University of Twente
Enschede - The Netherlands

University of Twente
Department of Electrical Engineering, Mathematics an Computer Science
Chair for Design and Analysis of Communication Systems

Detecting UDP attacks using packet symmetry with only flow data

by

Daan van der Sanden

I.O.O. report
Executed from September 2007 to Juli 2008

Supervisors: Dr.ir. A. Pras
Dr. R. Sadre
A. Sperotto M.Sc.

Detecting UDP attacks using packet symmetry with only flow data

Daan van der Sanden

July 7, 2008

Abstract

Attacks on the Internet are becoming a bigger problem since more users, companies and even complete societies rely on the correct functioning of the Internet. Existing packet inspection based monitoring systems do not scale well with the enrollment of Gbps speed network technology. This paper will show that with the use of flow based data and packet symmetry, UDP floods can be detected and the presence of "background" noise on the internet can easily be shown. The contribution of this paper is that it shows that even for large Gbps networks the symmetry in UDP packets with flow data is a good metric to detect attacks using UDP. The analysis of two real high-speed networks with Gbps links supports this statement.

1 Introduction

The Internet is growing: more people subscribe via broadband connections and networks speeds are increasing. The backside of this growth is the increasing dependency on the Internet. Because of this dependency, the misuse of the Internet will have a larger the impact on society. Examples of such misuses or attacks are port scans and Denial of Service (DoS) attacks. To detect these attacks most network operators use Network Intrusion Detection Systems (NIDS) on their network.

Most NIDS these days work with deep packet inspection. This gives problems when the network speed increases to 10 Gbps or higher, which are getting more common these days. Since those NIDS do not scale well to those speeds, another approach that is more scalable on high-speed network links is needed. One method analyzes aggregated data of the traffic in the form of flow information. The *de facto* standard at the moment for this is Ciscos NetFlow v5 and v9. The IETF IPFIX [2] working group is also working on standard for flow aggregation, which is based on NetFlow v9.

A flow is identified as a unidirectional stream of packets between a given source and destination. Specifically, a flow is identified as the combination of the following five key fields: source IP address, destination IP address, source port number (if applicable), destination port number (if applicable) and the layer four protocol type. With these unique five keys, some other information is stored for each flow, for example, when the first or last packet of a flow is switched/routed and how many bytes and packets are sent. It is also possible to store routing information in the flow records. To reduce the burden on the router, which aggregates the packets into flow information, it can be configured to sample the packets; one out of every n packets will be considered for storage in the flows.

Some research today focuses on detecting attacks in TCP traffic using packet inspection. Not much research is done at anomalies in UDP traffic yet. Administrators in the field have the same feeling we had, that attackers nowadays use the UDP protocol more than the TCP protocol for DoS attacks.

Two NIDS that are freely available on the web are SNORT [6] and BRO [5]. These NIDS use deep packet inspection to match against rules to detect intruders. They also use signature based

scanning to identify known viruses, worms and malware. Other research focuses on detecting attacks from the source [4, 3] where they try to stop a DoS attack before it even started. The systems decide, based on different metrics, if a host participates in an attack. If they do, they will be limited in the amount of bandwidth they are allowed to use. Lots of research is done on methods and detection mechanisms to identify suspicious traffic. Usually they are design to detect one specific attack, for example in [7] different metrics for detecting different DDoS attacks are investigated, while [8] focusing on detecting port scans.

The metric that [3] uses the symmetry between incoming and outgoing packets to differentiate legitimate from malicious traffic. It argues that if the flow is legitimate that the number of sent and received number of packets in the traffic is symmetric. MULTOPS [1] uses a similar approach, where the packet rates are compared and evaluated. However, the paper on MULTOPS focuses more on how to effectively store the ratios instead of analyzing the metric.

In this paper we will investigate how the *packet ratio* performs when we only have flow data available. The packet ratio is based on the principle that there is symmetry between the number of sent and received packets. It is defined as the number of received packets over the number of sent packets. This papers answers the following questions to get a better idea of the behavior of the packet ratio.

1. Is it (in principle) possible to use the packet ratio for detecting UDP attacks using only flow data?
2. At what granularities can packet symmetry be used to detect UDP attacks?
3. Is it also possible to use sampled flow data?
4. What UDP attacks are found in flow data from real networks?

To answer the first question a tool will build that calculates the packet ratio from the flow data. To validate the tool, it analyses a flow data set with several attacks in them. To answer the other two questions the tool analyses one week of flow data from two high speed networks, one flow set from the University of Twente (UT) and a sampled flow set from the universities Internet service provider SURFnet[1] and investigate what was going on during that week. The second, sampled, data set is also used to answer the second question regarding the sampling of the flow data.

This paper has the following organization: Section 2 describes the packet ratio, the metric that is investigated in this paper, in more detail. Section 3 discusses the tool that we developed to analyze the metric. In addition, the types of analysis that will be done on the data sets are described in Section 3. Section 4 gives a short explanation on how we validated the tool that implemented the metric, by using a reference data set. Section 5 presents the results of the described analysis on the real network data. Some found anomalies are also analyzed. In Section 6, we will present our conclusions.

2 Method

In [3] a method of selecting malicious traffic from legitimate traffic was proposed based on the packet symmetry. We will use the same metric to detect attacks in UDP traffic. Equation 1 is the metric proposed in [3], where tx is the number of sent packets and rx the number of received packets per unit of time.

$$S = \log_e \left(\frac{tx + 1}{rx + 1} \right) \tag{1}$$

We will use a slightly different format but the basic principle stays the same: the number of transmitted and received packets should be symmetric. Instead of using a logarithmic function

[1] SURFnet is the academic research network of the Netherlands.

where perfect symmetry in traffic will result in S having a value around zero, we just look at the ratio between the received packets and the sent packets. Because we drop the logarithmic function, we do not need to add any numbers to the numerator or the denominator. This addition introduced an unwanted error in the ratio of hosts not sending many packets. Since we are interested in attacks toward the network, we decided the received number of packets to be in the numerator instead of the denominator as presented in Equation 2. If the number of transmitted packets is zero, R_{UDP} will be defined as a large number.

$$R_{UDP} = \begin{cases} \frac{rx}{tx} & tx > 0 \\ M & tx = 0 \end{cases} \tag{2}$$

The packet ratio has one downside. When an attacker uses a host for a reflective attack and uses it as an amplifier the metric will not work. It is known that DNS servers are used for these kinds of attacks [9]. However, one should never rely on one metric to identify attacks. An alternative metric that will work in the case of a reflective attack is the byte ratio as proposed in [7], which can be used together with the packet ratio. This is quite easy to do with our tool, but will not be discussed in this paper.

In [3] multiple granularities were investigated and it was believed that host-to-host symmetry was a good starting point for further investigation. Because the data set of SURFnet we analyze, we expect the number of host-to-host ratios to become too large to analyze them easily. The smallest granularity that we will analyze is the packet ratio at host level.

3 Setup

For the analysis on the real network data, exported flow information from the network of the University of Twente and the network of SURFnet are used. Almost all traffic from and to the University of Twente goes via the network of SURFnet. The flow records were captured over one week of time between 27 July 2007, 0:00 until 3 August 2007, 0:00. For the SURFnet data set, the sampling mode in the routers was set to sample 1-to-100.

The build tool performs a few simple steps. In the first step, the tool filters out all NetFlow records with information of UDP unicast traffic. The tool stores the number of incoming and outgoing UDP packets and the number of incoming and outgoing octets per host of the network2 in a database. This is done in constant time bins starting at fixed times. If a flow overlaps two time bins it is split into two parts, proportional to the time the flow overlapped the time bins. When splitting up the flow we assume that the number of packets and bytes were uniformly distributed during a flow. The size of the time bins is chosen the same size as the active timer in the routers. For the University of Twente data set, this is one minute and five minutes for the SURFnet data set.

Two different analyses are performed on the data sets, a time series analysis and a statistical analysis. With the time series analysis, the UDP packet ratio of the whole network is calculated for every time bin and plotted over time. This analysis will be used to validate our developed tool. In order to answer the research question about the granularities, a statistical analysis is done. We make a Cumulative Percentage Plot for the ratios in the database for different granularities. The two granularities we will test is the size of the time bins and the number of hosts that are aggregated. By changing these values and investigating their impact on the cumulative percentage plots, we can gain a better insight how the ratio functions at different granularities.

4 Validation

To test the validity of the implementation of the method in our tool, a dataset with flow data of the University of Twente, that was captured during another project, was analyzed. During this project a honeypot on the university was set up to monitor the behavior of “hackers”. During

this project the "hacked" honeypot was used for various UDP floods on several hosts outside of the university. In order to test our tool we match the timestamps of the log file from the honeypot with the results of a time series representation of the UDP packet ratio.

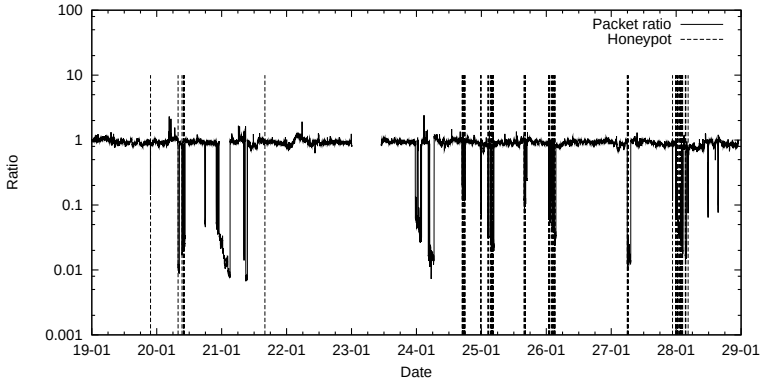

Figure 1: Time series representation of the packet ratio of the reference data set with the times marked when attacks were started. The gap on 23 January is a result of failure in the collector.

The result of the time series analysis is displayed in Figure 1. We marked the times when an attack was started from the honeypot with a dashed vertical line. Because of the way the honeypot was set up, the exact end times were not possible to determine. Figure 2 displays the packet ratio and the start times of the attacks from the honeypot during a small percentage of time.

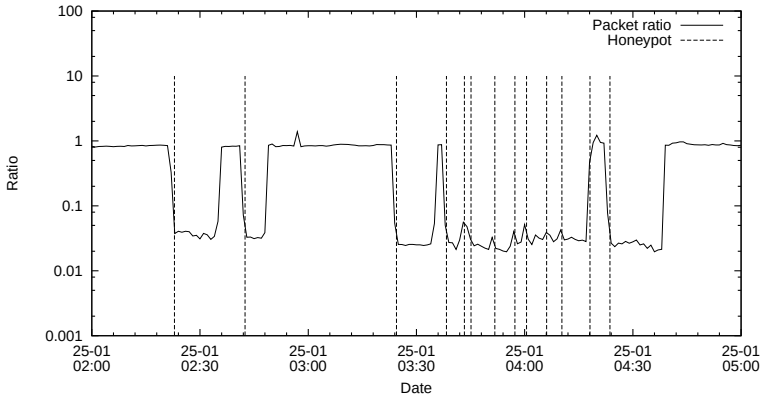

Figure 2: Zoomed in version of Figure 1, specifically the morning of 25th January to have a better view of the ratio in comparison with the times marked as the start of attacks.

The number of attacks that were initiated from the honeypot that were found in the log files was 68. With the designed tool using the packet ratio, 66 time bins, which were reported by the honeypot, were marked as time bins in which an attack took place. The two attacks that were missed only lasted for several seconds. We were using time bins of one minute; those short attacks did not contribute enough to disrupt the symmetry in UDP packets. Figure 1 shows that there are still several attacks (that also originated from the honeypot) that were detected by our tool, but that were not marked as start times of an attack according to the log file of the honeypot. At the moment of writing this paper, the honeypot is still analyzed to see how these other attacks were initiated.

	Sent packets	*Received packets*	*Sent bytes*	*Received bytes*
UT	3 894M	4 850M	1 014 GB	587 GB
SURFnet	38 080M	34 620M	14,1 TB	5,7 TB

Table 1: The number of sent and received UDP packets and bytes during the one week of capture. The number of packets are not

5 Real network analysis

The previous section showed that our tool is capable of detecting UDP floods in a flow data set. In this section, we use our tool to analyze one week of real traffic. In this section we define an active host as a host that sent at least one UDP packet during the week that was monitored. In the case of the UT (a /16 address range) there were 12 553 active hosts out of the 65 506 hosts in the data set. Because of the sampling in the SURFnet data set, such numbers cannot be given with that precision for the SURFnet data set. If we do the same analysis for all UT traffic in the SURFnet data set, 44 154 hosts were found. Of the 12 553 active in the UT data set, only 10 517 were found in the SURFnet data set. In Table 1, some basic information can be found of the two datasets. The numbers for SURFnet are corrected for the sampling by multiplying them with 100.

5.1 Time series analysis

The UDP packet ratio of the University of Twente can be found in Figure 3. In the night of 1 August, we can observe an ongoing attack towards the university. The attack consisted of 786M Packets and 39GB of data. 99,5% of the packets was exactly 46 bytes (88% of the total bandwidth consumption of the attack) and destined for port 53 (DNS). The other traffic consisted of packets of 1500 bytes sent to all possible UDP ports.

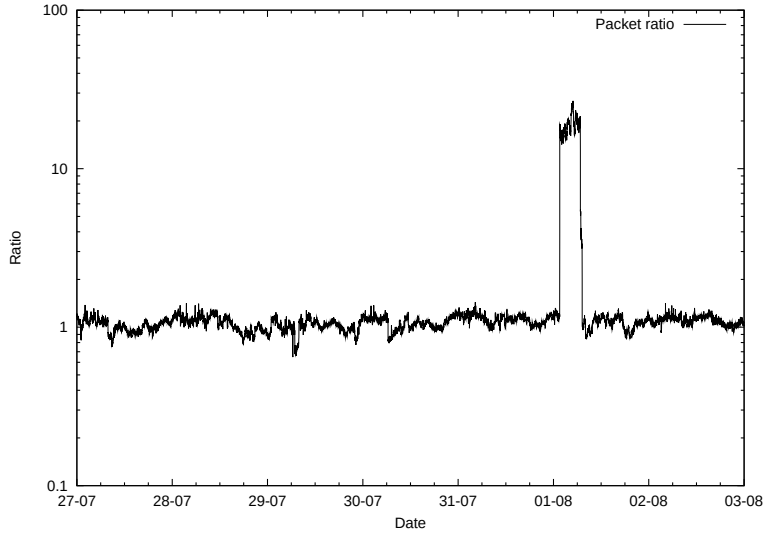

Figure 3: Time series representation of the packet ratio of the University of Twente.

When we make a time series plot of all the SURFnet traffic, we expect the attack in the night of 1 August towards the University of Twente also to become visible. Figure 4 shows the time series representation of the SURFnet data set. As we can see, the attack in the night of 1 August towards the University of Twente also became visible. The value of the ratio is not that

big in the SURFnet anymore, since there is more legitimate traffic than traffic generated by the attacks, however the ratio is still good enough to detect the three attacks. We can also identify two more attacks on hosts in the SURFnet network. One attack starting around 29 July 2007, 14:20 and the second attack starting around 31 July 2007, 23:35.The attack on 29 July lasted for about 7 minutes and 76M packets of exactly 43 bytes (3.0 GB) were used. The attack on 31 July lasted for about 9 minutes and 49M packets of exactly 43 bytes (2.0 GB) were used. Both attacks show a big similarity in behavior, but the victims were different and the attacks also originated from different sources.

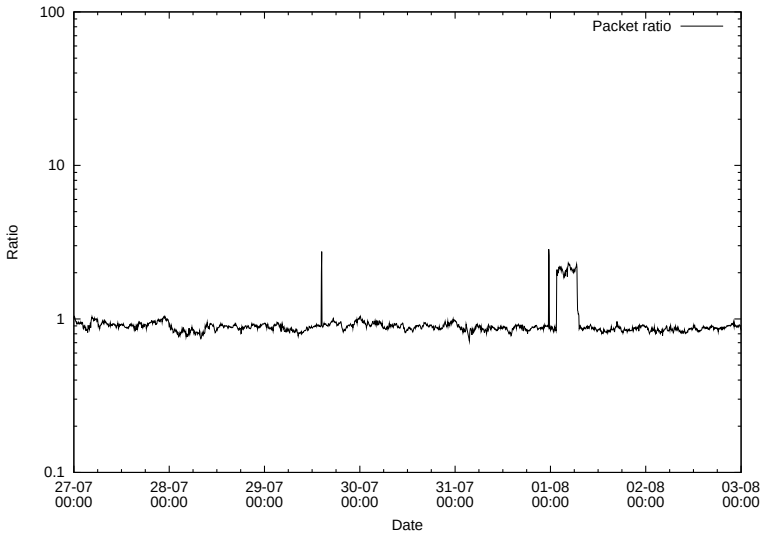

Figure 4: Time series representation of the packet ratio of SURFnet.

In order to see how sampling influences the packet ratio we consider only all UDP traffic to or from the University of Twente out of the SURFnet data set. Since for the whole network the two ratios are almost completely overlapping on a logarithmic scale, Figure 5 plots relative error percentage between the two. The influence of sampling is negligible when we only look at the overall UT ratio.

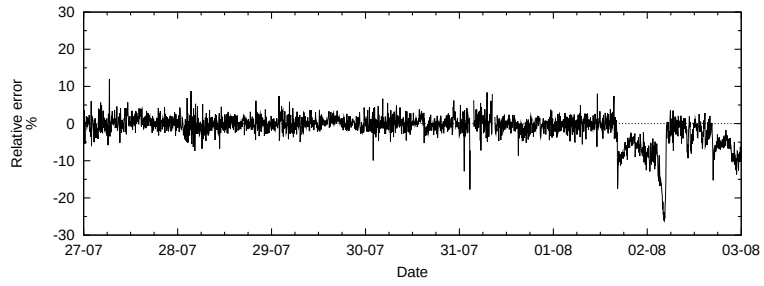

Figure 5: The relative error percentage between the sampled and unsampled ratio.

5.2 Granularity research

This Section investigates how the tool, and thus the metric, performs at different granularities. In the previous Section and in Section 4 we could see that when we consider the whole network it functions correctly at identifying flooding attacks. Now we will look at different granularities

to see how it performs. Since it is unfeasible to plot time series plots for all hosts, we plot the Cumulative Percentage. In other words, the percentage of all ratios smaller than a certain value is plot.

In Figure 6 it can be seen that when we use a host level granularity that more than 70% of the packet ratios of the UT data set are larger than 10. The University of Twente has a /16 network address range. However, in our data set we only had 12,5 thousand active hosts. All inactive hosts were visited[2] (on average) 190 times that week. That means that 55% of the ratios larger than 10 at host granularity, were because of traffic destined for inactive hosts. Because when an inactive host "receives" a packet R_{UDP} in Equation 2 will be defined as M. If we extrapolate this behavior of "background" noise to all hosts in the UT range, 47% of all ratios in the database with 1 minute time bins are as a result of this "background" noise. This can be assumed since only a few hundred hosts are active for the whole week; most of the hosts are not active constantly. A more detailed insight on the mix of this "background" noise is given in Section 5.3. This is also emphasized by the behavior of the CDF when we plot all packet ratios in our set when we use different time bins (see Figure 7). With larger time bins we see that more ratios in the database are larger than 10.

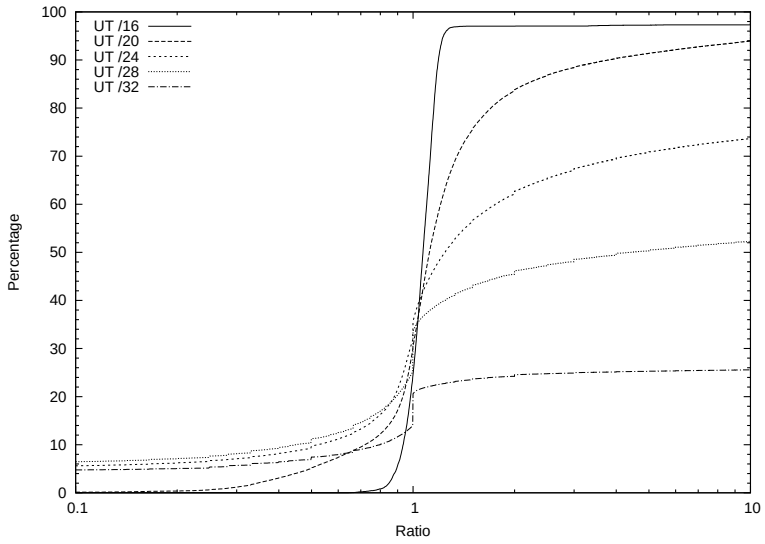

Figure 6: CDF plot of all ratios of a time window of one minute of the University of Twente. They were on a host basis (/32), and aggregated into different subnets (/28, /24, /20) and of the entire /16 network.

Figure 8 Shows that there are more ratios that are smaller than 0,1 in the SURFnet dataset, however this is expected behavior since in some valid small connections all off the reply traffic may be filtered out due to the sampling. However, we can also see that a fair amount of the background traffic is still in the dataset. When analyzing the sampled dataset we can see that 33,5 thousand of the inactive hosts are still in the data set, which were visited on average eight times during that week.

5.3 Background traffic

In Section 5.2 in all the cumulative percentage plots at host level granularity, a lot of traffic going to inactive hosts cause the plots not to distribute between 0 and 100 percent. A more thorough analysis was performed on the two days from the complete University of Twente dataset. From 1 August 2007, 00:00 and 2 August 2007, 23:59 a port distribution was made of all UDP packets

[2]During a time bin of one minute, they received at least on packet.

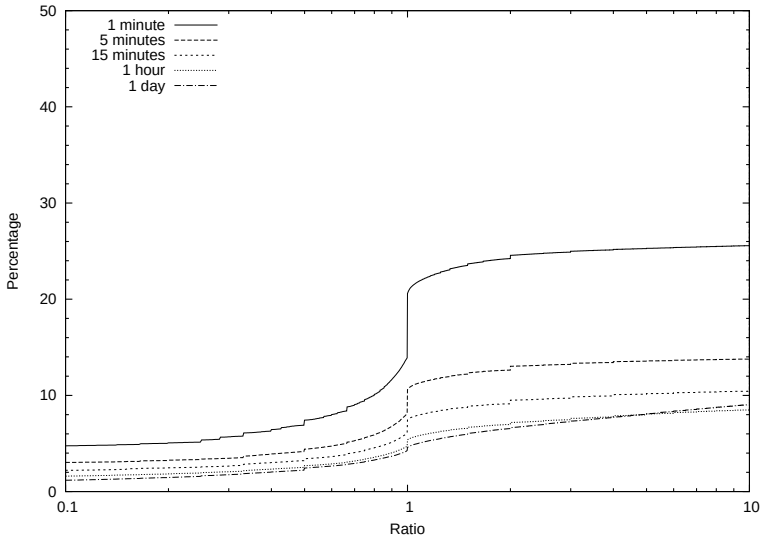

Figure 7: CDF plot of all ratios at host level granularity of the University of Twente. Done with different values for the time bin size.

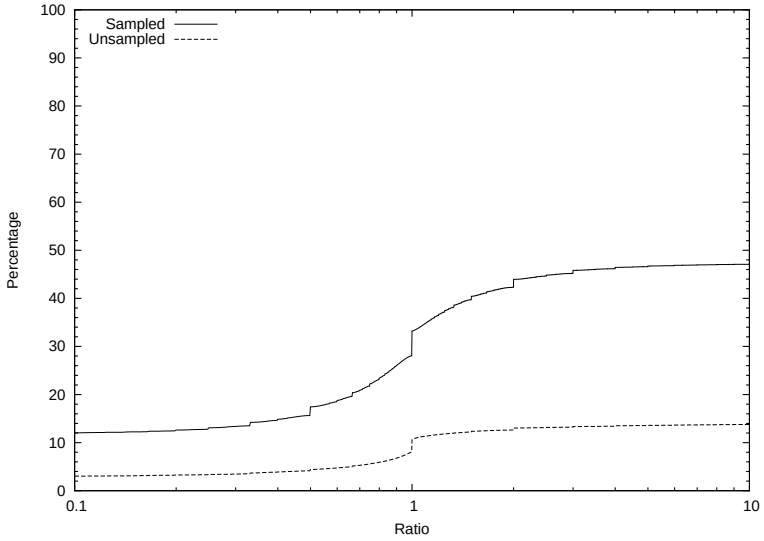

Figure 8: CDF of all ratios with a time window of five minute. From the UTwente dataset and the utwente traffic fitlered from the SURFnet dataset.

that were sent to inactive hosts. In this instance, all inactive hosts were hosts that did not send any packet (no matter what protocol) during those two days. The result of this can be found in Table 2.

Popular ports are the ports 1026/1027 also known as the ports that messenger services uses, and is misused for displaying pop up messages on the screen of the user. The other top ports belong to one destination address per port (and in the case of port 16443 two destination addresses). The connections came from many different hosts. On average around 100 different hosts per minute tried to connect. When we look at the activity of that host over the whole week, it can be seen that in the beginning of the week the host was active (and the ratio was in balance), when the host was turned off, the packets kept coming. The numbers of packets

Port	*Packets*	*Percentage*	*Extra information*
1 026	2 555 326	22,6%	Messenger Spam
1 027	1 015 623	9,0%	Messenger Spam
6 167	670 547	5,9%	One single host as the destination address
16 443	447 835	4,0%	Two hosts as the destination address
11 795	248 561	2,2%	One single host as the destination address
27 015	203 722	1,8%	Half Life server
	6 182 911	54,5%	Other port numbers
Total	**11 324 525**	**100%**	

Table 2: Distribution of packets sent to inactive hosts in exactly two days.

gradually decreased over time. Because of the great number of different hosts sending packets we think this is from a peer-to-peer application running on the hosts.

6 Conclusions

The research from Section 1 are repeated and answered in this section. Finally, some future work will be discussed on how to improve the tool by identifying some of its shortcomings.

Is it (in principle) possible to use the packet ratio for detecting UDP attacks using only flow data? In Section 4, we showed that analyzing the time series of the packet ratio is a good measure to detect UDP floods. We were able to detect 66 of 68 attacks that were initiated from the command prompt on the honeypot. We were also able to detect attacks that were not started from the command prompt. Analyzing the real data in Section 5.1 also showed that the packet ratio can be used for detecting floods, because we identified one attack in the University of Twente set. The same attack was also found in the SURFnet data set together with two more attacks.

At what granularities can packet symmetry be used to detect UDP attacks?
In Section 5.2, we look at the cumulative percentage plots at host granularity of all ratios in the database. We could see that most of the time the UPD ratio was not balanced. After further analysis it was shown that is was mostly due to scans and or spam activity (small packets to inactive hosts with big intervals). This statement is supported by the fact that when the time bins were enlarged the cumulative percentage plots got even worse. Because we investigated the behavior at host level granularity, the presence of this traffic was easily found. For flood detection, the best granularity level is to monitor bigger parts of the network at once. With the current implementation of the tool and only flow data, the method will not function on host level, because off the many inactive hosts that were contacted. For the time granularity, the choice of this value depends on the value of the active timer in the router, the higher this value, the later a flow record is exported the later we know something is wrong.

Is it also possible to use sampled flow data?
In both Section 5.1 and Section 5.2 we looked at the influence of sampling, by considering only traffic to and from the University of Twente out of the SURFnet data set and compared it with University of Twente data set. In the time series analysis of the entire University of Twente, we could see that influence of sampling was negligible.

However when we look at host granularity we can see some big differences. Some active hosts with small amounts of traffic are missed in the SURFnet data. In the different plots in Section 5.2, we can see that cumulative percentage plots are heavily influenced by sampling. This was also discussed in the first paragraph of Section 5 were we saw that only 84% of all active hosts University of Twente hosts was accounted for in the SURFnet data set. So sampled

flow data will only give good results when used at larger granularities such as a complete network for flooding detection, otherwise sampling is not desirable for intrusion detection.

What UDP attacks are found in flow data from real networks?
In Section 5.1, we saw that two of the attacks in SURFnet were relatively short (about 7 minutes) and consisted out of small packets of exactly 43 bytes. In addition, a large attack on host on the University of Twente was detected which lasted for several hours. This attack mainly consisted of small packets of 46 bytes exactly.

As shown in Section 5.3 ,there is still a lot of messenger spam send over the internet to hosts (port 1026/1027, messenger spam). It is unknown if it is actively used or just the residue of old viruses/worms still alive on the Internet. Another interesting thing we saw was the behavior of several peer-to-peer applications where after the host was turned off still hundreds of packets per minute were coming to that host after a few days from many different sources.

6.1 Future work

The ratio performs well at network granularity, to be able to use it at lower granularities such as host level granularity a method needs to be found to detect if a host is active. Therefore, for host level granularities we do not want to consider all inactive hosts. In order to this effectively a better method of defining an active host is needed. Instead of sent one packet during the whole week a shorter period can be used and instead of only looking at UDP also other protocols can be considered to define a hosts activity.

The tool is also not useable for detecting reflective attacks for which hosts inside the monitored network are used. In order to overcome this problem, it is also possible to look at the byte ratio (received bytes divided by sent bytes). Because a reflective attack is usually used to amplify an attacks bandwidth the byte ratio should be able to identify those attacks.

References

[1] Thomer M. Gil and Massimiliano Poletto. MULTOPS: A data-structure for bandwidth attack detection. In *Proceedings of the 10th USENIX Security Symposium*, August 2001.

[2] IP Flow Information Export Working Group. `http://www.ietf.org/html.charters/ipx-charter.html`, April 2008.

[3] C. Kreibich, A. Warfield, J. Crowcroft, S. Hand, and I. Pratt. Using packet symmetry to curtail malicious traffic. In *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[4] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of Ttenth IEEE International Conference on Network Protocols*, pages 312–321, November 2002.

[5] Vern Paxson. Bro: a system for detecting network intruders in real-time. In *SSYM'98: Proceedings of the 7th conference on USENIX Security Symposium*, pages 3–3, 1998.

[6] Martin Roesch. Snort - lightweight intrusion detection for networks. In *LISA '99: Proceedings of the 13th USENIX conference on System administration*, pages 229–238, 1999.

[7] C. Siaterlis and B. Maglaris. Detecting DDos attacks with passive measurement based heuristics. In *Proceedings of the Ninth International Symposium on Computers and Communications, 2004 (ISCC 2004)*, volume 1, pages 339–344, June/July 2004.

[8] Stuart Staniford, James A. Hoagland, and Joseph M. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10(1–2):105–136, 2002.

[9] US-CERT. The continuing denial of service threat posed by DNS recursion (v2.0). `http://www.us-cert.gov/reading_room/DNS-recursion033006.pdf`, 2006.