Project Darklight

Supervisor
Scott A. McIntyre

Project members
ing. M. Rave
ing. R. Buijs

29-06-2007

Universiteit van Amsterdam

Abstract

XS4ALL is a large Internet Service Provider in the Netherlands. By using a darknet, the XS4ALL Security Officer, Scott A. McIntyre, monitors traffic. A darknet captures traffic towards bogon IP addresses and not-yet allocated XS4ALL IP-space. Bogon IP adressess are IP blocks, not yet allocated by IANA and RIRs and also private or special use IP addresses.

The goal of this project is to find a method to get useful information from the darknet data streams and see if it is possible to use this as a zero day warning system. Argus is a flow monitoring tool that captures data streams to the darknet. By analyzing parts of data with use of the Argus client tools and handmade scripts, several patterns – IP-scans, portscans, time patterns, faulty assigned IP addresses – are found.

Most malicious traffic is coming from XS4ALL customers, who are sending traffic with a bogon IP address as destination and people from all over Europe sending traffic towards not-used XS4ALL address space. With the help of the darknet data, a trend analysis and a top N can be made. This can be used as a zero day warning system.

Acknowledgments

We would like to extend our gratitude to the following people and organizations for their assistance in our research.

XS4ALL for giving us the trust and resources to make this project possible.

Scott A. McIntyre for the data, support and feedback during this project.

Contents

Chapter 1

Introduction

XS4ALL is a large Internet Service Provider (ISP) in the Netherlands. XS4ALL has a good reputation in terms of security. By using a darknet they capture traffic towards bogon IP addresses and not used XS4ALL IP-space. Bogon IP adressess are IP blocks, not yet allocated by IANA and RIRs and also private or special use IP addresses [8, 9, 10]. The data streams obtained by the darknet, are monitored by a handmade script. This gives a basic overview of the most popular sorts of traffic and basic security issues on their network.

1.1 Research question

To extend the monitoring of the darknet data streams, this project is based on the following research question:

> *What information can be gained from the captured XS4ALL darknet streams, and could it be used as a zero day warning system?*

Four months of captured XS4ALL darknet data will be analysed in this research. This data is viewable with the help of the Argus client tools, which are described in section 3.1.

1.2 Research goal

The goal of this project is to find a method to get useful information from the captured darknet data streams. This method can then be used as a zero day warning system, as explained in section 4.4.

Chapter 2

XS4ALL Darknet

The term *darknet* is used to describe two different situations:

1. A private connection between two or more hosts, by which content is exchanged.

2. A server, which monitors or captures bogon IP traffic or traffic towards unused IP addresses in a network;

XS4ALL uses a darknet to capture bogon IP traffic and traffic towards unused IP addresses in a network, as described in item *2* above. The XS4ALL server uses Argus to capture data streams; Argus is explained in the next section.

Two types of IP addresses are routed to the darknet. The first is all the customer traffic routed to bogon IP-addresses, which include for example 192.168/16, 10/8, 1.0.0.0 or 1/8 [20]. All traffic of XS4ALL customers which is trying to reach one of these bogon addresses, will be routed to the darknet.

The second type are unallocated XS4ALL IP-addresses. These addresses can be reached from the internet, although there should be no traffic to these addresses since they are not assigned to any server, service or customer.

When traffic is sent to the IP address ranges described above, the traffic is routed to the darknet. When this traffic enters the darknet, it will be captured. The darknet does not respond to any traffic. Figure 2.1 illustrates the global flow of internet traffic. The red arrows indicate the direction of bogon traffic, while the green arrows represent the direction of ordinary internet traffic.

2.1 Argus

Argus (Audit Record Generation and Utilization System) is an open source project by Qosient [1] and is available for Linux, Solaris, FreeBSD, OpenBSD, NetBSD and Mac OS X.

Argus is a real-time flow monitor and can be used as a data stream capturing tool. The captured data is saved in the Argus-format, which is flexible and extensible [1], and can be viewed with the Argus client tools.

XS4ALL runs the Argus server application to capture the streams of data towards the darknet. There are several Argus tools available to analyze the data

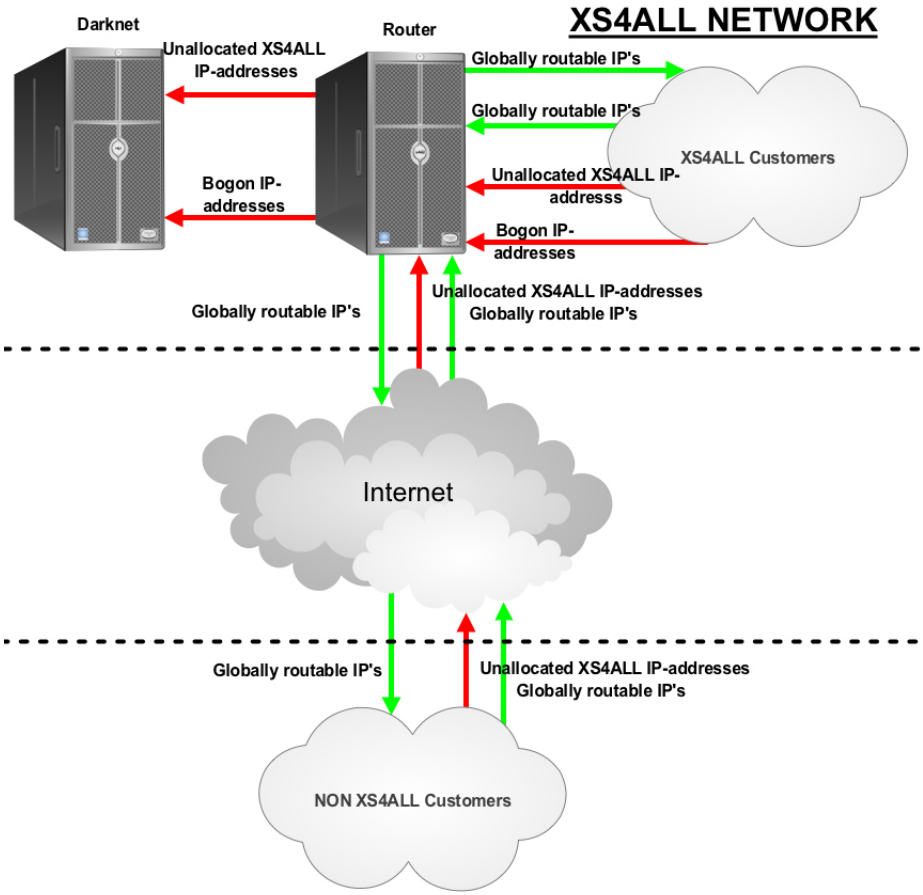

Figure 2.1: Global flow of internet traffic and traffic towards the XS4ALL darknet

the server captured, which are described in section 3.1. All Argus client tools are compatible with the main Argus tool, *ra*. In other words, the commands used by *ra* to filter data can also be used with the other tools.

2.2 Data

NetFlow is software developed by Cicso Systems. NetFlow analyses packets of an interface on which it is active. The analysis is done on seven fields: source and destination IP address, source and destination port, layer 3 protocol type, type-of-service byte and input logical interface. When two of the packets analyzed by NetFlow match all seven of the fields it assigns them to the same flow. The flow is then passed to a flow analyzer [11].

NetFlow is a specific technology which has several versions, the most utilised of which is "sampled netflow" where one packet out of every X packets is sampled from an interface and the data is gathered up, then exported to a NetFlow

collector on a periodic basis.

Most NetFlow setups will sample at 1:1000, or 1:10000, which means that it will select every 1000th or every 10000th packet. It will continue to do this and then every few minutes export a summary of all the packets it sampled to the collector. This approach works fine for a very busy interface.

Argus captures data streams and tries to work out if the packet is part of a new connection, or an existing connection (in the case of TCP) between systems. It also has the ability to do some packet inspection.

Since the traffic sent to the Darknet is often not very much, it is better to inspect everything that is sent there.

The captured data does not contain the actual packets, but only the streams, except for UDP messages. The first 712 bytes of an UDP message are captured. Due to the limits of this project we did not focus on this. Argus creates an entry for each data stream it captures, which contains the following fields:

Start time The time the stream started;

Flags Which flags are set for the stream;

Type Protocol type, TCP, UDP, ICMP and others;

Source address Address of the machine sending data to the darknet;

Source port Port where the packet is sent from;

Destination address Darknet routed IP address;

Destination port Port the sender tries to reach;

Packet size Size of the packet received;

State What state the connection is in, mostly timed out or initialization state, because the darknet does not respond.

Chapter 3

Analysis

The first phase of this project is to get a clear view of the Argus data streams. There are several tools available from the Argus client. Also, handmade scripts are used to analyze the darknet data streams. Both will be discussed in this chapter.

3.1 Tools

The tools that the Argus client contains can be used for filtering and sorting the captured darknet data. The basic tool *ra* is compatible with all other Argus tools. In other words, the options that are available in *ra* are also available in all the other Argus tools, these are [1, 17]:

RA gets it data from an Argus datafile and filters the records, which is done by a given filter expression. The output can be printed to stdout or to another Argus datafile;

RASORT produces an Argus stream after sorting the records of an Argus input file, based on command line criteria;

RAGATOR merges Argus flow activity records together by reading an Argus datafile;

RACOUNT prints various counts based on data of a given Argus datafile;

RAGRAPH produces graphs of selected fields of an Argus file. This tool uses the very complex *rrd-tool* for generating PNG-files, therefore for this report other graphing tools are used, those from Microsoft Excel;

RAMON aggregates records read from an Argus datafile and produces an Argus data that could be used as input for RMON2;

RAXML converts Argus data to a XML file.

From the tools above, *RA* and *RASORT* are used most in this project. The output of these tools is piped to handmade scripts, especially made for this project, see Appendix A.

3.2 Scripts

As our research will show, the presence of internet worms scanning darknet IP space and customer based infections can dramatically effect the amount of data collected in a darknet. In order to help process this data in a more efficient manner, a number of bespoke scripts were created to sort and categorize data into 'bins' which permitted more accurate analysis. Due to the time limitation of the project the scripts are quite basic, however, they can be extended to gather more information. Section 3.3 describes which patterns are found, and which handmade scripts have been used.

3.3 Patterns

When analyzing the captured data streams certain patterns could be found, which are described in the following subsections.

3.3.1 Portscans

The purpose of a portscan is to systematically determine the availability of network resources/services by testing the response of a target machine. It doesn't have to be vulnerable, it just has to exist, to be of interest to the person doing the scan. Portscanning is a form of information reconnaissance. You only look if there is an opportunity to enter a host. With a port scan a person can find out which services might be listening and can then decide what, if any, attacks to run. If this person knows there's a SSH daemon listening, and a FTP server, he then want to try to perform a brute force authentication attack against those services. If he knows there is a HTTP server, they may try to exploit vulnerable HTTP servers, or CGI programs.

Another method of scanning is "service scan". With service scans a lot of host are scanned for an specific port. If a hacker is only interested in hacking SSH daemons (22/tcp) then he would only do a service-scan for 22/tcp on lots of machines and then decide which machine to go back and hack later.

If he is interested in any weakness on one machine, then he is more likely to do a full portscan and get the complete list of possible things to attack next.

The Argus data lets us see both types of attacks. If you run the *ra* query looking at the source address alone, you can see what they were scanning, was it just a service, or was it a range of services, making it a portscan. This distinction is subtle, but this is how many of us in the security community see the difference.

When running a *ra* query looking at the source and destination, it is possible to see specifically what the attacker is interested in.

3.3.2 IP address scans

IP address scans are often initiated by hosts, which try to discover if a host behind a certain IP address can be reached. These scans frequently use ICMP[1]. Machines, which send a lot of these ICMP packets are easily discovered by checking the number of ICMP packets from a certain IP address. Detection of

[1]Internet Control Message Protocol, used to get a status reports of sent information

these scans can be used to give XS4ALL a warning that a host might be infected with some kind of malware or virus.

3.3.3 DDoS

DDoS attacks take many forms. You can have a Service-based DoS which sends a lot of valid requests to a server and tries to overwhelm it handling those requests. This would be something like an HTTP GET based attack, where thousands of requests to get a specific web page are send, causing the server to become slow trying to answer all of the requests.

Other forms may contain a lot of little packets, but at a very high rate of packeting. For example, when an attack of 1.500.000 packets per second are sent to one computer. Those don't have to be HTTP packets, they just have to overwhelm the operating system's ability to handle TCP/IP requests coming in, causing the system to become unresponsive, and thus, deny service to people who legitimately want to use service on that host..

Another type may be a large-packet attack, which often seeks to overwhelm the data pipe to the server. If you have a 8Mbit DSL line and you send 50Mbit of traffic to it, then the line is essentially unusable. DDoS's which use large packets often have this type of goal.

It is also possible to get a mixed DDoS, which has a lot of large packets (usually UDP packets), and small packets (tcp/udp/icmp) and at various rates that can overwhelm a lot of things.

In defending against some DDoS attacks, sometimes it is as easy as blocking just one host who is attacking. In other cases, depending on the attack, you may have to block hundreds or thousands, or find some other "fingerprint" which describes the attack which you can put into your firewalls to block.

When looking at the captured Argus data streams, DDoS attacks can be recognized, because the packets look like a *TCP SYN-ACK*, or *RST* packet, in which case, Argus would be seeing something known as a "backscatter".

Backscatter in a DDoS attack occurs when the attacker spoofs the IP address that he claims to be sending from, and sends a packet, such as a tcp-syn packet, to the victim machine. The victim machine then does what the TCP protocol says and tries to send a SYN-ACK packet to the IP address he thinks the SYN packet came from. If the IP address is spoofed, and the spoofed address happens to be from the darknet ranges, the darknet will suddenly see *SYN-ACK* packets coming in. These packets aren't sent by the darknet because the darknet IP addresses are never used to send a *SYN* packet to the internet on their own. Therefore a *SYN-ACK* to the darknet is always bad. ICMP can also cause backscatter, and the darknet would see that as echo-response packets *ECR* or some other packet besides a Echo-Request.

3.3.4 IP addresses

The handmade script – published in Appendix A, section 6.1.2 – generates a list of all IP destination addresses from the past 10 days, sorted on the total number of connections to each destination IP address. Also, the average number of connections per destination IP address is calculated and listed. We were hoping to see why there is a lot of traffic that does not belong there and what

information can be gained from the Argus streams. When sorting on the destination IP addresses with the highest average number of connections, there are a lot of IP addresses like 1.0.0.0 and 1.3.3.7, which are bogon IP addresses; these IP addresses appear frequently.

Some of the globally routable IP addresses are sending a steady amount of packets for a certain amout of time. After this period, no traffic from these hosts is seen. We assume this is caused by a malware infected host. An explanation for this behavior could be an infected host disconnected by its ISP or maintenance to the infected host.

3.3.5 Port patterns

During this research, large amounts of repeating port patterns are found. We noticed that all the destination ports available were used at least once every day. Therefore, sorting the data was important, the techniques for doing this are described in section 3.8. A list of most occurring ports and upcoming port numbers can be produced.

By analyzing the data, we observed various patterns of targeted destination ports. These patterns are frequently the same as port patterns seen by malware, described at the website of Dshield [4], for example. Results of the research done can be found in section 4.

3.4 Time

The amount of traffic is analyzed per day; every day about two to three million packets hit the XS4ALL darknet. There seems to be no special pattern per day of the week, it is always fluctuating.

When looking on an hourly base at the arriving packets, a clear pattern can be found, as showed in figure 3.1. Starting at 07:00AM CET [2], traffic is increasing, until 02:00PM. Then, until 05:00PM, the amount of traffic is steady, and from 05:00PM till 08:00PM it is increasing again. From 08:00PM till 0:00PM it is steady again. After 0:00PM, until 07:00AM, the amount of traffic is decreasing. This is interesting because of the geographic breakdown of the source of the packets, and the fact that the majority of them are from The Netherlands and indeed XS4ALL customers themselves

This indicates most of the senders are active in the afternoon and evening. We think the time pattern matches the home users habits quite well. Since most companies open earlier, at eight or nine in the morning, and most companies have professionals controlling the security of their systems, it might be assumed, a lot of the senders are home users. A small amount of them might be actually controlling the packets, and are doing a port scan, for example.

3.5 Protocols

Figure 3.2 illustrates the average protocol usage in the XS4ALL darknet. The largest amount of packets are TCP packets, followed by UDP packets and ICMP packets. ICMP is used to discover if hosts are active, for example. TCP and

[2] Central European Time

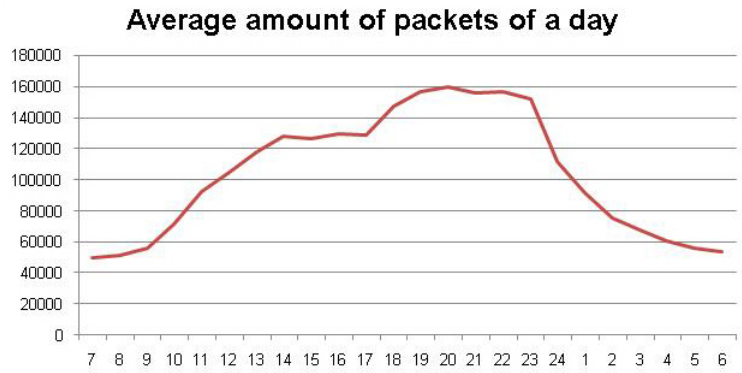

Figure 3.1: Average amount of packets per hour, hitting the XS4ALL Darknet

UDP packets are used for a wide variety of attacks and probe attempts such as the Windows popup spam. People sending Windows popup spam do not care about answers. They just want to send the message, and hope people click on the links and download possibly infected software. The remaining protocols were detected relatively infrequently.

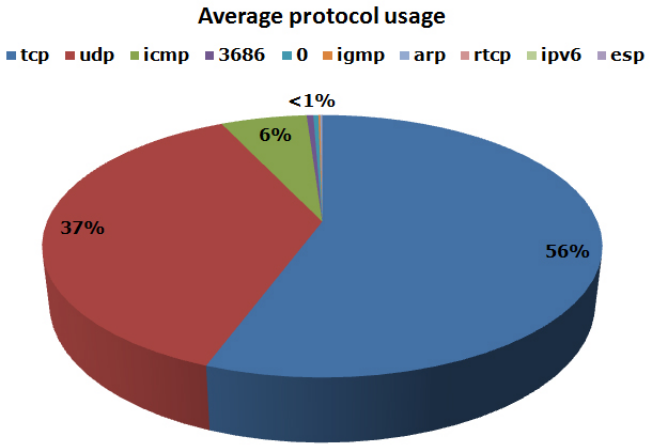

Figure 3.2: Average amount of protocols used by packets heading towards the XS4ALL Darknet

In RFC 790 [13, 14] all available protocols are numbered from 0 to 255. When looking at figure 3.2 a protocol with number 3686 is seen. Because protocols are numbered from 0 to 255 this is an incorrect protocol, and it might be a bug in the logging program Argus.

There is a small amount of IGMP traffic visible in the darknet – IGMP is used for multicasting. It can be used by a machine to assign itself to a multicast list on a router. It is possible that hosts are trying to get on these router lists by probing, which declares the traffic in the XS4ALL darknet.

Real Time Control Protocol (RTCP) is used periodically to transmit control

packets to participants in a streaming multimedia session. The primary function of RTCP is to provide feedback on the Quality of Service provided by Real-time Transport Protocol, which is a streaming media protocol [2]. Some lost pieces of this traffic enter the darknet, only a few packets a day.

A tiny amount of IPv6 traffic and Encapsulation Security Payload (ESP)[3] traffic is left, which is used to encapsulate data in a secure way.

3.6 Categories

The traffic flow towards the XS4ALL darknet – what data is coming from which direction – is shown in figure 3.3.

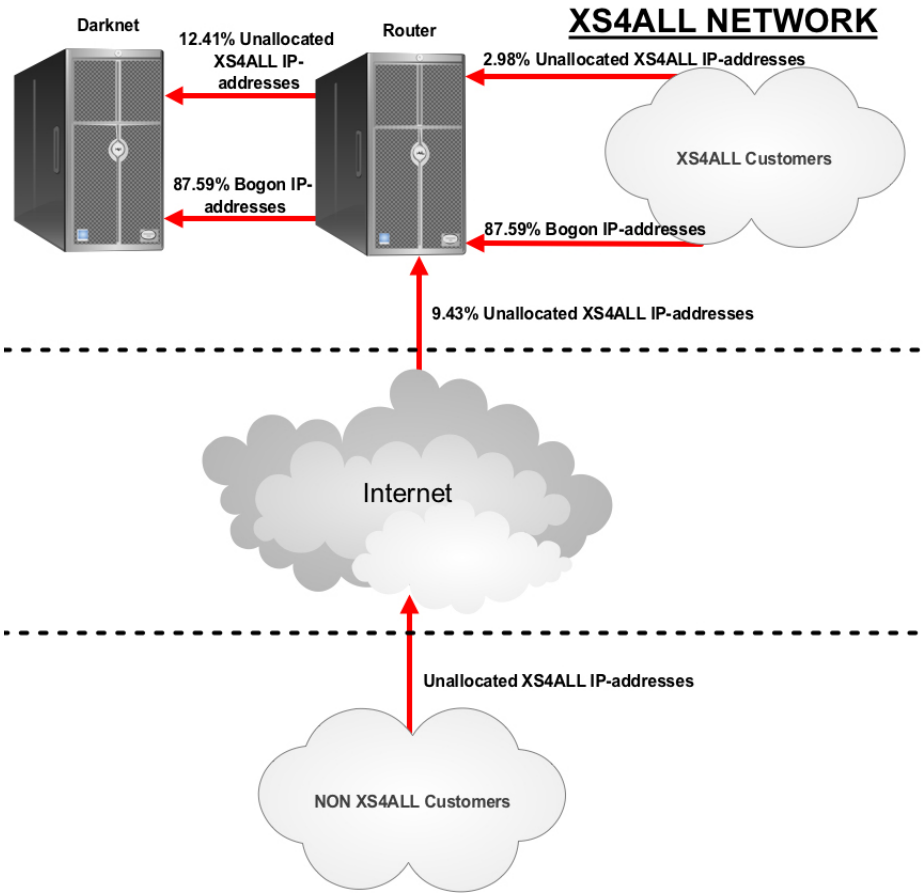

Figure 3.3: Illustration of the traffic towards the XS4ALL Darknet, with an indication of the average amount of data coming from which direction.

All the percentages shown in the illustration are percentages of the total amount of data hitting the darknet. Note the large amount of traffic com-

[3]ESP is described in RFC 2406

ing from XS4ALL customers to invalid IP space, which represents the largest amount of data of the darknet.

3.7 Origin

All data sent to the darknet is either data to bogon IP addresses or to unassigned XS4ALL IP addresses. In both cases it is data which should not be sent at all. Therefore, it is interesting which hosts send this data and where the data comes from. This section describes the origin of the traffic in the darknet.

3.7.1 Misconfiguration

One of the reasons hosts are sending packets to the XS4ALL darknet can be misconfiguration. Due to misconfiguration of (network) equipment, non-globally routable IP addresses could be routed from the customer to XS4ALL, for example 10.0.0.0/8 addresses. The amount of traffic currently hitting the darknet due to misconfiguration is unknown, however, we do know the darknet gets more than 50 percent of packets designated for addresses like these.

3.7.2 Viruses / Worms and other malware

It is likely that a lot of the traffic going to the darknet is sent by viruses, worms or other kinds of malware. By analyzing the traffic to the darknet, a lot of port patterns are found. These patterns, which are discussed in section 3.3.5, may be sent by viruses or worms; these packets are sent by malware trying to spread itself or trying to harm other hosts, for example.

3.7.3 IP origin

Outside hosts trying to connect to an address of the unallocated XS4ALL IP address-space will also be redirected to the darknet. By analyzing this data it is possible to retrieve some information like the origin and AS number. This information can be used for source analysis. By doing a *whois* lookup on the source IP address it is possible to gather this information. The script to do this can be found in Appendix A in section 6.1.3. There are two sorts of connections to the darknet. One is the XS4ALL customer and the other is an outsite host. A large amount of traffic is heading towards bogon IP addresses 3.6.

The first chart 3.4 shows the top 10 of countries connecting to the darknet, including traffic from the XS4ALL customers. By filtering only these 10 countries, more than 95 percent of the origin of the data is illustrated.

XS4ALL is located in the Netherlands, so the packets from XS4ALL customers are from the Netherlands too. Since it is interesting what traffic is actually heading towards the customers, and getting a more reliable view of what is hitting the darknet, the XS4ALL customers are filtered out in the second chart.

The second chart 3.5 shows the top 10 of countries connecting to the darknet, but without the connections of XS4ALL customers. This results in a different chart with 23% less connections from the Netherlands.

It is noticeable most traffice comes from Europe. Together with Scott A. McIntyre we came up with a few explanations for this.

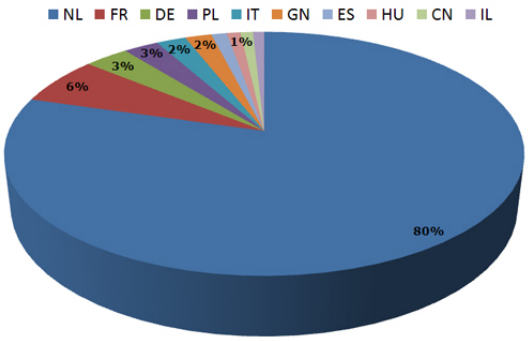

Figure 3.4: Number of connections per country connecting to the XS4ALL darknet, including XS4ALL customers

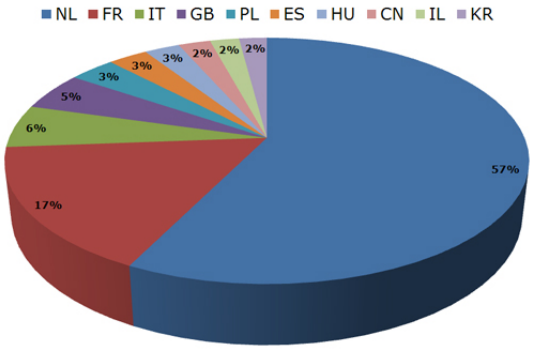

Figure 3.5: Number of connections per country connecting to the XS4ALL darknet without the XS4ALL customers

- Europe has a large broadband penetration, therefore a lot of always-on systems, which could be compromised more easily, and sending out malicious traffic because of that.
- IP addresses in Europe are allocated in blocks by RIPE. Infected machines might start spreading traffic beginning in or close to their own IP-address range.
- Another source is the possibility of the use of European proxy servers like Tor. Depending on the type of attack there may also be other sources. Traffic that hits port 22 and 5900 are typically direct attacks while traffic hitting port 139, 135 and 145 are more likely indirect attacks.

3.8 Types of analysis

During the research on the captured Argus data streams a few analysis techniques are used. These techniques are described in this section.

3.8.1 Top N

Because there are a lot of port numbers in the captured Argus data streams, it is not efficient to list them all every day. By generating a top N, for example top 25, of the ports with the highest activity, an efficient view is generated. The top N list is not the same as the trend list. The top N analysis generates a list of ports with the highest activity.

3.8.2 Baseline

To be able to make a trend analysis, which is described in section 3.8.3, it is necessary to have information to compare the new results with. The information to compare with is the baseline. The baseline is a number based on the average attacks on a port or IP for the last N minutes, hours or days. The baseline needs to be recalculated every time it is used to compare the new data with. The baseline indicates the average amount of packets to a certain port or IP address for the past N minutes, hours or days.

3.8.3 Trends

A trend is the increase of activity on a certain port. By monitoring trends it is possible to give an early warning about upcoming threats. Trends are based on a baseline 3.8.2. This baseline is then compared with the last N minutes, hours or days. When activity increases more than a certain specified percentage, it is a trend. Mostly, trends are only mentioned when they increase. Because it is also interesting to see when a trend is decreasing it is a good idea to monitor this as well.

As an example the raising of the the Symantic AntiVirus (SAV) worm[16] is used to show why monitoring trends are important. Figure 3.6 shows the raise of the SAV worm. The figure is originating from dshield.org[4]. Between

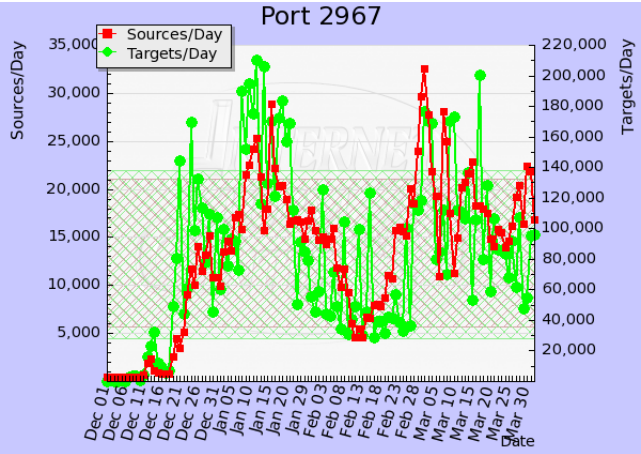

Figure 3.6: SAV worm spread

December 2006 and January 2007, there is a small amount of activity on the

SAV port, number 2967. This can indicate that an leak is found on that port and that hackers are trying some exploits. Around the 15th of January, there is much activity on the SAV port. This can be explained by hackers who are fine-tuning the exploit. Then around March 2007 there is an enormous increase of connections on the SAV port. This could indicate that there has been a successful worm that makes use of the SAV exploit. When the above information is know, it is possible to take the necessary measures to prevent an infection. The above example can also be used for XS4ALL. With the captured streams from there darknet it is possible to do trend analyze on the ports and IP addresses.

3.9 Analysis conclusion

Out of the XS4ALL captured data streams of their darknet, it is possible to get a lot of information, like destination and source IP addresses and ports. Based on this information it is possible to make analysis to discover upcoming trends, like new worms and virusses, gather information about the origin of an attacker, and use this information to send abuse messages to Internet Service Providers.

Chapter 4

Results

4.1 Statistics

This chapter describes the results of our research. Out of these results, various statistics can be shown.

4.1.1 Top Source IP addresses

To get a impression of what is going on in the darknet, a selection of top Source IP addresses is represented. For privacy concerns, these IP addresses are blanked out in this report. All the top five source addresses are XS4ALL customers.

Source A

This customer sends 16 UDP packets each second, increasing the destination IP address by one every time. The customer sends packets towards port 137, which is used by trojans like Chode, Qas and Msinit. During our research, we noticed it scans bogon IP space, that explains why the darknet receives these packets. The host started sending on the 1st of January at 20:34 to the bogon IP address 39.253.41.0. Everyday it sends around 10,000 packets, creating the most traffic in the darknet from one IP address. In February, the host quit transmitting.

Source B

This customer sends TCP packets to the bogon IP address 1.3.3.7 on port 80, around 9,000 a day. This behavior could indicate that the host is used in a botnet, and that it is waiting for a valid host IP address to connect to. In the meantime, it connect to a specified IP address, 1.3.3.7, which is also known as *leet*.

Source C

This customer sends ICMP packets to 'random' IP addresses. This could indicate that the host is scanning for other reachable hosts. The scanning can be initiated by malware, viruses and/or by user action.

Source D

This customer sends UDP packets to IP addresses 1.0.0.0 on port 0, 112.5.145.124 on ports 65535 and 48388. These ports are used by trojans like RC1, Sins and worms like Adore. There is a good chance, this host has been infected by at least one of these threats.

Source E

This customer sends TCP packets to 'random' IP addresses on 'random' ports. This IP address has a DNS record called xxxx.xs4all.nl. This is a host of XS4ALL itself, because XS4ALL controls the forward lookup, and in theory you should get an answer from the XS4ALL servers.

4.1.2 Top Destination IP addresses

A selection of top Destination IP addresses is represented in this section.

112.5.145.124

This destination IP address is bogon. Several XS4ALL customers send UDP packets on port 65535 to this IP address. This port is used by several worms and trojans as Sins, RC1 and Adore. This indicates that these customers are likely infected by at least one of these threats.

1.0.0.0

This is a bogon destination IP address. Several XS4ALL customers send UDP packets to this address on port zero. This is a reserved port, and should not be used for UDP or TCP traffic [19].

39.123.0.0

This IP destination address is bogon, according to RFC 3330 [10]. Several XS4ALL customers send UDP packets to this address on port 137. This port is used by the msinit-worm and the NETBIOS Name Service. NETBIOS is used for communicating in Windows environments.

1.3.4.9

This destination IP address is also bogon. Some XS4ALL customers send UDP packets to this address on port 8000. This port is used for the iRDMI protocol, or as an alternative for port 80 (HTTP).

1.3.3.7

A lot of customers are sending UDP packets to IP address 1.3.3.7 on port 80, which is already described in section 4.1.1. These customers may be compromised with malicious software which is attempting to "phone home".

4.2 Trend

To do a trend analysis for XS4ALL, a script is created for this research project. The script can be found in Appendix A in section 6.1.4. This script takes the average use of a port for a certain period of time and compares this with the new usage of the port. When it exceeds a given percentage it will display the port with the exceeded percentage. The output of this script may be similar to:

```
62.50% - Portnumber: 12149 - Port amount: 13 - Overallaverage: 8
40.00% - Portnumber: 12186 - Port amount: 7 - Overallaverage: 5
34.21% - Portnumber: 12183 - Port amount: 51 - Overallaverage: 38
25.00% - Portnumber: 12165 - Port amount: 10 - Overallaverage: 8
18.52% - Portnumber: 12210 - Port amount: 32 - Overallaverage: 27
18.18% - Portnumber: 12204 - Port amount: 13 - Overallaverage: 11
16.67% - Portnumber: 12188 - Port amount: 7 - Overallaverage: 6
```

Based on this information, it is possible to see which port has the most increasing usage. In the example above, this is port 12149, with a increased usage of 62.50%. This data may be processed into a chart.

4.3 Top N

As discussed in section 3.8.1 a top N of the most used ports is generated to give a good overall view. The script that is made for this job can be found in Appendix A in section 6.1.1. This script calculates the average usage per port. The output of this script may be similar to:

```
rank usage port average
1 2616494  4662  avg: 261649
2 1536178  137  avg: 153617
3 1513972  icmp  avg: 151397
4 1407002  65535  avg: 140700
5   986436  4672  avg: 98643
6   938564  0  avg: 93856
7   696205  80  avg: 69620
8   628577  135  avg: 62857
9   493594  139  avg: 49359
10 488282  8000  avg: 48828
```

With this generated data, it is possible to make a graph that displays the port activity, as shown in figure 4.1.

A lot of traffice is heading towards ports 4662 and 4672. These ports belong to a program called Emule. Emule is a peer-to-peer[1] program used for file exchange. Because Emule is used to exchange files, it is an easy way of spreading malware. When a lot of Emule traffic hits the Darknet it could mean that a lot of people are using Emule and/or are infected. Other ports mainly used are ports 137 and 139, which belong to NetBIOS. Attacks are seen on port 80 – which represents web servers – and on the ICMP protocol, which is used to check the availability of a specified host. Remarkable is port 0:

[1] http://en.wikipedia.org/wiki/Peer-to-peer

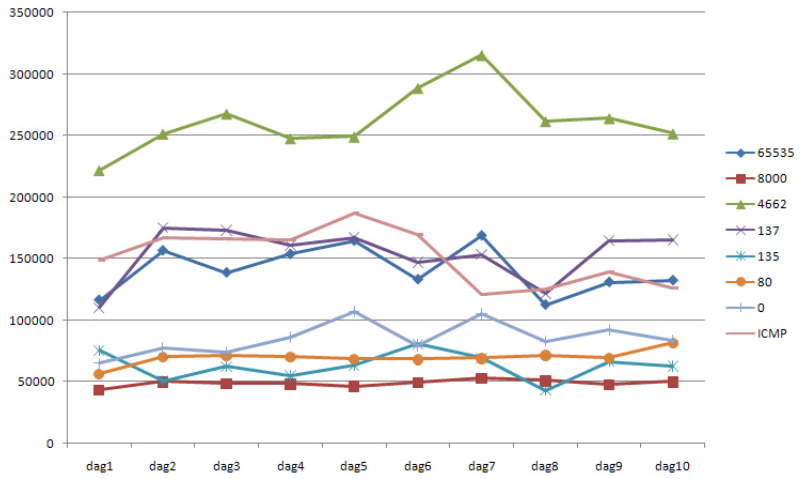

Figure 4.1: Top 8 most used ports for 10 days

> "Port 0 is officially a reserved port in TCP/IP networking, meaning that it should not be used for any TCP or UDP network communications. However, port 0 sometimes takes on a special meaning in network programming, particularly Unix socket programming. In this environment, port 0 is a programming technique for specifying system-allocated (dynamic) ports."[19]

4.4 Zero day warning system

A zero day warning systems exists in various shapes and sorts, all with the same goal: quickly identify and notify upon new upcoming events or dangers. In a network, a zero day warning system gives an alert for new upcoming worms, virussen, e.c.

A XS4ALL zero day warning system should quickly identify and notify upon new infections hitting the XS4ALL network. This way, XS4ALL and their customers are warned of upcoming threats in an early stage. With this zero day warning system, the necessary measures can be taken at an early stage to prevent infection at XS4ALL's customers.

A zero day warning system for XS4ALL can be realized by using the trend analysis, described in section 3.8.3 and with use of the Top N analysis, described in section 3.8.1. By collecting captured streams and compare them with and average of the last 30 days – for example – it is possible to see new upcoming worms, virusses, e.c. In addition to that, the techniques described in the following section could be used for this system.

4.5 Internet security index

XS4ALL is interested in a security index, which they can present on their website, to make customers aware of computer security, and the dangers of the

internet. Dshield offers their visitors a security index like this as well. Their mechanism of calculating the index is proprietary; contacting them with questions about their mechanism did not result in any information on how the system works. XS4ALL was wondering if the darknet data can be used for such a security index. For a security index for the XS4ALL customers, we are mainly interested in the traffic heading towards the unallocated XS4ALL IP addresses, since this is data heading towards their IP space.

Several ways of creating a security index with the help of the darknet data are possible:

Total amount Since all the data going to the darknet is malicious, the amount of traffic heading towards it might be an indicator for how much malicious things happen on the internet.

Rapid increase Rapid increase of traffic to certain ports can indicate a new worm or trojan being active. If one or more ports exceed a certain activity level, the security index should be marked unsecure. This level should be calculated in further research.

Port rating Some ports are known ports used by worms for malicious activities. To make the security index more accurate, a port rating system could be used. So the security index will be marked insecure earlier, when ports with a high rating appear, and it will react less when 'trusted' ports are increasing. The downside of port rating is that an exploit of an unknown port does not get recognized as high rated, because it is unknown. A solution to this could be a high rating for unknown ports, and only low for 'trusted' ports.

IP rating If any connection between IP origin and amount of threats can be made, it could be possible to mark traffic from these origins as more likely to be malicious. The security index increases when traffic from certain origins appear.

A combination of the above suggestions could be used to build a internet security index.

Chapter 5

Conclusion

In this chapter, the conclusion as well as the future research and development of the XS4ALL zero day warning system will be described. The future work is a proposal towards future developments for the XS4ALL darknet project.

5.1 XS4ALL Darknet

During this research project, a lot of information is gathered from the captured Argus data streams. The information is analysed as discussed in section 3, and the results can be found in section 4. At the end of this project the research question can be answered:

> *What information can be gained from the captured XS4ALL darknet streams, and could it be used as a zero day warning system?*

From the captured XS4ALL darknet streams it is possible to gain valuable information:

- IP origin: the country of the IP address;
- Protocol usage;
- Number of packets per hour;
- Destination and source IP address;
- Destination and source port.

With the available captured data streams it is possible to build a reliable zero day warning system for the XS4ALL network. By monitoring the trend analysis every day, and compare it with the calculated baseline, it is possible to see new worms, virusses and/or malware coming in an early stage. As additive for the zero day warning system, a top N can be used to show the most attacked ports at the moment.

5.2 Future Work

During this project, several scripts are built in order to get interesting data out of the darknet. These scripts could be improved and put together to form a system, which reports its findings on a daily base to the an administrator.

Further, an internet security index and a zero day warning system could be built, by using the darknet data, and possibly data from companies – like dshield for example.

Both of these suggestions are interesting, however, during our research there was unfortunally not enough time available to develop these systems. Scripts, which automatically recognize patterns – which we described in chapter 3 – would be interesting for the zero day warning system and the internet security index.

5.2.1 Coöperation with dshield

Sites like dshield [4] and the Internet Storm Center [5] already have an overall warning system with trend analysis and a top 10 of most attacked ports. Their information is based on reports and logfiles. It could be a good idea to coöperate with these organizations in order to compare the data. This way it is possible to see if the XS4ALL network faces the same threats as other networks.

5.2.2 Bind port with the help of CVE

Common Vulnerabilities and Exposures (CVE) [6] is a list of standardized names for vulnerabilities and other information, such as security exposures, and is worldwide used by companies like Cisco and IBM. The CVE list gives a good and standardized overview of the vulnerabilities and exposures currently present.

5.2.3 Abuse Messages

When looking up the origin of an IP address, it is possible to lookup the ISP and their abuse email address. With this information, an automated email can be sent to the abuse email address of the found ISP, which reports the abuse of the IP address. This way the ISP can take measures.

5.2.4 Security Index Calculation

The exact way of calculating a security index is not available at this moment. We advice further research is conducted on this issue.

Bibliography

[1] Argus homepage,
http://qosient.com/argus/index.htm

[2] RTCP,
http://www.javvin.com/protocolRTCP.html

[3] Protocol numbers,
http://www.iana.org/assignments/protocol-numbers

[4] DShield homepage,
http://www.dshield.org

[5] Internet Storm Center homepage,
http://isc.sans.org

[6] Common Vulnerabilities and Exposures,
http://cve.mitre.org

[7] Wikipedia about darknets,
http://en.wikipedia.org/wiki/Darknet

[8] Bogon IPs,
http://www.completewhois.com/bogons/

[9] RFC 1918,
http://www.ietf.org/rfc/rfc1918.txt

[10] RFC 3330,
http://www.ietf.org/rfc/rfc3330.txt

[11] What is NetFlow?
http://www.flukenetworks.com/fnet/en-us/supportAndDownloads/
KB/IT+Networking/ReporterAnalyzer/What_is_NetFlow_.htm

[12] RFC 1883,
http://www.ietf.org/rfc/rfc1883.txt

[13] RFC 790,
http://www.ietf.org/rfc/rfc790.txt

[14] IANA Protocol Numbers,
http://www.iana.org/assignments/protocol-numbers

[15] IANA Ethernet Numbers,
http://www.iana.org/assignments/ethernet-numbers

[16] Arbor Network, explaintion of the SAV worm,
http://asert.arbornetworks.com/2006/11/
that-new-bot-irc-bot-attacking-symantec-overflow

[17] Argus tool ragraph,
http://www.penguin-soft.com/penguin/man/1/ragraph.html

[18] Port 135, Windows popup spam,
http://www.secureworks.com/research/threats/popup-spam/

[19] Port 0 usage,
http://compnetworking.about.com/od/tcpip/l/blports_0.htm

[20] List of bogon IP's towards the XS4ALL Darknet,
http://www.cymru.com/Documents/bogon-bn-agg.txt

Chapter 6

Appendix

6.1 Appendix A

The scripts used in this research project can be found in this appendix. The scripts are written in either Ruby or Python.

6.1.1 Average use of the ports

The Python script below calculates the average usage per port with use of the input generated by the shellscript below.

Input

The shellscript below reads the Argus data with use of *rasort* and writes the output to the total_avg_dx files.

```
rasort -n -s dport -r dag2 | sort | uniq -c | sort -nr | cat -n > total_avg_d2 &&
rasort -n -s dport -r dag3 | sort | uniq -c | sort -nr | cat -n > total_avg_d3 &&
rasort -n -s dport -r dag4 | sort | uniq -c | sort -nr | cat -n > total_avg_d4 &&
rasort -n -s dport -r dag5 | sort | uniq -c | sort -nr | cat -n > total_avg_d5 &&
rasort -n -s dport -r dag6 | sort | uniq -c | sort -nr | cat -n > total_avg_d6 &&
rasort -n -s dport -r dag7 | sort | uniq -c | sort -nr | cat -n > total_avg_d7 &&
rasort -n -s dport -r dag8 | sort | uniq -c | sort -nr | cat -n > total_avg_d8 &&
rasort -n -s dport -r dag9 | sort | uniq -c | sort -nr | cat -n > total_avg_d9 &&
rasort -n -s dport -r dag10 | sort | uniq -c | sort -nr | cat -n > total_avg_d10
```

Python script

This script uses the output of the previous shellscript for input and prints the output to stdout.

```
res = {}
baseName = 'total_avg_d'
firstFileNo = 1
lastFileNo = 10

for x in range(firstFileNo,lastFileNo+1):
```

```
  f = open(baseName+str(x))
  file = f.readline()
  while file:
    file = file.split()
    file = [int(y,0) for y in file]
    if len(file) == 2:
      res['icmp'] = file[1] + res.get('icmp',0)
    else:
      res[file[2]] = file[1] + res.get(file[2],0)
    file = f.readline()
  file = file.split()

for x in res:
  print res[x], x, 'avg:',res[x]/(lastFileNo-firstFileNo+1)
```

6.1.2 Average use of a destination IP address

The Python script below calculates the average usage per destination IP address with use of the input generated by the shellscript below.

Input

The shellscript below reads the Argus data with use of *rasort* and writes the output to the daddr_dagx files.

```
rasort -n -s daddr -r dag2 | sort | uniq -c | sort -nr | cat -n > daddr_dag2 &&
rasort -n -s daddr -r dag3 | sort | uniq -c | sort -nr | cat -n > daddr_dag3 &&
rasort -n -s daddr -r dag4 | sort | uniq -c | sort -nr | cat -n > daddr_dag4 &&
rasort -n -s daddr -r dag5 | sort | uniq -c | sort -nr | cat -n > daddr_dag5 &&
rasort -n -s daddr -r dag6 | sort | uniq -c | sort -nr | cat -n > daddr_dag6 &&
rasort -n -s daddr -r dag7 | sort | uniq -c | sort -nr | cat -n > daddr_dag7 &&
rasort -n -s daddr -r dag8 | sort | uniq -c | sort -nr | cat -n > daddr_dag8 &&
rasort -n -s daddr -r dag9 | sort | uniq -c | sort -nr | cat -n > daddr_dag9 &&
rasort -n -s daddr -r dag10 | sort | uniq -c | sort -nr | cat -n > daddr_dag10
```

Python script

This script uses the output of the previous shellscript for input and prints the output to stdout.

```
res = {}
baseName = 'daddr_dag'
firstFileNo = 1
lastFileNo = 10

for x in range(firstFileNo,lastFileNo+1):
  f = open(baseName+str(x))
  file = f.readline()
  while file:
    file = file.split()
    file[:2] = [int(y,0) for y in file[:2]]
```

```
    res[file[2]] = res.get(file[2],0) + file[1]
    file = f.readline()

  file = file.split()

for x in res:
  print res[x], x, 'avg:',res[x]/(lastFileNo-firstFileNo+1)
```

6.1.3 Get the origin of an IP address

The Python script below determines the origin of an IP address with use of the input generated by the shellscript below.

Input

The shellscript below reads the Argus data with use of *rasort* and writes the output to the saddr_dagx files.

```
rasort -n -s saddr -r dag1 | sort | uniq -c | sort -nr | cat -n > saddr_dag1 &&
rasort -n -s saddr -r dag2 | sort | uniq -c | sort -nr | cat -n > saddr_dag2 &&
rasort -n -s saddr -r dag3 | sort | uniq -c | sort -nr | cat -n > saddr_dag3 &&
rasort -n -s saddr -r dag4 | sort | uniq -c | sort -nr | cat -n > saddr_dag4 &&
rasort -n -s saddr -r dag5 | sort | uniq -c | sort -nr | cat -n > saddr_dag5 &&
rasort -n -s saddr -r dag6 | sort | uniq -c | sort -nr | cat -n > saddr_dag6 &&
rasort -n -s saddr -r dag7 | sort | uniq -c | sort -nr | cat -n > saddr_dag7 &&
rasort -n -s saddr -r dag8 | sort | uniq -c | sort -nr | cat -n > saddr_dag8 &&
 rasort -n -s saddr -r dag9 | sort | uniq -c | sort -nr | cat -n > saddr_dag9 &&
 rasort -n -s saddr -r dag10 | sort | uniq -c | sort -nr | cat -n > saddr_dag10
```

Python script

This script uses the output of the previous shellscript for input and prints the output to stdout.

```
res = {}
baseName = 'saddr_dag'
firstFileNo = 1
lastFileNo = 10

for x in range(firstFileNo,lastFileNo+1):
  f = open(baseName+str(x))
  file = f.readline()
  while file:
    file = file.split()
    file[:2] = [int(y,0) for y in file[:2]]
    res[file[2]] = res.get(file[2],0) + file[1]
    file = f.readline()

  file = file.split()

for x in res:
```

```
  print res[x], x, 'avg:',res[x]/(lastFileNo-firstFileNo+1)
```

6.1.4 Trend analysis

This script takes the average use of a port for a certain period of time and compares this with the new usage of the port. When it exceeds a given percentage it will display the port with the exceeded percentage. This scripts takes the total file generated by the average script, as described in section 6.1.1.

Ruby script

```
#!/usr/bin/ruby -w
#part[0] = amount
#part[1] = port
#port[2] = "avg:"
#port[3] = average

#Percentage over average
#Portnummer
#Port amount

average = Array.new
newday = Array.new
exceed = 1.1 #1.1 = 10%, 1.19 = 19% ect.

f = File.new("average")
while (line = f.gets)
part = line.split
portNumber = part[1].to_i
average[portNumber] = part[3]
end
f.close

f = File.new("avg_d11")
while (line = f.gets)
part = line.split
portAmount = part[0].to_i
portNumber = part[1].to_i
portAverage = part[3].to_i
newday[portNumber] = portAmount
average[portNumber].each do |overallAverage|
portExceed = overallAverage.to_f * exceed
if portAmount > portExceed
if overallAverage == "0"
puts "Infinity - Poortnumber: #{portNumber} - Portamount: #{portAmount}
 - Overallaverage: #{overallAverage}"
else
exceedPercentage = portAmount / overallAverage.to_f
exceedPercentage = exceedPercentage * 100
exceedPercentage = exceedPercentage - 100
```

```
#exceedPercentage = portAmount / overallAverage.to_i * 100 - 100
puts "#{exceedPercentage} end
end
end
end

f.close
```