Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Evaluating and Improving the Detection of Internet Background Radiation

Konstantinos Karampogias (10-937-498)

Semester Thesis
Computer Engineering and Networks Laboratory, ETH Zurich
Tutor: Prof. Bernhard Plattner
Advisors: Prof. Eduard Glatz / Dr. Xenofontas Dimitropoulos
October 2010 to February 2011

Contents

Chapter 1

Introduction

1.1 Motivation

The primary goal of every network analysis tool is to identify malicious or unproductive traffic, also known as "Internet Background Radiation". To this end, examining the traffic in flow level has been proved to be especially effective. The reason is that flow data aggregates the necessary traffic information into a more compact and compressed structure which enables fast and efficient analysis of high volume datasets. To extent the effectiveness of flow level analysis, we can further decompose the flows into two non-overlapping categories, named "two-way" flows vs. "one-way" flows. A flow belongs to the first category if there is bidirectional traffic between communication endpoints and to the second category if there is only traffic in one direction. This distinction between flows should be considered an improvement, since interaction between hosts is taken into consideration. This is, without doubt, an initial strong indication whether a flow is benign or malicious, productive or unproductive. Besides, one-way flows constitute a portion of the total traffic and, hence, network analysis is possible even over larger datasets. Nevertheless, this type of flows have not been thoroughly examined in literature and it has not been proved that they always relate to malicious/unproductive traffic. Whether there are benign one-way flows remains an open question and this semester projects targets at giving a quantified answer. In order to accomplish that, a classification scheme was implemented and run over the SWITCH flow datasets archived by ETH covering a period of seven years.

1.2 The Task of the semester project

The task of the project was to examine the sources of one-way flows, propose and implement a classification scheme and run this scheme over the SWITCH flow dataset archived at ETH. The categories need to be more descriptive than just benign or malicious, the implementation should be optimized because we analyse flow dataset of significant size and the result should finally be verified in some way. The ultimate goal is to define an upper bound on unknown one-way flows.

1.3 Prerequisites

A single data flow is a unidirectional sequence of packets which shared the same characteristic values and which were transmitted close in time. Usually a flow consist of the 5-tuple (Source Host, Source Port, Destination Host, Destination Port, Protocol). In our case we have NetFlow records that include the above standard 5-tuple, along with size/packet counter and timestamps. The records were collected data at ETH, and come from all border routers of the academic and research network of Switzerland (SWITCH). They are unsampled netflow flow traces captured from February 2004 until August 2010 (7 years). A single data flow (unidirectional) is one-half of a network conversation, which cannot characterize network activity

or performance, thus it is useful to decompose further the original sample into two-way flows and one-way flows. The decomposition procedure is simple, every flow in our dataset has an endpoint inside the SWITCH network (local endpoint) and an endpoint located outside of SWITCH network (remote endpoint). Two-way flows are produced by pairing unidirectional flows in opposite direction that use identical endpoints and protocol, one-way flows are the rest flows for which no pair was observed. In more formal description:

- Two-way flows (or biflows): are flows for which it had been observed unidirectional traffic for both direction.
- One-way flows: are flows for which it has been observed packets in only one direction.
  The one-way flows can further be divided into:
  1. Inflow: if there is a flow from the remote endpoint to the local endpoint only.
  2. Outflow: if there is a flow from the remote endpoint to the local endpoint only.

1.4 Report Overview

Chapter 2 describes the theory of One-Way flows explaining the cases where one-way flows should be observed. Chapter 3 describes the criteria which we can use in order to distinguish the flows, as well as the methods for validation and further analysis. Chapter 4 describes the implementation and the output classification scheme. In chapter 5, we present the graph produced by the result of our scheme after running it for a period of seven years. Finally, in chapter 5 we conclude giving some ideas for future improvements.

Chapter 2

Theory of one-way flows

2.1 Sources of one-way flows

The first step before a classification is to analyse the potential source of one way flows. We searched literature in order to identify the sources of one-way flows and build a comprehensive list of one-way flow types. Potential cases in which one-way flows are observed in our netflow datasets are the following:

- A failed connection attempt due to a malicious activity.
- An unsolicited packet or packets due to a malicious activity.
- A failed connection attempt due to a benign activity.
- A benign UDP application/ special protocol/ experimental protocol.
- A false positive, when the bidirectionality of the communication pair was not captured.

In more details, we describe the following situations

1. Scanning activity: it is considered to be the main source of one-way flows due to the fact that there are a high number of unsuccessful connection attempts. The sources of scanning can be sub categorized as follows:
   (a) Worms/virus/botnet activity[1]: A worm in its propagation step probes a set of addresses (randomly or targeted) in order to find vulnerable host to which it can be spread. A virus in order to stop antiVirus tools from functioning may map the update virus server to a random IP address. Further, there are botnets in which bots are programmed to use some static IP addresses in order to communicate with the Master bot in order to receive commands.
   (b) Malicious usage: an attacker using automated tools (e.g NMAP - Network Scanning tool) performs scanning over a random address space in order to discover alive hosts, acquire network topology information or identify potential vulnerable targets.
   (c) Discovery Services: nowadays software vendors try to automate the procedure and this means that they include service discovery activities (e.g automate discovery of mail gateways, automate discover of similar devices in the same network) which are based on scanning [3]. Usually these services function at local network, nevertheless sometime can create one-way flows. For instance if a static IP address is saved for accessing a log server (e.g syslog to 50.153.199.194 described at [12]) or for accessing an update server.

[1] information where taken from the course "Network Security"

(d) Testing reasons: Classic network debugging tools like ping, traceroute are still widely used for analysing network issues by the network operators.

(e) Other benign usage: Sometimes scanning can be benign, for instance "some search engines use not only 'spidering' (following embedded links), but also port scanning in order to find Web servers to index" [5].

2. Malicious activity:

(a) Denial of Service attack: During a DoS attack, which is based on source starvation technique by flooding service requests[2], the target server is unable to respond, thus a significant number of packet remain without answer and result in one-way flows.

(b) Backscattering: As a result of an attack (e.g by spoofing the source IP address), a legitimate host creates flows towards non-existent IP addresses or towards not listening nodes.

(c) Other (e.g ping of death)

3. Of Benign purpose: these one-way flows are not created by a malicious activity, but from a benign. Nevertheless, they cannot be considered productive traffic.

(a) Legitimate flows failed to be two-way flows due to firewalls/ network congestion/ network misconfiguration. It is quite often for a packet to be blocked/rejected/dropped by a gateway firewall, or by the target host software firewalls. Thus it is not possible to exist a reply to that packet and eventually this packet can not be part of a two-way flow[7]. The fact that these flows consist for more than one packets are due to the retransmission mechanism of the transport layer protocols.

(b) Hosts which are trying to reach legitimate servers create one-way flows. Today, web application are built primarily based on a Client-Server Infrastructure, thus the well-function of the server side is crucial, but it is not rare a "failure" to occur. The reasons why a server might be down are numerous, e.g heavy legitimate load, DoS attack, maintenance, upgrade/patch procedure, human mistake etc, but the result is one: there is no answer to the client request and a one-way flow is created. A characteristic example is one-way flows that are called "Web Retries", it is the case where a web server fails to address the heavy load, and consequently leaves without answer the constant[3] tries of an end host to connect to it.

(c) P2P protocols: In a peer-to-peer infrastructure peers discovery stage creates one-way flows since the peer network is very dynamic. These one-way flows are called "P2P reconnections attempt" and are produced when a legitimate peer reconnects to the network and tries to determine which peers are also accessible. We should mention that this one-way flow may be productive and useful traffic for the peer-to-peer application[4].

(d) Mobility: one-way flows are created also due to the mobility of the hosts. Today, it is often that mobile devices constantly change their access point to the internet, every time this happens the software (operating system, browser, mail client, update server etc) tries to re-establish previous connections, which led to the generation of one way flows since the destination hosts are not still accessible from the new location [3].

(e) Misconfiguration in network settings, or buggy application. Configuration of network devices have been quite complicate procedure and hence it is possible that there are false entries that create packets towards nodes that will never answer or that will never pass through firewalls. Also, there are application that are not thoroughly tested and show a not well network-behavior. Example is the protocol MSDP (Multicast Source Discovery Protocol Overview, RFC 3618) which is responsible for one-way flows in spite of the fact that it is based on bidirectional communication.

[2] e.g the DoS on web server sending a huge number of syn packets

[3] more than one one-way flows are created

[4] Peer-to-peer application may depend on whether there is an answer or not in order to make further actions

4. Benign Activity: These are one-way flow that indeed carry benign traffic.

   (a) Special protocols or UDP application: There are UDP application that do not require acknowledge packets, they mostly send or receive packets in one direction only and there is not need for the target host to acknowledge the received packet and thus create the bidirectional flow. For instance are the protocol NTP[7]

   NTP: "The NTP protocol specification (RFC 958) defines a symmetric mode in which a group of peers periodically send each other a copy of the latest timestamp without expecting an acknowledgement.In this mode the protocol may trigger benign one-way flows"

   (b) Application using different communication channels: There are applications which use different ports for transmitting the acknowledgement packets. Example is the commercial Application "Lansource" which has been observed in our datasets.

   (c) Experimental protocols/future architectures: like Mobile IP which in theory should create one-way flows.

   Mobile IP: [5] When a Correspondent Node (CN) wants to reach a user A which uses Mobile IP protocol, it must send a packet to home agent, the home agent forwards packet to mobile user locating in visiting network. Then the mobile user creates packets using his home IP address, sends the reply directly to the Correspondent Node while being in the visiting network (Triangular routing). The CN is going to answer to the home network and hence an one-way flow is created to the visiting network.

5. False positive:

   (a) Fragmentation: packets that are fragmented are an important source of producing one-way flows. The reason why this happens is that in a fragmented sequence of packets, the first packet carries the full 5-tuple header which is required for the aggregation procedure. Thus the data collection mechanism can only put the first packet in the right flow in the flow table and for the rest packets creates new flows entries. Obviously these flows are counted as one-way flows. In other words, it is not possible to decide if a fragmented packet belongs to an existing flow.

   (b) Sampling/measurement errors: The procedure of decomposing flows into two-way and one-way flows is a classification scheme which has without doubt false positive. For instance, "start-up" finish-off" effects[6][7], sampling techniques which capture only a portion of the traffic[7], host pairs which communicate in a very rare time frame, protocols[8] which provide security at the IP layer because they obscure the header of the transport layer.

6. Flows towards or from unallocated IP space: Flows that have as a destination IP or source IP address[9] an IP which belongs to a non-used, not allocated IP space.

[5] Information where taken from the course "Advanced Topic In Communication Networks"

[6] In the decomposition of our data into one-way or two-way flows, this was taken into account

[7] This also does not apply on our datasets, since no sampling technique was used

[8] IPsec, VPN technologies, ESP

[9] called Bogon which is an informal name for an IP packet on the public Internet that claims to be from an area of the IP address space reserved, but not yet allocated or delegated by the Internet Assigned Numbers Authority (IANA) or a delegated Regional Internet Registry (RIR). The areas of unallocated address space are called the bogon space.

2.2 Non Over-Lapping Categories

Analysing one-way flows and categorize them into Malicious flows (A1) vs Benign flows (A2) definitely gives a intuitive insight of the network traffic. Nevertheless, a network operator might be interested in a different classification scheme. We propose the following potential alternative black and white categories:

- Unproductive flows (B1) vs Productive flows (B2)
  Flows that are a result of a failed connection of legitimate user to a web server is an example of unproductive one-way flow. Nevertheless it is not always straightforward in which category belongs and is subjective to the decision maker. For instance, when the commercial voIP application skype connects to the internet, sends scanning probes to the saved IP addresses of the host list friend in order to determine if they are alive, this procedure creates one-way flows if a friend has changed his IP. If these one-way flows belong to the productive or to the unproductive category is questionable.

- Host unreachable (C1) vs. Host reachable (C2)
  This category is created by answering the question "Why there is no opposite traffic to that flow" There are two possible reasons, firstly the Host unreachable case in which the packet did not manage to reach the target host (e.g due to a firewall) and hence a reply from that host is not possible. Secondly, the Host reachable case, when the packet reaches the target host but for some reason the host does not reply (e.g. due to a failure).

- true One-way flows (D1) vs false One-way flows (D2)
  This category stems from questioning whether the one-way flows capture the interactivity between hosts. We distinguish that there are cases that the bidirectionality of a communication is not just a simple packet in the opposite direction which will create the two-way flow. Examples are when target host answers in different flow, does not answer at all (no need for acknowledge packet), answers in different way (e.g the reply is in fragmented packets), or it takes too long time to answer. Generally there are cases where the creation of one-way flow fails.

Chapter 3

One-Way Flows Classification Scheme

Having analysed in the previous chapter the sources/cases of the one-way flows we proceed to the analysis of the potential criteria that we can use in order to differentiate these flows. We discuss what will be the general approach (e.g source host behavior), what will be the characteristics (e.g a how many one-way flows has a source host), and what will be the final thresholds and criteria (e.g if a source host have more than X one-way flows) that we will take into account before characterize a one-way flow. At the end of this chapter we present the categories as well as our option for further analysis and validation.

3.1 Approaches

Decision based on the flow attributes

The initial technique is applying some static rules in our datasets, in other words comparing the flow attribute to a known signatures (e.g destination port of the flow vs a known port table). This approach is simple, useful but it has some major drawbacks and it will be avoided in our scheme:

- Cannot be very flexible in flow level since the attribute per flow are limited.
- Static rules enable application being hide (e.g the commercial P2P application skype uses the well known port 80). Only by checking the payload can be considered a valid[1] method.
- It is difficult to collect a complete and valid set of signatures.

In our scheme we applying some intrinsic rules to the attributes which are produced by the aggregate behavior of the packets (e.g the packet count, the byte size and the duration). These rules are useful to detect some extreme benign or malicious cases.

Decision based on the aggregate behavior

The alternative more flexible approach is to base our decision on the aggregate network behaviour, by examining the distribution characteristics of the traffic. In simple words, we choose an element X from the attribute of the flow, we aggregate the behavior of the element X over a time window, and then we examine the information. In particular, the element X can be one or any combination of the following attributes

(*protocol*, *srcIP*, *srcPort*, *dstIP*, *dstPort*,
srcAS, *dstAS*, *size*, *packets*, (Flow Attributes)
type, *starttime*, *duration*)

[1]The validity depends on the deep of analysis e.g comparing the destination port with known port is not very reliable, checking tcp flags is quite reliable, comparing the payload of every single packet with known signature is considered to be very reliable

Example, if we choose X to be the source host of the flow we are able to produce the following variables:

$[X].totalFlows$: how many flows in total has created the X
$[X].oneFlows$: how many one way flows in has created the X
$[X].totalBytes_k$: how many bytes created the X
$[X].totalPackets_k$: how many packets has sent in total the X
$[X].dstIPdegree_k$: with how many distinct destination IPs has the X communicate
$[X].srcIPdegree_k$: with how many distinct destination IPs has the X communicate
$[X].dstPORTdegree_k$: with how many distinct destination ports has the X communicate
$[X].srcPORTdegree_k$: how many distinct source ports has the X used

The $k = 2$ if it refers to two-way communication or $k = 1$ if it refers to one-way communication. In other words, these are the histograms/ aggregate behaviour of X. In this project we have chosen as X the srcIP and we followed a host classification approach because it is well defined in the literature. Based on that information we proceed to the flow classification based on source host behavior.

The important parameters that we should take into consideration when we aggregate the behavior are:

1. The time interval over which we aggregate the behavior.
   The interval time defines in great extent the accuracy of the analysis[2] and it is restricted by the resources of the system or by a not-optimized software. For instance we cannot aggregate the host behaviour for a time interval more than a day using optimized program because of the memory consumption.

2. The chosen variable X.
   The decision should be credible, meaningful (e.g cannot be the start time), and able to be scale in the chosen time window. For instance, we can aggregate the behaviour of the triple source host, destination host, duration which will give a good knowledge of the communication level between hosts, but it will not scale. In this step, using Data Frequency Algorithms can be of extreme help in order to chose an efficient variable X.

3.2 Characteristics

Scanning

As we have seen before, scanning is a major source of one-way, due to the fact that in most of the cases there are no successful connections. The issue of identifying scanning traffic by statistical analysis have been thoroughly studied in literature [11] [5] [1], [4] and it is based on source host behaviour analysis.

According to the [2] scanning traffic can classified into four categories with the following characteristics:

Port Scan: The traffic characteristic of a host which operates a port scan[3] is that probes scanning packet at a specific IP address at many different destinations ports. Thus, the ratio between the distinct destination IP addresses which the host contacts and the different destination ports should be small.

Horizontal Scan: The traffic characteristic of a host which performs a sweep scan[4] is that scans a lot of distinct IPs addresses to certain/limited number of ports. Thus by evaluating the ratio between the distinct destination IPs and the number of destination ports which the source host contacts versus a threshold, we can decide whether there is a sweep scan or not.

[2] defines the True Positive as well as the False Positive cases
[3] or Vertical scan: pings a specific IP at lot ports
[4] or Horizontal scan: pings a specific port at lot IPs

Hybrid Scan:[11] Hosts that scan multiple destination ports on multiple destination IP addresses. In this case, there are no specific characteristic, apart from the fact that these hosts are more active than a benign source host.

Coordinated Scans or Distributed Scan: are more complex scans which performed by multiple distinct hosts. For instance bots performs a sweep scan for vulnerable alive hosts and hand in the result to the master. The characteristic that might distinguish this type of scan is that usually scans a whole network. Hence if we aggregate the behavior of destination host in network level[5] might be able to get some interesting result.

Stealth Scans: are hosts which perform vertical or horizontal scans but using extended time intervals (scanning speed) between probing in order to evade Intrusion Detection/Prevention Systems which use statistical/behavioral traffic analysis. Thus the characteristics of this scanning type remain the same as before, but the time window over which the source host behavior is aggregated should be enlarged enough in order to be captured[6]. This approach has the short come that when the processing window is known it is easy for the scanner to adjust, by increasing their scanning interval and evade detection [5]. Nevertheless, the fact, that we examine the datasets offline in one-way flow level, makes possible the size of the time window to be enough large[7] and if a host performs scanning in an interval that evades the rules, then the number of flows added to the network are significantly small[8]. What is more hosts that perform stealth scan usually do not create in parallel useful bidirectional communication, thus it is highly possible that these hosts have only one-way flow and no two-way flows. A final characteristic of stealth automated scanning will be to have some static patterns, for instance the duration between the flow that a source host creates should be stable.

Malicious

Denial Of Service attacks is without doubt an activity which can create an immense number of failed connection and thus they constitute a significant source of One-Way flows. The main characteristic which a source host exhibits when it performs DoS attack is that he creates a lot of traffic in order to hammer its target destination node. On the other hand, the main characteristic that the target host exhibits is that it is the destination address of a numerous one-way flows. By taking these two characteristics into account, we can detect when a Denial of Service attack take place and which flows are part of that attack. Nevertheless, we mention firstly that usually there is a Distributed DoS and consequently the behavior of the host might not be abnormal enough to be recognized, and secondly that even in the case of single DoS the host usually spoofs his source IP address and again the it evades the host classification scheme.

Finding a unique characteristic in flows created by the **backscattering** is not easy. Without certainty, we can say that the flow might have only one packet, might target at an IP which is highly unlike to have other flows originating from it, the packet size should be small (usually it is an ack) and the protocol should be TCP in most cases.

[5]Choosing a mask, this will also limit the potential observation space and might help scaling

[6]When the time window is small then the scheme is only able to capture hosts which performs noise scanning. An example of noisy port scan executed by the NMAP network scanning tool[ref] should look like this: "nmap -sS 192.168.0/24".
While the time window is increased it also possible to capture hosts that perform a stealth scan. An example of stealth scan, that is performed over several weeks and is nearly impossible to be identified, is the following

$$for\ target\ in\ 192.168.1.0/24;\ do$$
$$nmap\ --scan-delay 1155\ --max-hostgroup\ 1\ -f\ -g\ 53\ -n\ -vv\ -PS 21,22,23,25,53,80,113,45943$$
$$-PA 80,113,443,45943\ \$target;\ usleep\ 1075000;\ done$$

More information can be found at [10]

[7]This is possible because one-way flows are a portion of the total traffic and the resources, that need to be allocated for storage and processing, can be available.

[8]For instance, if a host needs to probe one scanning packet every ten minutes in order to remain undetected, this is translated to one-way flow over around 600 thousand flows

Benign

Identifying one-flows that indeed carry benign data and are not a reconnection attempt from a not malicious host is not an easy task. The main characteristics can be that these flows should carry TCP data, that they carry a significant amount of data and have a significant number of packet (as well as duration).

On the other hand, one-way flows that belongs to a benign reconnection attempt do not exhibit the above characteristics. One indication of that type of flows would be the destination address to be a well behaved destination host[9] inside SWITCH network. In any case, we can say that this characteristic excludes the flow from being malicious.

A more flexible approach would be to investigate the history between this two communication pair. In other words to aggregate the communication taken place by every pair (local IP $\Longleftrightarrow$ remote IP) and show that the flow is benign since there is previous bidirectional communication.

False positive

As we have previously seen, ICMP flows evade the decomposition scheme into two-way and one-way flows. What it is more this flows can be can be benign as an ordinary fragmentation procedure, but also can be malicious since scanning uses the ICMP protocol frequently[10]. A possible characteristic of benign ICMP over malicious can if the size per packet is close to MTU value. Other protocols like ESP that also evade decomposition can be easily recognised and consider to be benign.

3.3 Criteria

Identifying malicious traffic

We adopted the criteria that are expressed by Wolfgang at [4] in order to identify scanning hosts. Comparing the behaviour of the source host by using some heuristic criteria we can recognize hosts that perform sweep scanning, port scanning. Furthermore, we express some intrinsic misbehavior characteristic with which we expect to capture the rest malicious traffic (backscattering, stealth scanning, etc).

- **SWEEP SCAN:**

$$\frac{[IP].dstIPdegree_1}{[IP].dstPORTdegree_1} > SWEEP_{THRESHOLD} \qquad \text{(Sweep Scan Filter)}$$

The threshold value that [4] suggest for the (Sweep Scan Filter) is 30.

- **PORT SCAN:**

$$\frac{[IP].dstIPdegree_1}{[IP].dstPORTdegree_1} < SCAN1_{THRESHOLD} \quad \&\& \quad [IP].dstIPdegree < SCAN2_{THRESHOLD} \qquad \text{(Port Scan Filter)}$$

A further check that distinct destination addresses is a small number can also be applied in order to increase the certainty that the host is indeed scanning. The thresholds that [4] suggests is $SCAN1_{thr} = 0.33$ and $SCAN2_{thr} = 5$. We mention that we expect to number of hosts(and flows) that perform port scanning to be smaller than the number of sweep scans. The reason is that before a port scan, a sweep scan usually takes place.

[9] e.g a Web Server at port 80, a DNS at port 53, a NTP server at port 37 or 123, a mail server at port 25 etc ...

[10] There are other malicious activities that use ICMP like firewall penetration, icmp tunelling

- **Stealth SCAN:** Based on the idea that the host who performs scanning is not well behaved. Whether a source host is behaved well or not depends on whether he produce no bidirectional communication. We apply the following filter:

$$[IP].twoFlows == 0 \ over \ 30 \ min \ time \ interval \quad \text{(Malicious 1)}$$

- **Other malicious:** Flow which has only one packet is potential malicious flow.

$$number \ of \ packet == 1 \ over \ 30 - min \ time \ interval \quad \text{(Malicious 2)}$$

- **Denial of Service Attack:** The Wolfgang at [4] states that a heuristic rule which indicates when a host joins a DoS attack, based on connection behaviour. This rule remains applicable to one-way flow because as it had been explained, DoS attack is source of one-way flows. In particular we have the following rule: the following rule :

$$[IP].dstIPdegree_1 < DoS1_{THRESHOLD} \ \ \&\& \\ [IP].dstPORTdegree_1 < DoS2_{THRESHOLD} \quad \text{(DoS Filter a)}$$

which should be applied in combination with

$$\&\& \ \ [IP].(oneFlows) < DoS3_{THRESHOLD} \ over \ 6 \ sec \ interval \quad \text{(DoS Filter b)}$$

The thresholds that [4] suggests is $DoS1_{thr} = 5$, $DoS2_{thr} = 5$, and $DoS3_{thr} = 6$ *connection per second*. As we can see the last rule of the above filter is an aggregate behaviour over a small time window (6 sec). In our analysis, the window processing time is 30 minutes, so we applied the following the rule (DoS Filter b')

$$\&\& \ \ [IP].(oneFlows) < DoSe'_{THRESHOLD} \ over \ 30 \ min \ interval \quad \text{(DoS Filter b')}$$

where $DoS3'_{thr} = 3 \times 3600 \ one - way \ flows \ per \ interval$

Identifying benign traffic

We proceed to the criteria which are going to capture the benign traffic. The first idea is to characterize an one-way flow by characterizing its destination address. The procedure is simple: *we identify a list of IP addresses that during a broad observation interval are considered to be safe destination IP addresses and then we examine the destination host per flow if belongs to that list.*

There are some remarks:

- The lists of IP addresses is produced by examining the two-way flows which are considered to be of benign use and credible.
- The classification/ identification of source behaviour is a well known subject.
- IP addresses are stronger rule than just testing against port number because public address can not change easily.
- The observation time window plays obviously important role to the false positive ratio.

Identifying Safe destinations hosts

Our approach here depends on the information which can be extracted from examining all bidirectional flows. If we find a pattern to recognise safe servers, then we have a complimentary rule that a flow which target at a safe server then it may be potential. Searching literature we found patterns which can recognize with validity the following benign hosts:

WEB servers

WWW is the dominant application of the internet and thus web retries to connect to a server which is not temporally accessible might be quite numerous. Thus it is useful to find pattern that recognises web servers, and then check if one-way flows are indeed attempts of a host trying to connect to a legitimate web server.

A static pattern (dstPORT 80, 8080, 443) is not very useful rule since a lot of application hide themselves behind that door, for instance the voIP application Skype. Nevertheless, a comparison versus this ports can extend the validity of the following filter. Wolfgang and Karagiannis proposed at [4], [6] rules that can identify web servers with accuracy. Nevertheless for simplicity reasons, and due to the fact that accuracy at finding web servers is not the target of the projects, we adopted the pattern which Perenyi at [9] stated:

According to [9], web servers can be distinguished by the following pattern

Hosts with parallel connections to a http
ports are considered to be webservers (Web Server pattern)

The above pattern can be transformed to the following rule

$\#biflows\ in\ dstPort\ 80 > WEB_{thr}\ \ over\ 10\ min\ interval$ (WebServer rule)

As we picked the value 22 for the thresholds and for the interval was only 10 minutes. We tested these values manually[11] and the results were satisfactory.

Potential P2P hosts

According to the previous chapter, P2P reconnection constitute a source of one-way flows. Thus, we need pattern and rules that identify peers of a P2P network. The basic idea is that the majority of P2P applications use both TCP and UDP protocol (TCP for transfer, UDP signaling) [6]. Thus we can create a list of potential P2P hosts by observing for that in the biflow sample.

The filter is the following

$(src.IP, dstIP)\ :\ \exists\ tcp\ \&\ udp\ communication\ over\ 10\ min\ interval\ \in (P2Phosts)$ (P2P pattern 1)

Apart from this in the literature there are other four patterns have been proposed that add accuracy to the identification: Briefly, in the p2p application, it is possible for the source host to repeat the source port within 60 sec, contrary to the normal applications. In normal applications it is very unusual that same source port is used in such a small period of time. [9].

$[IP].srcPORT the\ same\ over\ 60sec\ then\ \in (P2Phosts)$ (P2P pattern 2)

According to [6], each P2P peer *"the number of distinct IPs connected to it will be equal to the number of distinct ports used to connect to it"*. According to [4] *"has at least the same number of distinct IP addresses, number of distinct ports connected to it"*.

$[IP].dstIPdegree_2 \geq [IP].srcPort_2\ then\ IP\ \in (P2Phosts)$ (P2P pattern 3)

[11] For a sample day of Aug 2010, we saved all IP addresses that were characterized as web server and using the NMAP tool we validated that 90% of them answers to an HTTP request.

A further pattern is compare flows with a known table of ports used by the P2P applications. The argument in order to make this check is that "it seems disadvantageous for non-p2p applications to deliberately use well known P2P ports for their services, since traffic on these ports is often blocked "[4]

Benign using strong intrinsic characteristics

Apart from the destination approach we can use consider flow with significant size, packets and duration as being benign. We chose the following filter

$$number\ of\ packet > 10 \ \ || \ \ size > 10Kb \qquad \text{(Benign 1)}$$

There was not an extensive test about the thresholds values because the flows captured by this filter was a very small portion of the total traffic.

3.4 Categories

The issue with the above criteria is that they do not have the same priority, and lead to overlapping categories. For example a host may perform sweep scanning sending flow with one packet to a known peer-to-peer host. Consequently, we need to put priorities to the criteria in order to create distinct and able to be visual categories[12]. The priorities and the relevant categories are the followings:

1. Scanning & Dos flows: The category with top priority is combining the criteria which we took by the literature. The flows which belong to this class are the flows for which their source IP address is classified as malicious because it performs sweep scan, port scan or DoS attack.

2. Stealth Scanning flows: The flows which belongs to this class are the flows whose source IP address is considered to be malicious since it has not created any bidirectional communication with a host inside switch. Potential malicious activity is the stealth scanning in which a source host probes at long time intervals.

3. Other malicious flows: Flows which have one-packet size are also considered malicious since any potential use should include more packets. For instance it cannot be a reconnection attempt because when a benign host is trying to reconnect to a server or peer, it sends more than one packet.

4. Benign flows: are the flows which have intrinsic benign characteristic or belong to other protocols

5. Potential Benign flows: Flows that have evaded from the above filters and that targeted at a safe host inside SWITCH are considered as being potential benign flows. Nevertheless, the number as we will see is always not significant.

6. Fragmentation: flows that carry ICMP traffic and belong to a fragmented sequence of packets.

7. Unknown flows: the rest flows which where not captured from the above filter.

More information about the classification scheme i given to the next chapter.

[12]Nevertheless, in our implementation the value of number of flow for all criteria was saved in the output file, as well as their intersection

3.5 Validation - Further analysis

The next step after we have classified the one-way flows is to find a way to validate the results. A first efficient way would be to built custom datasets and validate that our analysis captures the type of traffic we have manually inserted. Nevertheless this is not an option, since we examine backbone data and building dataset of that size, including all potential forms of traffic, is not possible. However, we used synthetic datasets to verify the correctness of the code during the development phase. Our next two options which we used is comparing our scheme with a network telescope and by manual analysing samples using Frequency Itemset Mining Algorithms.

Building a Network telescope

A Network telescope[13] is a set of IP addresses which are reachable (exist on the global routing tables) but are not being used by any host. Hence flows towards to that IP space are going to be one-way flows. Traffic inside network telescope have been thoroughly examined by analysing datasets from IP Telescope [8]. Worm activity and backscattering are proved to be the primary reasons for generating this type of one way flows. Apart from worms and according to [12], network protocol vulnerabilities, misconfigured network servers, services, and devices, misconfigured attack tools, misconfigured peer-to-peer network software and various other software programming bugs can also be blamed. The bottom line is that one-way flows observed inside a network telescope is malicious traffic.

In this project, we built a telescope since we had in our disposal the necessary datasets. The procedure was simple: since we have the traffic we can look for hosts that have not create any traffic at all during the observation period. These hosts are probably unused IP address and constitute a darknet inside the SWITCH network. Thus by acquiring this list of the IP addresses and by checking the destination address of the inflows if it belongs to that list we obtain a first clue about our validity.

Manual Inspection through FIM

Frequent Item Set Minning (FIM) is fundamental problem in data mining, motivated by the decision support problem and has many applications like Association Rules Induction. FIM is also effective in finding the root-cause of network anomalies observed in network traffic and thus will be a useful tool in order to extract frequent itemsets and conclusion about what are the prevalent flow characteristic of one-way flows. Nevertheless, the are some restraints, firstly the analysis should be manual since the procedure cannot be automated. Secondly, FIM algorithms cannot be applied to large datasets and thus we need to pick indicative samples firstly. Finally, the result might be time dependant and rules or conclusion that we make for one period would have small impact to other time periods.

[13] or Darknet or blackhole monitor or network sink

Chapter 4

Implementation

4.1 Description

The program is split into three stages:

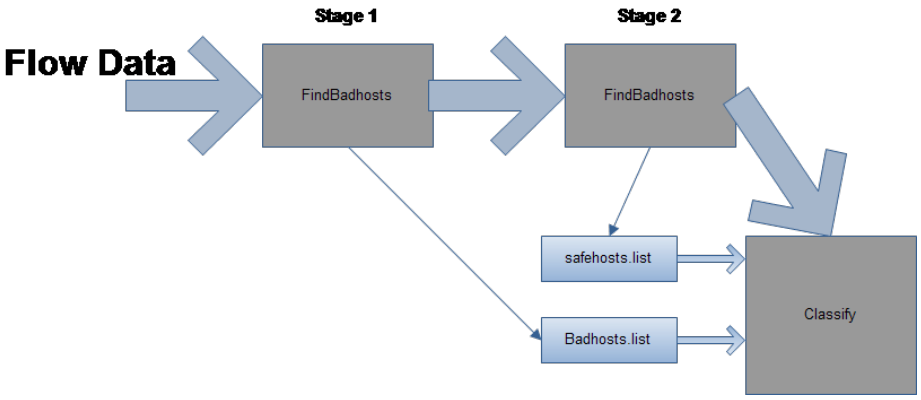

Figure 4.1

Stage 1: Find Safehosts: Given a time period A, we apply the filters described in the previous chapter and we produce a list of IP addresses which are considered to be safe destination hosts. In particular we use

1. the filter "parallel" connection at port 80 over a 10 minutes interval to distinguish the Web Servers.[1]
2. the filter both tcp and udp connection between pairs in order to distinguish peer-to-peers hosts.[2]

The safe host list are saved in an external file for convenience.

Stage 2: Find Badhosts: In order to find hosts that belongs to an unused IP space, we examine IP address, inside SWITCH network which belong to an inflow as destination address, whether they have created any flow over the two week period. Hosts that have no biflow or outflow are considered to be not valid and marked as badhosts. The list is also saved in en external files.

Stage 3: Classify One-way flows: We process three samples at once. This means that we take three samples of 10 minutes, we concatenate them and aggregate the behavior of the source host over that period. Then we apply our filters to the aggregated information in order to characterize the host. The

[1] How parallel is the connection does not play role according to tests.

[2] There are other application which use both tcp and udp like DNS, but since they are also of benign use, there is no need to distinguish them.

final classification of the One-Way flows is dependant on the source host behaviour, on some strong benign criteria and whether the destination point belongs to the safe host list.

The classes, which are implemented combining the criteria expressed to the previous chapter, are the following:

1. Scanning & Dos flows
2. Stealth Scanning flows
3. Other malicious flow
4. Benign flows
5. Potential Benign flows
6. Fragmentation
7. Unknown flows

In the following figure 4.2, the internal structure of the classify stage is given, as well as how the classification scheme is applied.

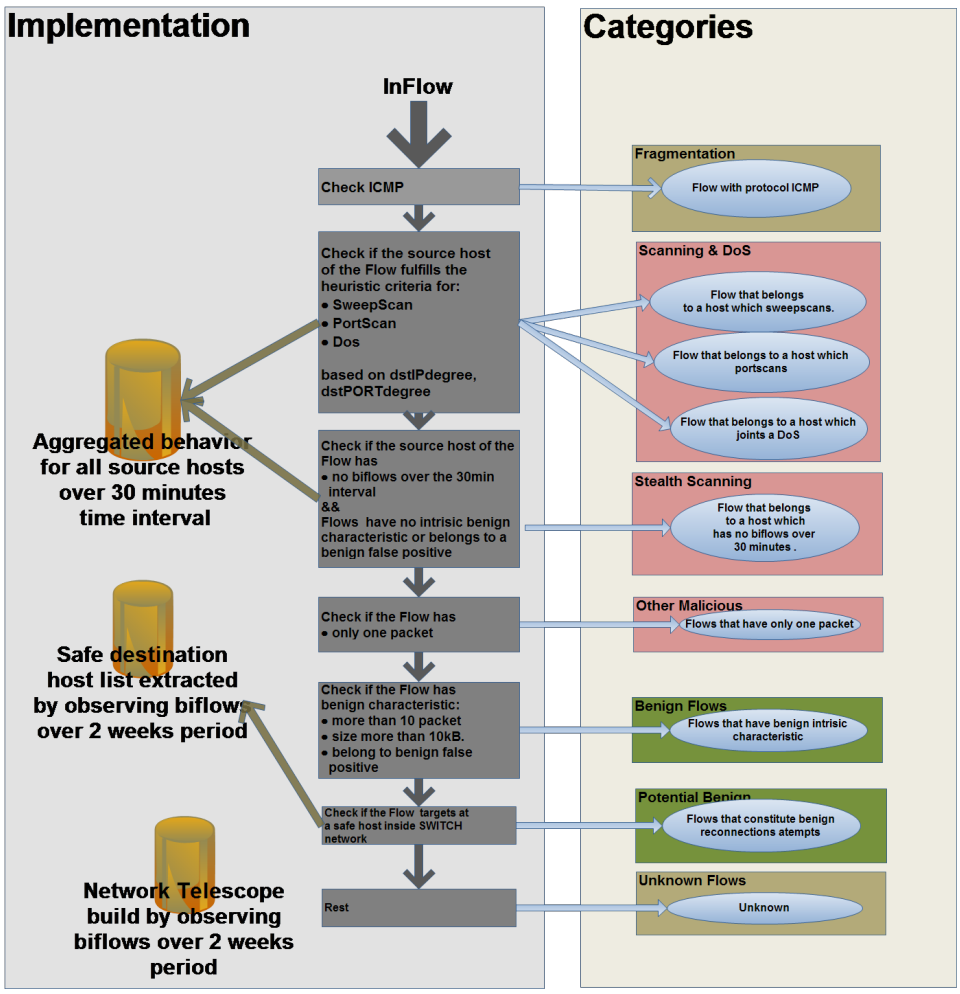

Figure 4.2: How the classification works

4.2 Input/Output

Input: The program is given as an input a file which contains the location of all 10 minutes samples that belongs to the observation period. The total size is irrelevant but we used a period of two weeks, which should be a file containing the location of around 2040 compressed 10 minute samples.

Output: The program produces as an output some log files and CSV files, which include the following informations:

- The IP addresses which belong to the network telescope (Badhost.list)
- The IP addresses which considered to web servers or P2P peers (Safehost.list)
- Performance Profile (P)
    1. Memory consumption
    2. Cpu consumption
- Statistics (S)
    1. Per flows (window, totalflows, biflows, inflows, outflow)
- Classification of One-Way (C)
    1. Per flows (window, totalOneWayflows, flowCategories)

From that logs files we produce some visual graphs.

4.3 Performance

It is a challenge to handle and analyse so large datasets. In the following table, we give a sense of how our program performs as far as the execution time and memory consumption is concerned.

Period	type	Stage 1	Stage 2	Stage 3	Total	unit
1hour	time	60	67	229	356	sec
	memory	0.15	0.20	7.0	1.05	Gb
24hours	time	937	1237	4239	6413 (1.7hour)	sec
	memory	0.52	0.48	7.7	8.7	Gb
4hours	time	20113	31056	84076	135245(37hours)	sec
	memory	0.90	0.90	12	13.2	Gb

We analyse our datasets significantly more rapidly than real time, the average speedup is 12. In other words 400 hours sample is evaluated at $400/12 = 33$ hours.

4.4 1 Day - Example output

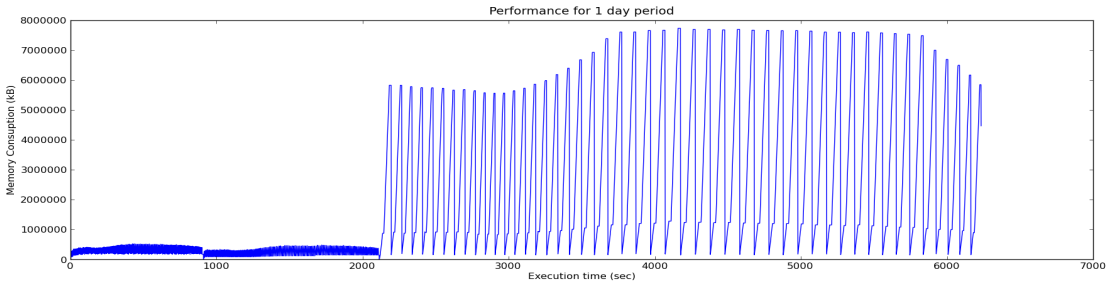

Figure 4.3: Performance Graphs

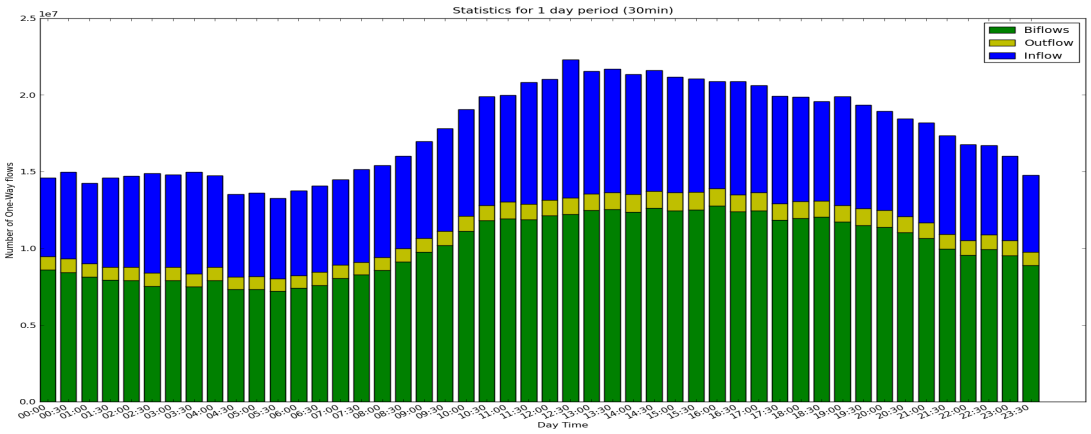

Figure 4.4: Statistic Graph

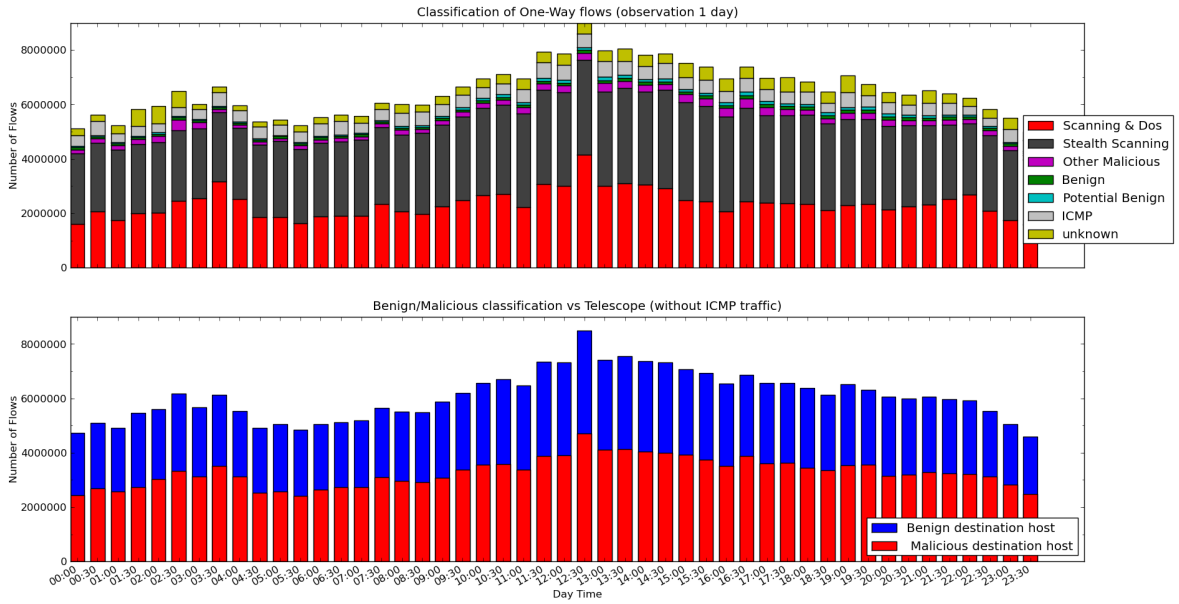

Figure 4.5: Classification Graph

Chapter 5

Results

5.1 Test - Samples

We would like to make conclusions by examining a period of 7 years (2004 until 2010). In spite of the good speed-up we could not run over the total time because this would have taken around 730 hours per year (almost a month)[1]. So we had to take a sample per year, in particular every year is represented by two two-week samples taken on month February and on August[2].

Period	10 min samples	One-way/Two-way	Safehosts	Unused IPs
aug10a	2406	1.97	38017	1501437
feb10a	2347	2.02	36795	1622712
aug09a	2402	1.16	32112	2398170
feb09a	2395	1.64	30250	1594881
aug08a	2398	0.83	26942	1536211
feb08a	2397	0.83	31661	1606444
aug07a	2398	0.97	90626	1508468
feb07a	2283	0.59	100281	1610628
aug06a	2387	0.51	20716	1568208
feb06a	2357	0.55	98951	1609704
aug05a	2400	0.62	80737	1635055
feb05a	2400	0.55	86108	1592791
aug04a	2396	0.60	13272	1779790
feb04a	2391	0.49	32684	1560175

Figure 5.1: Sample summary and statistics

[1] Furthermore, the original archived data are not decomposed into one and two-way flows.
[2] We examined the exact datasets that used at "On the Evolution of One-way Flows in the Internet"

5.2 Evolution of One-Way flows

In the following figure 5.2 we see the evolution in total number as well as in percentage of one-way over the last 7 years. Observations:

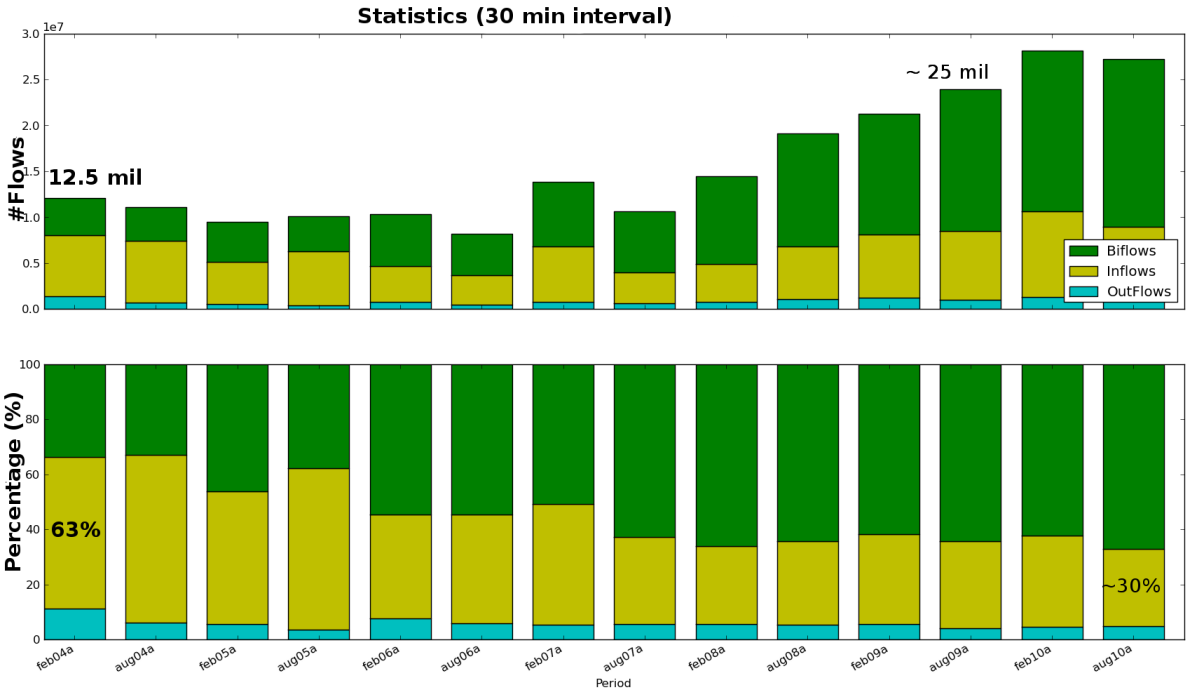

Figure 5.2: Evolution of One-Way flows

- We observe an increase of the total traffic over the year. This increase is a physical consequent of the internet penetration we faced the last years.
- The percentage (one-way flows over two way flows) declined since 2004.
- The percentage is stable since 2008[3]

[3]More information "on the evolution of One-Way Flows in the Internet"

5.3 Approach- Result Explanation

First we graph the result exactly as our classification scheme implement. Secondly we split the flows into flows which are going inside the telescope and into flows that are going outside telescope. In that point, we produce three graphs which make a comparison with the telescope. The first graph is total flows that are going inside vs the flows that are going outside, in the second graph we give the values and the percentage per category and final we give in a graph the result of our scheme applied only to the flows that belong to the telescope. All flows with a small exception can be considered malicious. As far as the flows outside the telescope is concerned, we give a graph with our classification result. Then in order to have a more complete insight we applied a FIM analysis before we proceed to their final categorisation as malicious, benign or unknown. In the following figure our approach, along with the graphs, in order to reach a final conclusion is visualized.

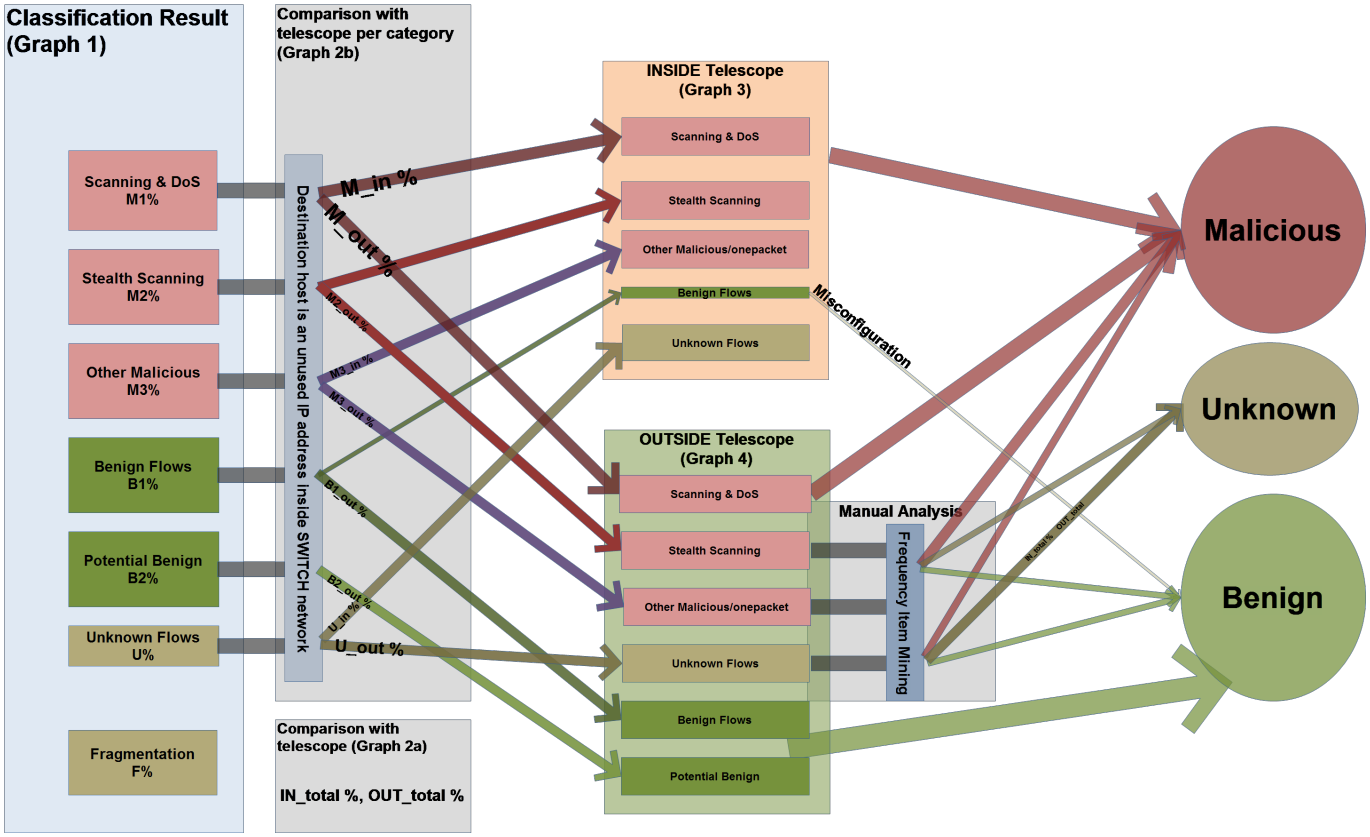

Figure 5.3: Approach - Graphs

5.4 Graphs

5.4.1 Graph1: Classification on ALL flows

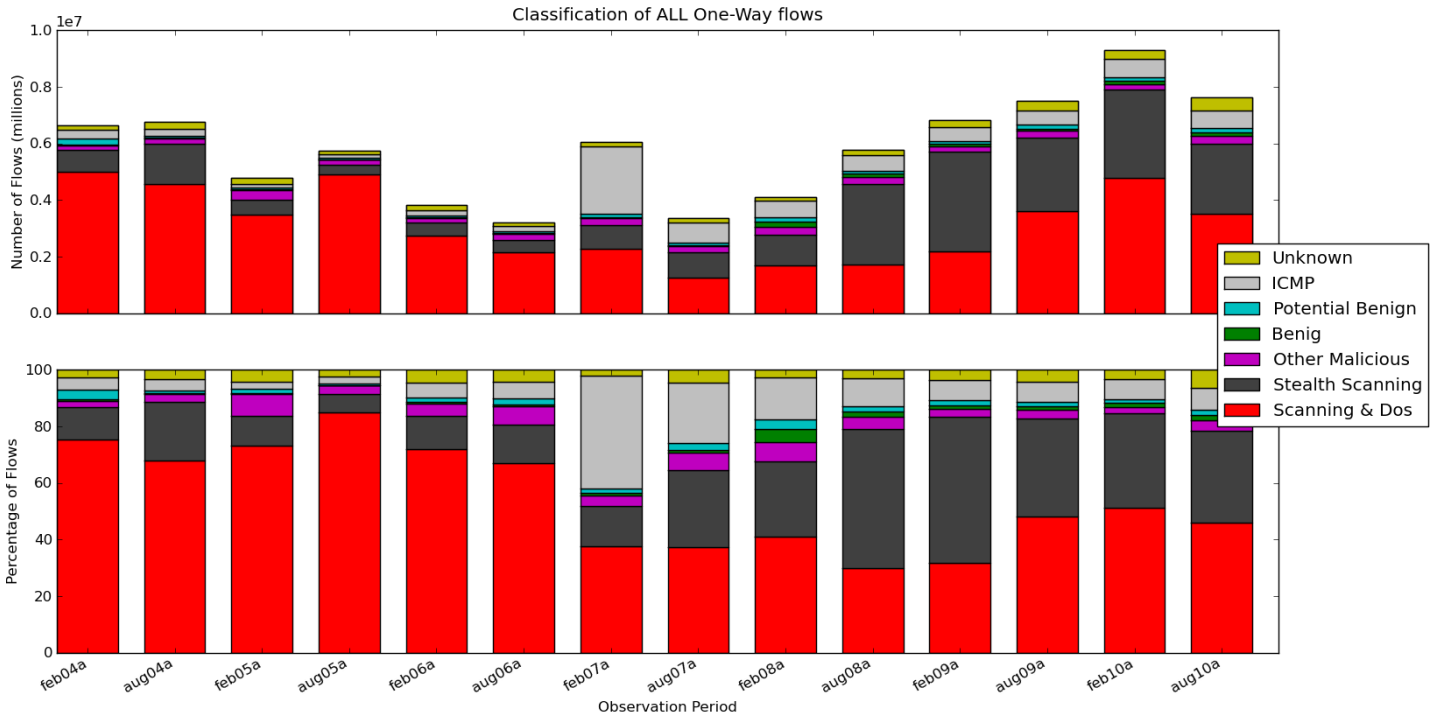

Figure 5.4: Classification on ALL flows

Observations:

- Scanning percentage has been declined
- More stealthy scanning has been increased
- In February 2007 we observed a lot of ICMP traffic (5.5). This can be connected to worm outbreak (e.g alliaple worm).
- There is an increase in the benign categories over the last years

What we keep from the graph:

- Scanning&DoS flows: 54%,
- StealthScan flows: 23%,
- OtherMalicious flows: 4.4%,
- Benign flows: 1.1%,
- Potential Benign: 1.9%,
- Fragmentation/ICMP: 10%,
- Unknown flows: 4%

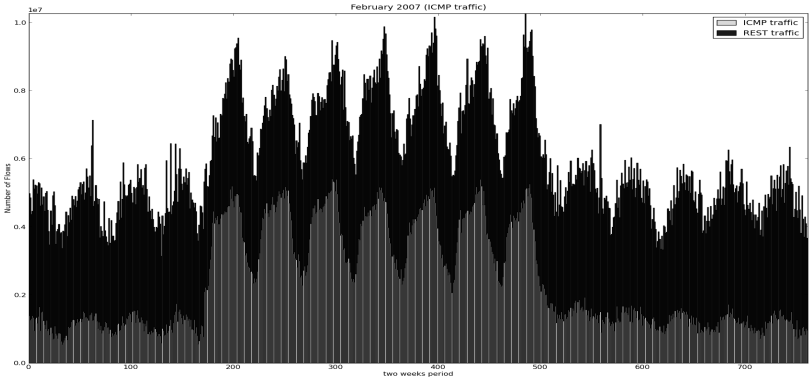

Figure 5.5: ICMP traffic on Feb 2007

5.4.2 Graph2a: In comparison with telescope (in total)

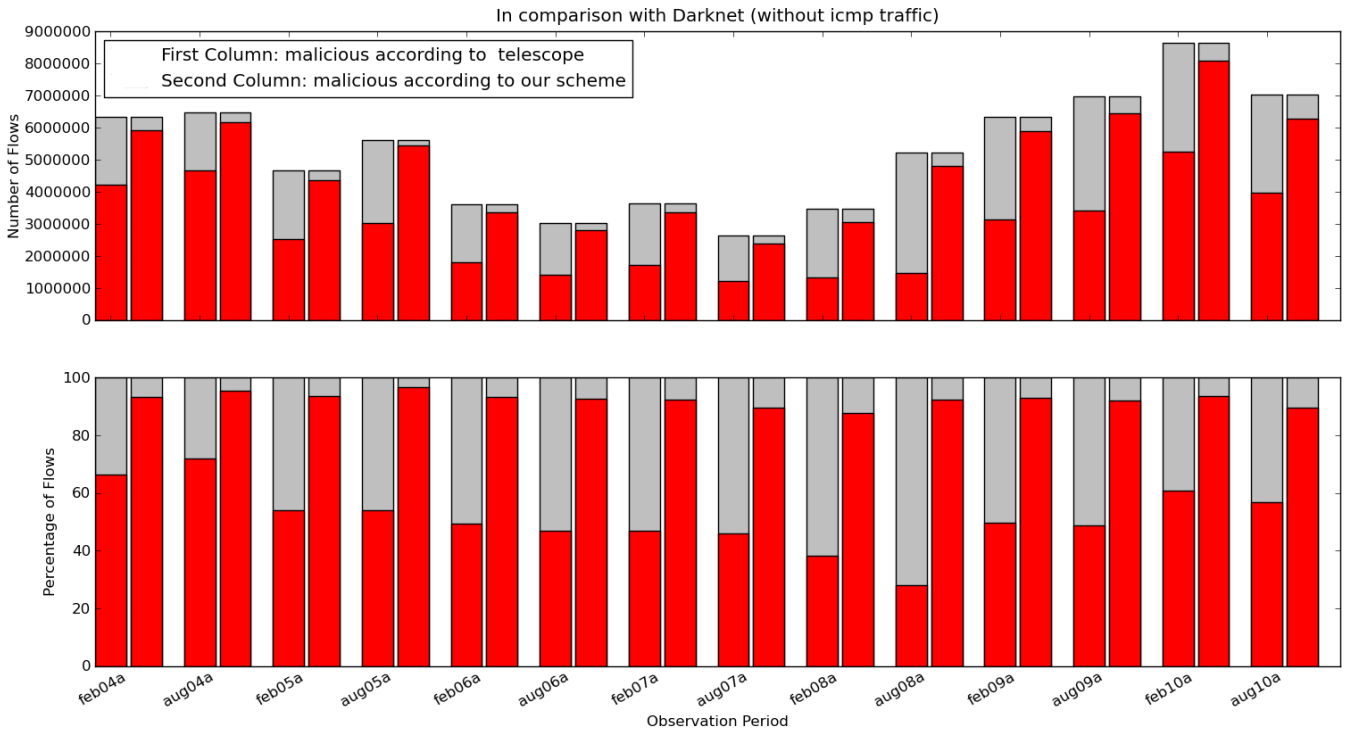

Figure 5.6: In comparison with telescope

- Traffic to dark net is proved to be malicious.
- At least 51% of inflows are malicious wrt telescope.
- At least 91% of inflows are malicious wrt our scheme.
- In Aug 2008 we observe the lowest percentage.
- According to our scheme the last years there is a small reduction of the malicious percentage.

5.4.3 Graph2b: In comparison with telescope (per category)

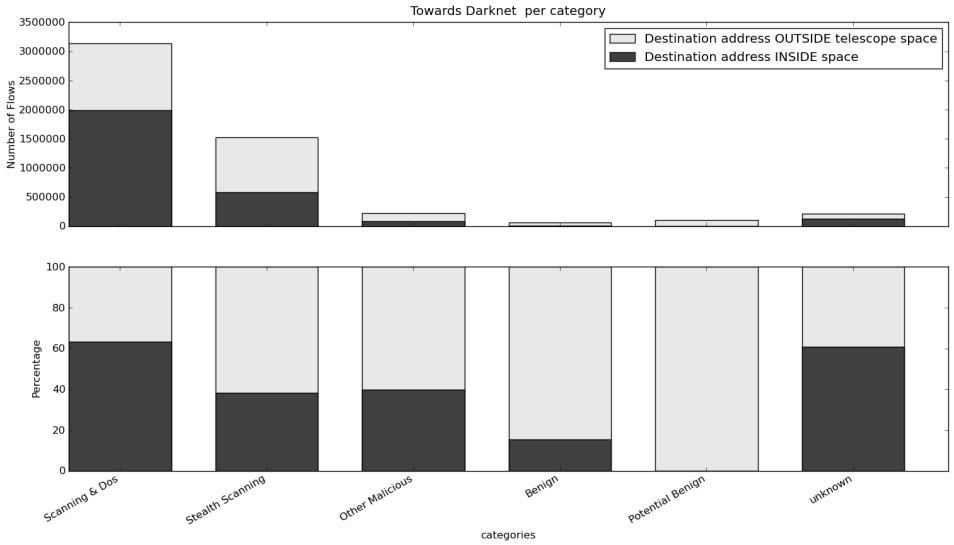

Figure 5.7: Per category

The number of flows which our scheme has categorized as Scanning & DoS, and thus malicious, are around 3 million. 2 million of them (63%) are targeted at an unused IP address and thus they are indeed malicious flow. The fact that in our scheme we have 90 thousand benign flows from which 10 thousand belongs to network telescope shows that the benign filter have big ratio of false positive, nevertheless these flow is a small percentage (only 0.001%) and we could consider them misconfiguration)

5.4.4 Graph3: Classification INSIDE telescope

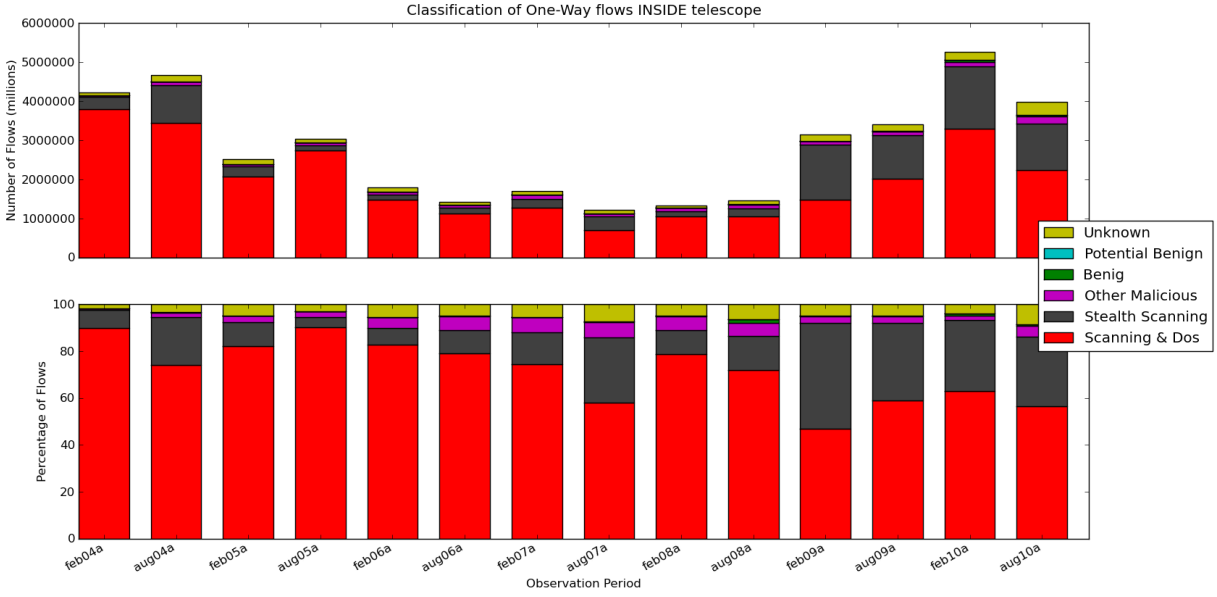

Figure 5.8: INSIDE

- 93% of flows towards telescope are captured by the three malicious filters

5.4.5 Graph4: Classification OUTSIDE telescope

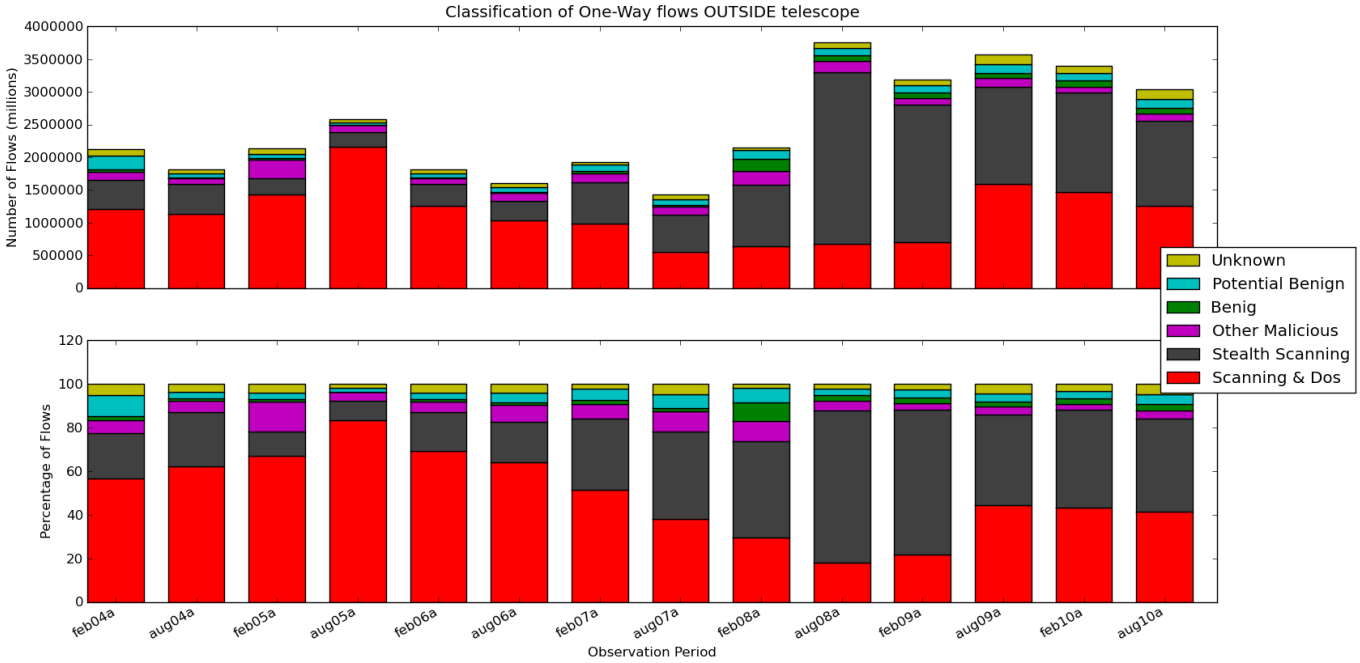

Figure 5.9: OUTSIDE

Remarks:

- in August 2008 we observe a significant percentage of stealth scanning . A potential explanation might be the fact that this period the Olympics took place at China.
- Scanning and DoS criteria taken from literature and we are convinced that these flows are malicious
- We should examine more what traffic is inside Stealth, Malicious and Unknown categories.

5.5 Frequency Item Set Mining

Mining Outside Telescope

Observing that the categories Stealth Scanning, Other malicious, and Unknown is quite significant portion and that the criteria used were not validated by a literature reference, we applied Frequency ItemSet Mining algorithms in order to acquire a better insight.

In particular, we adjust our program in order to save in files all the flows from the above categories. The time window was 3 hours. We searched for maximal itemset with minimum support (frequency) greater than 7-10 %.

The output format is the following:

```
LIP130.60.113.126,LPORT445,RIP1.1.1.11,RPORT4581,TCP,192BYTES,4PACKETS,3293msec
LIP130.60.178.106,LPORT445,RIP1.1.1.11,RPORT2581,TCP,96BYTES,2PACKETS,7msec
LIP130.92.94.118,LPORT445,RIP1.1.1.11,RPORT1307,TCP,192BYTES,4PACKETS,3532msec
LIP130.125.224.65,LPORT445,RIP1.1.1.11,RPORT3908,TCP,96BYTES,2PACKETS,40msec
LIP147.87.80.72,LPORT445,RIP1.1.1.11,RPORT1540,TCP,192BYTES,4PACKETS,3691msec
LIP147.88.127.9,LPORT445,RIP1.1.1.11,RPORT3901,TCP,96BYTES,2PACKETS,3008msec
```

where LIP stands for Local IP and in our case is the destination IP address. Whereas LPORT stands for Local port and it is the destination port of the flow.

Mining Stealth Scanning flows

Mining the flows that

- target to a host inside SWITCH network (inflow),
- the destination host does not belong to the telescope,
- the source host was not identified as a scanner

we have the following frequent itemsets

Sample	Frequent Itemsets	Comment
feb07	(405744 flows) • LPORT37 LIP129.132.2.21 (10.8%) • 3PACKETS TCP (10.6%) • 1512BYTES 2PACKETS UDP (14.0%) • 1PACKETS 0msec UDP (18.4%) • LPORT1026 2PACKETS UDP (23.7%) • TCP 2PACKETS (12.8%)	Port 37 is the NTP protocol and the IP 129.132.2.21 a NTP server Swisstime.ethz.ch and swisstime.ee.ethz.ch. The rest do not indicate benign use. (benign 10%- malicious 90%)
aug07	(3507907 flows) • 3PACKETS TCP (10.1%) • LIP129.132.2.21 LPORT37 TCP (10.4%) • 2PACKETS TCP (12.6%) • 2PACKETS UDP (16.6%) • 1PACKETS 0msec UDP (33.0%)	Again the NTP service is a significant portion. Here also is TCP protocol thus might mean that the NTP server was unavailable. (benign 10%- malicious 90%)

feb08	(6601202 flows) • 3PACKETS (10.8%) • TCP (11.3%) • 2PACKETS UDP (13.1%) • 46BYTES LPORT4246 UDP 1PACKETS(13.9%) • LIP82.130.102.218 LPORT4246 UDP (16.7%) • LIP82.130.102.161 LPORT4246 UDP (21.3%) • 32BYTES LPORT37 LIP129.132.2.21 1PACKETS UDP (24.5%)	Again UDP request to NTP server 129.132.2.21 creates a significant percentage. The destination port 4246 is known to be used as listening port of an emule Server. (benign 65%- malicious 35%)
aug08	(10880868 flows) • 760BYTES 10PACKETS RPORT123 LIP129.132.2.21 LPORT123 UDP (13.9%) • 2PACKETS UDP (14.4%) • 46BYTES LPORT37 LIP129.132.2.21 UDP 1PACKETS 0msec (26.9%)	(benign 40% - malicious 60%)
feb09	(12473013 flows) • LPORT37 LIP129.132.2.21 UDP (10.6%) • 4PACKETS (11.1%) • LPORT445 TCP (13.7%) • 1PACKETS 0msec LIP129.132.2.21 UDP (11.4%) • 2PACKETS UDP (16.8%) • RPORT123 LPORT123 LIP129.132.2.21 UDP (41.9%)	port 445 can be considered malicious. (55% benign - 45% malicious)
aug09	(10109514 flows) • 96BYTES 2PACKETS TCP (12.6%) • 4PACKETS TCP (12.6%) • 46BYTES LPORT37 LIP129.132.2.21 UDP 1PACKETS (12.3%) • LPORT445 2PACKETS TCP (13.8%) • 2PACKETS UDP (24.4%)	(10% benign 90%malicious)
feb10	(8161185 flows) • 96BYTES LPORT445 2PACKETS TCP (12.7%) • 4PACKETS (14.0%) • LPORT123 LIP129.132.2.21 UDP (13.7%) • 46BYTES LPORT37 LIP129.132.2.21 UDP 1PACKETS(18.1%) • 2PACKETS UDP (20.0%)	(32% benign - 68% malicious)
aug10	8161185 flows • 96BYTES TCP 2PACKETS (12.3%) • 4PACKETS (14.7%) • 1PACKETS 0msec UDP (15.7%) • 92BYTES LPORT37 LIP129.132.2.21 UDP 2PACKETS (19.1%) • LPORT445 TCP 2PACKETS (13.9%)	(20% benign -80 %malicious)

In our classification scheme should include a rule as far as the NTP traffic is concerned, a filter for the port445/tcp, as well as for DNS protocol. On overall, we concluded that a 30% of the stealth scanning outside might be of benign purpose, but all the traffic seems to be unproductive.

OnePacket Traffic

Mining the flows that

- target to a host inside SWITCH network (inflow),
- the destination host does not belong to the telescope,
- the source host was not identified as a scanner,
- the source host has created a bidirectional communication over the last 30 minutes
- has only one packet.

we have

Sample	Frequent Itemsets	Comment
feb07	(735047 flows) • RIP62.2.17.60 LPORT53 UDP (10.5%) • LIP129.132.2.21 LPORT123 76BYTES UDP (10.4%) • TCP (12.7%)	There is some benign use from DNS and NTP protocol. 20% benign
aug07	(644562 flows) • TCP (10.2%) • 76BYTES UDP (10.8%) • RPORT31415 60BYTES UDP (17.7%) • LPORT53 UDP (22.6%)	22.6% benign%
feb08	(1407070 flows) • LPORT53 UDP (7.9%) • TCP (8.5%) • LPORT4246 46BYTES UDP (10.7%) • LIP82.130.102.218 LPORT4254 46BYTES UDP (17.1%) • LIP82.130.102.161 LPORT4254 46BYTES UDP (21.6%)	According to the http://forum.emule-project.net the IP addresses 82.130.102.218, 82.130.102.161. where declared as fake emule server, the message posted at Posted 10 October 2007 - 10:02 PM.
aug08	(819847 flows) • 60BYTES (7.4%) • LPORT123 76BYTES UDP (7.8%) • LPORT37 LIP129.132.2.21 46BYTES UDP (8.6%) • LPORT53 UDP (32.3%)	45% benign
feb09	(1223353 flows) • LIP130.59.1.80 LPORT53 UDP (10.2) • 76BYTES UDP (11.2) • 46BYTES (11.2) • TCP (12.6)	10% benign
aug09	(957625 flows) • LPORT123 76BYTES UDP (5.1%) • TCP 46BYTES (13.7%) • 46BYTES UDP (5.1%) • LPORT53 UDP (33.0%)	40% benign

feb10	(782506 flows) • RPORT80 TCP (10.4%) • 76BYTES UDP (10.8%) • 46BYTES (10.9%) • 48BYTES TCP (11.4%) • LPORT53 UDP (45.6%)	45% benign
aug10	(931460 flows) • TCP (10.3) • LIP130.223.4.6 LPORT53 UDP (11.9) • LIP130.223.8.21 LPORT53 UDP (12.0)	20% benign

Unknown Traffic

Mining the flows that

- target to a host inside SWITCH network (inflow),
- the destination host does not belong to the telescope,
- the source host was not identified as a scanner,
- the source host was have created a bidirectional communication over the last 30 minutes
- the flow consist more that one packet.
- the flow does not have intrinsic benign attributes (e.g size more than 10kB)

we have

Sample	Frequent Itemsets	Comment
feb07	(258615 flows) • RPORT80 TCP (7.4%) • LPORT80 TCP (7.6%) • LPORT445 TCP (7.7%) • 96BYTES TCP 2PACKETS (8.0%) • LPORT5180 LIP128.178.50.170 TCP (8.3%) • 9024msec 3PACKETS TCP (8.2%) • 8960msec 3PACKETS TCP (8.7%) • 288BYTES 6PACKETS TCP (9.0%) • 4PACKETS TCP (9.0%) • 144BYTES 3PACKETS TCP (24.2%) • UDP 2PACKETS (13.7%)	• Source port 80 looks for backscattering • destination port 80 looks for a web retry • destination port 444/tcp is the known MS window Service protocol • From Symantec, "Backdoor.Peeper is a Trojan horse that allows its creator to control an infected computer. By default, it listens on TCP port 5180" the page was last updated at 13 Feb 2007 • a 40% carries tcp traffic with size/packet 48 bytes 20% Malicious - 10% Benign
aug07	(256876 flow) • 92BYTES 2PACKETS (7.9%) • 288BYTES 6PACKETS TCP (8.1%) • 4PACKETS (10.1%) • LPORT53 UDP (10.3%) • LIP192.33.214.12 2PACKETS UDP (10.8%) • LPORT25 TCP (11.4%) • 144BYTES 3PACKETS TCP (17.7%) • 2PACKETS TCP (11.7%)	• 10% DNS traffic • 12% SMTP protocol • 10.8% NTP (192.33.214.12 is an NTP server) • 26% 48bytes per packet 30% benign

feb08	(291994 flows) • LIP128.178.50.170 LPORT5180 TCP (9.3%) • LPORT53 UDP (10.4%) • 6PACKETS TCP (10.4%) • 4PACKETS (13.3%) • 144BYTES 3PACKETS TCP (19.8%) • 2PACKETS UDP (18.0%) • 2PACKETS TCP (12.5%)	• DNS • 20% 48bytes per packet • Backdoor.Peeper again on the same host 10% Malicious - 10% Benign
aug08	(549417 flows) • 144BYTES 3PACKETS TCP (10.0%) • LPORT25 TCP (15.9%) • 6PACKETS TCP (14.4%) • LPORT53 2PACKETS UDP (20.5%)	A 35% seems benign
feb09	(848250 flows) • 96BYTES TCP 2PACKETS (10.2%) • 6PACKETS (12.4%) • 3PACKETS (12.8%) • 4PACKETS TCP (10.8%) • LPORT445 TCP (18.3%) • LPORT53 UDP 2PACKETS (25.7)	25% benign, 18% Malicious
aug09	(1113320 flows) • 288BYTES 6PACKETS TCP (10.3%) • 192BYTES 4PACKETS TCP (11.6%) • 3PACKETS TCP (10.3%) • 96BYTES LPORT445 2PACKETS TCP (12.1%) • 4PACKETS LPORT445 TCP (10.9%) • LPORT53 UDP 2PACKETS (17.0%)	17% benign, 23% Malicious (20% 48BYTES/packet)
feb10	(990953 flows) • 192BYTES 4PACKETS TCP (11.0%) • LPORT25 TCP (12.3%) • 96BYTES LPORT445 2PACKETS TCP (11.2%) • 6PACKETS TCP (13.8%) • 4PACKETS LPORT445 TCP (11.5%) • 3PACKETS TCP (14.5%) • LPORT53 UDP 2PACKETS (26.7%)	35% Benign 20% Malicious (11% 48 bytes/packet)
aug10	flows • 6PACKETS TCP (11.4) • 192BYTES 4PACKETS LPORT445 TCP(11.4) • 96BYTES LPORT445 2PACKETS TCP(12.4) • 142BYTES LPORT53 UDP 2PACKETS (19.1)	20% Benign 23% Malicious

From the above results, we observed that there should be added a rule about port445/tcp because flows to that port are a significant portion of the unknown traffic. What is more, we observed tcp flows with

48 Bytes per packet, this might be an acknowledgement packet[4] but it is not safe if we not first filter out port445/tcp as it has the same size per packet (itemset 192BYTES 4PACKETS LPORT445 TCP(11.4) from Aug2010). Further, a static rule for the protocols DNS (udp at port 53), SMPT (tcp at port 25) and NTP may also help to further shrink the class unknown.

On overall conclusion

As result of the above manual analysis, we could say that a 30% of the flows classified as Stealth Scanning might be benign, this is significant percentage of the total flows (7%). A 25% of one packet might be benign but the number of these flows are very small in total (1%). As far as the Unknown flows is concerned, we concluded that a $\sim$ 24% might be benign and a $\sim$ 13% might be malicious.

Mining inside telescope

During the mining procedure, we looked into the flows that are going to network telescope to verify that they are indeed malicious traffic. The results are given to the following tables and are satisfactory.

Stealth scanning

Flows	%	Comment
5256156	100%	Total flows captured by the Stealth scanning filter
4030374	76%	DstPORT=445/tcp (MS Window Services)
345461	6.5%	Total flows with one packet (281668 is UDP traffic)
879375	16%	Unknown Flows

One-Packet scanning

Flows	%	Comment
529291	100%	Total flows with 1packet which escaped other filtered
310714	59%	Flow (srcPORT=80/tcp) that created by an IP located in China
79436	15%	Flows with srcPORT=80 (backscattering)
14386	3.5%	DstPORT=445/tcp (MS Window Services)
14338	3.5%	TCP protocol
27959	5.2%	DstPORT=53
18582	3.5%	Originating from a planet lab node at Bern
55290	1%	Unknown

Unknown traffic

Flows	%	Comment
968360	100%	Total flows classified as unknown
430098	44%	DstPORT=445/tcp (MS Window Services)
204646	21%	Flows with srcPORT=80 (backscattering)
80931	8%	IP from China with srcPORT10008
56651	5.8%	DstPORT=53
62698	6.4%	A known bot agent?
133336	14%	Unknown

[4] 40 bytes is the minimum size of a TCP packet, no options, these are usually ACKs or RSTs, 52 byte packets are ACKs that include timestamps and 48 byte packets are SYN packets setting up a connection

Chapter 6

Conclusion

Taking everything into account we conclude that every 30 minutes, 7.5 million flows are reaching SWITCH network which don't receive an answer. According to our study:

- malicious 81%
- benign 7.5%
- unknown 1.5%
- ICMP traffic 10%

The procedure with wich we computed the above values is shown at the figure 6.1

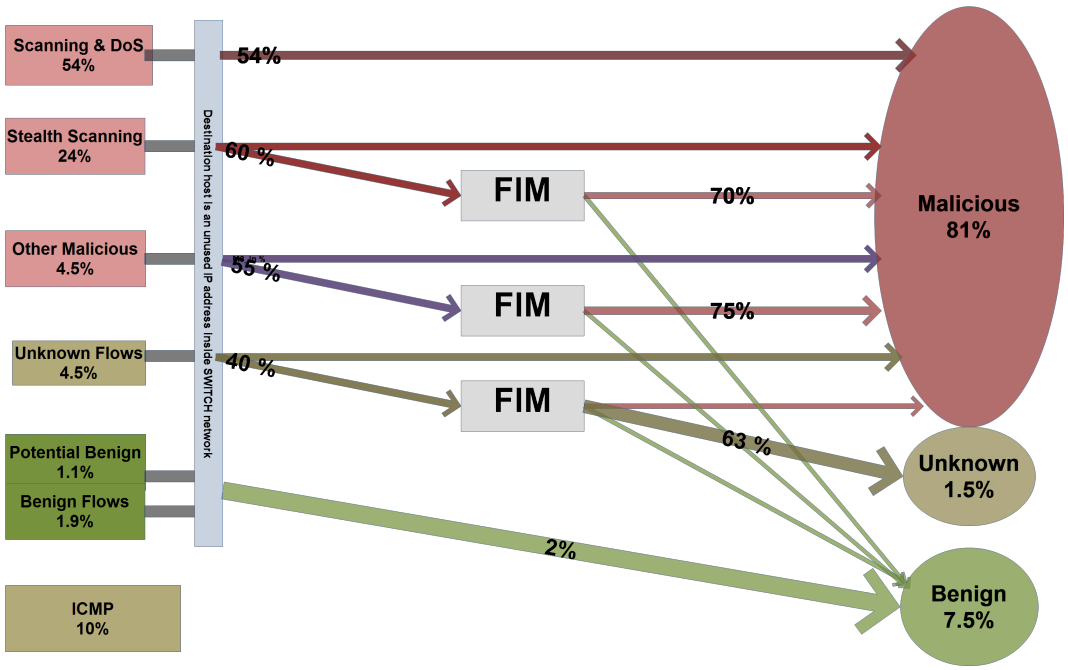

Figure 6.1: The overall conclusion

6.1 Future work - Improvements

- There should be more manual analysis using data frequency itemset mining algorithms. This might help to extract meaningful static and time invariant rules which will classify a significant portion of the one-way flows. Nevertheless, the analysis should be made after we filter out the known malicious traffic in order to scale and to be more efficient. Our implementation will be of great use in that end.
- Increase the processing window from 30 minutes to an hour. These will help firstly to increase the credibility of the criteria which are based to the aggregated behavior and also will decrease significantly the execution time. The impact of processing window over the performance is shown to the figure 6.2

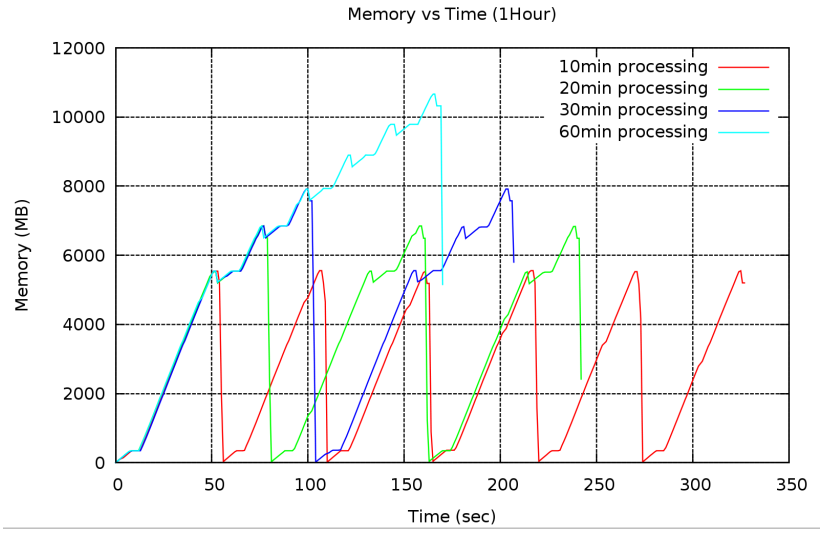

Figure 6.2: Impact of processing window to the performance

However, an increase to the window should be also accompanied with a threshold adjustification.

- Another idea is to base the classification on the destination host behavior. In this project, we recognise without high accuracy destination addresses that are web browser or peers of a peer-to-peer application. A better approach will be to characterize every host inside SWITCH network according to how many biflows/outflows creates and how many inflows he received along with a tag like web server, DNS server etc. This, without doubt, will help to have a further indication of whether the flows are malicious or benign/productive, what is more, since we will investigate only the IP addresses inside SWITCH which create traffic, it probably will scale[1].
- A more complicate but more valid idea is to classify the flows by examining the communication history between communication pairs. This demands significant programming work and might also not scale.
- In our analysis, we had excluded from the beginning the ICMP traffic because we considered them false positive and also small portion of the total traffic. Nevertheless, the results showed that this traffic has significant percentage and should be included to the analysis.

[1] We expect to have around 800 thousand IPs and should be able to recognise all servers dns server, mail server, ntp server

Bibliography

[1] Mark Allman, Vern Paxson, and Jeff Terrell. A brief history of scanning. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 77–82, New York, NY, USA, 2007. ACM.

[2] Richard J. Barnett and Barry Irwin. Towards a taxonomy of network scanning techniques. In Reinhardt A. Botha and Charmain Cilliers, editors, *SAICSIT Conf*, volume 338 of *ACM International Conference Proceeding Series*, pages 1–7. ACM, 2008.

[3] Saikat Guha, Jaideep Chandrashekar, Nina Taft, and Konstantina Papagiannaki. How healthy are today's enterprise networks? In Konstantina Papagiannaki and Zhi-Li Zhang, editors, *Internet Measurement Conference*, pages 145–150. ACM, 2008.

[4] Wolfgang John and Sven Tafvelin. Heuristics to classify internet backbone traffic. 2006.

[5] Jaeyeon Jung, Vern Paxson, Arthur W. Berger, and Hari Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *IEEE Symposium on Security and Privacy 2004*, Oakland, CA, May 2004.

[6] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of P2P traffic. In *4th Internet Measurement Conf. (IMC)*, pages 121–134, October 2004.

[7] DongJin Lee and Nevil Brownlee. Passive measurement of one-way and two-way flow lifetimes. *SIGCOMM Comput. Commun. Rev.*, 37(3):17–28, 2007.

[8] kc claffy Nevil Brwnlee. Network telescope traffic: What can we see today.

[9] Marcell Perényi, Trang Dinh Dang, András Gefferth, and Sándor Molnár. Identification and analysis of peer-to-peer traffic. *JCM*, 1(7):36–46, 2006.

[10] Ryan Trost. *Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century.* Addison-Wesley Professional, 1st edition, 2009.

[11] Alif Wahid, Christopher Leckie, and Chenfeng Zhou. Characterising the evolution in scanning activity of suspicious hosts. *Network and System Security, International Conference on*, 0:344–350, 2009.

[12] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. Internet background radiation revisited. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 62–74, New York, NY, USA, 2010. ACM.