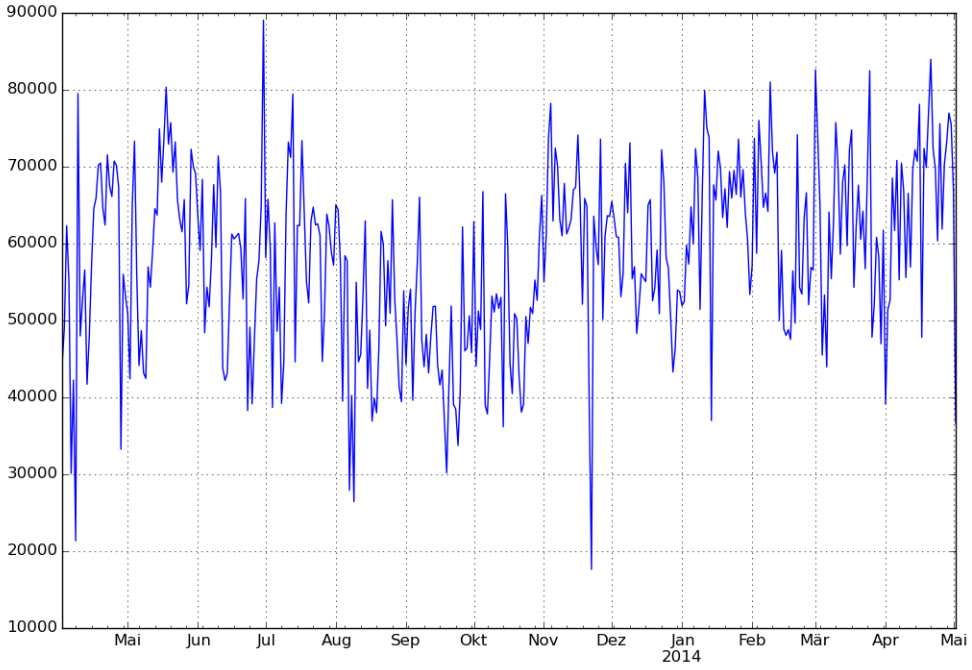


Figure 7: DSHIELD attack data showing unique sources scanning for port tcp/445.

Application of ARMA Models to DSHIELD Data

To test the applicability of ARMA models for attack statistics we applied these models to DSHIELD data ([19]). This data set is provided by the SANS institute and contains attack data gathered by firewalls. One of its main advantages is that SANS collects long-term time series of attacks dating back for more than a year. We decided to use the attack statistics related to the TCP port 445 which is frequently scanned by botnets and other malware for vulnerabilities in Microsoft Windows. For example, the W32.Conficker worm exploited a vulnerability that has been accessible on this port. The time series is presented in Fig. 7 containing unique sources scanning for port tcp/445. The time interval is from 5th April 2013 to 5th May 2014.

As described in Section 2.2.4 the ACF and PACF indicate applicable ARMA models. Both are shown in Fig. 8. While the ACF exhibits a slow decay the PACF indicates a direct correlation between the first two lags. Both the ARMA(2,0) and ARMA(2,1) models provided good results and minimized the AIC criterion. Finally, we selected the ARMA(2,1) model because this model had a slightly lower AIC value.

An indication of the quality of the model fitting is provided by analyzing of the differences between the prediction by the model and the data (residuals). A lack of fit is indicated if the residuals are not randomly distributed, for instance, if the ACF of the residuals shows a correlation between lags. The ACF of the residuals (first graph) as well as the prediction of the ARMA(2,1) model (graph below) are presented in Fig. 9 (first graph). Since the ACF of the residuals does not indicate any significant

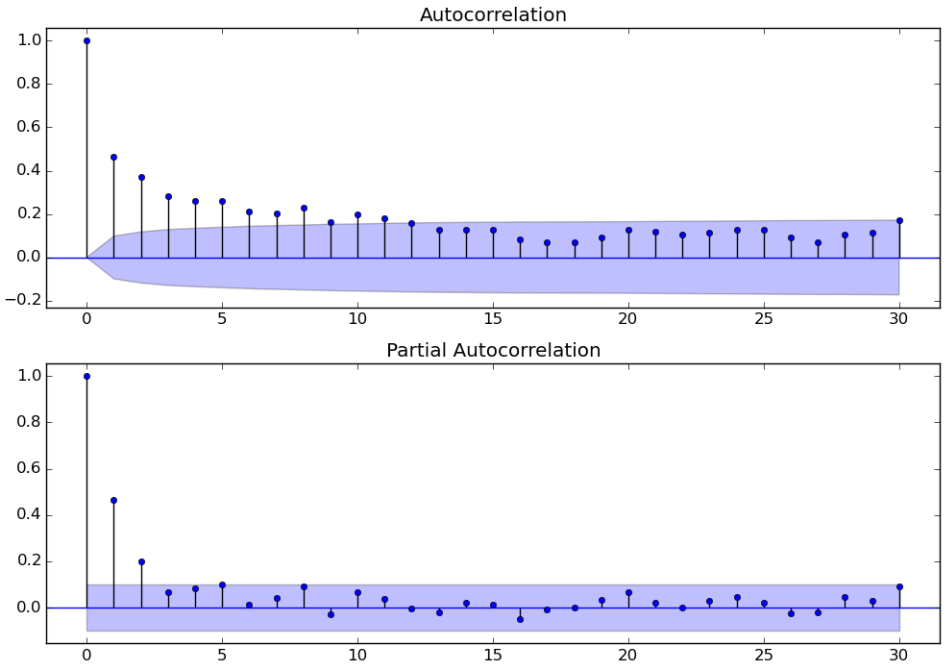


Figure 8: ACF and PACF of the DSHIELD attack data showing unique sources scanning for port tcp/445.

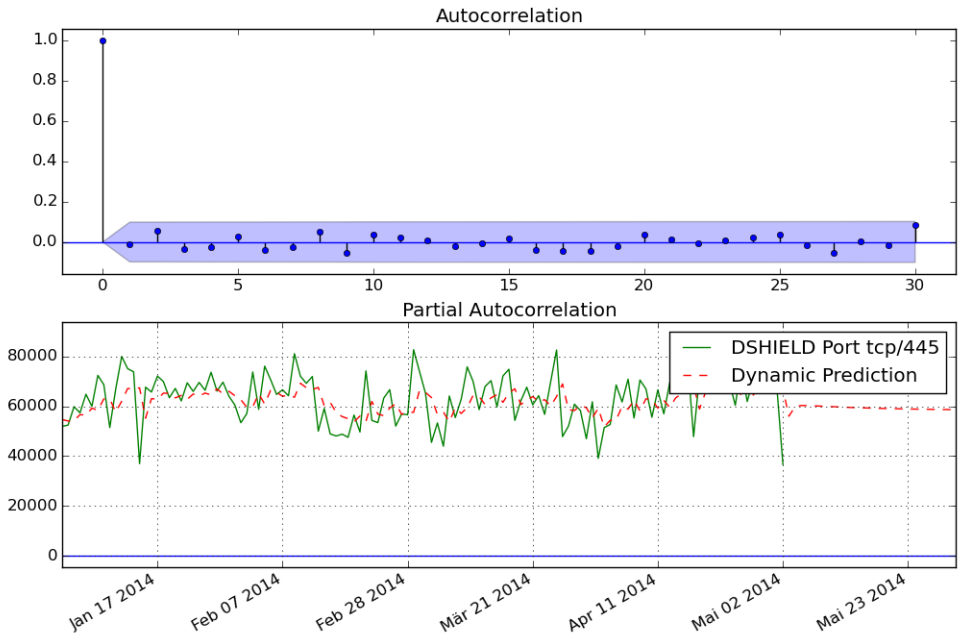


Figure 9: DSHIELD attack data showing unique sources scanning for port tcp/445.

auto-correlation, it good fit of that model can be assumed.

In summary, the main findings are:

- The ARMA(2,1) and ARMA(2,0) models provides a good fit for the DSHIELD time series of the attacks against port TCP/445. Thus, there is a significant correlation between the current value and the two subsequent values.

- The fluctuations are not completely random as assumed by the simple model in (1). Therefore, the ARMA models provide a better characterization of the mathematical properties of the time series which includes a better prediction of future values. It is reasonable to assume that other attack data also benefit from ARMA models.

- The applicability of the ARMA models indicates that the time series is stationary. Thus, there is no significant trend in this time period.

- The residuals are the difference of the model prediction (red dotted line in Fig. 9) and the time series (green solid line). As presented in Fig. 9 the ACF of the residuals are uncorrelated and allow in analogy to (1) to estimate the mathematical properties of the remaining random fluctuations.

2.3 Data Aggregation

Aggregation is a process in which data sets that share some properties are combined. A lot of work including, for example, [20] have been done in this area. The aims of aggregation are manifold:

Data reduction Some applications do not require all the details of the raw data. For example, metrics that count attacks may not need all consecutive connection attempts. Instead, related connections could be aggregated to a single event "Portscan". This prevents, for example, flooding in case of DDoS attacks. However, some approaches to analyze the data might require the raw data.

Data enrichment If multiple different data sets share the same IP address this data can be combined into a single set. This is especially valuable if the data originates from different sources and varies in the technical details. Alternatively, a set of related connection attempts can be grouped into a single event such as "Portscan" or "DDoS". Here, additional semantic information regarding the type of attack are added to the event.

Prevention of false-positives Moreover, aggregation can improve the trustworthiness of the data. An event that is reported from multiple independent sources or different sensors is less prone to a false-positive alert. Moreover, the aggregation allows to determine false positives if the aggregated data contains contradicting events.

Normalization of the granularity of data The majority of metrics are used to count attacks, compromised systems or malware. This requires a comparative representation of the data. For example, a data provider might deliver already aggregated data that cannot be directly compared to other raw data. Additionally, this step could involve the normalization of the data such as the malware and attack notation. For example, some AV vendors denote the Conficker as "W32.Conficker" and others as "Downup" or "Kido" worm. Furthermore, a lot of different variants exist lacking a precise discrimination.

Improvement of the stability Aggregation can be used to improve the stability of the data. For example, this applies to the sum of infected systems if the variations are random and are independent from each other[8].

It is crucial to specify aggregation criteria that conform to the further data processing in WP4. They specify common features that are used to group the data. For example, it is frequently used to group all connections that share the same source IP address. As introduced in [20] and formalised for IDS alert in [21] we use a data model to specify the specific aggregation criteria. Basically, the data model consists of a hierarchy of classes. The entities of each class specify the criteria used to group data. Furthermore, the model might consider relations between classes. We propose to use the following classes for the data model:

[8] This is resulting from the central limit theorem

Malware This class considers common features that are related to malware attacks. For example, an entity could be a unique malware family resulting from the normalized malware prevalence analysis of Task 1.1.8.

Attacks Entities of this class are, for example, portscans, DDoS attacks, Spam emails, and other attacks that are specified by the ACDC project members. Optionally, this class could consider hierarchical relations between entities. For example, DDoS attacks are frequently conducted by botnets. A detailed analysis of the relationships between ISPs, botnets, and spam abuse has been conducted in [22].

Infected systems This class considers common properties of incident data related to infected systems. For example, these properties regarding the source and target of these attacks that include IP addresses, networks, organizations, and nationalities. As previously mentioned, the DHCP churn and NAT are important issues for this class.

Criteria related to different classes can be combined. For example, this applies to the number of infected systems suffering from a specific malware (e.g. Trojan of the Zeus family). As mentioned above, this model is used to derive the specific aggregation criteria.

There are at least two methods that can be used for aggregating data that can be grouped into lossless and not lossless approaches. The first approach clusters a list of messages into a single message whereas all information is preserved. The second alternative summarizes the data omitting the original messages. For example, multiple connection attempts are aggregated into a single portscan report. Although this compresses the data, information is lost.

It is important to note, that the aggregation criteria heavily depends on the metrics that are detailed in Sec. 5. Furthermore, the experiments in WP3 also consider an aggregation and correlation of the data. Therefore, these processes are synchronized with the proposed metrics. For the first iteration we propose to consider the following criteria for aggregation:

Attacks These criteria are shared with WP3 experiments and relate to the previously mentioned attack class. It is important to note, that the experiments are work in progress and the resulting incident data will be dynamically extended. Therefore, additional criteria might be added to this class. Currently, we consider the following points that are reflected by the "report_type" field of the data format (see [23] for further information)

- Spam related attacks (e.g. all reports that share, for example, an identical spam message).
- Distributed denial of service (DDoS) attacks (e.g. all DDoS originating from a botnet C&C server
- Malicious websites (e.g. that share a domain)
- Fast Flux domains

- Mobile bots

Attack Sources These criteria consider the source of the malware or attacks

- Common IP address or network: All events are aggregated that share the same IP source address or network. Duplicates (identical data and time stamp) are omitted. It is important to note, that this aggregation criteria requires to cope with NAT and DHCP churn. If the data protection guidelines prevent aggregating IP addresses, network addresses (e.g. class-c network) could be used instead.
- Common AS: All events are aggregated that share the same autonomous system (AS) as source.
- Company Network: All events are aggregated that share the same company network as source.

Malware Families This criteria relates to a common malware family that is associated with a botnet. Starting point is the malware prevalence list of WP1. Crucial for this criteria is to agree on a normalized malware nomenclature.

As previously mentioned, we consider a dynamic feedback loop regarding the data processing. It is likely, that additional requirements arise during the evaluation of the data analysis. For that reason, the criteria are to be adapted dynamically to future needs.

2.4 Processes and Data Flows in WP4.1

An overview of the data flows is shown in Fig. 10 which presents two alternatives. Both diagrams have in common that the sensor data is retrieved from of the central clearing house (CCH). In the left alternative the data is pre-processed by a built-in component in the CCH whereas the processing on the right is implemented by a separate component. From there, the data is passed to the quality control and the data analysis.

Both alternatives have individual advantages and disadvantages. If long term data has to be aggregated, a separate component as shown in Fig. 10 (right) is advantageous because this method could better cope with potentially limited resources in the CCH. Furthermore, more complex algorithms that are, for example, used to correlate data could be better applied in an external component. On the other hand, a pre-processing component that is part of the CCH is easier to implement because existing features of the CCH are reused. Furthermore, no raw data has to be transmitted to external components. This is an important advantage because it allows to comply with data protection requirements.

Fig. 11 presents an overview of the data processing in WP 4.1. As described in Section 2.2.1 the first step is sorting out all obvious bogus data. After that, the data is normalized. As previously pointed out, the most critical issue of the data normalization is to cope with different naming conventions of malware and botnets. In the next step

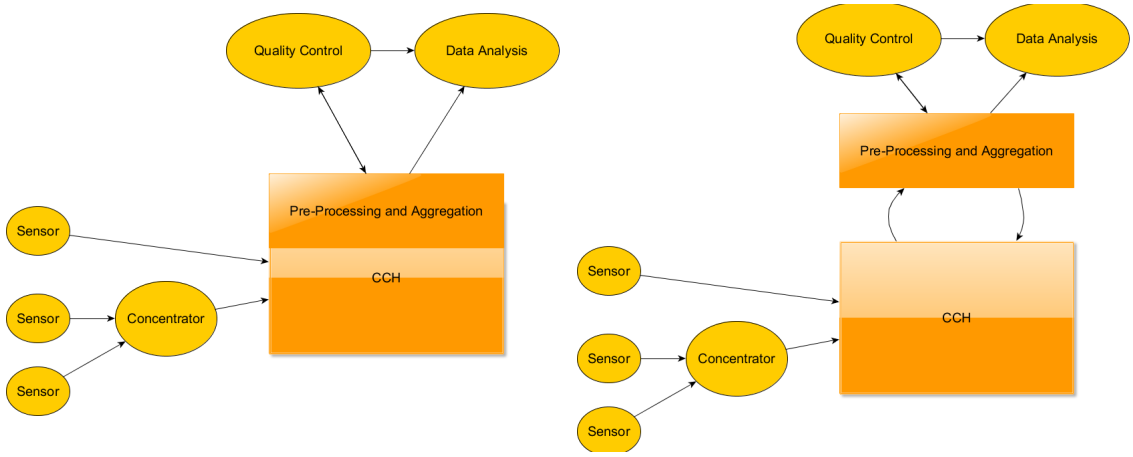


Figure 10: Overview of the Data Flow in WP4.1

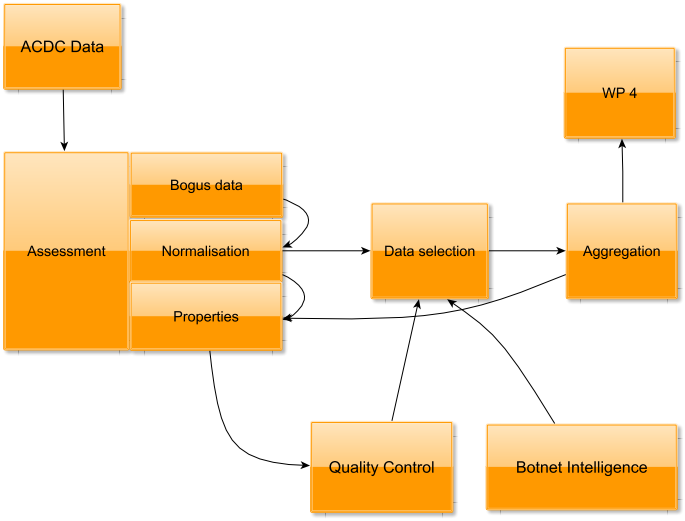


Figure 11: Overview of the Data Processing in WP4.1

the statistical properties are computed that relate to the quality control of the data. Furthermore, this component identifies gaps in the data. Optionally and only in cases in which prediction is applicable, gaps in the data could be closed by predicting missing data.

In combination with botnet intelligence the data is used to extract stable parts that fulfills the quality requirements. Botnet intelligence can contribute to the quality control by adding information about the reliability with which bot are detected. For example, a botnet might migrate the bots or change its communication behavior resulting in artifacts in the botnet detection.

Before applying the metrics, the data is aggregated according to some of the previously specified criteria. Furthermore, duplicates are detected that share exactly the same data fields containing redundant information. As previously pointed out the definition of duplicates depends on the metrics. Thus, at this step a less strict definition of

duplicates is not reasonable.

Calibration Phase

There are many complex relationships between the data pre-processing and the subsequent analysis. For that reason, we use a calibration phase to test the previously introduced procedures. According to our expectation, the calibration process is very important and has to be thoroughly tested. It is required to test the applicability of the aggregation criteria and quality control.

In the first phase, all raw data will be delivered to the data analysis in WP4.2. The experiences will be used to step-wise refine the aggregation criteria and quality control:

- Do the aggregation criteria comply to the security metrics?
- Does the data fulfill the statistical requirements?

It is likely, that data characteristic as well as data quality change within the project duration. Therefore, we use the metrics that are detailed in Sec. 2.5.4 to assess the properties of the provided data. If the assumptions detailed above proves to the true other, more specific, quality metrics could be added to address more specific requirements.

Data Exchange Formats

A data exchange format is required that enables the transmission of pre-processed data to the analysis in WP4.2. Critical requirements are the capability to handle aggregated data and the flexibility to support most of the properties of the other data formats that are used in the ACDC project. Thus, an extensible data exchange format is advantageous. Basically, there are two alternatives: a proprietary format based on JSON and the XML based format IDMEF.

The assessment of data formats for ACDC in Task 1.1.7 resulted in multiple different formats that focus on different aspects. To standardize the information, a set of common data fields have been proposed that is defined by a JSON schema. The first alternative is to extend this schema to address aggregated data. Technically, this requires to add cross-references and data types (e.g. “array”) for aggregated sources or targets. For example, the minimal dataset as defined in [23] provides the field “additional_data” where an additional JSON object could be stored. This solution requires only a minor effort for the implementation and supports the available solution of the project. Furthermore, the JSON schema can be adjusted to new requirements rather easily.

An alternative is IDMEF which supports aggregation and correlation, and provides a quasi standard for the exchange of IDS alerts and malware attacks. This makes it easy to exchange the resulting data set (e.g. fully anonymized and filtered) with third parties associated to the project. However, the data format is very complex and requires some effort to support it.

Data Protection

As presented in Fig. 10 the data pre-processing of WP4.1 consists of gathering, processing and exchanging the ACDC data related to security events. Since all these tasks involve potentially person related data such as IP addresses their handling either requires a legal basis or the data has to be anonymised prior to its usage. Even if the metrics rely on aggregated data which omits all person related information the process in which the aggregation takes place has to process the raw data that may contain person related information. It is important to note, that the gathering, processing, and data exchange have to be considered separately if multiple parties are involved:

Data Gathering WP4 relies on data that is submitted to the CCH whereas a legal basis exists for the gathering and processing of this data related to incident handling. A critical point is whether the legal allowance can be extended to use the data for the intended statistical analyses[9] or not. If not, the data has to be anonymized prior to statistical analysis. Alternatively, the anonymization can be implemented by the aggregating component (e.g. by summarizing all events that originate from a specific network).

Data Processing As shown in Fig. 10 there are two alternative data flows differing in the location of the data processing. The left alternative proposes that the data processing which includes the data aggregation is conducted in the CCH. This is advantageous because there is no transmission of the data that may violate the data protection guidelines. The right alternative considers to exchange the raw data between the operator of the CCH and other project members where the data is processed. This step is considerable and requires a legal basis unless the data is anonymized before the transport. However, a full anonymization might render the data useless for a further analysis.

Data Exchange The data exchange to subtask WP4.2 is the most critical point because it means sharing potentially personally identifiable information (PII) with other ACDC partners for the purpose of a statistical analysis of this data. It is reasonable to consider technical means to anonymize or pseudonymize the data before the transport. As previously mentioned, the data aggregation could be used to anonymize the data. It is important to synchronize this step with the specification of the metrics.

2.5 Implementation of the Data Processing

Based on the requirements of the data processing outline introduced in Section 2.4 we next detail the implementation covering the processing of the data and workflows to introduce new metrics.

[9] In legal terms this is a change of purpose.

2.5.1 Acquiring and Pre-Processing the Data

Application of metrics rely on attack data that is submitted by the partners of the ACDC project. It is important to note, that the data has to comply with the requirements of the specific metrics which comprise the quality, availability, and completeness of the data.

The overall aim of the workpackage 4 is to assess the impact of the project pertaining the countermeasures against botnet threats. It is reasonable to assume that this goal can be achieved without the need to process person related data in general, and IP addresses in particular. If required, pseudonyms could be used that replace IP addresses or other sensitive data. Here, we adopt the "**Research**" workflow that has been detailed in Deliverable 1.7.2 in [23] to acquire data.

Application of the metrics implies requirements pertaining the quality of the raw data. Since this pre-processing does not depend on a specific metric it is reasonable to define processes that precede the application of the metrics.

Reduction of False-Positives by discarding known non-malicious scanners This process aims at reducing the number of known false-positives. An increasing number of security researchers are scanning the Internet for services and vulnerabilities distorting the reports and statistics of attacks and sources. Reports pertaining these system have to be discarded.

Plausibility Tests Most importantly, the plausibility of the timestamp has to be checked to discard all events that have a timestamp that is either in the future or outdated.

Reduction of Duplicates In this case duplicates are defined as events that are accidentally submitted multiple times. It is important to note, that a broader definition of duplicates could depend on the applied metrics.

Data Standardization Standardization addresses especially the specification of malware and botnets. This standardization is enforced by the CCH during import.

Data Normalization This affects the granularity of the data. Selected data sets shall be normalized to enhance comparability.

This also includes timestamps. They should all be set to the same timezone making it easier for comparison. This may be included in Data Standardization during import into the CCH.

Data Enrichment Data Enrichment includes e.g. adding the AS-number to a dataset.

Quality Control Quality control excludes obviously bogus data such as private IP address ranges.

2.5.2 Unified Data Format

This section proposes an additional data exchange format that addresses the specific requirements for the data exchange of data in WP4.1 and 4.2 and takes the specific metrics in Sec. 5.1 and Sec. 2.5.4 into account. It will be added to the Deliverable D1.7.2 after it reached its final version. It is important to note, that this format is specifically defined for data that is delivered by the "Research" Workflow. As such, no fields are provided that contains person related or sensitive data such as IP addresses.

Statistical Data Exchange Format		
This data format is designed to extract statistical data from the CCH.		
Required Fields		
report_category	string	The category of the report. This links the report to one of ACDCś schemata. A report category has the format `eu.acdc.aggr.<identifier>`.
report_type	string	The type of the report. This is a free text field characterizing the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb this should not be longer than one sentence.
timestamp	string: format timestamp	The timestamp when the first reported observation took place. This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.
aggregation_type	Boolean	True, if data is aggregated, False if not
Required Fields for Aggregated Data		
These fields apply only for aggregated data		
measurement_window	integer minimum: 0	The time interval in hours in which the aggregated data falls into, e.g. 24h.
aggregation_field	string enum, e.g: Submission Key, Report Category, Source Value	The Field that is used to aggregate the data. For example, "Report Category" state that all reports that have the same value are enumerated in the measurement window.
aggregation_criteria	string	Value of the criteria: e.g. AS680
aggregation_value	Integer	Number of aggregated data entities
Required Fields for non-Aggregated Data		
These fields apply only for non-aggregared data		

source_ASN	string	Number of ASN the attack originated from
source_type	string enum: hash, pseudonym, network	The type of the reported object. Since this format is associated with the Research Workflow, strict data protection is considered. This requires to either fully anonymise the data (e.g. by discarding the last Byte of the IP) or to replace the IP with an pseudonym.
source_value	string	The identifier of the reported object like its pseudonymised IP address or hash. If the source type "pseudonym" is selected, a prefix preserving psudomysation algorithm is applied that preserves the data type for IP addresses.
Optional Fields for Specification of the Source		
These fields apply for a further specification of the source.		
organization	string	Source of the attack, e.g. Deutsche Telekom A.G.
country	string	Source of the attack, e.g. Germany (Geolocation DB).
city	string	Source of the attack, e.g. Munich (Geolocation DB).
Optional Fields		
Other optional fields		
report_id	Integer	The ID of the report in the CCH. This will be set by the CCH.
reported_at	string: format timestamp	The timestamp when the report was submitted to or created by the CCH.
report_subcategory	string	The subcategory of the report. This is used to categorise different types of similar reports that have mostly the same fields. It is defined as an enum in the schema of the report category.
aggregation_description	string	Free text description of the aggregation criteria, e.g. shared ASN of network
ip_protocol_number	integer minimum: 0 maximum: 255	The RFC 790 decimal internet protocol number of the connection.

src_port	integer	The source port of the connection. This is always the remote port from the perspective of the attacking system.
dst_port	integer	The destination port of the connection. This is always the remote port from the perspective of the reported system (i.e., the one identified by source_value). It can for example be the port of a honeypot that was contacted to infect it.
type_of_connection	string	Type of the connection according to Maxmind connection DB, e.g. DSL
toplevel_domain	string	Top-level domain according to reverse DNS data , e.g. dtag.de

Table 1: Data format specification to extract data from the CCH

2.5.3 Methodology and Method for Adding Metrics

The application of new metrics cannot be achieved without an appropriate pre-processing of the data. We here introduce a workflow and specification for metrics that is used to implement the required pre-processing and data export in the CCH. It is important to note, that different partners associated with different workpackages are required to smoothly set up the data processing. For that reason, a specific and unambiguous workflow is necessary to achieve this aim. Moreover, all metrics have specific demands pertaining the required input data and quality. If these requirements are not satisfied the results are either invalid or may be misleading. For example, a fluctuation of a data source might be erroneously interpreted as a change in the number of bots that belong to a specific botnet. That is why care has to be taken if a new metric is added.

Before the CCH can provide a data feed for a metric several tasks must be fulfilled. We follow a two-man-rule-based methodology. Precisely it should be a multi-step collaborative four-eyes principle with both parties being experts in the field of processing and metric definition. This includes a third party reviewing and evaluating changes or new subjects before they are submitted to the operator of the CCH. The basic idea for this procedure is to reduce errors, misconceptions and misconduct. Reviewing the intention and the proposed algorithm for processing can also reveal and remove possible ambiguity before the operator of the CCH is asked to implement the metric and its processing steps.

Preliminary Workflow

1. All required information for a metric must be defined. This typically includes the data defined for the template as provided by [24].

2. Before the defined processing is forwarded to the operator for implementation a review of its pseudo code shall be conducted to remove possible ambiguity.

3. If any changes are advised by the reviewer on either the metrics definition or the processing pseudo code the metric's creator shall check the proposed changes whether they comply with his intentions.

   If the metric's creator accepts the changes the metric and its processing pseudo code can be transmitted to the CCH's operator for implementation.

   Otherwise, if the metric's creator does not accept the changes the process starts again.

4. If required or requested by the metric's definition the operator of the CCH must enrich the data as specified. If the requested additional data cannot be supplied by the operator the operator shall follow the guidelines to be specified in that very section. In any case in which those instructions are not supplied the default action 'DROP' shall be applied resulting in excluding that data set from further processing.

5. If required, the operator of the CCH must implement the pseudo code of processing to prevent any ambiguity. If ambiguity is detected the metric's implementation shall be stopped until the ambiguity is resolved.

6. If application of the metric is applied external to the CCH the operator of the CCH must implement the data exchange format to be specified. The metric's creator must also define its workflow. During the project's lifespan Deliverable D1.7.2 shall be updated to include the data exchange format.

7. If all previous steps are successfully performed the metric can go live.

2.5.4 Processes and Metrics for Quality control

In this section we present processes and metrics that are devoted to the quality control process of the data. The metrics are intended to gather data for a statistical analysis using the algorithms described in Section 2.2 for time series and similarity measures.

Process for Quality control

Objective of the Process This process aims at reducing the number of known false-positives and to drop all data that does not conform to quality requirements. It comprises the following steps:

Black-listing non-malicious systems Since an increasing number of security researchers are scanning the Internet for services and vulnerabilities these events

distort the reports and statistics of attacks and sources. These systems are omitted from the attack data[10].

Validation of Timestamps Timestamps are valid if they are within a configurable time window. For example, it is reasonable to discard all data whose timestamp is either in the future[11] or exceed a backward limitation of 1 week.

Validation of IP addresses All events that contain an IP address that is within private or unassigned IP address space should be discarded.

Legal Statement Since the data processing is performed in the CCH, no legal issues are expected.

Data Selection The white-list is applied to all data that is submitted to the CCH.

Data Enrichment Not applicable

Data Exchange Format Not applicable

Semantics of the Process The semantics of the data are not changed.

Application of the Process The process is applied as a filter to all data that is submitted to the CCH. Thus data processing takes place in the CCH.

Quality Assessment: Number of Reports

The Metric's Objective The metric aims at identifying gaps or anomalies in the data that might be caused by a failure of the data submission or the sensor. This is achieved by computing the total number of reports pertaining all data sources in a specific time interval. Data sources are unique keys that are used to submit data to the CCH. In the context of the metrics, gaps are time intervals where no reports are submitted or where the number is significantly less than the average number of reports.

Legal Statement Since only statistical data is collected legal issues do not arise from this metric.

Data Selection and Quality Criteria This metric is applied to all data sources (CCH keys) that submit data to the CCH. Since this metric is used to assess the data quality, no quality criteria are expected.

[10] As long as their IP addresses are known.
[11] Considering the timezone!

Data Exchange Format Sample for Quality Metrics I	
Required Fields	
report_category	`eu.acdc.aggr.quality1.`
report_type	Statistical data to identify gaps and anomalies in the quantity of submitted reports.
Time Stamp	string: format timestamp
aggregation_type	True
measurement_window	24
aggregation_field	`Submission Key`
aggregation_criteria	`<Instance of Submission Key>`
aggregation_value	Integer
aggregation_description	Aim is to count all reports that has been submitted with a unique Submission Key
Optional Fields	
Report ID	Integer
Reported at	string: format timestamp

Table 2: Data Exchange Format Sample for Quality Metrics I

Data Enrichment Not applicable

Data Exchange Format The data exchange format as specified in Section 2.5.2 is used. See Table 2.

The Metric's Semantics This metric measures the number of events that are submitted by the different data source. Here, a data source is associated with a unique CCH key that is used to submit data.

Quality Assessment: Distribution of Reports

The Metric's Objective The metric aims at identifying gaps or anomalies in the data that pertain the distribution of reported systems. The metrics compute the number of reports that are associated to ASNs and if feasible networks. The assumption is that anomalies and gaps distort the statistical stability of the data.

Legal Statement Since only statistical data is collected legal issues do not arise from this metric.

Data Selection and Quality Criteria This metric is applied to all data sources (CCH keys) that submit data to the CCH. Since this metric is used to assess the data quality, no quality criteria are expected.

Data Exchange Format Sample for Quality Metrics II	
Required Fields	
report_category	`eu.acdc.aggr.quality2.`
report_type	Statistical data to identify gaps and anomalies in the source distribution of submitted reports (ASN or network).
Time Stamp	string: format timestamp
aggregation_type	True
measurement_window	24
aggregation_field	`ASN` or `source_value`
aggregation_criteria	`<Instance of ASN or Network>`
aggregation_value	Integer
aggregation_description	Aim is to count all reports sharing a common ASN or network as source of the attack
Optional Fields	
Report ID	Integer
Reported at	string: format timestamp

Table 3: Data Exchange Format Sample for Quality Metrics II

Data Enrichment The ASN and optionally network has to be derived from the source.

Data Exchange Format The data exchange format as specified in Section 2.5.2 is used. See Table 3 for details.

The Metric's Semantics This metric measures the number of events that are associated with an ASN or network as source of the attack.

2.6 Summary

The overall aim of the data processing is to provide a statistically stable data set that allows to derive the requested results of the propsed metrics. In this section we presented requirements and statistical approaches to achieve these aims. Furthermore, we presented collaborative workflows to specify and to implement the data exchange and the application of metrics. To address data protection requirements the data processing will be conducted as far as this is possible in the CCH, which mitigates the problem of transfering sensitive data.

3 Botnet Metrics

A critical problem that all botnet mitigation efforts face is this the lack of consistent metrics to measure the impact of countermeasures across networks and over time. The absence of metrics also undermines the incentives of market actors to act against botnets.

It is costly for ISPs and other market players to mitigate botnets. Previous research suggests that most ISPs are not attacking the problem at the scale at which it currently exists. The effectiveness of mitigation measure cannot be established without accurate and reliable reputation metrics [25]. Without such metrics, there is only anecdotal evidence that cannot be reliably interpreted.

As the "U.S. Anti-Bot Code of Conduct (ABC) for Internet Services Providers (ISPs)" states, "the current inability to uniformly measure the bot population and the results of activities to reduce bots" [26] is a key barrier to implement broad botnet remediation initiatives [27].

The lack of metrics also create an information asymmetry which impedes the functioning of markets and may even result in market failure. If consumers, businesses, regulators and other stakeholders cannot reliably tell more secure ISPs from less secure ISPs, then the market incentives to invest in mitigation are weakened.

Therefore, in WP4, we will investigate what kind of network measurement data is required to statistically account for botnet population in the networks of ISPs, and how this data can be expressed in mature and comparative reputation metrics at firm and country level that provide more transparency regarding the efforts of providers and countries in mitigating botnets.

Comparative metrics will be developed at the level of countries and ISPs, focusing on, *e.g.*, the number of bots per user in access networks, persistence of those bots, C&C infrastructure density, and other metrics.

The goal of this chapter is to provide the foundations upon which we can build to define robust comparative botnet metrics. It is structured as follows: In Section 3.1, we present background information on metrics in computer science – from the networking and software engineering disciplines – which have a story of development and have been widely used and validated, which we base upon while defining our botnet metrics. Then, in Section 3.2 the requirements we envision for botnet metrics. After that, (Sec. 3.3) we present a survey on the current botnet metrics. Finally, in Section 3.4, we map the requirements that the existing botnet metrics fail to meet.

3.1 Background: Metrics in Networking and Software Engineering

In general, a metric can be defined as "a standard of measurement". In computer science, metrics have been a frequent term and research topic in mostly networking and software engineering disciplines. In this section, we cover how metrics are addressed by these two disciplines.

3.1.1 Networking

In networking, metrics have been usually associated with performance and routing. For example, RFC 2330 [28] – “Framework for IP Performance Metrics”, has the goal of “achieve a situation in which users and providers of Internet transport service have an accurate common understanding of the performance and reliability of the Internet component that they use/provide.” It’s definition of quantitative metric is as follows: “In the operational Internet, there are several quantities related to the performance and reliability of the Internet that we’d like to know the value of. When such a quantity is carefully specified, we term the quantity a metric.”

It also establishes the criteria that performance and reliability metrics must satisfy:

- The metrics must be concrete and well-defined,
- A methodology for a metric should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, the same measurements should result in the same measurements.
- The metrics must exhibit no bias for IP clouds implemented with identical technology,
- The metrics must exhibit understood and fair bias for IP clouds implemented with non-identical technology,
- The metrics must be useful to users and providers in understanding the performance they experience or provide,
- The metrics must avoid inducing artificial performance goals.

RFC 2544 [29], on the other hand, defines a “ benchmarking methodology for network interconnect devices”. In addition, metrics have been used in network routing, as a value to be used by a routing protocol to determine whether one particular route should be chosen over another (e.g., hop count, latency, packet loss, etc.).

3.1.2 Software Engineering

Software engineering encompasses a series of software metrics, which servers as a measure of a property of the particular software code or its specifications [30, 31]. Software metrics elaborate some of the characteristics which can be generalized and applied to botnet metrics. For instance in [30] Kaner *et. al* proposes a framework to evaluate defined metrics based on some questions which include defining the purpose, scope , scale, variability, among others. Moreover, the IEEE body has defined a standard for a Software Quality Metrics Methodology (IEEE 1061) [31], in which the subject is extensively covered.

3.2 Botnet Metrics Requirements

In order to be useful, the botnet metrics must fulfill a series of requirements. The more fundamental requirements are related to the software engineering discipline, from the IEEE Standard for a Software Quality Metrics Methodology (IEEE 1061) [31]. Kaner *et al.* [30] summarizes the requirements as follows:

- Correlation. The metric should be linearly related to the quality factor as measured by the statistical correlation between the metric and the corresponding quality factor.
- Consistency. Let F be the quality factor variable and Y be the output of the metrics function, $M : F- > Y$. M must be a monotonic function. That is, if $f_1 > f_2 > f_3$, then we must obtain $y_1 > y_2 > y_3$
- Tracking. For metrics function, $M : F- > Y$. As F changes from f_1 to f_2 in real time, M(f) should change promptly from y_1 to y_2 .
- Predictability. For metrics function, $M : F- > Y$. If we know the value of Y at some point in time, we should be able to predict the value of F.
- Reliability. "A metric shall demonstrate the correlation, tracking, consistency, predictability, and discriminative power properties for at least P% of the application of the metric.

Besides these fundamental requirements, we add extra requirements for the botnet metrics:

1. Metrics are comparative across networks.

   Metrics have to be useful to compare networks. This implies not only that they are based on data that is collected across networks, but also that the metric is normalized to take network properties into account when calculating infection levels.

2. Metrics are comparative over time.

   Measurements might fluctuate because of the rise and fall of specific botnets, changes in the criminal business models behind botnets, improvements in detection techniques, or better evasion strategies by attackers.

3. Metrics are able to take input of differing data sources with different biases.

   Given that all data sources have intrinsic limitations, a metric ideally would take different sources as its input and produce a consistent rate as its output.

4. Metrics are reliable, which means that they must have an acceptable levels of random error

   Some data sources, such as spam traps, are basically sampling strategies. To generalize from these samples to a metric for the whole network might introduce

significant random error, which means the metric would fluctuate even if the infected population remains the same over time. RFC 2330 further highlights some of the common sampling strategies which can assist in correctly identifying measurement window.

5. Metrics are valid, which means that they accurately represent the actual infection level in a network. In other words, they do not have systematic errors.

   Many data sources suffer from the problem of capturing only a partial view. A sinkhole might see all bots in a botnet, but only for that botnet. A spam trap sees bots from different botnets, but never all of them. Each source suffers from different systematic biases. For example, the geography of the spam trap may influence which infected hosts try to reach it. The geography of a host may impact the probability it is attacked with a certain type of botnet infection and, hence, whether it would show up in a sinkhole of that botnet.

6. Metrics are normalized, so that they express the infection level, rather than the size of the network or other properties.

   If we count bots per country, the country with more Internet subscriber will typically have more bots. This does not mean it has higher relative infection rates, e.g., more infections per subscriber. To compare infection levels we need to normalize number of infections by subscribers and potentially other factors.

7. Botnet metrics take into account impact on users.

   Not all botnet infections are equally active or dangerous. For example, some machines are persistently infected but may not be active because the botnet is abandoned. Those infections that are active should be given higher weight. Similarly, among active bots, those involved in more impacting activities could also be given higher weight, e.g., being infected with spambot infection might be considered less of a threat to the user impact than an infection with a banking Trojans. We might also consider taking the potential to do damage into account by looking at their geographical location, their bandwidth or their presence in sensitive networks like banks, military or government systems.

3.3 State-of-the art on Botnet Metrics

In this section, we cover the state-of-the-art on botnet metrics. We have carried an extensive literature review on the current metrics and cover and present an analysis on the current metrics.

Moreover, we also propose a classification of these metrics into three categories, as follows:

- IP-based: metrics that use the originating IP address of traffic related to infected machines. For example, a standard metric is to count the number of unique IP addresses sending out spam from ISPs' network, where each unique address is interpreted to represent at least one spambot. However, due to effects of DHCP

and NAT, IP addresses are far from perfect proxies for the number of compromised hosts [12].

- Host-based: metrics that are build based on data that directly and reliably identifies individual hosts on the Internet. An example metric is the number of bots in ISP networks as seen in data from anti-virus software deployed at customer end point devices or in data from sinkholes for those botnets that assign unique identifiers to its bots

- Proxy-based: metrics that are estimations based on traffic volume associated with botnets (spam, ssh attacks, DDoS, etc). We refer to them as proxy-based methods as the data does not have any identifiers (neither IP address nor host), but rather assumes a correlation between the observed activity and the number of infected hosts in a network.

Each individual metric that falls into these categories can be further extended, by aggregation (per AS, country, etc.), by normalization (e.g., number of spamming IP addresses divided by size of IP pool or user population), or by being turned into a rating based on a different scale than the original metric. All of these types metrics allow for some type of comparison of different Internet intermediaries. Figure 12 summarizes this relationship.

Figure 12 shows these different metrics as consecutive steps away from the raw data captured at collection points. While these different types of metrics do typically follow each other, this is not necessarily the case, however. Once could produce a ranking metric without normalizing the botnet counts, for example. We would consider such metrics to be rather useless for evaluating and incentivizing mitigation, as they do not take into account important differences among networks and countries. That said, such drawbacks don't stop some security vendors from producing ranking of the most spamming or the most infected countries in the world.

Table 4 summarizes the state-of-the-art of existing metrics and their respective reference. It presents a list of current metrics used in the research on botnets. We also classify them accordingly to the measurement window that was used, the underlying data source, the metric's aggregation level, whether normalization was applied (e.g., considering the size of the AS) and if a final ranking was produced. These metrics cover both industrial and academic communities.

In Table 4, we can see that many metrics are based on IP addresses or traffic volume, even though host-based metrics are more precise, and often offer the additional benefit of including more precise information on the detected malware strains. This reflects the wealth and accessibility of data on which metrics can be based. The data required for host-based metrics is much more scarce and often derived from a very specific collection mechanism with unknown biases – i.e., the anti-virus client of a specific vendor, who does not reveal how many clients they have in each network. To the extent that the data is available, it is often proprietary and less accessible to independent researchers or intermediaries interested in transparent metrics based on (semi) open data.

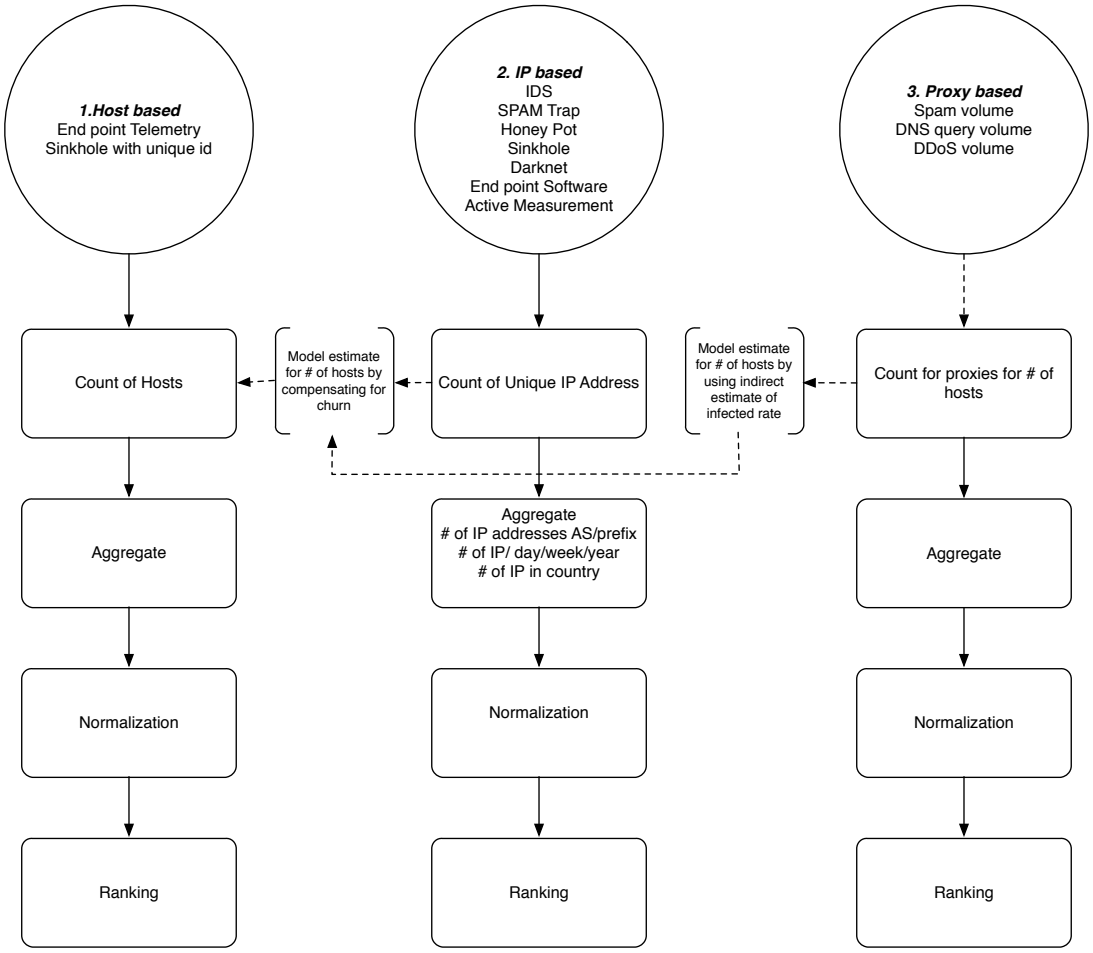


Figure 12: Taxonomy of botnet metrics

We can also see that aggregation of mostly done at the level of ASes or countries. These might reflect research questions that needs to be addressed, but can also be attributed to the fact that these aggregations are relatively easy to execute. Several publicly available datasets which can help map IP addresses to geogrpahical location and ASN [32, 33, 34, 35]. This stands in contrast to the lack of aggregation to the ISP level (where one ISP may operate several or even many ASes). This is unfortunate, as the bulk of the infections are concentrated in ISP networks and ISPs have become critical players in botnet mitigation [36, 37, 38, 39]. The dearth of ISP-level comparisons might be due to unavailability of available tools to reliably and historically accurately map IP addresses to ASes to ISP.

Metric	**Type**	**Measurement Window**	**Data source**	**Agg.**	**Normalized**	**Ranking**
estimated #_of_hosts [40, 41]	IP	per hour / per day	Sinkhole	-	-	
extrapolated # of bots [42]	IP	per day	Honeynet and Darknet	# of Source ASes	Avg, number of IP scanned per botnet	
# of bots per AS [43]	IP	per day	Spam email	ASes, BGP		Top 20 AS and countries sending spam
Malscore [44]	IP	60 days	IRC-based botnets HTTP-based botnets	ASes	Size of AS	AS Ranking
Botnet activity [45]	IP	per day	Spam Data	ISP	# of subscribers per ISP	ISPs
CCM [46]	Host	quarter	Malwares cleaned	Country	Number of computers cleaned for every 1,000 unique computers executing the Malicious Software Removal Tool	Countries
Unique malicious objects [47]	Host	quarter	Malwares detected	Country, % of unique attacked users		Countries

spam_volume [48]	Host	quarter	spam, Web Exploits, Malware, DDoS	Themes for spam, Platform (Windows, Linux, Mobile) Country		Countries, Platform
# bot IDs per countries [12]	Host	10 days, per hour , per day	Sinkhole	Country		Countries
Suspiciousness score [49]	Proxy	per day	recursive DNS (RDNS), spam			
# of malicious domains [50]	Proxy	1.2 days	DNS , spam			
Active Size [51]	Proxy	per day	Spam emails	Clustered emails into spam campaigns / # of countries participated in sending spam		
Badness score [52]	Proxy	per day	Click-spam	Search Ad Network, Mobile Ad Network, Contextual and Social Ad Networks		
AS_{rank}[53]	IP	per day	malware	ASes	Size of AS	AS
max_spam _vol_per_asn min_spam _vol_per_asn [25]	IP	per day	spam	ASes, Country	Size of AS	Country
%_malicious _hosts _per_asn [54]	IP	30 day	Phishing, malware, spam	ASes	Size of AS	% of malicious hosts per asn
% spam caught [55]	IP	per day	spam	ASes	Size of AS, Size of subnet	reputation _subnet reputation_asn
cluster based reputation [56]	IP	per day	spam emails	BGP prefix cluster , DNS cluster		
% spam caught [55]	IP	per day	spam	ASes	Size of AS, Size of subnet	reputation _subnet reputation_asn
# of infected domain clusters [57]	Proxy	per day	DNS	DNS cluster		

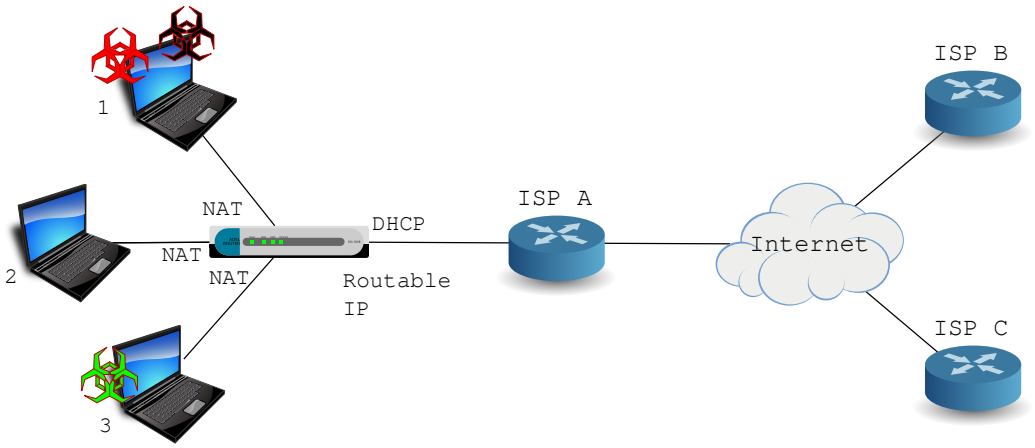


Figure 13: Relationship between ISPs, botnet and home users

# of bots per timezone [58]	IP	per day	sinkhole	bots per continent	total number of bots	number of syn connections by botnet sent per continent
# of unq ip per spam campaign [59]	IP	per hour	spam emails	Countries, ISPs		Top-20 Countries with the Most Bot IPs, Top-20 ISPs that Host the Most Bot IPs
# of unq suspected bots [60]	IP	per day	sinkhole	flows (src ip,dest ip,src port,dest port)		

Table 4: Summary of Current Botnet Metrics

3.4 Issues with Current Botnet Metrics

In this section, we present in more detail the main problems with current botnet metrics. In the next chapter, we can then explore innovative methods and approaches to overcome them, allowing us to produce more reliable metrics.

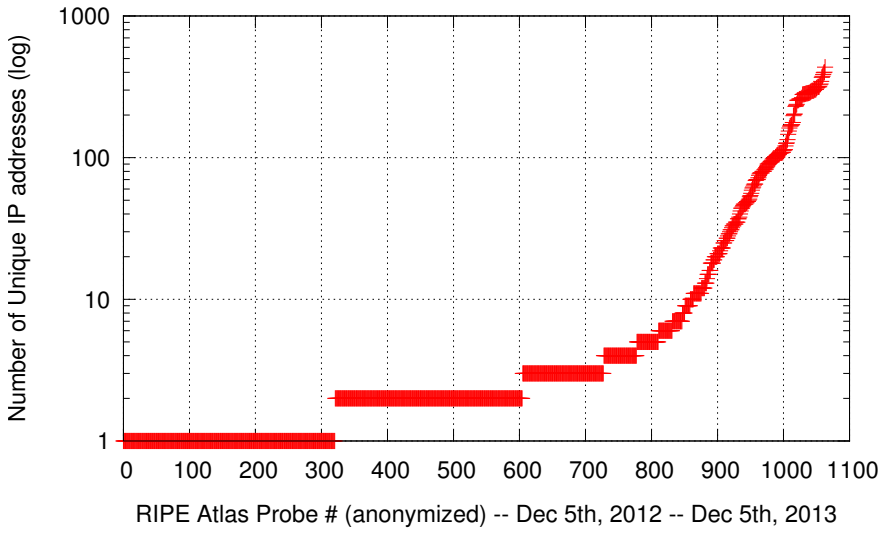


Figure 14: Number of unique IP addresses per RIPE probe

3.4.1 IP-based metrics

Currently, most of botnet metrics are build upon counts of compromised IP addresses, including all the metrics of the type "IP" in Table 4. Typically, all the IP type metrics void several of our requirements: consistency, tracking, predictability, reliability, among others.

The reasons for that is due to DHCP and NAT effects. To illustrate this, consider Figure 13. In this Figure, we see that a subscriber of ISP A is using a home router (with DHCP and NAT) to connect three laptops to the Internet. Laptop 1 has two malware instances running, while laptop 3 has one and laptop 2 has none. There are three bots which are operating from two different laptops and are hiding behind a single public routable IP address.

This exemplifies how complex it is to count botnet presence in ISP networks, and how IP addresses do not correspond to the number of botted computers. To show how the number of IP addresses may significantly differ from the actual number of hosts, we have analyzed the variation on the number of IP addresses of 1,064 RIPE Atlas probes [61][12], over 1 year period. As can be seen, there is a significant variation among the probes and, on average, each probe had 24 IP addresses (1:24). In another study, Stone-Gross *et al.* [12] hijacked the Torpig botnet for 10 days, and found that on average, each bot had (1:7) IP addresses, varying significantly according to ISP and country.

In the next chapter, we present an active measurement approach to estimate DHCP churn in different ISP networks and subnetworks.

[12]Atlas probes are small hardware devices distributed all over the world and used to measure Internet connectivity and reachability, developed and maintained by the *Réseaux IP Européens* Network Coordination Centre (RIPE NCC).

3.4.2 Host-based metrics

Host-based metrics are built on data that can reliably identify individual hosts on the Internet, regardless their location and IP address.

Typically, such measurements are very hard to obtained, since it typically requires some form of access to the hosts themselves. This requires either highly intrusive probing from the network, much more intrusive than would be deemed acceptable according to current practices, or the owner of the machine to run software that produces this data. One of example of the latter is a metric generated by Microsoft, based on telemetry of more than 600 million end user machines that have opted in to run an anti-malware tool.

Microsoft Security Intelligence Reports include a metric for infection rates called ”Computers Cleaned per Mille” (CCM). This counts from how many hosts malware was removed for every 1,000 unique hosts executing the Malicious Software Removal Tool (MSRT), a free tool for malware removal that is distributed as part of Microsoft Automatic Updates. Similarly, Trend Micro publishes a quarterly report where they report counts of the number of unique malware infections reported for mobile, desktop/laptop computers and point-of-sale systems. In each of the categories they also rank the malware relative to each other.

Even though these metrics tend to more precise than IP-based ones (fulfilling the requirements specified on IEEE 1061), the main issue with them is that access to them is either restricted – meaning ACDC parters are not able to obtain it – or they are presented to the public in aggregated levels – e.g., country or AS level.

3.4.3 Proxy-based metrics

The main issue with proxy-based metrics is that they basically express estimates, rather than direct data, on the number of infected machines. The volume of spam or DDoS traffic leaving a network has been shown to be correlated to the number of infected machines in that network, but it can never be very precise. Many factors can influence these measurements. While the correlations might be useful at the aggregate level, we cannot use it to compare individual ISPs. For example, a small ISP is a highly-connected country may generate a larger DDoS attack than a bigger provider in a country with less connectivity, even if the latter has many more infected computers.

Therefore, proxy-metrics should only used for purposes that fit with their shortcomings. Within ACDC, they might be helpful to triangulate other measuruments, but they cannot provide a precise picture of infection rates.

3.5 Summary

In this section we have covered the state-of-the-art of botnet metric and presented background information on metrics in the disciplines of networking and software engineering. We have also presented the requirement for botnet metrics and discussed how the current metrics (in groups) fail to address those requirements.

In the next chapter, we present a methodology and an experiment to deal with the issue of DHCP/NAT of IP-based metrics. The next (and final) version of this deliverable will cover the other issues with other metrics as well.

4 Dealing with DHCP Churn and NAT Effects

In this chapter we focus on a critical problem with current IP-type botnet metrics: the impact of DHCP and NAT effects on counting infected machines. Other problems, as covered in Section 3.4, are dealt with in the specification of the metrics.

4.1 Introduction

There have been a number of measurement studies into the usage of the IPv4 addressing space, mostly focusing on the degree in which allocated address space is actually used [62, 63], on quantifying statically versus dynamically managed address space [64] and, to a lesser extent, on the duration of use of addresses [65]. Relatively little work has been done on measuring the relationship between addresses and hosts [66, 67, 68, 69], especially for large-scale for highly dynamical managed networks of Internet Services Providers (ISPs).

The differences among networks are substantial, and such usage variation within and among ISPs poses a challenge to any research that, in lack of a more precise solution, relies upon IP addresses as a surrogate for the unique identifiers of hosts, as is the case in much research in Internet security. Metrics produced with this surrogate can be dramatically wrong. Take the example of counting the number of compromised computers (bots) [8] in an ISP: it is well-known that the number of IP addresses observed in malicious activity correspond to a completely different number of hosts for different ISP networks [70]. By hijacking the Torpig botnet, for example, Stone-Gross *et al.* [12] were able to observe that infected hosts at German ISPs changed their IP addresses faster than others, on average around 13.4 times over a 10 day period, compared to 2.9 times for American bots and 1.8 times for Dutch bots. So within just 10 days, using IP addresses to count malicious hosts can be off by one order of magnitude. This number of addresses per host over time is commonly referred to as DHCP churn rates, regardless of the underlying technology used to assign the IP address. In the absence of a reliable way to measure this churn [70], it is very difficult to develop reliable security metrics, as well as other host counts based on IP addresses.

To understand why there is such variation among and within ISPs, we have first to understand the relation between ISPs and IP addresses. To connect customers, ISPs are *allocated* with network prefixes [71] by their respective Regional Internet Registrar (RIR) [72], which in turn, receive those prefixes from the Internet Assigned Numbers Authority (IANA) [73]. ISPs then advertise their prefixes to other ISPs usually employing the Border Gateway Protocol (BGP) [74]. ISPs and organizations that provide connectivity to hosts are free to decide how/what prefixes should be assigned to end users in light of their business requirements. Using a variety of technologies (e.g. DHCP [75], RADIUS [76], BRAS), IP addresses are assigned (statically or dynamically) to hosts.

Each of these technologies, in turn, may be configured with a different set of parameters, such as the size of the IP address pools and lease time (the time an address is assigned to a client) [77]. In an ISP, we expect sub-prefixes to exhibit different usage patterns from one another [68]: prefixes assigned to business customers is likely to

differ from a wireless hot-spots block or a home DSL block in terms of session times duration and prefix usage, which are more likely to be managed at /24 [64]. We also expect the usage patterns to vary when comparing different ISPs.

Due to the large number of ISPs and privacy/security issues, currently there is no authoritative way to measure DHCP churn across multiple ISPs/countries/jurisdictions. In this paper, therefore, we present an scalable active measurement-based methodology based on [65] and employ it to measure the dynamics of all prefixes of several Autonomous Systems (ASes). Even though session durations have been previously measured for random prefixes [65], it is unclear to which degree such active measurement-based methods are capable to capture the dynamics of all addresses allocated to different ISPs/ASes. We assess the precision of our methodology by comparing our measurements against ground truth data from an 1 million IP addresses ISP.

We make the following contributions: (*i*) we present and assess the precision a scalable measurement methodology to measure session times of all active IPs within an AS (Section 4.3); (*ii*) we then apply the methodology to ASes of four large ISPs and show how their IP usage/visibility, session duration and inactivity varies, and how this results can be used to profile /24 level policies based on their IP address usage (Section 4.4). We then (*iii*) develop a statistical model (Section 4.5) to estimate the number of different users behind IP addresses (DHCP churn rates) over the monitoring period and validate it.

4.2 Measurement Methodology

We first provide in this section background information on IP addresses assignment and then present our methodology.

4.2.1 Background: IP Address Assignment

Various technologies can be deployed by ISPs to assign IP addresses to hosts. These include (i) Dynamic Host Configuration Protocol (DHCP) [75] servers, (ii) Remote Authentication Dial In User Service (RADIUS) servers [76], (iii) IP pools managed by Broadband Remote Access Servers (BRAS), among others. The latter are often deployed with Point-to-Point Protocol (PPP) [78]. Due to space constrains, we do not delve into the specifics of these protocols, and instead look at IP assignment in generic terms.

ISPs assign either static or dynamic IP addresses to clients with a certain *lease time*, which is the time interval that an IP addresses is assigned, which can also be extended – a DHCP client may issue a request message to extend the duration of the lease time, which typically happens after half of lease has been expired. Moreover, determining the most appropriate lease time for is far from being an obvious tasks: short leases times lead to high volume of broadcast traffic, while long lease times can lead to exhaustion on the address pool space [68].

Figure 15 shows an example: a device has been assigned with the address `2.2.2.2` at T_s, having L_1 as initial lease with default lease time (L_1). These lease

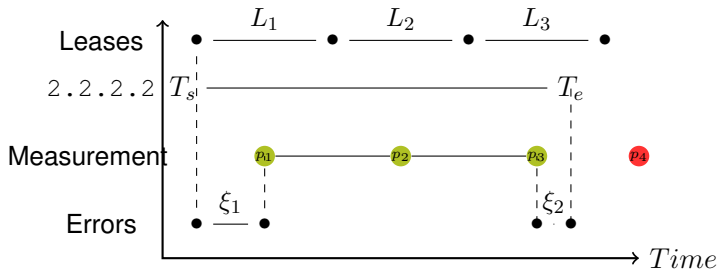


Figure 15: Session Duration and Errors

was then renewed twice (L_2, L_3). The device disconnects at T_e. The *session duration* of a device in a network is a function of the default lease time ($Lease$, in time units):

$$Session(Lease) = n \times Lease - (Ln_{end} - T_e) \tag{22}$$

in which n is equal the number of leases ($n \in \aleph | n \in [0...\infty]$) and Ln_{end} is the time in which the last lease ends (these might differ since DHCP, for example, does not mandate a client to inform a server when it disconnects [75], therefore active leases might be allocated to offline devices). Ultimately, the session duration of a device therefore is not only influenced by the way IP pools and leases are configured, but also by human behavior (e.g., users deciding when to connect or disconnect from the network) as well as external-factors, such as network failures and power outages [79].

4.2.2 Method and Metrics

Figure 15 summarizes the relationship between our probing method and the measurement session duration of an IP address. A random device uses the IP `2.2.2.2` for the time interval $T_e - T_s$. To actively estimate this session, we send four probes (p_n). In this example, three probes were successfully replied, indicating the device was active and reachable, while p_4 did not succeed, to which we assume the device disconnected from the network.

We define *measured session duration* of an IP address to a device as the interpolation of the timestamps of continuously acknowledged probes (ACK). For `2.2.2.2` this is the time difference between the timestamps of the ACK messages of p_3 and p_1 (we disregard the time interval between the time the packet is sent and received). The measured session is an approximation of the *actual session duration*, which, for the same figure, is $Te - Ts$.

Whenever a hosts disconnects, its former IP address might be reassigned to a different user. As a consequence, an IP address may have multiple users over a measured period of time. In our method, we also calculate the *number of sessions* each IP has been assigned. Finally, based on the same method, we can also calculate for each IP the *time in between sessions*, i.e., how long it takes for an IP address to be re-assigned.

4.2.3 Probe Design and Measurement Setup

The two main requirements for our probing design is to be *ISP-independent* and *scalable*. It must enable probing entire ISP's address ranges in a short time frame, without requiring a large amount of computer resources. In this sense, we employ active probing as a measurement technique to be ISP-independent and employ `ZMap` [80], a high-performance network scanner to achieve scalability. Additionally, the design should minimize traffic footprint and respect user's privacy, i.e., collect the minimum information necessary about the probed systems. Next we present our choices for the probing design:

Measurement Protocol: Several protocols can be used to probe the state of an IP address (probes p_n in Figure 15). We chose to use ICMP [81] echo request/reply messages (types 8 and 1) over TCP and UDP since ICMP has proved to be less firewalled, generated less abuse messages (and usually considered "benign traffic"), and be more accurate than TPC and UDP [65, 82]. ICMP also generates a smaller traffic footprint, and better respects user's privacy, since no information other than system status is obtained.

Number of Probes: As discussed in [79], "one ping is not enough". Whenever an ICMP packet reaches a router that does not know the MAC address of the destination, the ARP RFC [83] states that the router should drop the packet and then, send a ARP request instead, impacting our results. Therefore, we choose to send two probes instead per IP per measurement. More probes could possibly lead to more accurate results, at expenses of increased traffic footprint.

Measurement Tool: Standard Unix measurement tools, such as `nmap` [84], `ping`, and `hping3` can be used as probing tool in our design. However, none of these aforementioned tools is designed with scalability as a main requirement. Therefore, we employ `ZMap` [80], an open-source network scanner that enable to scan the entire IPv4 address space within one hour time. Besides being more scalable, `ZMap` outperforms `nmap` in accuracy, since it has a higher connection timeout when waiting for echo reply messages. In our measurements, we could easily probe more than 400K IP addresses per second, using one single computer.

Frequency of Measurements: The frequency of the measurements play a crucial role in network measurements. One common sampling scheme is to send the probe packets separated by a fixed sampling interval. However, using a uniform sampling interval the probes might not capture the true system behavior. Due to DCHP polices and user behavior, there is a possibility that periodic samples may be synchronized with a periodicity in the system under observation. Moreover, commonly used uniform sampling misses high-frequency components and causes aliasing in low-frequency components. Some sampling problems can occur where the samples and system periodicities are not synchronized.

Random sampling is an important step towards more accurate network measurements [85]. It has long been recognized that one way to overcome aliasing in sampling is to sample at random intervals rather than at uniform intervals. Therefore, our approach is based on random additive sampling: samples are separated by independent, randomly generated intervals that have a common statistical distribution $G(t)$. $G(t)$ is

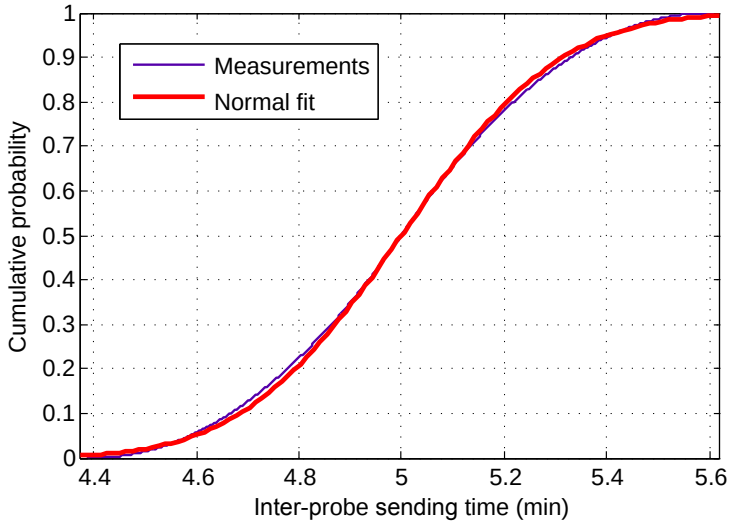


Figure 16: CDF of the inter-probes sending interval

defined by ZMap randomization algorithm. ZMap selects addresses using a random permutation of a cyclic multiplicative group of integers modulo a primer and generating a new primitive root (generator) for each scan. To verify this, we have carried 144 measurements (1 every 5 minutes) over 1 million IP addresses and analyzed $G(t)$, and obtained the timestamps from the outgoing IP packets from the pcap files.

Figure 16 shows the empirical cumulative distribution function of $G(t)$ compared to a normal distribution. As expected, ZMap sends the probes randomly according to a normal distribution with mean equal to the sending interval, i.e., $\mathcal{N}(5, 0.06)$. Thus, by using ZMap we achieve a Non-Uniform Probabilistic sampling strategy avoiding phase-lock problems while being non-intrusive [85].

We have determined empirically the most suitable interval in between each measurements in Section 4.3.1. Thus, we run the scans every 10 minutes with an average inter-probes sending interval equal to $\overline{G(t)}$.

It is important to emphasize the difference between our work and [65]. Contrarily to theirs, our probe selection method is random and does not have bias to active portions of the address space. Moreover, we probe entire IP address spaces of ISPs, while they use sampling instead (24K /24 prefixes, 9,200 probes/s), at more than 400K probes/s. In addition, their interval between measurements is 11 minutes while we employ 10 minutes, and we employ an open-source tool (Zmap) as measurement tool.

Measurement Setup: Our probing setup was configured in a Ubuntu 12.04 Server edition, in a Kernel-based Virtual Machine (KVM), with 6 3.3GHz Xeon cores and 8GB of RAM. The measurements were originated from the network of our university of Delft University of Technology (TU Delft, AS 1128), which has SURFNet (AS 1103) as upstream provider.

In this setup, the most demanded resource is CPU power – our average network throughput was $\sim$ 25Mpbs on a 1Gbps line. We found that the version of ZMap we used did not guarantee packets transmission[13], and had to run it with three threads only to

[13] See https://github.com/ZMap/ZMap/issues/136

avoid packets being dropped on our side. We probed and logged the IP address and the timestamp at of the corresponding ICMP echo response (`SRC_IP, timestamp`).

4.2.4 Limitations

There are several limitations in this method that ultimately impact the precision of the results:

Visible IP Addresses: As discussed in [65], any active probing method can only account for the "visible" part of pool of probed addresses. Many online hosts are expected to be located behind network/application firewalls, network address translators (NAT) which may block all probes destined to a certain network. Moreover, when not behind network firewalls, hosts/customer-premises equipment (CPE) may have their own firewall, and block probes.

Transient Errors and α threshold: Packet losses due to network failures (e.g., poor wireless links), limiting-rate network firewalls, intrusion detection and preventions (IDPS) systems, may also lead to incorrect measurements. In Figure 15, if probe p_2 would have been lost, the ACK message related to p_2 would not be received, and therefore there would be two sessions, $p1 - p_1$ and $p_3 - p_3$, instead of $p_3 - p_1$. To cope with error incurred by transient failures, we introduce a tolerance threshold α. This threshold defines how much longer (in seconds) the algorithm should wait before considering a host offline whenever a probe is not acknowledged. By definition, the algorithm waits for the fixed period of measurements ($1/f$). We added α seconds to this period in order to cope with such errors.

Sampling Errors: Our measurements are subject to random sampling errors. Uncertainties associated with the divergence due to sampling errors are generally small compared to the average measured magnitude. For example, measurement may start after a session has been initiated on the DHCP server, and therefore, not measure it ($m_1 - T_s$) in Fig. 15. Similarly, it may miss the ending of a session, which leads to other errors - ($T_e - m_3$) and ($T_e - m_3$) and ($m_4 - \xi_2$). To mitigate such errors, we weight each timestamp with a uniform distribution of mean $1/f$ as in [86].

4.3 Validation

Any active measurement method requires its precision to be verified. In our case, it requires us to rely on sources that have ground truth data on the session durations. We collaborated with ISP, a mid-size ISP with approximately one million IP addresses, for this research. ISP is the largest privately held broadband service provider in Iran providing a range of services, mostly based on DSL technologies [87]. We carried out the measurements and provided the ISP staff with the results from `ZMap`; they then compared this against their customer IP log files, and provided us aggregated information on the results. For privacy reasons, we were not given access to the session logs directly, and data processing was performed at the ISP.

ISP DHCP logs - Sessions Duration (h)

	m-0	m-600	%-m0	%-m600
5min	29,560,569.95	33,071,437.91	58.58%	65.54%
10min	29,248,506.23	31,594,275.51	57.97%	62.61%
20min	28,630,164.97	28,630,164.97	56.74%	56.74%
30min	28,233,964.92	28,233,964.92	55.95%	62.95%

ISP DHCP - Sessions

	m-0	m-600	%-m0	%-m600
5min	26,536,848	41,899,400	226.62%	357.29%
10min	15,192,248	8,143,872	129,55%	69.44%
20min	9,324,855	9,324,855	79.51%	79.51%
30min	7,179,625	7,179,62	61.22%	61.22%

Table 5: Results of Interval in Between Measurements

4.3.1 Interval in between Measurements

The measurements described in this section all leverage the fact that probes are intrusive. To determine the interval in between the measurements, we first obtained the BGP prefixes announced by ISP from RIPE Routeviews [88], which have corresponded to 1,081,344 unique IPv4 addresses. After that, we have probed twice every IP allocated to ISP every 5 minutes, for one week (May 22–28, 2014), using the methodology described in Section 4.2. After processing all `ZMap` output files, we generated a file that reconstructs the DHCP sessions of the ISP users (m-0). We repeated the procedure adding an offset α of 600s (threshold parameter, (m-600), Section 4.2.4). We choose this value in order to cope with possible transient/network errors.

Then, we vary the probing rate and observe how the accuracy of the results changes. To that end, we process `ZMap` output files for 10, 20, and 30 minutes probing interval. We compare the accuracy of the results for each probing interval with the DHCP log files (ground truth, % w.r.t to it), as can be seen in Table 5. On the one hand, short probing intervals (i.e., < 5m) lead to underestimate the number of session ($\sim$35%). On the other hand, larger intervals ($>$10m) increase only marginally the precision in terms of session hours. Thus, there are trade-offs in among the measurement accuracy, the probe rate, and the overhead on the network. Increasing the probe rate beyond 10 minutes might lead to the situation that the probes themselves skew the results. We therefore choose 10 minutes intervals and use it the remainder of this paper.

4.3.2 Address and Prefix Visibility and Usage

After determining the 10 minutes time in between measurements, we employ a measurement dataset that we have generated for 2 continuous weeks (March 22nd – April 5th, 2014). That lead to a file having a total 14,533,525 measured sessions (m-0), and m-600 with 8,215,301 measured sessions (m-600).

Before evaluating the precision of our method, we first need to determine whether our ICMP-based method is able to obtain response from a significant part of addresses

ISP Session Logs

	# IP addresses
Measurements	714,139
Session Logs	752,098
Measurements ∩ Logs	709,586 (94.95%)
Only Session Logs	42,510
Only Measurement	4,551

Table 6: Validation Datasets

allocated. We have sent 4,641,128,448 probes (two probes per IP, per measurement), to which 356,805,959 IPs (non-unique, total) responded. In average, 166,266 of the $\sim$1M IP addresses responded per measurement, which shows that the ISP, at any given time, has in use 15% of its pool – which was confirmed by the ISP, providing an insight on how the ISP (re)use its pool of IP addresses.

Table 6 shows the number of unique IP addresses observed on the measurement and on the ground truth. As can be seen, our method was capable to obtain response from 94.95% of the addresses employed by ISP (the ratio of intersecting IPs between measurement and DHCP logs) during the measured period. The remainder IPs in the DHCP log files of ISP (42,510) did not respond either because of firewalls or because our probes might have missed those IPs due to sampling rate.

Interestingly, there were 4,551 IP addresses only found in our measurements: those are assigned to devices such as routers and servers that do not have their IP addresses recorded in the ground truth we employed in this paper. In addition, part of these addresses were allocated to business customers, which, in turn, maintain their own independent DHCP servers, therefore not included in the ground truth.

Figure 17 shows the time series of the number of unique IP addresses, in which each point represents the total number of unique IP addresses observed until that measurement. As can be seen, there is an asymptotic curve towards the number of used IP addresses over time. The 1st derivative at a point P determines the ratio of new IP addresses being used at the time P took place. This supports our probing method, in which we probe entire IP address spaces, instead of sampling certain portions of IP addresses as in previous works, which might lead to incorrect results with regards the usage of the address pool space.

4.3.3 Session Duration Distribution

We processed the output file generated by our software at the ISP, which aggregates information per IP addresses in the format of the tuple (`SRC_IP, NSessions(Measur.), NSessions(DHCP), SumDuration(Measur.), SumDuration(DHPC)`), for $\alpha = 0$ and $\alpha = 600$ (m-0 and m-600, respectively). This tuple provides us enough information for validation while keeping privacy of users.

Table 7 summarizes the results. In the first line of the table ($\sum$ **all**), we show the sum of the duration of all sessions for the measurements and ground truth. Then, in the second line $\sum \cap_{IPs}$, we show what portion of these measured hours overlap in time with

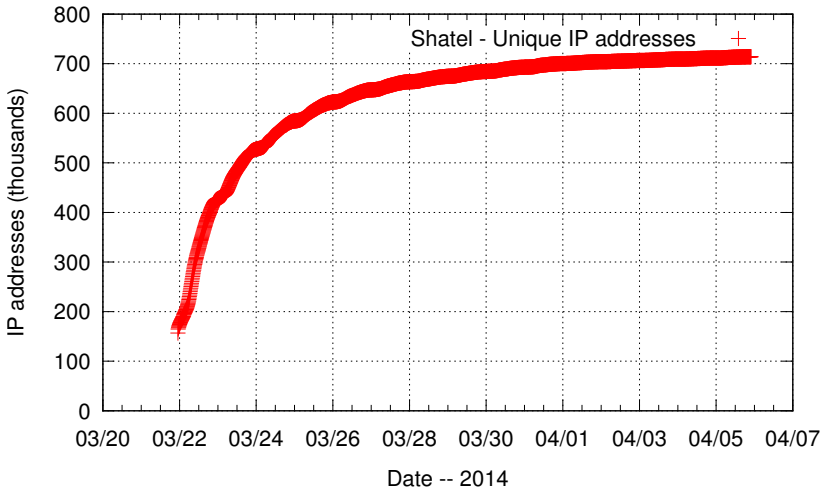


Figure 17: Time Series of Unique IP addresses

ISP DHCP logs - Sessions Duration (h)

	m-0	**m-600**	**DHCP**
$\sum$ **all**	59,676,305.56	60,733,610.81	96,899,976.59
$\sum \cap_{IPs}$	59,336,733.11	59,374,111.53	90,874,619.90
Ratio	65.29%	65.33%	
Error	0.29	0.28	

DHCP - Sessions

	m-0	**m-600**	**DHCP**
$\sum$ **all**	14,533,525	8,215,301	19,877,570
$\sum \cap_{IPs}$	14,432,133	8,182,572	18,498,448
Ratio	78.01%	44.23%	
Error	0.42	0.50	

Table 7: Validation Results

the measured hours from the session log files – that is, that captured correctly online time intervals of the intersecting IP addresses. This is shown in the Ratio line, which is obtained by dividing $\sum \cap_{IPs}$ of the measurements by the $\sum \cap_{IPs}$. As can be seen, for both measurement files (m-0 and m-600), our method was able to account for $\sim$ 65% of the online time of all the observed IP addresses. It is important to highlight the meaning of these findings: by only sending frequent ICMP messages, we were able to infer correctly 65% of all of the sessions duration observed at ISP. Due to our sampling rate, we technically miss all sessions that start and end in between two consecutive time intervals. Increasing the frequency would lead to better results, however at the price of increasing traffic footprint.

Another finding is that the threshold parameter α only slightly improves the accuracy of the session durations. To understand why, we further analyze the number of sessions by comparing m-0 to m-600. We can see that m-600 in fact reduced the total number of measured sessions, by merging two distinct sessions. This, in turn, has led to 37,378.42 more hours being correctly estimated, which is a small fraction of the total

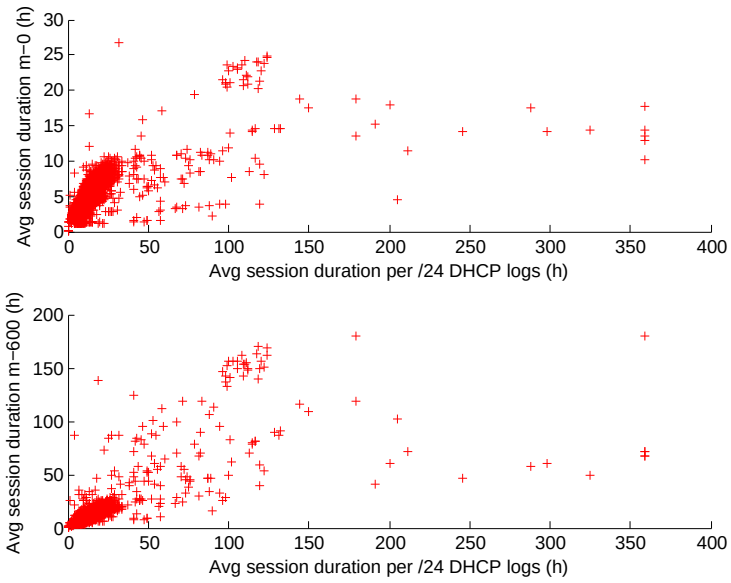


Figure 18: Scatter plot of DHCP logs vs. m-0/m-600

hours ($\sum$ all) at cost of a higher error in the number of sessions.

However, when comparing the average session duration per IP (`SumDuration(Measur.)/NSessions(Measur.)`), m-600 outperforms m-0 in estimating the average session time on IP addresses, as can be seen in Figures 18 and 19. Fig. 19 shows the histogram of these results: m-600 follows closer the shape of the ground truth ($R^2 = 0.69$), and is capable to estimate average sessions if from IPs having long average duration sessions. m-0, on the other hand, is sensitive to any packet loss, and estimates a larger number of very short average sessions ($R^2 = 0.50$). Comparing Fig.19(b) to Fig.19(a), we can see that m-0 performs poorly in estimating the correct number of sessions with duration inferior to 50 hours, which is explained by the fact that those IPs did not respond to the probes while they were actually online. The reasons for that are hard to pinpoint, but include either real-time firewall/IDPS, probe loss, transient error, graylisting, as discussed in Section 4.2.4.

4.4 Analyzing Larger ISPs

Having assessed the precision of our methodology, in this section we apply it to ASes of four major ISPs from different countries: AT&T, British Telecom, Deutsche Telekom, and Orange, as can be seen in Table 8. Using the same setup described in Section 4.2.3 we probed the ISPs for 17 continuous days (March 13th – March 29th, 2014), which amount to a total of 134 million IPv4 addresses.

Before starting the measurements, however, we met with the Security Incident Response team of our university and coordinated how the measurement would be arranged. First, we ran in the same measurement VM the `micro-httpd` web server with a web page describing the project goal, our credentials, and how users could opt-out of our measurements. We have received a total of 35 e-mails requesting IP addresses

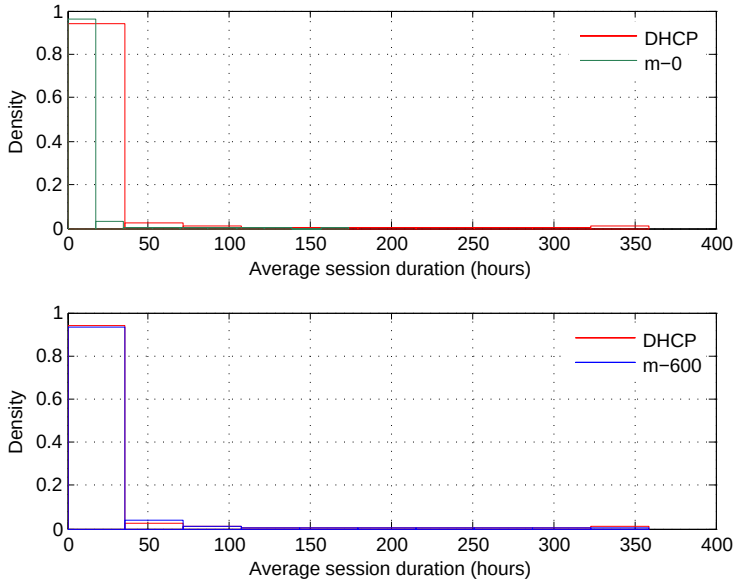


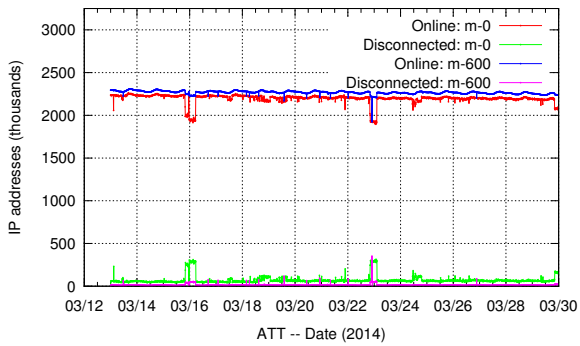
Figure 19: Histogram of Average Session Duration per IP for ISP

to be removed from our measurements, which we did immediately upon received the messages. All the complaints we received were from system administrators in small-businesses and few tech savvy home users. In only one instance one user wrongly thought we were carrying out a denial-of-service attack (DoS) on his server, which was not the case and we have confirmed him once he shared with us his intrusion detection system (IDS) log files, which showed we sent only 2 ICMP echo-requests per IP per measurements. In general the users were understanding and supportive; they only requested few of their IPs to be removed from our measurements.

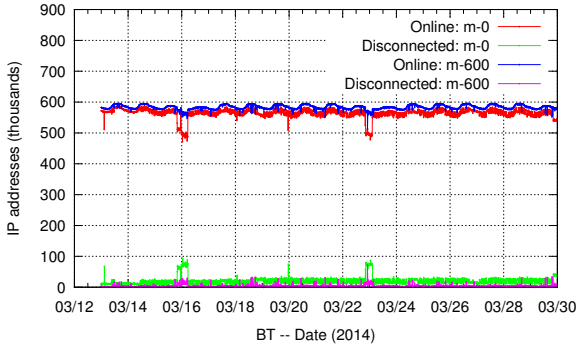
4.4.1 Address and Prefix Visibility & Usage

Using our measurements, we can determine the lower bound on the address space in use in those ISPs (Section 4.2.4) by determining the percentage of visible IPs with regards the total announced by the ISPs. Table 8 presents the *addresses visibility*, in which only a small fraction of IP addresses are on average reachable via ICMP. However, when considering all IP addresses that have responded over the measurement period, the percentage increases - in the case of Deutsche Telekom this value goes to 51%, a reflect on how fast ISPs recycle IP addresses.

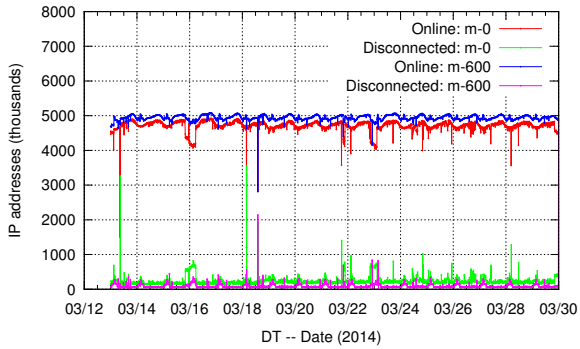
Figure 20 shows the time series of the IP addresses that are online (active) and that become offline (disconnected) for the ISPs discussed in Section 4.4. For each ISP, we also include the time series when using α threshold parameter as 600 seconds. We can observe a diurnal pattern in all the time series, i.e., the peaks in the plots represent late mornings to early afternoons while the falls correspond to nights. Due to few transient network outages during the measured period, the time series suffered from two sudden drops in the number of active IPs. This effect is diminished when using the α threshold. As expected, the m-600 reduces the number of disconnecting IPs.



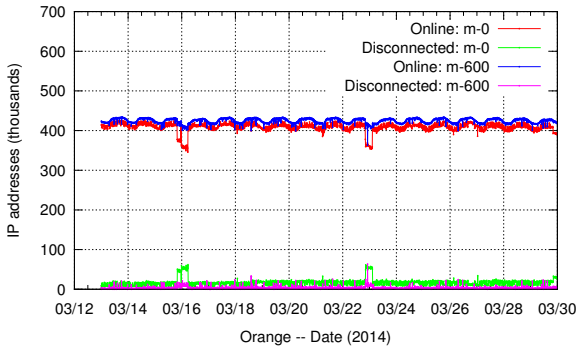
(a) AT&T



(b) British Telecom



(c) Deutsche Telekom



(d) Orange

Figure 20: Time Series of online IPs per ISP

Servers that assign IP addresses are configured using a default lease time for multiple prefixes. Figure 21(a) shows the number of visible/active IP addresses per /24 prefix for the evaluated ISPs. In this Figure, we can see how Deutsche Telekom, faster than the other ISPs, quickly re-uses its IP addresses, while most of prefixes from AT&T present low-visibility/usage.

4.4.2 Session Duration Distribution

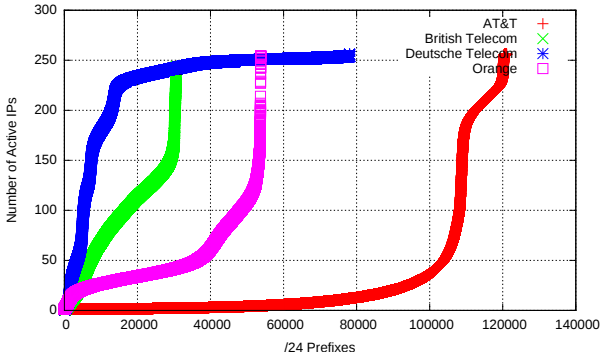
Table 9 shows the results of session duration for each ISP. We have shown in Section 4.3.3 that $\alpha = 600$ lead to more precise results than $\alpha = 0$. Analyzing Tab. 9, we can see all m-600 files were able to generate at least one session duration equal to the monitoring period (17 days $\sim$ 408 h). Since some IP addresses in these ISPs are expected to belong to network devices/servers with high availability, we can expect that m-600 tend, as in Section 4.3.3, to be more precise with regards estimating the session durations. However, only a comparison with ground truth could precisely answer this question.

Figure 22 shows the empirical cumulative mass function of average session duration per IP for each ISP for m-600. On average, a bot would have its IP address renewed every 61, 20, 10, and 14 hours for AT&T, British Telecom, Deutsche Telekom, and Orange, respectively, which wind up inflating at different rates the actual number of compromised computers per ISP. We can also observe that most of the "visible" IP addresses have an average lease time of less than 50 hours, and that for AT&T, we see spikes around $t = 75, 100, 140$ hours, which indicates that large portion of the IP addresses are managed by DHCP servers that enforce IP address changes after reaching these session durations. We can see how the average session duration varies according to /24 prefix on Figure 21(b).

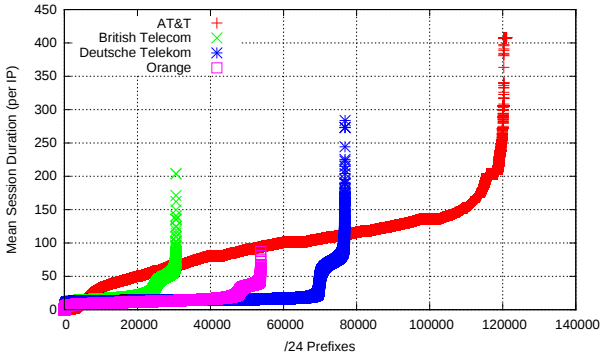
This information can be also used to profile prefixes i.e., classify them according to what they are used. Figure 21(c) shows the average inactivity time per /24. For all ISPs, the majority of IPs for these prefixes have a low average inactivity time ($< 25h$), while some have higher values. Prefixes with short lease times and large number of users are likely to have a low inactivity time; while prefixes with long lease time and long inactivity time are more likely to belong to ranges from static IP addresses.

4.5 Towards Churn Rates

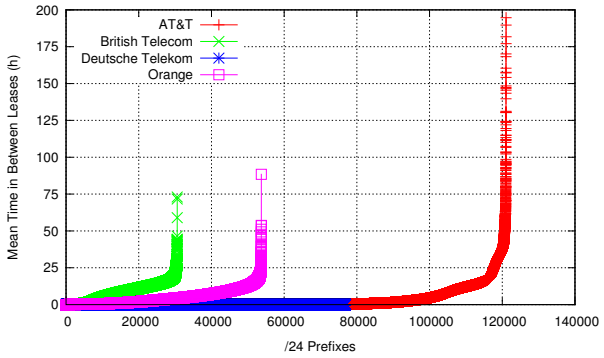
Estimating churn rates would allow us to normalize the bot count in the networks of ISPs. By using our methodology, we were able to account for 78% and 44% of the observed sessions for our ground truth datasets. If each new session would be associated to a new user, counting the number of sessions an IP address has exhibit would yield to the number of distinct users that IP has been assigned too. However, a same user might be re-assigned to the same IP multiple times over the measured period. To cope with that, one could employ device fingerprinting [89], but this approach requires a large number of packets to be sent in order to measure clock skews, which is hard to scale to when probing entire ISPs address pools, not to mention the privacy implications.



(a) Maximum Number of Visible IPs



(b) Mean Session Duration



(c) Mean Time In Between Sessions

Figure 21: Distribution of Prefixes

We envision an approach to statistically estimate the churn rates of IP address. Previous works either used passive data to estimate this churn rate [90] or described complex stochastic models that are not able to capture the whole nature of the dynamic allocation of addresses [91]. However, none of these models was able to establish a methodology valid for the whole Internet. Contrarily, our methodology is scalable and valid for any network.

Our churn estimator is based on the activity rate of an IP address to approximate the number of users behind it. Instead of analyzing IPs individually, we group them

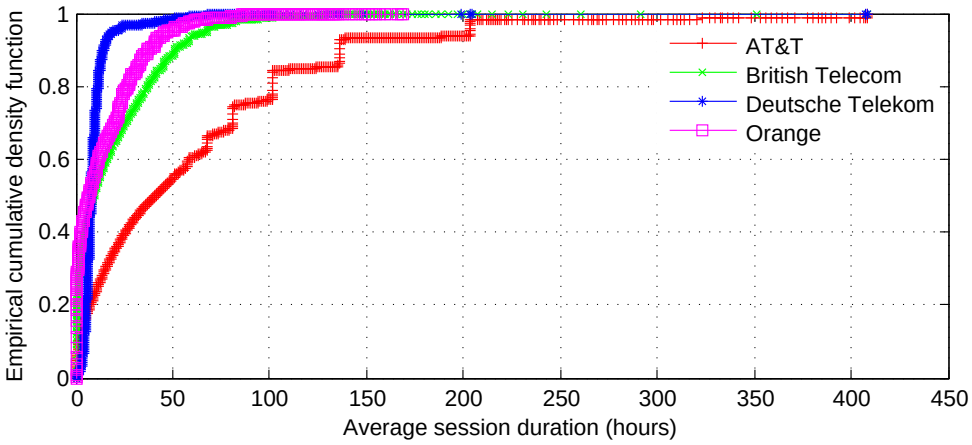


Figure 22: ECDF: mean session duration/IP (m-600)

according to the Network Access Server (NAS) the users connects to. DHCP/RADIUS servers are commonly configured with multiple /prefixes in a NAS group which leads to similar behavior (See Fig. 21). For any session, as the churn rate is verified to follow a Poisson distribution [90], then from the properties of the distribution, the number of IPs per user can be estimated based on its rate. Consider a Poisson process $\{\mathcal{A}(t)\}_{t\geq 0}$ that counts the number of active IPs in intervals of $[0, t]$. Assuming that all users were online at some time $t' < t$, we can use the intensity process λ of the Poisson process $\{\mathcal{A}(t)\}_{t\geq 0}$ as an estimator churn rate, i.e., we estimate the number of new added IPs per measurement as the churn rate.

Figure 23 shows the mean number of IPs per user per day for those NAS groups that are visible in the same ISP used in Sect. 4.3. We observe that around 2% of the groups have at most one IP per user per day in average; while the mean number of IPs per user per day is around 5. It is also worth noting more than 60% of the groups have 10 or more IPs per user per day. By measuring the leverage and the Cook's distance we can detect two outliers (NAS-183 and NAS-214). Removing these outliers, the root mean square relative error is equal to 0.27 which confirms the notable accuracy of our estimation. Figure 24 shows the normal probability plot (NPP) of the error of the churn estimation. Despite a short curvature in the NPP, the probability plot seems reasonably straight, meaning an accurate fit to normally distributed residuals. The F-statistic of the linear fit versus the constant model is 4.19, with a p-value of 0.049. Hence the model is significant at the 5% significance level.

4.6 Related Work

To the best of our knowledge, this is the first study that employs high-performance large-scale probing with the goal of estimating session durations and dynamics of IP addresses of entire ISPs. Heidemann *et al.* [65] have analyzed the session time of random 24,000 /24 prefixes. We extend their methodology to reconstruct sessions and validate against a mid-size ISP, and then apply it to the entire IP addresses of four major ASes ($\sim$ 280,000 /24). We show how the distribution of sessions varies within and for

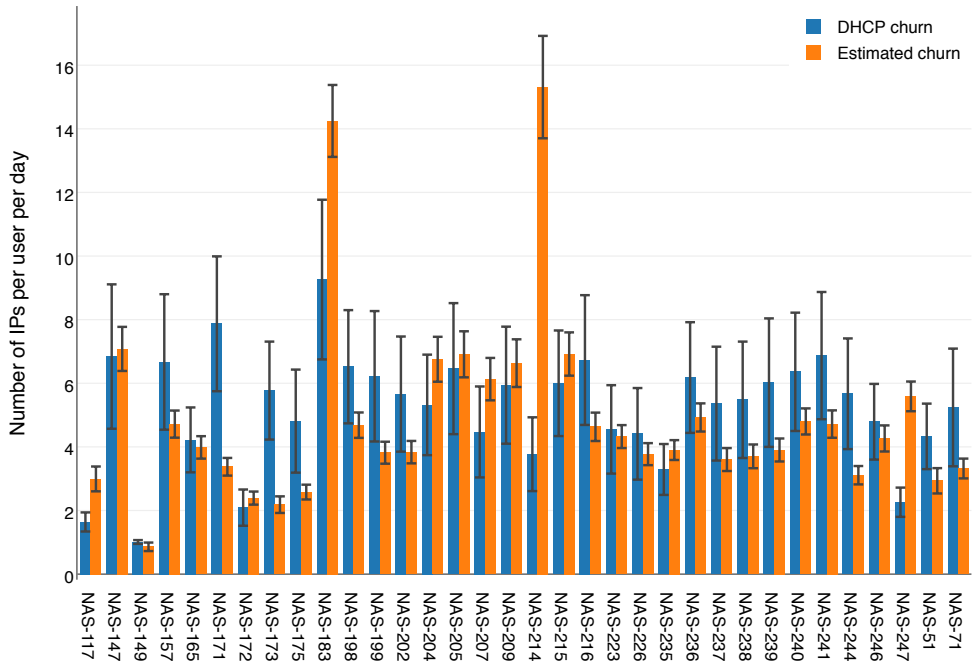


Figure 23: Number of IPs per user per day

different ISPs. Another Internet-wide probing was carried out by anonymous authors in [82]. Since the use of their datasets of is controversial [92] (data was obtained by hacking users' CPEs) and its validity questionable, we therefore refrain from compare our works to their. Schulman *et al.* [79] have employed ICMP-based measurements to detect network failures incurred by the weather.

DHCP leases have been analyzed in previous works [66, 67, 68, 69]. However, they have employed DHCP and http server logs. Brik *et al.*, for example, monitored DHCP servers of University of Wisconsin-Madison for 3 weeks, while Khadilkar *et al.* [67], analyze four days of DHCP logs at George Tech. Papapanagiotou *et al.* [68], have monitored two networks for less than 6 weeks, having less than 6,000 active IP addresses. Finally, Xie *et al.* [69] analyzed the http log files for MSN Hotmail, which also included user login information for one month period. Our method differs with these since it enables session duration estimation independent of an ISP and does not require access to log files, making it scalable to the entire Internet.

It is also important to highlight our measurements do not allow to monitor/fingerprint individual users, since the information we collect (IP address, timestamp) is not enough to single out unique users. Active probing has been used to perform device fingerprint. Kohno *et al.* [89] have employed ICMP and TCP-based active measurements to measure clock-skews of devices, which ultimately may allow fingerprint. However, their method requires a vast number of probes per individual IP to be sent, and it is not easily scalable. Eckersley [93], in turn, develop a method to measure the entropy of a users' browser, based on the parameter automatically provided by the user's browser. However, in this case, is a passive measurement approach, in which users must voluntarily access websites that may fingerprint their browsers.

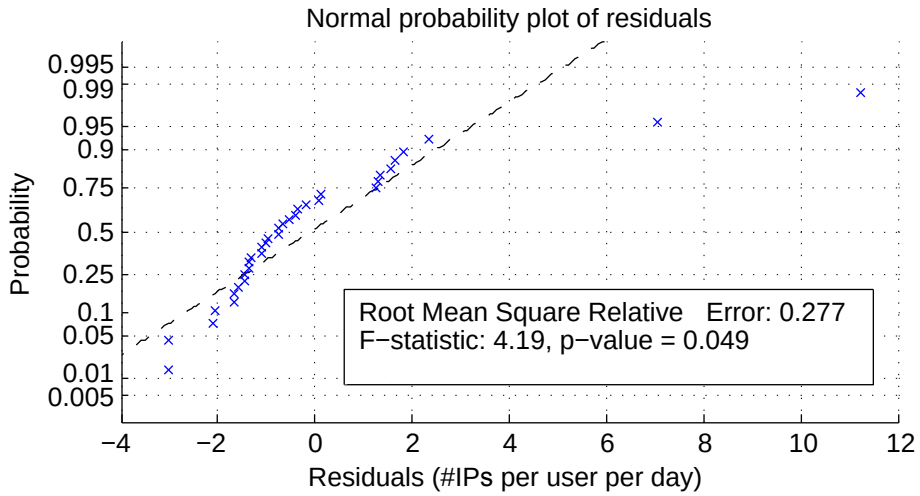


Figure 24: Error estimation

4.7 Conclusions

We have shown in this chapter how, in a scalable and ISP-independent way, we can measure session durations of entire ISPs with precision. Using the same methodology we have profiled and analyzed several ISPs gaining insights about their IP address allocation policies – how often IP addresses remain online, are re-used, and remain inactive. We have shown how these vary for different ISPs.

Moreover, we have developed a simple but rigorous statistical model for estimating the number of users behind active IP addresses. Our validation using DHCP logs from a mid-size ISP proves the accuracy of the methodology. The proposed methodology is generic, and can be applied to a wide spectrum of applications and determine the number of different hosts behind IP addresses – which has a direct application in normalizing bot counts across ISPs.

As future work, we will carry measurements on the networks of the biggest ISPs on the Internet and apply the churn estimation methodology to norrmalize metrics related to several types of abuse traffic including spam, DDoS attacks or ad-click fraud which will lead to better estimation of botnet size. We will explore the possibility to create more sophisticated abuse detection techniques as well as redefine mitigation policies such as IP blacklisting.

AS	CC	ISP	IPv4	Visible (total)	Visible (mean)	σ	Size
7018	US	AT&T	73,820,672	3,836,880 (5.19%)	2,195,068 (2.97%)	56,437.0	29GB
2856	UK	British Telecom	11,352,576	2,673,034 (23.54%)	563,635 (4.96%)	15,439.3	6.7GB
3320	DE	Deutsche Telekom	34,404,352	17,450,601 (50.72%)	4,705,551 (13.67%)	176,638.3	59GB
3215	FR	Orange	15,273,728	2,680,682 (17.55%)	408,537 (2.67%)	11,374.9	5GB
Total:			**134,851,328**	**26,641,197 (19.75%)**	**7,872,791 (5.83%)**	–	99.7GB

Table 8: Evaluated ISPs – March 13th–26, 2014

	AT&T		British Telecom		Deutsche Telekom		Orange	
	m-0	m-600	m-0	m-600	m-0	m-600	m-0	m-600
Mean	4.854	61.140	3.280	19.451	3.169	9.900	2.769	14.115
Median	5.354	40.725	3.416	9.247	3.113	7.914	2.716	6.816
Max	68.669	408.212	67.195	408.159	27.679	408.158	50.071	408.018
Min	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Std Dev	2.259	68.634	2.496	24.473	0.905	10.249	2.383	17.967
Coeff Var	0.465	1.123	0.761	1.258	0.286	1.035	0.861	1.273
Trimmed Mean (95%)	4.848	55.723	3.150	17.634	3.157	8.751	2.630	12.769

Table 9: Statistics summary session duration in hours.

5 Employed Metrics for Evaluation

We divide this chapter into three main parts: in Section 5.1, we show how data collected from the ACDC experiments should be enriched and anonymized, so we can produce the results for our comparative botnet metrics, which we show in Section 5.2. Then, in Section 5.3, to bridge the work on metrics development to actual evaluation (the next phase of WP4), we carry out an example of an evaluation using our currently available datasets.

5.1 Data collection and enrichment

In this section, we specify in more details the fields required to produce botnet metrics that will be employed in the evaluation of European ISPs. We also show how data should be collected. This also takes into account the legal and privacy concerns involved in WP4 and in the project.

Table 10 shows what fields should be capture by the sensors and what fields should be added to it before being imported from the CCH, as well as the anonimization of the IP address. It is important to emphasize that the enriched fields must be added to each logged event *before* the anonimization of the IP address takes place.

Captured Fields

Field	Header/Layer	Example
Timestamp	1/2	1422352309
IP address	3/IP header	10.10.10.10 – not the one of the sensor
Protocol	3/IP header	0x06 (TCP)
Source Port	4/ TCP header	25 (SMTP)
Destination port	4/TCP header	9292 (client port)
Application	5/depends on the application	botid, in case of botnet protocol

Enriched Fields

Field	Source	Example
AS number	BGP feeds/Maxmind db	AS3320
Organization	Maxmind Org DB [32]/`whois` [94]	Deustche Telekom A.G.
Country	Geolocation DB [32]	Germany
City	Geolocation DB [32]	Munich
Type of Connection	Maxmind Connection DB [32]	DSL
Top-level Domain	Reverse DNS	dtag.de

Anonymized Fields

Field	Source	Example
IP address	Must be prefix preserving	depends on the algorithm

Table 10: Capture Fields and Enriched fields

Finally, each event will then be imported from the CCH in the following format: `<Timestamp,IP address (anonymized), Protocol,Source Port, Destination Port, AS Number,Organization,Country,City,top-level domain, Type Of Connection>`. These fields are as well specified in the CCH

data exhanged format described in Table 1, Section 2.3, that we will use to extract aggregated and non-aggregated data from the CCH.

5.2 Comparable Botnet metrics

In Chapter 3, we have covered extensively the state-of-the art of current botnet metrics, the requirements, and presented a survey of existing metrics. In addition, we have presented in Figure 12 (Section 3.3) a taxonomy for botnet metrics, based on types of data sources available.

In this section, we employ the data fields described in Section 5.1 to produce our botnet metrics. Each metric is composed of a summation part and a normalization part. The summation refers to the sum of unique fields. The normalization part, however, refers to the part in which the summation is normalized in order to make it comparable across different ISPs/countries etc. For example, it is expected that Germany has many more bots than Finland, since Germany's population is many times larger than the Finish. We therefore normalized the bot counts by the number of Internet users in the country, so we they can be compared.

5.2.1 Host-based metrics

Host-based metrics, in our taxonomy, refers to metrics that count infected hosts within networks of ISPS – for example, botnet sinkholes with unique identifiers. Datasets that provide such type of data are harder to obtain: one must first sinkhole or hijack a botnet, and the botnet must have unique identifiers in its protocol, and the protocol must be not encrypted or poorly encrypted.

Table 11 shows the host-metrics we will employ in our evaluation. First of all, these metrics will produced on a daily basis. For each bot id found on a day, we will aggregate them into countries, Autonomous System Numbers (ASN), and ISPs (using TU Delft ISP mappings). To make it comparable across countries, we will normalize it to remove external factors such as size of population from the rates.

#	Metric	Summation	Normalized by	Time Interval
1	Daily botIDs/country-user	`for country in i:` $\sum BotIds \in i$	population× Internet Penetration Rate	Daily
2	Daily botIDs/ASN-IP	`for ASN in i:` $\sum BotIds \in i$	$\sum IPs \in ASN$	Daily
3	Daily botIDs/ISP-subs.	`for ISP in i:` $\sum BotIds \in i$	$\sum Subscribers \in ISP$	Daily

Table 11: Host-based Metrics used in the evaluation

It is important to highlight that the normalization uses variables that change with time – e.g., population of country, subscribers base. Therefore, in this process, we use the timestamp (Table 10) of the event to determine what is the closest values we have for Population, Internet Penetration Rate, Number of IPs announced by an

ASN on daily BGP feeds, and number of subscribers obtained from the Telegeography database [95].

5.2.2 IP-based metrics

Table 12 shows the IP-based metrics we will use in our evaluation. Similarly to the host based-ones, in this one we sum IP addresses instead of bot-ids. While host-based metrics may not be able to be produced (due to the fact botIds are not always possible to be obtained or even may not even exist), IP-based metrics should be always be able to be produced in the comparison.

#	Metric	Summation	Normalized by	Time Interval
4	Daily IP/country-user	`for country in i:` $\sum IP \in i$	population× Internet Penetration Rate	Daily
5	Daily IP/ASN-IP	`for ASN in i:` $\sum IP \in i$	$\sum IPs \in ASN$	Daily
6	Daily IP/ISP-subs.	`for ISP in i:` $\sum IP \in i$	$\sum Subscribers \in ISP$	Daily

Table 12: Host-based Metrics used in the evaluation

5.2.3 Proxy-based metrics

Depending on the type of attack, we can also explore compare the impact of the attack for different ISPs. For example, in the case of spam, one metric which is also important is the number of spam messages each bot has sent, and total number of bots. Table 13 shows the list of metrics we will use.

#	Metric	Summation	Normalized by	Time Interval
7	Daily Events/country-user	`for country in i:` $\sum Events \in i$	population× Internet Penetration Rate	Daily
8	Daily Events/ASN-IP	`for ASN in i:` $\sum Events \in i$	$\sum IPs \in ASN$	Daily
9	Daily Events/ISP-subs.	`for ISP in i:` $\sum Events \in i$	$\sum Subscribers \in ISP$	Daily

Table 13: Proxy-based Metrics used in the evaluation

5.2.4 Normalization by DHCP churn rates

In Chapter 4, we have a method to measure DHCP churn rates for entire ISPs. We are currently working on producing a model that employs these measurements and profiling using domain names associated to IP addresses to determine, per /24 prefix, what is the average churn rate. Therefore, after that, we will be able to normalize each metric by how often their IP addresses change per ISP.

5.3 Performance Evaluation

In this section, we employ a series of datasets to show how part of metrics we propose can be used to evaluate the performance of ISPs. Even though this Deliverable D4.1 does not require any performance evaluation results, we include it to demonstrate the functionality. However, in this section, we focus on comparing only countries against each other. First, we cover the employed datasets we employ, then on the measured results. Please notice that this results do not capture the impact of ACDC yet.

5.3.1 Datasets

Shadowserver Sinkhole Conficker data (Conficker)

Established in 2004, the Shadowserver Foundation comprises volunteer security professionals that "gathers intelligence on the darker side of the Internet". They have created the Conficker working group, which provides reports and data on "the widespread infection and propagation of Conficker bots" [96].

Several members of the working group run sinkholes that continuously log the IP addresses of Conficker bots. The sinkholes work in this fashion: computers infected with Conficker frequently attempt to connect to command and control servers to receive new payloads (i.e., instructions). In order to protect the botnet from being shut down, Conficker attempts to connect to different C&C domains every day. The working group has succeeded in registering some of these domain names and logging all connections made to them. Since these domains do not host any content, all these connections are initiated by bots. Therefore, we can reliably identify the IP addresses of the Conficker bots.

The Conficker dataset is unique in several ways. First of all, unlike the other two datasets, it is not a small sample of a much larger population, but rather captures the universe of its kin. This is because of the way the bot works most of them will eventually contact one of the sinkholes. Second, this dataset is basically free from false positives, as, apart from bots, no other machine contacts the sinkholes. These features make the dataset more reliable than the spam or DShield datasets. The difference, however, is that the dataset is only indicative of the patterns applicable to one specific botnet, namely Conficker. Although Conficker has managed to replicate very successfully, with around several million active bots at any given moment, it has not been used for any large-scale malicious purposes or at least no such uses have been detected yet. This means ISPs and other market players may have less powerful incentives to mitigate these infections, different from spam bots, for example. These differences make the Conficker dataset complementary to the two other sets.

Overall, the Conficker dataset adds a fresh, robust and complimentary perspective to our other two datasets and brings more insight into the population of infected machines worldwide.

Zeus Gameover Botnet (Peer and Proxy)

Zeus botnet started making headlines in 2007, as a credential stealing botnet. The first version of Zeus was based on centralized command and control (C&C) servers. The botnet was studied by various security researchers and multiple versions were also tracked [97, 98, 99, 100].

In recent years Zeus has transformed, into more robust and fault tolerant peer-to-peer (P2P) botnet, known as P2P Zues or Gameover. The botnet supports several features including RC4 encryption, multiple peers to communicate stolen information, anti-poising and auto blacklist. It also can be divided into *sub-botnet*, based on BotIDs , where each sub-botnet can be used to carryout diverse task controlled by different botmasters.

The botnet is divided into three sub-layers, which provide following functionality.

- **Zeus P2P Layer (Peer):** This is the bottom most layer and contains information of infected machines. Bots in P2P layer exchange peer list with each other in order to maintain updated information about compromised machines.
- **Zeus Proxy Layer (Proxy) :** A subset of bots from P2P layer are assigned the status of proxy bots. This is done manually by the botmaster by sending proxy announcement message. Proxy bots are used by Peer-to-peer layer bots to fetch new commands and drop stolen information.
- **Domain Generation Algorithm Layer:** DGA layer provides fall backup mechanism, if a bot cannot reach any of its peers, or the bot cannot fetch updates for a week. Zeus algorithm generates 1000 unique domain names per week. Bots which lose connection with all connected peers search trough these domains until they connect to live domain.

  More details about architecture and functioning of the botnet can be found in literature [101, 102].

This dataset is sub-divided into three feeds, GameOver Peer, GameOver Proxy and GameOver DGA. The botnet is spread in around 212 countries with on average 95K unique IP addresses per day. Hence it is gives us insight of botnet infection level at global level, and compare various countries and ISPs.

ZeroAccess

ZeroAccess is a Trojan horse, which uses a rootkit to hide itself on Microsoft Windows Operating Systems. The botnet is used to download more malware and open backdoor for botmaster to carry out various attacks including click fraud and bitcoin mining.

The botnet is propagated and updated through various channels including compromised website redirecting traffic and dropping rootkit at potential host or updating the already compromised host through P2P network.

ZeroAccess also provide global view with bots in around 220 countries with an average of about 12K unique IP addresses per day.

Morto Botnet

Morto is a worm that exploits the Remote Desktop Protocol (RDP) on Windows machines to compromise its victims. It uses a dictionary attack for passwords to connect as Windows Administrator over RDP with vulnerable machines in the network. After successfully finding a vulnerable machine, it executes a dropper and installs the payload.

We have a time series data of Morto for past 4 years with an average of 5k daily unique IP addresses distributed globally. This is relatively small, but it complements our other data sources by providing a longitudinal perspective.

Spam trap dataset (Spam)

Spam data are obtained from a spamtrap we (TU Delft) have access to. It might not be fully representative of overall spamming trends, and also there is no guarantee that the listed spam sources are indeed originating from botnets, though so far that is still the main platform for distribution. The more important limitation is that the spam has become a less important part of the botnet economy, as witnessed in the substantial drop in overall spam level. The reports of security firms seem to confirm these overall trends. Symantec reported a significant decrease in the volume of spam messages, from highs of 6 trillion messages sent per month to just below 1 trillion [103] until 2012. Cisco, TrendMicro and Kaspersky show that the spam volume since that period has been fluctuating, but staying at more or less the same level (see [104] and [105]). All of this means that the source is becoming less representative of overall infection levels. Regardless, we employ this source and compare against the other ones as well.

5.3.2 Mapping offending IP addresses to EU ISPs

For each unique IP address that was logged in one of our data sources, we looked up the Autonomous System Number (ASN) and the country where it was located. The ASN is relevant, because it allows us to identify what entity connects the IP address to the wider Internet – and whether that entity is an ISP or not.

However, there are some ISPs that operate in various countries across Europe. We employ IP-geolocation databases [106] from Maxmind [32] to single out IP addresses used in The Netherlands from the other European countries when classifying the attacking IP addresses from each ISPs.

As both ASN and geoIP information change over time, we used historical records to establish the orgin for the specific moment in time when an IP address was logged in one of our data sources (e.g., the moment when a spam message was received or network attack was detected). This effort resulted in time series for all the variables in the datasets, both at an ASN level and at a country level. The different variables are useful to balance some of the shortcomings of each a point to which we will return in a moment.

We then set out to identify which of the ASNs from which the trap received spam belonged to ISPs. To the best of our knowledge, there is no existing database that

maps ASNs onto ISPs. This is not surprising. Estimates of the number of ISPs vary from around 4,000 based on the number of ASNs that provide transit services to as many as 100,000 companies that self-identify as ISPs many of whom are virtual ISPs or resellers of other ISPs' capacity.

So we adopted a variety of strategies to connect ASNs to ISPs. First, we used historical market data on ISPs wireline, wireless and broadband from TeleGeographys GlobalComms database 2013 [95] . We extracted the data on all ISPs in the database listed as operating in a set of 40 countries, namely all 34 members of the Organization for Economic Co-operation and Development (OECD), plus one "accession candidate" and five so-called "enhanced-engagement" countries.

The process of mapping ASNs to ISPs was done manually. First, using the GeoIP data, we could identify which ASNs were located in each of the 40 countries. ASNs with one percent of their IP addresses mapped to one of the 40 countries were included in our analysis. For each of these countries, we listed all ASNs that were above a threshold of 0.5 percent of total spam volume for that country.

We used historical WHOIS records to lookup the name of the entity that administers each ASN in a country. We then consulted a variety of sources such as industry reports, market analyses and news media to see which, if any, of the ISPs in the country it matches. In many cases, the mapping was straightforward. In other cases, additional information was needed for example, in case of ASNs named after an ISP that had since been acquired by another ISP. In those cases, we mapped the ASN to its current parent company.

5.4 Comparison Results

In this section, we use data sets (Section 5.3.1) to rank several EU countries against each other, and against the US and Japan, developed nations comparable to EU countries.

Table 14 shows the average number of daily unique IP addresses for each global feed we have analyzed, for both top 10 countries with the highest number and for the countries of interest we have mentioned before. Analyzing this table, we can see that, there is a significant difference among the number EU countries and other top 10 countries, due also to the large difference in the population in these countries.

As discussed in Section 5.2, we have then to normalize this total counts to make these metrics comparable.To compensate for this, we have produced a ranking in which the number of unique IP addresses seen in the infection data is normalized by the Internet user population of the country (metric # 4 in Table 12). The results can be seen in Table 15. As can be seen, EU countries perform relatively well to others; but there is a significant difference between the Italy and Finland, for example. We need to determine with active measurements the probability of DHCP churn rates differences impacting the results. However, Finland is renowned for having highly effective mitigation initiatives, which can be seen in Table 15. This shows that there is a lot of potential for ACDC to bring EU rates closer to the ones observed for Finland.

Top 10 Countries

#	GameOver Peer		GameOver Proxy		Conficker		Morto		ZeroAccess		Spam	
	IPs	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**
1	GB	4536.67	UK	3652.43	CN	307422.54	CN	351.76	US	1830.03	IN	27733.18
2	JP	4251.1	IT	1317.69	BR	187905.77	IR	137.92	ES	894.54	RU	13590.24
3	IT	3754.36	JP	1106.8	RU	128092.49	BR	78.4	TR	830.66	VN	8535.64
4	US	2542.71	BY	997.17	IN	118912.98	TR	66.41	IN	762.3	BR	8199.17
5	UA	2016.02	GB	760.24	VN	111371.68	JP	44.39	IT	721.99	US	8006.29
6	IN	1690.31	KZ	697.95	KR	69882.05	TW	40.4	JP	623.31	PK	7320.34
7	UK	1665.18	UA	668.14	IT	69629.65	US	38.62	BR	443.99	ID	5697.34
8	FR	1210.62	IR	626.79	AR	62840.21	RU	34.61	MY	439.65	BY	5105.02
9	ID	948.74	ID	620.12	TW	62652.1	DE	34.61	TH	414.16	UA	4940.40
10	KR	842.48	VN	504.19	ID	62375.01	TH	32.57	VN	390.93	SA	4493.54

Countries of Interest

#	GameOver Peer		GameOver Proxy		Conficker		Morto		ZeroAccess		Spam	
CC	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**
NL	**62**	52.98	**132**	10.43	**85**	1299.95	**36**	9.27	**52**	50.17	**40**	683.47
DE	19	398.24	31	152.53	22	31574	9	34.61	14	326.91	21	2489.28
GB	1	4536.67	5	760.24	29	18076.93	19	17.33	15	310.82	26	2042.28
FR	8	1210.62	22	226.65	28	18761.83	35	9.29	11	365.54	32	1256.59
FI	173	3.15	183	2.37	135	118.03	166	1.79	122	6.24	FI	58.35
IT	3	3754.36	2	1317.69	7	69629.65	15	22.47	5	721.99	22	2451.62
ES	28	270.24	54	69.13	15	50135.93	23	13.58	2	894.54	17	2997.32
US	4	2542.71	23	207.49	12	57109.84	7	38.62	1	1830.03	5	8006.29
JP	2	4251.1	3	1106.8	20	32216.55	5	44.39	6	623.31	18	2875.94

Table 14: Average Daily Unique IP addresses ranking

Top 10 Countries

#	GameOver Peer		GameOver Proxy		Conficker		Morto		ZeroAccess		Spam	
	IPs	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**
1	SS	26700.00	SS	13100.00	SS	12900.00	PW	2	SS	500000.00	SS	30600.0
2	VA	4166.67	SH	625.00	RO	5402.16	SM	1.87	AS	14473.68	VA	4375.0
3	MP	241.55	MS	351.25	BG	4651.16	MC	2.58	CK	8666.67	NR	3571.42
4	TC	239.84	GE	219.28	VA	4000.00	GI	1.71	PW	6707.32	TK	2500.0
5	AI	239.83	SM	196.47	TW	3352.54	VU	1.95	VG	5471.96	NU	1818.18
6	MH	233.10	BY	191.59	MK	3199.93	AW	5.01	GP	4525.12	WF	1757.66
7	SM	213.53	MP	190.86	HU	2946.08	GD	1.94	MC	4372.81	AS	1694.07
8	GE	195.21	TO	170.45	AL	2874.16	IM	1.77	BG	4362.52	SH	1250.0
9	MC	172.38	MC	157.57	MP	2747.81	KY	1.75	ES	4058.59	BY	980.85
10	KI	166.44	TC	152.44	VN	2715.58	VI	1.99	GE	3561.21	MS	800.84

Countries of Interest

#	GameOver Peer		GameOver Proxy		Conficker		Morto		ZeroAccess		Spam	
CC	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**	**#**	**IPs**
NL	**185**	3.34	**206**	0.65	**192**	81.97	**141**	0.58	**127**	395.44	**150**	43.09
DE	165	5.7	190	2.18	112	452.48	148	0.49	78	796.42	163	35.67
GB	24	79.22	115	13.27	140	315.66	171	0.3	72	955	164	35.66
FR	77	23.18	168	4.34	135	359.27	176	0.17	57	1238.9	187	24.06
FI	207	0.6	208	0.49	220	24.48	161	0.37	188	47.91	211	12.10
IT	15	104.11	57	36.54	21	1931.03	139	0.62	12	3483.9	120	67.99
ES	144	7.5	194	1.93	38	1404.13	159	0.38	9	4058.59	93	8394
US	140	9.1	202	0.74	161	206.02	180	0.13	63	1168.5	178	28.88
JP	56	38.77	126	10.09	147	293.87	156	0.4	68	1006.37	183	26.23

Table 15: IP addresses/Million Internet Users Ranking (metric #4)

CC	Conficker				Morto				Spam			
	2011	2012	2013	2014	2011	2012	2013	2014	2011	2012	2013	2014
NL	176	183	196	195	68	75	94	76	108	179	156	144
DE	91	105	124	144	31	56	62	80	158	144	115	148
GB	126	135	145	149	64	76	97	115	146	152	133	136
FR	113	128	134	142	115	121	128	133	163	193	162	173
FI	209	214	217	219	141	151	149	143	199	205	206	213
IT	13	22	22	27	103	155	111	111	103	124	60	100
ES	31	34	23	37	57	59	70	70	129	53	39	114
US	150	154	161	162	79	93	93	99	178	164	136	92
JP	138	134	139	152	132	53	157	152	184	157	165	156

Table 16: Countries Yearly Ranking (normalized by each countries' Internet Users numbers, metric #4)

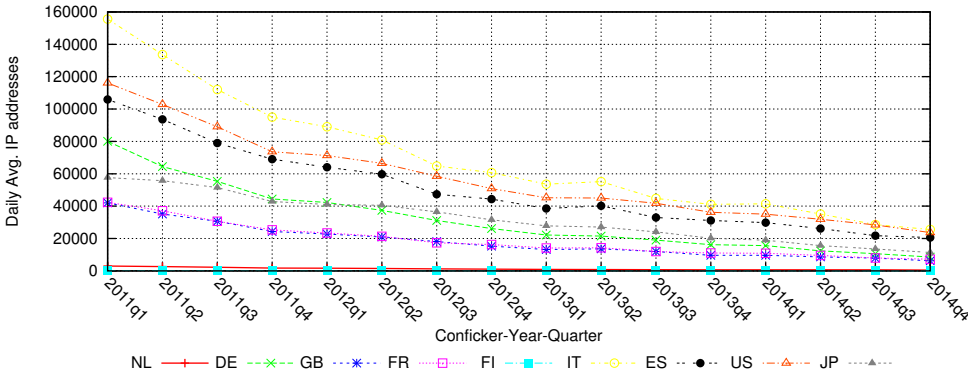


Figure 25: Conficker Countries - Daily Average

5.4.1 Country performance over time

Table 16 shows the evolution of the ranking of the countries of interest for the global feeds, broken into years. First and foremost, we can observe that most of the reference countries have improved over time, with a few exceptions.

We have also looked at the speed of clean-up across the reference countries. Figures 25–29 shows the time series of daily unique IP addresses the chosen countries. These figures show that how the total number of infected IPs changed over time, as also shown in Tables 14 and 15. As can be seen, for these botnets, most of the infection numbers were reduced over time.

To get a better sense of the relative speed of clean up, we have generated indexed time series. Figures 31 – 33 show the infection rates of the reference countries all index at 1 at start of the measurement period – i.e., we have divided all daily averages by the first daily average of the first measurement. We only performed this for the data sources that span more than one year (Morto, spam, and Conficker). In this way, all countries start with a value equal to 1 and their variation shows the percentage of infections that have increased or reduced. As can be seen, we can see that Germany and Finland are the ones that have improved more proportionally for conficker.

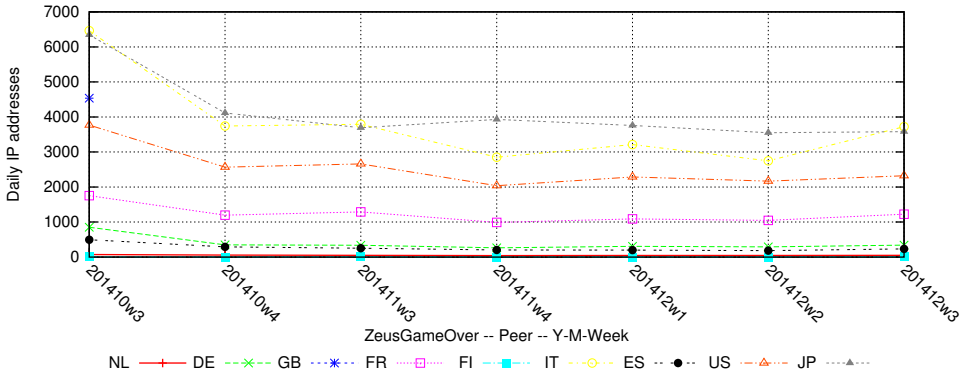


Figure 26: GameOver Peer Countries - Daily Average

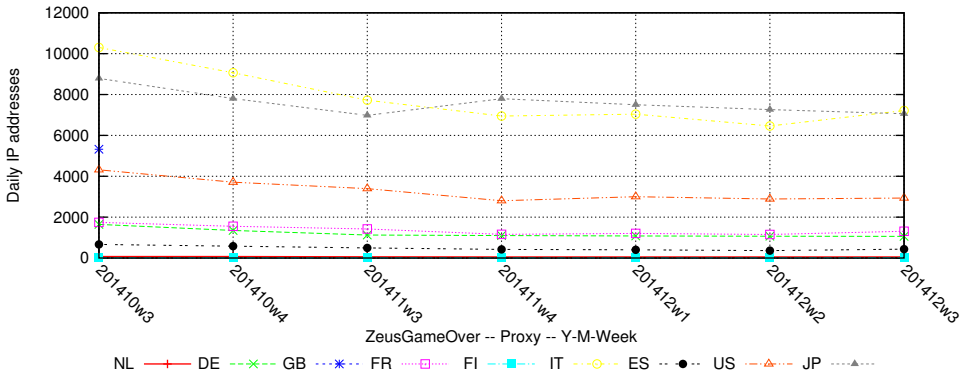


Figure 27: GameOver Proxy Countries - Daily Average

5.5 Next steps

We have covered in this chapter how measurement data has to first be captured, enriched, anonymized, and shared with ACDC CCH so we can produce the botnet metrics to evaluate the performance of countries and ISPs.

Then, we have show an example on how one of these metrics (Metric #4) can be employed to compare performance of countries with regards botnet infections. In the next deliverable, we will present an extensive evaluation not only of countries with regards all the produced metrics, but of the individual ISPs as well. We will continue on working on normalizing the metrics by measuring average churn rates of IP addresses.

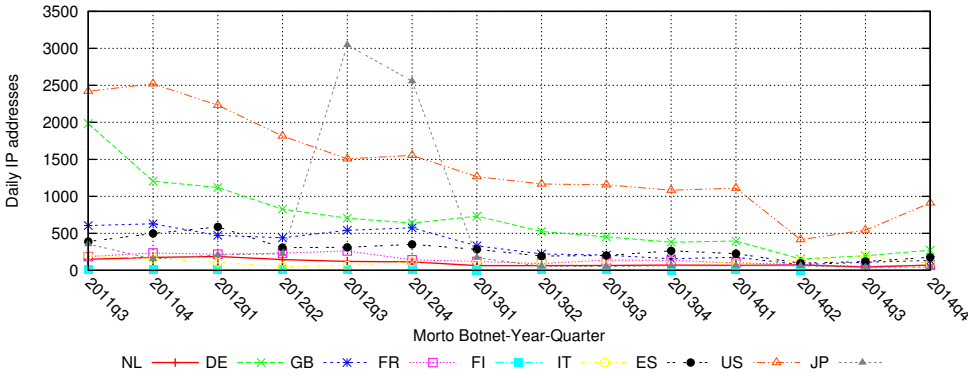


Figure 28: Morto Countries - Daily Average

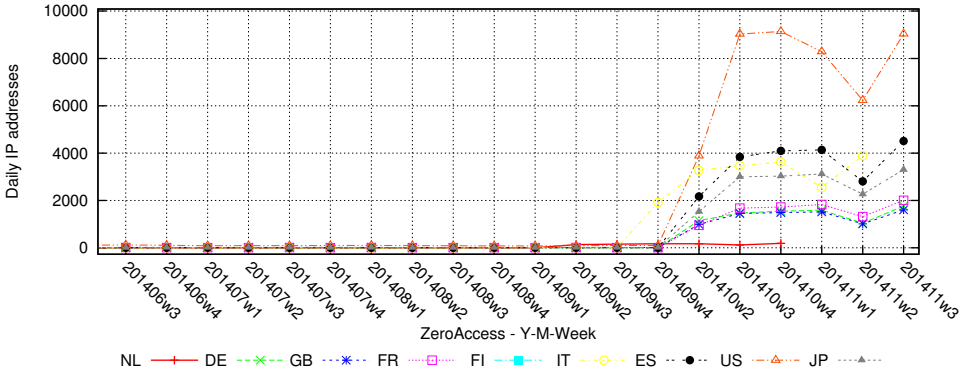


Figure 29: ZeroAccess Countries - Daily Average

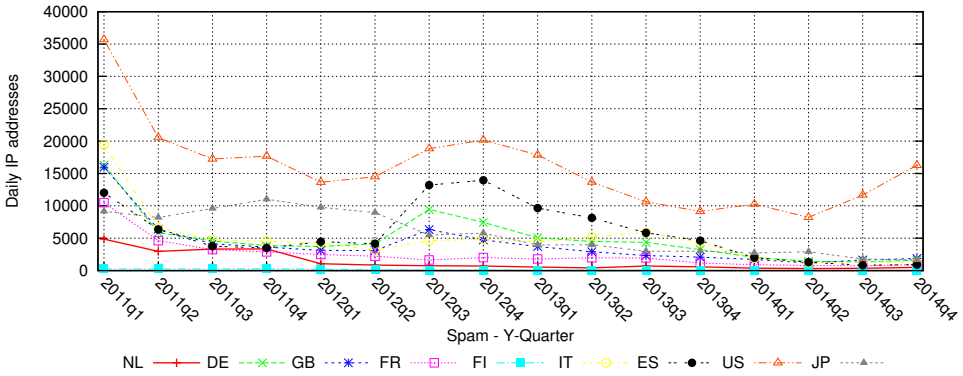


Figure 30: Spam Countries - Daily Average

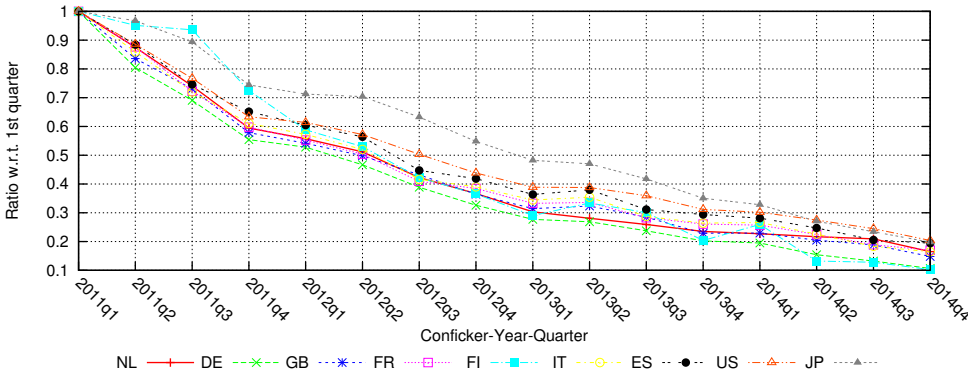


Figure 31: Conficker Countries - Indexed w.r.t. first quarter

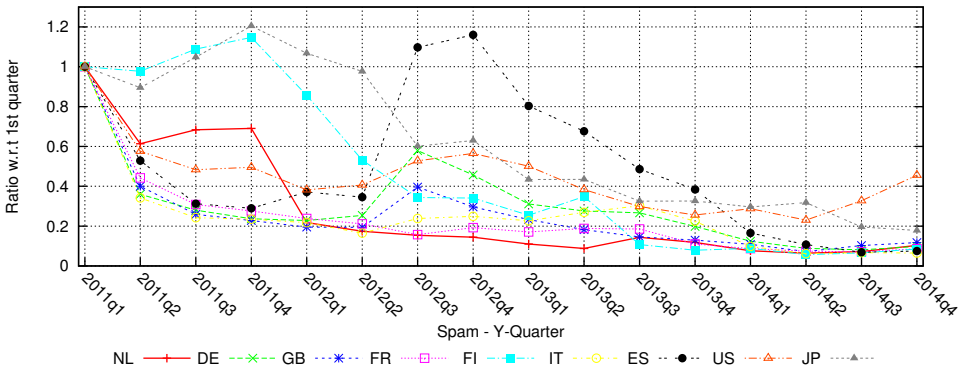


Figure 32: Spam Countries - Indexed w.r.t. first quarter

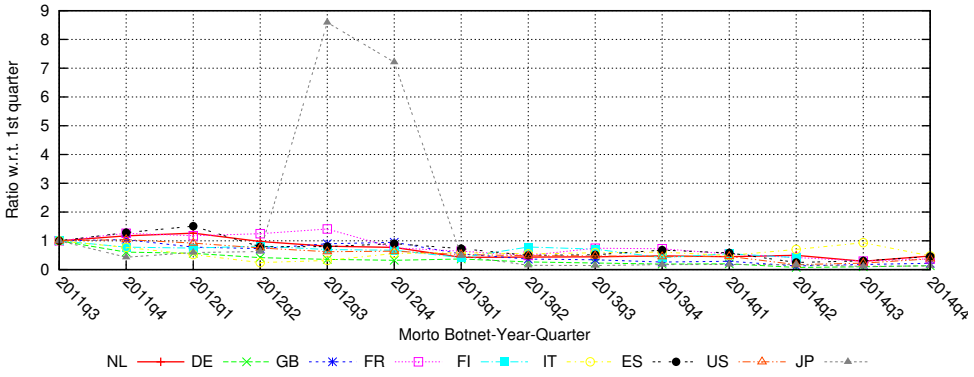


Figure 33: Morto Countries - Indexed w.r.t. first quarter

6 Summary

The ACDC project, more specifically WP4, has committed itself to develop comparative metrics that capture the number of bots, their command & control structures as well as related botnet infrastructure across networks.

In this report, we have covered how to quality control on data obtained from the ACDC CCH will be executed. After that, we have presented a survey of the state-of-the-art on botnet metrics. We have discussed how current solutions fall short of the requirements for comparative metrics across ISPs.

We then proposed a novel active-measurement based approach to deal with one critical problem: the impact of dynamic IP address allocation on deriving bot counts from IP-based infection data. These counts are skewed due to effects of Dynamic Host Configuration Protocol (DHCP) and Network Address Translation (NAT). We demonstrate the feasibility of this active-measurement approach by applying it to several large ISP networks.

Next, we have specified which metrics will be employed in the evaluation. We have shown (i) which botnet metrics we will employ in the ISPs evaluation with regards botnet infections, and (ii) how they can be enriched, anonymized, and shared using ACDC's CCH. We have also shown an example in which we compare the performance of various countries for real-world botnet datasets we have obtained, to illustrate the usage of one of the metrics.

The activities carried out in WP4 have lead to the following dissemination activities:

- Lone, Q. Moura, G. C. M. , Van Eeten, M.: Towards Incentivizing ISPs To Mitigate Botnets. In: 8th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2014 – Ph.D. track), Brno, Czech Republic, June 30-July 3 2014 [27]

- Lone, Q: Towards Incentivizing ISPs To Mitigate Botnets. Poster presented at 4th PhD School on Traffic Monitoring and Analysis (TMA), 2014.

In addition, we have submitted a paper based on Chapter 4 to the IFIP Networking 2015 conference, which is currently under review.

The main contribution of this report was to document the data processing and what metrics will be used in the next deliverable " D4.2 – Statistical evaluation of the impact ofthe Pilot", in which we will evaluate the performance of the ACDC sharing solution in reducing overall botnet infection accross Europe, in six month time from the writing of D4.1.

References

[1] R. A. Clarke and R. Knake, *Cyber War: The Next Threat to National Security and What to Do About It*. New York, NY, USA: HarperCollins Publishers, 2010.

[2] Cisco Systems, "Cisco IronPort SenderBase Security Network," 05 2012.

[3] MAAWG, "Messaging Anti-Abuse Workign Group - E-mail Metrics Program: The Network Operator's Perpective – Report # 15," November 2011.

[4] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, G. M. Voelker, S. Savage, and K. Levchenko, "PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs," in *Proceedings of the 21st USENIX Security Symposium*, (Bellevue, Washington, USA), USENIX Association, August 2012.

[5] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage, "Spamalytics: an empirical analysis of spam marketing conversion," in *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, (New York, NY, USA), pp. 3–14, ACM, 2008.

[6] B. Krebs, "The Scrap Value of a Hacked PC," May 2009.

[7] J. Soma, P. Singer, and J. Hurd, "SPAM Still Pays: The Failure of the CAN-SPAM Act of 2003 and Proposed Legal Solutions," *Harv. J. on Legis.*, vol. 45, pp. 165–619, 2008.

[8] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: understanding, detecting, and disrupting botnets," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, (Berkeley, CA, USA), pp. 6–6, USENIX Association, 2005.

[9] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A Survey of Botnet Technology and Defenses," in *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, CATCH '09, (Washington, DC, USA), pp. 299–304, IEEE Computer Society, 2009.

[10] M. van Eeten, H. Asghari, J. Bauer, and S. Tabatabie, "ISPs and Botnet Mitigation: A Fact-Finding Study on the Dutch Market," tech. rep., Duthc Ministry of Economic Affairs, The Haghe, The Netherlands, 2011.

[11] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, and J. D. Tygar, "Characterizing Botnets from Email Spam Records," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, (Berkeley, CA, USA), USENIX Association, 2008.

[12] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet

takeover," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 635–647, ACM, 2009.

[13] J. Hamilton, *Time series analysis*. Princeton, NJ: Princeton Univ. Press, 1994.

[14] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2nd ed., May 2006.

[15] "The team cymru bogon project." *http://www.team-cymru.org/Services/Bogons/*.

[16] N. Mantel, "Chi-square tests with one degree of freedom; extensions of the mantel- haenszel procedure," *Journal of the American Statistical Association*, vol. 58, no. 303, pp. 690–700, 1963.

[17] "Anubis networks: Unknowndga17 the mevade connection." *http://www.anubisnetworks.com/unknowndga17-the-mevade-connection/* .

[18] F. Soldo, A. Le, and A. Markopoulou, "Blacklisting recommendation system: Using spatio-temporal patterns to predict future attacks.," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1423–1437, 2011.

[19] "Dshield attack data." *http://www.dshield.org/port.html* .

[20] H. Debar and A. Wespi, "Aggregation and correlation of intrusion-detection alerts," in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, RAID '00, (London, UK, UK), pp. 85–103, Springer-Verlag, 2001.

[21] B. Morin, L. Mé, H. Debar, and M. Ducassé, "M2d2: A formal data model for ids alert correlation," in *Recent Advances in Intrusion Detection* (A. Wespi, G. Vigna, and L. Deri, eds.), vol. 2516 of *Lecture Notes in Computer Science*, pp. 115–137, Springer Berlin Heidelberg, 2002.

[22] M. van Eeten, J. M. Bauer, H. Asghari, S. Tabatabaie, and D. Rand, "The role of internet service providers in botnet mitigation: An empirical analysis based on spam data.," in *WEIS*, 2010.

[23] "D1.7.2 data format specification." TBP: `https://workspace.acdc-project.eu/`, 2015.

[24] "Developing processing and metrics for acdc wp4." TBP: `https://workspace.acdc-project.eu/`, 2015.

[25] Q. Tang, L. Linden, J. Quarterman, and A. Whinston, "Improving internet security through social information and social comparison: A field quasi-experiment," *WEIS 2013*, 2013.

[26] WORKING GROUP 7 - Botnet Remediation, "U.S. Anti-Bot Code of Conduct (ABC) for Internet Services Providers (ISPs) – barrier and metric considerations," tech. rep., FCC – http://www.fcc.gov/bureaus/pshs/advisory/csric3/CSRIC_III_WG7_Report_March_%202013.pdf, US, Mar 2013.

[27] Q. Lone, G. Moura, and M. Van Eeten, "Towards incentivizing isps to mitigate botnets," in *Monitoring and Securing Virtualized Networks and Services* (A. Sperotto, G. Doyen, S. Latr, M. Charalambides, and B. Stiller, eds.), vol. 8508 of *Lecture Notes in Computer Science*, pp. 57–62, Springer Berlin Heidelberg, 2014.

[28] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics." RFC 2330 (Informational), May 1998.

[29] S. Bradner and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices." RFC 2544 (Informational), Mar. 1999. Updated by RFCs 6201, 6815.

[30] C. Kaner, S. Member, and W. P. Bond, "Software engineering metrics: What do they measure and how do we know?," in *METRICS 2004, IEEE CS*, Press, 2004.

[31] I. S. Dept, "IEEE Standard for a Software Quality Metrics Methodology," tech. rep., Dec. 1998.

[32] Maxmind, "Maxmind." http://www.maxmind.com/, 2015.

[33] European Internet Exchange Association, "ASNs present at 10 or more IXPs," August 2012.

[34] Maxmind, "GeoLite Autonomous System Number Database," 2012.

[35] Team Cymru Community Services, "Ip to asn mapping," 2012.

[36] M. van Eeten, J. M. Bauerb, H. Asgharia, S. Tabatabaiea, and D. Randc, "The Role of Internet Service Providers in Botnet Mitigation: An Empirical Analysis Based on Spam Data," in *WEIS 2010: Ninth Workshop on the Economics of Information Security*, 2010.

[37] G. C. M. Moura, *Internet Bad Neighborhoods*. PhD thesis, University of Twente, Enschede, The Netherlands, March 2013.

[38] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane, "Using Uncleanliness to Predict Future Botnet Addresses," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, (New York, NY, USA), pp. 93–104, ACM, 2007.

[39] W. van Wanrooij and A. Pras, "Filtering Spam from Bad Neighborhoods," *International Journal of Network Management*, vol. 20, pp. 433–444, November 2010.

[40] R. Weaver, "A probabilistic population study of the conficker-c botnet," in *Passive and Active Measurement*, pp. 181–190, Springer, 2010.

[41] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage, "The heisenbot uncertainty problem: Challenges in separating bots from chaff.," *LEET*, vol. 8, pp. 1–9, 2008.

[42] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Automating analysis of large-scale botnet probing events," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 11–22, ACM, 2009.

[43] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 36, pp. 291–302, ACM New York, NY, USA, 2006.

[44] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda, "Fire: Finding rogue networks," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pp. 231–240, IEEE, 2009.

[45] M. Van Eeten, J. Bauer, H. Asghari, S. Tabatabaie, and D. Rand, "The role of internet service providers in botnet mitigation an empirical analysis based on spam data," in *Proc. of The Ninth Workshop on the Economics of Information Security (WEIS 2010)*, TPRC, 2010.

[46] Microsoft, " Microsoft Security Intelligence Report ," 2014.

[47] mcafee, " McAfee Labs Threat Report June 2014," 2014.

[48] CISCO, " Cisco 2014 Annual Security Report ," 2014.

[49] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting malicious flux service networks through passive analysis of recursive dns traces," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pp. 311–320, IEEE, 2009.

[50] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive dns analysis.," in *NDSS*, 2011.

[51] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, I. Osipkov, and J. D. Tygar, "Characterizing botnets from email spam records.," *LEET*, vol. 8, pp. 1–9, 2008.

[52] V. Dave, S. Guha, and Y. Zhang, "Measuring and fingerprinting click-spam in ad networks," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 175–186, ACM, 2012.

[53] C. Wagner, J. François, R. State, A. Dulaunoy, T. Engel, and G. Massen, "Asmatra: Ranking ass providing transit service to malware hosters," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pp. 260–268, IEEE, 2013.

[54] A. J. Kalafut, C. A. Shue, and M. Gupta, "Malicious hubs: detecting abnormally malicious autonomous systems," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–5, IEEE, 2010.

[55] A. G. West, A. J. Aviv, J. Chang, and I. Lee, "Spam mitigation using spatio-temporal reputations from blacklist history," in *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 161–170, ACM, 2010.

[56] Z. Qian, Z. M. Mao, Y. Xie, and F. Yu, "On network-level clusters for spam detection.," in *NDSS*, 2010.

[57] M. Thomas and A. Mohaisen, "Kindred domains: detecting and clustering botnet domains using dns traffic," in *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pp. 707–712, International World Wide Web Conferences Steering Committee, 2014.

[58] D. Dagon, C. C. Zou, and W. Lee, "Modeling botnet propagation using time zones.," in *NDSS*, vol. 6, pp. 2–13, 2006.

[59] J. Choi, J. Kang, J. Lee, C. Song, Q. Jin, S. Lee, and J. Uh, "Mining botnets and their evolution patterns," *Journal of Computer Science and Technology*, vol. 28, no. 4, pp. 605–615, 2013.

[60] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, vol. 7, Cambridge, MA, 2007.

[61] RIPE Network Coordination Centre, "RIPE Atlas." `https://atlas.ripe.net`.

[62] S. Zander, L. Andrew, G. Armitage, and G. Huston, "Estimating ipv4 address space usage with capture-recapture," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*, pp. 1010–1017, Oct 2013.

[63] A. Dainotti, K. Benson, A. King, k. claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos, "Estimating internet address space usage through passive measurements," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 42–49, Dec. 2013.

[64] X. Cai and J. Heidemann, "Understanding block-level address usage in the visible Internet (extended)," Tech. Rep. ISI-TR-2009-665, USC/Information Sciences Institute, June 2010.

[65] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, "Census and Survey of the Visible Internet," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, (New York, NY, USA), pp. 169–182, ACM, 2008.

[66] V. Brik, J. Stroik, and S. Banerjee, "Debugging DHCP Performance," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, (New York, NY, USA), pp. 257–262, ACM, 2004.

[67] M. Khadilkar, N. Feamster, M. Sanders, and R. Clark, "Usage-based DHCP Lease Time Optimization," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, (New York, NY, USA), pp. 71–76, ACM, 2007.

[68] I. Papapanagiotou, E. M. Nahum, and V. Pappas, "Configuring DHCP Leases in the Smartphone Era," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, (New York, NY, USA), pp. 365–370, ACM, 2012.

[69] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How dynamic are IP addresses?," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 301–312, 2007.

[70] M. A. R. J. Z. Fabian and M. A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging," in *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, USA*, 2007.

[71] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan." RFC 4632 (Best Current Practice), Aug. 2006.

[72] K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg, and J. Postel, "Internet Registry IP Allocation Guidelines." RFC 2050 (Historic), Nov. 1996. Obsoleted by RFC 7020.

[73] Internet Assigned Numbers Authority. `http://www.iana.org`, 2014.

[74] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Draft Standard), Jan. 2006. Updated by RFCs 6286, 6608, 6793.

[75] R. Droms, "Dynamic Host Configuration Protocol." RFC 2131 (Draft Standard), Mar. 1997. Updated by RFCs 3396, 4361, 5494, 6842.

[76] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)." RFC 2865 (Draft Standard), June 2000. Updated by RFCs 2868, 3575, 5080, 6929.

[77] Cisco Systems, "Configuring the Cisco IOS DHCP Server," 2014.

[78] W. Simpson, "The Point-to-Point Protocol (PPP)." RFC 1661 (INTERNET STANDARD), July 1994. Updated by RFC 2153.

[79] A. Schulman and N. Spring, "Pingin' in the Rain," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, (New York, NY, USA), pp. 19–28, ACM, 2011.

[80] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, 2013.

[81] J. Postel, "Internet Control Message Protocol." RFC 792 (INTERNET STANDARD), Sept. 1981. Updated by RFCs 950, 4884, 6633, 6918.

[82] Carna Botnet, "Internet Census 2012–Port scanning /0 using insecure embedded devices ." `http://internetcensus2012.bitbucket.org/paper.html`, 2012.

[83] D. Plummer, "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware." RFC 826 (INTERNET STANDARD), Nov. 1982. Updated by RFCs 5227, 5494.

[84] nmap, "Nmap - Free Security Scanner For Network Exploration & Security Audits." `http://www.nmap.org`, 2014.

[85] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection." RFC 5475 (Proposed Standard), Mar. 2009.

[86] S. Zander, L. L. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Mitigating sampling error when measuring internet client ipv6 capabilities," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pp. 87–100, 2012.

[87] Shatel. `http://en.shatel.ir`, 2014.

[88] Reseaux IP Europeens Network Coordination Centre (RIPE NCC), "Routing Information Service (RIS)." `http://www.ripe.net/data-tools/stats/ris`, 2014.

[89] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 93–108, 2005.

[90] A. Metwally and M. Paduano, "Estimating the number of users behind ip addresses for combating abusive traffic," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 249–257, 2011.

[91] A. Wahid, C. Leckie, and C. Zhou, "Estimating the number of hosts corresponding to an address while preserving anonymity," in *Proceedings of the 6th International Conference on Network and System Security*, NSS'12, pp. 166–179, 2012.

[92] M. Bailey, L. Bauer, L. J. C. S. Dietrich, and D. McCoy, "Conducting Research Using Data of Questionable Provenance." https://www.usenix.org/conference/cset13/workshop-program/presentation/bailey, 2013.

[93] P. Eckersley, "How unique is your web browser?," in *Privacy Enhancing Technologies*, pp. 1–18, Springer, 2010.

[94] L. Daigle, "WHOIS Protocol Specification." RFC 3912 (Draft Standard), Sept. 2004.

[95] TeleGeography, "GlobalComms Database Service." http://www.telegeography.com/research-services/globalcomms-database-service/, Mar. 2015.

[96] ShadowServer, "Conficker Working Group," December 2014.

[97] ABUSE.ch, "Zeus Tracker." https://zeustracker.abuse.ch/.

[98] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang, "On the analysis of the zeus botnet crimeware toolkit," in *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*, pp. 31–38, IEEE, 2010.

[99] M. Riccardi, D. Oro, J. Luna, M. Cremonini, and M. Vilanova, "A framework for financial botnet analysis," in *eCrime Researchers Summit (eCrime), 2010*, pp. 1–7, IEEE, 2010.

[100] N. Falliere and E. Chien, "Zeus: King of the bots," *Symantec Security Respons e (http://bit. ly/3VyFV1)*, 2009.

[101] Shadowserver , "Gameover Zeus." http://blog.shadowserver.org/2014/06/08/gameover-zeus-cryptolocker/.

[102] D. Andriesse and H. Bos, "An analysis of the zeus peer-to-peer protocol," 2013.

[103] B. Krebs, "Spam Volumes: Past & Present, Global & Local," Jan 2013.

[104] Cisco Systems, "Spam overview - SenderBase," 2014.

[105] TrendMicro USA, "Global Spam Map." http://www.trendmicro.com/us/security-intelligence/current-threat-activity/global-spam-map/, 2015.

[106] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 53–56, Apr. 2011.