BGP (D)DoS Diversion

Wouter Borremans & Ruben Valke

July, 2005

Abstract

By using BGP DDoS diversion, ISPs will have a mechanism to prevent their entire network becoming unreachable as the result of a DDoS attack on part of their network. An external BGP diversion mechanism will be used to announce a specific part of the provider's network to (a part of) the Internet. Announcing a specific part of this network will divert the DDoS traffic and thereby prevent other parts of the provider's network becoming unreachable. This gives the provider the ability to continue providing services to the rest of his customers.

Contents

1 Introduction

This report will describe a new way to reduce the impact of a DDoS attack. In no way this project is *the* solution to prevent a DDoS attack, but from a practical point of view it is a very good way to keep the provider from effectively losing his connection with the Internet and limit the financial impact of an attack.

In this document, the BGP[4] DDoS diversion project will be described in detail. Since BGP DDoS diversion is a new technique, the description will be mostly theoretical but it will also have a practical view, since a prototype was build to test the BGP DDoS diversion technique as a whole.

The goal of this document is to give an overview of the specific settings needed to implement a BGP DDoS diversion mechanism and which issues have to be solved to make BGP DDoS diversion work.

July 2005,
Ruben Valke[20]
Wouter Borremans[21]

1.1 Why was this project initiated?

This project was initiated by the NL-ix[7] primarily to fill the need of DDoS protection for ISP's connected to the NL-ix Internet exchanges. DDoS attacks can cost ISP's a lot of money since they have to pay variable fee's for the usage based transit[1] traffic on their lines. A mechanism is needed to actively reduce the impact of a DDoS attack on a provider's network. This results in saving costs for transit traffic and network reachability. Currently there is a great need for an anti DDoS mechanism, many providers are often under attack.

1.2 What is a (D)DoS attack?

A (**D**istributed) **D**enial of **S**ervice attack is intended to flood a target network with useless traffic in such a way that it will become unreachable. (D)DoS attacks are one of the largest problems on the Internet infrastructure at the moment. It causes ISP's to loose a lot of money by not letting them have connectivity to the outside world and letting them pay fee's for traffic generated by the attack. A (D)DoS attack exploit's vulnerabilities in the TCP/IP[1] protocol. There are two types of a (D)DoS attack, namely;

Non distributed DoS attacks (1) are attacks carried out by a single computer. These attacks appear in a peer-to-peer form. Well known non distributed tools are: Ping of Death, Teardrop, Winnuke, Land, Bonk, Snork and Smurf.

Distributed DoS attacks (2) are attacks which are carried out by multiple 'zombie' pc's all pointing to one destination (the destination can be a network or a specific host) often cracked by an exploit and waiting for instructions to start sending packets to a specific destination. Well known Distributed Denial of Service tools are: trinoo, Tribe FloodNet (TFN), and TFN2.

[1] Globally routed internet

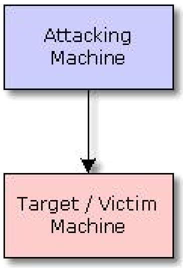

Figure 1: Non distributed attack

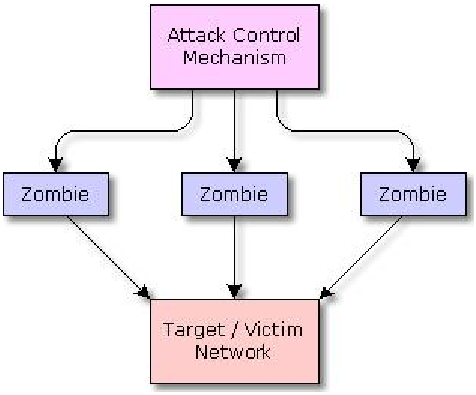

Figure 2: Distributed attack

2 How to detect a (D)DoS attack

A key problem when addressing (D)DoS attacks is detection. In this section we will discuss a few methods to detect (D)DoS attacks.

- Detection by traffic patterns

  Every network has a specific network traffic pattern. This pattern repeats itself on and on with a deviation. When you learn the traffic patterns for a certain amount of time you are able to distinguish non regular traffic from the standard pattern. To be sure you are really dealing with a strange pattern, you can compare historical data which you acquired in the past. (Think of historical data of yesterday, last week, or last month) When there is a big difference in the pattern triggers can be set to notify other systems or to enforce human interference.
- Detection by (sudden) traffic increase

  This method is quite simple, but in some cases effective. A mechanism continuously monitors the traffic on the uplink of a provider. A trigger can be configured when a specific threshold is overrun. This method has a very high possibility of a false positive, think of situations where a specific threshold is overrun due to the popularity of a website.

Besides only detecting a (D)DoS attack, you also want to be able to trace back the origin of the attack itself. This can be very difficult. As an ISP you first want to know which part of your network is under attack. After that you can take counter measures leading in blocking parts of your network.

Several tools are available which are able to show the content on traffic for your network. It would be preferable to use tools like *tcpdump*[13] to be able to show essential information on network activity. Based on information provided by network tools you can see if an attack is carried out by a set of compromised hosts or just a single one. In case of a single host which carry's out the attack you can notify the provider or the subscriber self. In many cases the source IP's are spoofed since the attacker does not want to get caught. It is very important that you acquire evidence to proof that you were under attack, log files or screen dumps can be very effective getting attackers sentenced. More and more companies try to trace back attackers since (D)DoS attacks cost a lot of money because continuation of service is disrupted.

3 Defending against a (D)DoS attack

Once a DDoS attack is started it is very difficult to stop it. Hundreds or even more hosts that have been compromised to join the attack are extremely difficult to trace back since in most cases the IP-address has been spoofed. Many organisations think that it is possible to defend against DDoS attack with a firewall. This is in most cases nonsense since the rate of packets will flood the connection until it is completely full.

In the following paragraphs we will discuss several methods from both a theoretical and a practical point of view.

3.1 Basics

There are a lot of different techniques to defend against a (D)DoS attack. All techniques have in common that they can be implemented at different levels of a network. The next paragraph will explain these levels.

3.1.1 Diverting levels

In the context of anti (D)DoS mechanisms diverting means the act of leading unwanted traffic away from a network, preventing the network to become unreachable. There are three levels of diverting traffic. To understand these three levels we first explain where to place these levels.

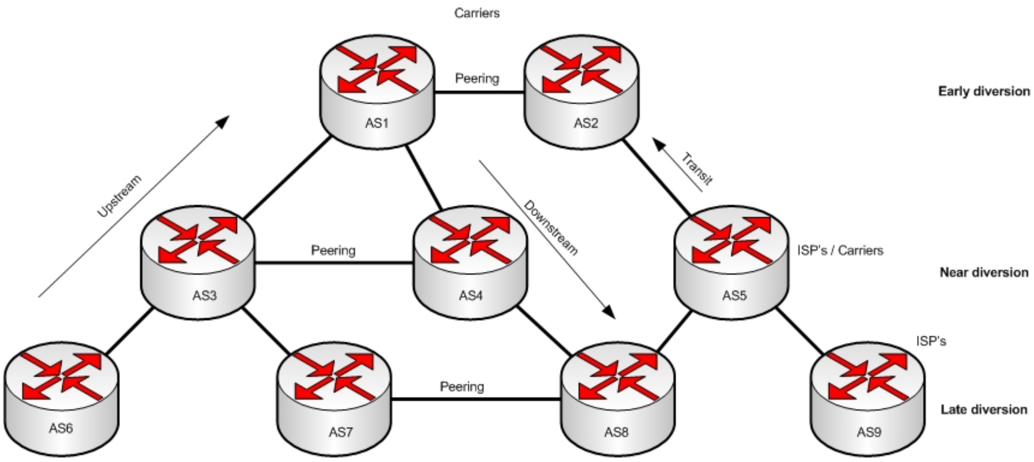

Figure 3: Levels of traffic diversion

- **Early diversion**
  Early diversion is carried out at carrier level. This means that at the most upper level (upstream path) of the hierarchy (See figure (3)) the diversion mechanism is implemented. Looking from a global perspective, an Internet hierarchy does not look like a regular binary tree but more like a chaos of interconnected routers. (D)DoS attacks can flow from top to bottom and back again. The key advantage of early level diversion is that the (D)DoS attack will be stopped at the highest for the targeted hierarchical level possible, this means that the (D)DoS attack won't arrive at the transit provider or below it. This prevents ISP's loosing money on extra traffic fees.

- **Near diversion**
  Near diversion is implemented in the direct network neighborhood of the ISP near its upstreams and peers. This means that the transit and upstream providers *will* receive the useless (D)DoS traffic in case of an attack. The attack will not arrive at a targeted ISP. Good agreements between providers and it's transit providers are needed in this case. Near diversion takes place in the downstream path, paid by the targeted ISP.
- **Late diversion**
  As the name already states, late diversion is implemented as near as possible to the ISP. This means that if a (D)DoS attack takes place, the traffic flow comes through the carrier provider and transit provider resulting in financial loss.

It is crucial to find a good spot to implement your anti (D)DoS mechanism, this can save you a lot of money avoiding high variable traffic fees.

3.2 Anti (D)DoS techniques

In the next paragraphs different techniques to defend against a (D)DoS attack will be discussed. To get a clear view of all the different techniques a comparison is made in paragraph 3.3.

3.2.1 Rate limiting

When an ISP is not able to completely filter out the DDoS traffic because it would affect more then just one user or service, it could be an option to rate limit the DDoS traffic. Rate limiting can only be carried out on specific ports or types of traffic. This method still will affect other users or services on the network because their traffic cannot be seperated with the rate limiting feature on the ISP's backbone. By looking on the problems DDoS attacks already create, the side effects of rate limiting can be neglected. The way rate limiting is implemented depends on the type of attack carried out. Rate limiting is especially effective using it as a mechanism to stop 'SYN flood'[5] attacks. Some providers have a rate limit on their backbones, this prevents them from high traffic bills caused by (D)DoS attacks by limiting their traffic for example at 200Mbit/s to be able to burst on top of their normal traffic pattern. This way of rate limiting is called 'near rate limiting' and takes places within the network of the provider which is always-on or on-demand.

3.2.2 Oversizing

One of the most expensive ways to defend yourself against a (D)DoS attacks must be oversizeing. With oversizing, you make sure that you always have enough bandwith and routing capacity, so you can even serve a massive (D)DoS attack. Oversizing means that you will only need for example at about 10% of your total bandwitdh for daily operations. All the extra capacity will only be used in case of an emergency. By using this kind of countermeasure you are sure that you will always have enough spare bandwidth to keep providing your services during a (D)DoS attack. Oversizing is not a full protection against denail of service attacks. Routers and other network equipment are still vulnerable for specific types of packets,think of a specially manually crafted packets which shutdown the router or let it come into a state in which it is not possible anymore to handle packets. Another way to slow down, or even shutdown a router is to sent traffic which will be handled by the router's CPU. The traffic handling of the router itself will be slow down dramatically.

3.2.3 Firewalling (TCP/UDP blocking)

Another way to stop DDoS attacks is to simply block IP addresses. There are two ways of blocking IP-addresses namely; blocking source addresses and blocking destination IP's taking out the service behind the targeted IP address. In practice it is almost impossible to stop DDoS attacks by blocking a source IP address since most DDoS attacks appear in a distributed form. Thousands of IP addresses will flood the targeted network or host. It is almost impossible to filter traffic based on all different IP addresses.

Another way to block incoming DDoS traffic is filtering on port numbers. At router level, the packets will be dropped according to the filter rules. Obviously, this method will not work for important services where most of the attacks are targeted against. Shutting down the attacked port is an effective way to stop the attack resulting in doing exactly what the attacker wants: disrupting the availability of your services.

3.2.4 External BGP diversion

A very effective way to slow down and maybe even stop DDoS attacks to a network is by using BGP[4] (Border Gateway Protocol) diversion. Using BGP as a anti DDoS mechanism can be used when an attack really gets bad, resulting in no or very limited connectivity for the connected party. There mechanism works as follow:

- An independent DDoS diversion AS[2] will announce the IP-space that is under attack.
- BGP Peers will send all traffic to the diferted network to the diversion mechanism, since that route is more-specific.
- The diversion mechanism will null-route the DDoS traffic.
- The isp who was under attack, has his bandwitdh available for the rest of his network

BGP community dropping

BGP community dropping is a way to identify specific parts of a network. Many upstream carriers offer specific communities to let ISP's telling the carrier which traffic to drop. This way of traffic dropping takes places near the ISP and can be placed under 'near diversion' (see figure (**??**)). For more information on BGP communities, refer to chapter (4).

Using the BGP protocol as an anti DDoS mechanism is very effective and relatively simple. This document will discuss this technique in more detail in chapter (5) and further.

3.2.5 Stop announcing (BGP)

Stop announcing IP-address space, can also be a solution to defend against a DDoS attack. In this case, the ISP will stop announcing his IP-address space to

[2] Autonomous System

it's peers, which results in the fact that globaly there is no route to this addresss space anymore.

There are two types of not announcing address spaces, namely:

- Stop announcing the affected address block (e.g. /24), assuming that only a portion of your network is under attack of a (D)DoS
- Stop announcing your complete address space, and start announcing the smaller parts that aren't under attack. This mechanism works even when an ISP only has one block of IP addresses.
  Example: An ISP has one /24 network (e.q. 1.2.3.0/24), the first 120 ip's are under attack, in this case the ISP can announce 1.2.3.128/25, keeping to other part of his network up.

3.2.6 Isolation

Some providers have special network blocks assigned to isolate DDoS sensitive hosts. (such as IRC servers or mission critical hosts) These networks cannot be directly contacted from outside the core network. Alternatives of isolating your network from other networks:

- Seperating domestic and international traffic
  Purchase seperate traffic from different carriers/transit providers. When you are being attacked from a specific network region, you could stop announcing that part of the network, isolating you from specific parts of the Internet. You will still be reachable by the other networks you're connected to.
- Isolate traffic from peers or peer groups
  By isolating peers or peer groups you can configure your network in such a way that you won't be reachable from specific locations like cable or DSL providers.
- Isolate traffic based on cable or DSL ip's This method requires a very good administration of ip spaces from providers which are used for endusers.
- Seperating (D)DoS traffic from regular traffic In this case you need two lines to your upstream. In case of an attack you announce a more specific part of your network over the first line and the other (D)DoS traffic will continue to travel over your other line where it can be analyzed or null routed by another (second) router.

3.2.7 Commercial implementations

In this section we will shortly discuss two commercial implementations of (D)DoS diversion / protection mechanisms. Initally, this project was not started to do research on commercial implementations. Just to give an idea on what is available, you can find two selected products below.

Cisco Guard

The Cisco Guard [2] is a solution to analyze and actively defend a network from a DDoS attack. The device diverts traffic away from the targeted host and can identify and analyse incoming traffic without affecting the good traffic flows. This is a very powerful appliance because mission critical systems need continuous network connectivity. The Cisco Guard can deliver multigigabit performance protecting against malicious traffic. At the moment there are two models available for shipping

- Cisco Traffic Anomaly Detector XT 5600

  This anomaly detector is able to detect (D)DoS attacks, worms and other forms of attacks by looking at traffic patterns in combination with active packet monitoring. This enables the customer to prevent even so called 'day zero' attacks which are not yet identified.
- Cisco Guard XT 5650

  The 5650 is able to perform per-traffic-flow analyses. The 5650 can divert traffic away from targeted hosts by using MVP (MultiVerification process) The device is designed to analyse data at multiple levels ensuring the best possible security. It helps to achieve business continuation during an attack.

  Due to its great performance large organizations will be able to handle attacks of multi gigabits safely.

Riverhead

The Riverhead Guard[3] is a stand-alone device which filters the network traffic on malicious traffic. When a threat is detected, only traffic which is targeted to a by the Guard monitored host is diverted to the Guard for treatment allowing traffic addressed to non-targeted hosts to continue. Traffic which is detected as malicious will be subjected to a inspection process based on Riverhead's MVP (Multi Verification Process) architecture. The architecture completely removes malicous traffic allowing genuine traffic to pass.

The Riverhead Guard is a very effective tool for DDoS prevention. Its capabilities for traffic filtering offer effective protection against DDoS attacks.

The Riverhead Guard can be installed adjecent to a router or switch on a seperate network interface. It delivers a high-end solution for enterprises and hosting companies. The Guard uses the principe of diversion, filtering and analyses to provide full protection against DDoS attacks.

- Diversion

  When a DDoS attack is detected, the Guard redirects the targeted host's traffic off the main path. (The Guard has to be deployed as adjacent to the core-routers or switches) Malicious traffic is filtered in which packets are identified and removed if necessary. Diversion is performed on demand for different targets at different times, and can protect any element web servers, routers, switches, and DNS servers on the provider's backbone or in the data center.
- Filtering and Analysis

  The diverted traffic is subjected to a number of tests and statistical analyses, anti-spoofing and anomaly recognition modules. These modules are able to identify malicious sources and reveal abnormally behavior. The Guard is able to learn the traffic behavior of the network to which it is connected. Based on this statistical information the guard can determine when a the network is under a DDoS attack.

The Riverhead Guard is able to handle multiple gigabit traffic, this makes the appliance attractive for implementation. It's statistical traffic pattern learning makes it very effective against detecting and preventing networks of being attacked by a DDoS attack.

3.3 Comparison

The comparison table will give an impression on the possibilities of the different DDoS defend solutions. A specific solution can be effective, but nobody will use it if it will cost a fortune. This table will give a clear impression on the different DDoS defend solutions, and their advantages and disadvantages. Within this table, 1 means Low and 10 means high.

	Rate limiting	**Oversizing**	**Firewalling**	**BGP Diversion**	**Stop announcing (BGP)**	**Isolation**
Costs	7	10	6	3	1	6
Efficiëncy	5	8	7	8	5	7
Reliability	6	9	6	7	7	7
Simplicity	7	6	3	6	8	6
Scalability	3	3	4	7	8	5
Rechability network	4	8	7	9	1	6
Reachablity hosts	4	8	3	1	1	2
Overall	**3**	**7**	**6**	**8**	**6**	**5**
Divert level	late	late	late	early/near	early/near	late
Operational mode	manual	always-on	always-on/on-demand	on-demand	on-demand	always-on

Table 1: Comparison of different diversion techniques

Looking at the overall score of the techniques listed in the table we can conclude that BGP diversion and oversizing are the most effective ways dealing with (D)DoS attacks.

4 What is external BGP DDoS diversion?

Using the BGP routing protocol ensures efficient path selection, choosing the most optimal route to a destination. Traffic flows depend on routes determined by BGP. BGP announces routes to other directly connected routers (by peering or transit). (see figure (3) for an overview of types of links) Each router has its own responsibility for networks configured on its interfaces, these networks often have a boundary like /8, /16 or /24.

When a network is under a (D)DoS attack, large amounts of useless packets travel from a start- to an endpoint. The target of a (D)DoS attack often lies in the core of an ISP's network. The flow of packets generated by attacker(s) cause the ISP to become unreachable. The flooding of network with useless packets fills the backbone of an ISP until no spare bandwidth is left letting the connectivity of an ISP disappear from the Internet. Since (D)DoS attacks cost ISP's a lot of money, ISP's want to be able to stop or lead the (D)DoS traffic from their networks. One of the most effective methods is the use of BGP (D)DoS diversion.

With BGP (D)DoS diversion you are able to lead the traffic away from a targeted host or network. The (D)DoS traffic will be taken to a place where it is not able to do any harm. The BGP protocol talks to its peers on a regular interval announcing to them which networks can be reached via him. BGP uses several types of data (metrics) to determine its best route to a destination. The most important metric is the selection of the *most specific route*. Routers always choose the most specific route to a destination. BGP (D)DoS diversion is based on this principle.

The BGP diversion mechanism leads (D)DoS traffic away from a target by announcing a very specific network which has by preference a /32 boundary, representing one ip address. The diversion mechanism is implemented as a separate system which can be placed on several levels within a network topology. (See chapter (3)). The diversion system represents an AS. More details on the system itself can be found in chapter (5) and (7).

5 External BGP DDoS diversion issues

In the previous section, the external BGP DDoS diversion technique is discussed as a whole. This section will discuss different aspects of implementing this technique, and the problems that could arise when implementing this mechanism.

5.1 Announcing

The main idea behind the whole external BGP DDoS diversion is based on announcing a more specific route via an external and independent BGP router. This sounds pretty straight forward, but it has a few strings attached to it. These strings will be discussed in the next paragraphs, including solutions for these problems.

5.2 Small network filtering

It has become quite normal for large ISPs to filter networks announcements that are smaller than a /24 network[3]. This is done by the ISPs to keep their routing table small. This can be a problem when trying to divert a DDoS attack via a more specific route, since these diversion routes often are /32 announcements.

Solutions for this issue could be:

- agree on acceptance of /32 announcements via peering/transit contracts for diversion purposes;
- agree on acceptance of /32 announcements based on a RFC, and preferably only if accompanied with a corresponding community-tag (see (5.6));
- create a special contract which states that you are willing to except /32 announcements for diversion purposes (see (5.5.1)) from the diversion mechanism;

Especially in the beginning, not everyone will accept the /32 announcements of the diversion mechanism, but if other parties discover the usability of this mechanism this will become less of a problem.

5.3 AS number

Before the router can announce the to be diverted routes, it needs an AS number to make the announcements with. For this a new AS number for this project has to be assigned by RIPE[14]. The major advantage of a separate mechanism, with its own AS number, is that the diversion mechanism is totally independent. When people do not want to null-route traffic based on a community, they can always choice to drop any traffic to the route which has the AS of the diversion mechanism in its AS-path.

5.4 Ripe database

If the external BGP DDoS diversion mechanism is announcing a network that has to be diverted, it is actually violating a rule of ripe. This rule describes that an AS may not announce a network that is not in the ripe RPLS database[4] This issue can be covered by a special contract as discussed in the paragraph above, or by a to be written RFC as discussed in the paragraph below.

[3] for example a /25 would be filtered out

[4] The ripe RPLS database stores the routing policies of an AS, and the routes that this AS announces.

5.5 Contracts

When a peering or transit connection is made, a contract will be signed by both parties. Such a contract coverts some rules both parties will obey. Examples of such rules are:

- only announce your own IP-space;
- do not announce networks smaller than /24;
- we shall not pay each other for traffic (in case of a peering agreement);
- you will pay for the traffic we exchange (in case of a transit agreement).

These sorts of contracts could get in the way to let the diversion mechanism work. The main two solutions are:

- describe the diversion mechanism in a RFC, and ask other parties to co-operate in the support for this RFC (see (5.7) for more info).
- sign additional contract, which agrees on allowing small network announcements (see (5.5.1)), and the use of a communities to drop the traffic (see 5.6).

5.5.1 Participant contract

A participant contract can be used as an in-between solution during the start of the project. Until there is a RFC which describes the diversion mechanism, a participant contract can be used to agree that you are willing to participate in the diversion mechanism project.

Such a contract will cover the following:

- you will accept small network announcements form the diversion mechanism;
- you will advertise the attached community to your upstream providers;
- you will try to drop the traffic according to attached community.

5.6 Community

When the diversion mechanism is announcing a route, it can tag the announced route with a community tag. With this tag, the diversion mechanism can tell its upstream peers to take action according to this community. In this situation this community will mean: Please drop or null-route traffic destined this attached route.

Upstream providers Upstream providers, might not participate in the usage of the community, because they have to drop the traffic for their customers. This is traffic which they cannot bill to their customer.

Redistribution Upstream providers, often remove communities that where attached by their peers. In this situation, an upstream provider will have to redistribute the community to their upstream providers and their (downstream) providers. If they redistribute the community, the DDoS traffic can be dropped as early as possible.

Standarization Standarization is always a big issue in Internet protocols. As will be described in the next paragraph, writing a RFC can help further introducing the diversion mechanism to the internet community. This RFC will describe the use of pre-defined communities for dropping traffic.

5.7 RFC

As discussed before, writing a RFC will help further introducing of the diversion technique to the world. This is because normally, people and hardware manufacturers will follow the advisories from a RFC.

This RFC will describe the use of a community which specifies to the receiver that traffic to that route can be null routed. The community tag will probably have the following layout: *FFFFF:0005*. This means that the community is not bound to a specific AS, and has the id 5. ID 5 is currently not assigned by the IANA.

Before a global community for dropping can be assigned by IANA[15], a RFC needs to be written. This RFC needs to be accepted by the IETF[16], and finally IANA will assign a community number to this project. This process has a very long timeline, and can take up to a few years, before it gets accepted. Waiting before the RFC gets accepted means that the project can be introduced in a few years. In the mean time, an arbitrary community tag will be used to introduce the mechanism, and to bring the idea to the internet community.

Hopefully, when the RFC gets approved, hardware vendors will start implementing the RFC into their systems, when they do this, the chance that the diversion community will become a success will be a lot bigger.

5.8 AS number

Physically, the external BGP DDoS diversion mechanism is a BGP capable router. This router needs an AS number to announce the to be diverted routes.

5.9 Diversion levels

The external BGP DDoS diversion technique can supply different levels of diversion (see (3.1.1) for details). How far a DDoSed network will be announced into the global internet, mostly depends on how the diversion mechanism will be implemented, and for how far the global community will except this technique.

In the initial state, the diversion level will be the equivalent of near diversion. If the general community for dropping traffic will be assigned by the IANA(see 5.7) the diversion level can be compared to early diversion, since all the traffic will be dropped at or very near the source.

5.10 Usability

To encourage the participation in the project, NL-ix / OpenPeering needs to actively talk to organizations connected to their network (or outside their network) to create the awareness of the importance to cooperate with this project. This is the only way for the project to become a big success in defeating DDoS attacks.

6 Requirements

Implementing BGP DDoS diversion mechanism, can be divided into two distinct subsets; functional and logical design. Both will be discussed in the next two paragraphs, and will give a detailed description of the specific elements that are needed for the mechanism.

6.1 Functional

The functional part of the diversion mechanism, are all non-technical issues of the project. These are the parts that eventually will be used by the end-users of the diversion mechanism, whereas the logical part will be function in the background.

To control the diversion mechanism in an easy way, two (web) interfaces should be provided. A third interface provides general information on the diversion project.

6.1.1 Interface1: Management

The management interface, allows the maintainer of the diversion mechanism to control specific parts of the mechanism. The following tasks can be done via this interface:

- add/edit/delete users;
- add/edit/delete ip-space that a specific user can announce;
- compare user ip-space with ripe-db[5].

6.1.2 Interface2: User

The user interface will allow users to announce their DDoS'ed ip-space so that it can be diverted via the diversion mechanism. Underlying mechanisms will take care of the following:

- check if ip-space may be announced by this user;
- modify bgp session to start announcing the DdoS'ed ip-space of the user;
- null-route DDoS traffic on dedicated physical port (see section 6.2.2 for more details).

6.1.3 Interface3: Information

Apart from the first two interfaces, a third interface is necessary. This interface will supply general information about the project. This interface doesn't need any connection with the above two interfaces.

6.2 Logical

Logical elements are all the technical parts needed to implement the diversion mechanism. In this chapter the following logical elements will be discussed; equipment needed, physical interfaces for this equipment, software needed, connectivity that has to be provided and requirements for the location of the equipment.

[5] Via RPLS, the ripe database can be queried to check if the user is owner of the ip-space.

6.2.1 Equipment

To implement a BGP diversion mechanism, a minimum of two servers is required. One machine will be used for the diversion mechanism itself, the other machine will serve a website with general information concerning the diversion project.

Server1-Diversion box The first machine, will serve the physical diversion mechanism. The following elements will be implemented on this machine:

- functional management interface (see paragraph 6.1.1);
- functional user interface (see paragraph 6.1.2);
- physical interface 1: User web interface (see paragraph 6.2.2);
- physical interface 2: BGP routing interface (see paragraph 6.2.2);
- physical interface 3: Blackhole interface (see paragraph 6.2.2) ;
- OS, routing and interfacing software (discussed in paragraph6.2.3).

Server2- web server The second machine is of less importance. It provides general information concerning the project. It is necessary to host this website on a separate machine (and preferably also on a different network), since in might be DDoS'ed itself if attackers find out what the project does to prevent total denial of service.

6.2.2 Interfaces

The BGP diversion mechanism consists out of a lot of different physical interfaces which all serve different services. To make the whole idea behind the BGP diversion mechanism more clear, we will describe the different physical interfaces that are needed to make the diversion mechanism work. Within this project a physical interface means an Ethernet interface at 10/100 Mbit or faster.

Interface1: User/management web interface (Server1) The first physical interface will serve the user and management interface to the users of the diversion mechanism. How this interface must be accessible, is still a point of discussion. The following options are available.

- username/password on a global accessible website;
- special vlan on the NL-ix infrastructure;
- telephone dail-up.

The first option, will be the most easy one, since the interface can be accessed from anywhere, but it has the major disadvantage that an evil person can also DDoS this interface, rendering the DDoS diversion box useless.

The second option, is very secure, because no one else than registered users can access the user interface. But it has the disadvantage that a user that is under a DDoS attach, can't probably access this special vlan either.

The third option will offer a dail-up service, which only registered users can access. It has the advantage that it's secure, because the interface itself can't be DDoS'ed. But the disadvantage is that this method might be received be the users as an outdated technique.

Interface2: BGP routing interface (Server1) The second interface is connected to an Internet Exchange, and will communicate in BGP with its peers. On this interface the diversion mechanism will announce the routes that have to be diverted.

Interface3: Blackhole interface (Server1/Server3) This interface will null-route the diverted DDoS traffic. Null routing the traffic can be implemented in a variety of ways. The following points have to be taken into account:

- will the null-routing interface generate interrupts on a server;
- how will flow-control react on a switch;
- how will a static mac configuration on a switch withhold in combination with proxy-arp.

Interface4: General website (Server2) This interface is available on the second server, and will serve the general website of this project. This interface needs global Internet connectivity, so that the page is reachable for everybody. It is preferable that this machine is located at an independent provider, so that this site can run completely independent of the diversion mechanism itself.

6.2.3 Software

To make the diversion mechanism work, software is needed to provide the necessary configuration possibilities. The software can be divided into three pieces; Server OS, routing software and interface software. All will be discussed in the next three chapters.

Operating System In the initial stage, GNU debian linux will be used as OS for the diversion mechanism in combination with Quagga as routing daemon. As discussed in the next paragraph, it might be useful to test other OS'es and routing daemons for there usability.

Routing software The BGP routing software is used to communicate in BGP with its peers. If a specific IP-address of a user is under attach of a DDoS, the BGP routing software will announce the specific IP-address to its peers, allowing the DDoS traffic to flow to the blackhole interface (see 6.2.2).

Routing software candidates are: Zebra/Quagga, XORP and bgpd (bsd bdp daemon). In the experimental stage Zebra will be used, but in the future other daemons have to be tested to ensure the most efficient daemon is used. A few important points to test the software for:

- ability to interface with the routing daemon;
- cpu load in havy load environment;
- support for the use of communities;
- feature list.

web interface software An interface for users is needed, so that users have an easy way to let the diversion mechanism divert their DdoS'ed IP-addresses. This software has to be specifically designed for this project, since no software of this kind is available. Other NL-ix/OpenPeering projects are all based on LAMP[6] configurations, so it's best to also use LAMP for this interface since experience of these software configurations is already available.

6.2.4 Location

The location(datacenter) where the diversion mechanism has to be physically located and to which upstream provider('s) it has to be connected, depends primarily on the market the BGP DDoS diversion is directed to. In this case there are two options; the dutch market only, or the global Internet. The idea is to introduce the diversion mechanism in three fases:

- fase one: (small): only use peerings for announcing DDoS'ed routes;
- fase two (medium): also use transit for announcing DDoS'ed routes;
- fase three (large): force the use of communities for dropping DdoS'ed routes.

Fase 1 In the initial situation, the diversion mechanism is only directed to the dutch market, since this project is initiated in the Netherlands. Connectivity is not a big issue in this case, because OpenPeering[7]can offer the diversion mechanism about 98% of all dutch routes. In this initial fase, not 100% of the DDoS traffic will be diverted, since the provider who is using the mechanism will still receive a portion of the DDoS via its transit upstreams.

Advantages:

- no transit necessary;
- service can be provided for free to dutch users;
- start small and learn.

Disadvantages:

- smaller audience;
- the diversion mechanism will not divert 100% of the DDoS attack.

Fase 2 In this fase the diversion mechanism will be positioned as a service to the whole Internet. This means that for example an ISP who is connected to an Internet Exchange in London, can use the diversion mechanism (which is located in the Netherlands). This has the disadvantage that the diversion mechanism needs transit traffic, since it's impossible to reach every destination via peering for a small AS as the BGP diversion mechanism. Providing the diversion mechanism with transit has the huge advantage that a 100% diversion is possible, because every destination can be reached.

Advantages:

- every ISP can use it;
- can provide 100% diversion.

Disadvantages:

- service probably can't be provided for free, since transit costs have to be payed.

[6] Linux, Apache, MySQL and PHP

[7] OpenPeering is a sister company of NL-ix

Fase 3 In the final fase, an RFC is written which indicates how the special drop community has to be treated (see **??** for more details) by the global Internet community. In this situation transit is still needed for the mechanism to announce the to be diverted IP-space, but the advantage is that the actual DDoS traffic isn't received over this link since it's dropped at the source of the DDoS.

Advantages:

- every ISP can use it;
- can provide 100% diversion;
- can be provided for free;
- a diversion can be initiated by some else than the diversion mechanism.

Disadvantages:

- the majority of the Internet community has to support the drop community;
- harder to detect if the DDoS is finished.

6.2.5 Connectivity

As discussed before, it has both advantages and disadvantages to provide the diversion mechanism with Transit. In the initial situation, only peering traffic will be used when DDoS traffic needs to be diverted. This makes the mechanism less effective, but is has the great advantage that the project can start small. Starting the project small makes it easier to debug startup bugs, and makes it possible to further fine-tune the setup before introduction of the diversion mechanism to the world.

7 Test environment

In this section we will discuss our test setup which is used to proof the concept of BGP (D)DoS diversion. Below (figure (4) you can see an logical overview of our setup. We will discuss each part of the network, explaining it's function. A real life impression of our test setup can be found at the appendix (H).

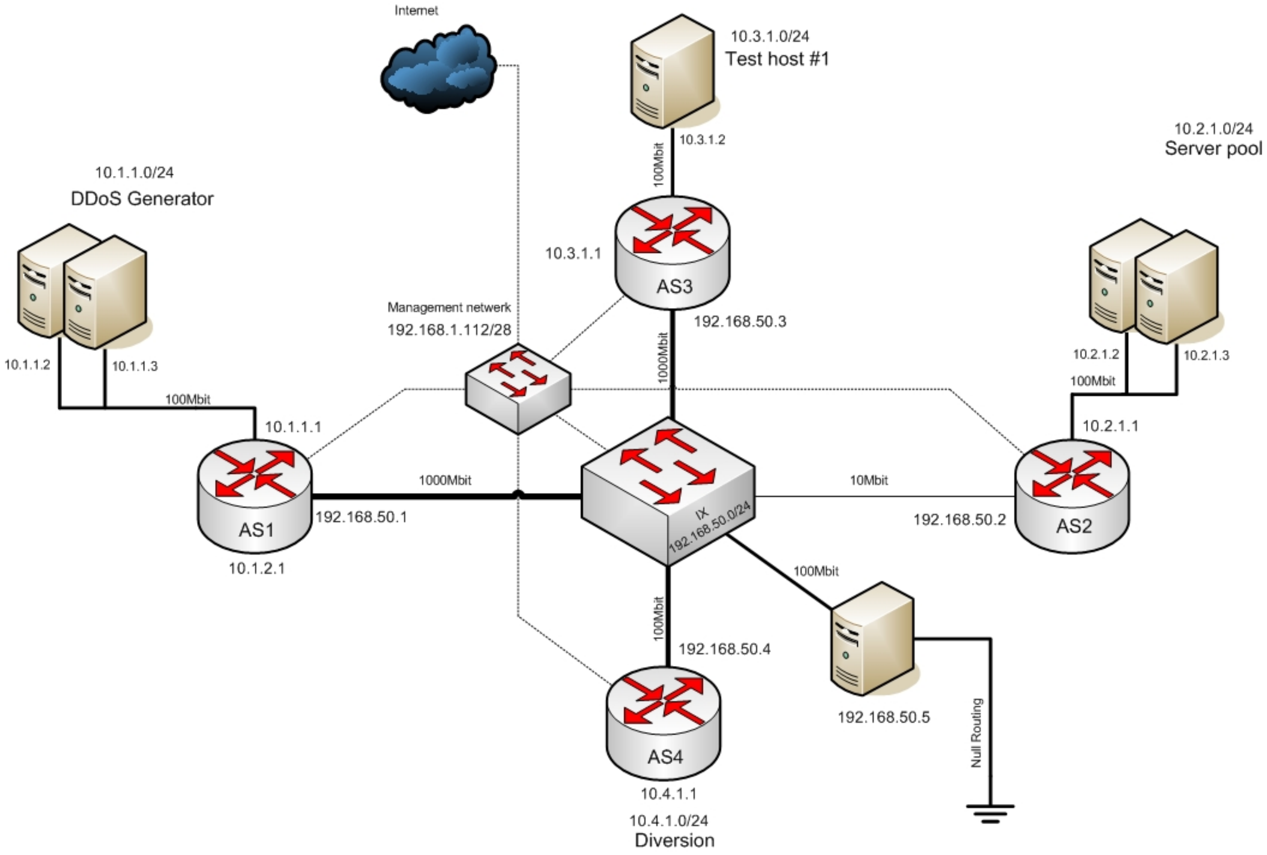

Figure 4: Test setup

AS1 represents the network in which the (D)DoS generator hosts are located. All the (D)DoS attacks carried out will be initiated from AS1.

AS2 represents the network in which the (D)DoS target hosts are located. This AS is connected to the Internet exchange by a 10Mbit line to simulate the impact of a large (100Mbit) (D)DoS attack to a connection with less bandwidth. Within AS2, two hosts are located. This is done to test connectivity during a (D)DoS attack to a host next to a host which is under attack by a (D)DoS attack.

AS3 represents the network in which one or more test hosts are located. We used this AS to test connectivity between AS3 and AS2 during a (D)DoS attack to AS2.

AS4 represents the network in which the (D)DoS diversion mechanism is implemented. The AS4 router announces a specific network and diverts the traffic to a Null Route interface.

7.1 Test equipment

In the table below, an overview with all the test equipment used is given. The test setup was quite impressive, as can be seen in appendix H.

Part of AS	Brand/model	Function	IP-address
1	Foundry BigIron 4000 Router Dell P4 Desktop Acer P4 Laptop	AS1 Router DDoS generator 100Mbit/s DDoS generator 100Mbit/s	192.168.50.1/24 10.1.1.2/24 10.1.1.3/24
2	Foundry BigIron 4000 Router Topline P3 laptop Compaq P4 laptop	AS2 Router DDoS Target other host in ddosed isp network	192.168.50.2/24 10.2.1.2/24 10.2.1.3/24
3	Foundry BigIron 4000 Router Sony P4 laptop	AS3 Router Other party representative	192.168.50.3 10.3.1.2/24
4	1U Server unknown brand	AS4 Zebra Software Router	192.168.50.4
4	Toshiba P3 laptop	blackhole device	192.168.50.5
(n.a.)	Foundry BigIron 8000 Switch	Internet Exchange	

7.2 Generating DoS traffic

To test if BGP diversion works, we needed to generate a DDoS our self. We used two different techniques to generate a lot of traffic to the attacked host. From the two tools used, floodping was the most effective in making a machine unworkable. The second tool: Mtools was a lot more effective when in came to generating traffic.

7.2.1 Floodping

Initially we used ping with the -f (flood) and -l (preload) options to generate an excessive traffic flow. The command looks as follow:

```
# ping -f -l 5000 10.2.1.2
```

In our case, we ran four instances of this floodping to the same host. With dis we could generate up to 70Mbit/s of traffic to the target. This floodping to the target host was most effective, since the target machine was unworkable when it was booted into windows XP and the floodping was running.

The major disadvantage was that this floodping was not reliable, because the traffic flow would fluctuate a lot. We are not certain what caused this, be we think it has something to do with the low priority ICMP traffic is handled by the TCP/IP stack.

7.2.2 Mtools

The ICMP floodping solution worked, but was not very reliable. It was better to generate UDP traffic, since this type of traffic has a higher priority in the TCP/IP stack.

The result was Mtools[19]. Mtools is a package for testing the performance of a network with Linux. Mtools consist out of a sender and a receiver part. We only used to sender part to send traffic to a host on port 80. In our case it doesnt matter if the destination port is open/refusing or closed. The traffic will still flow to the host.

To be able to run the program for a long time, we used the following script:

```
#!/bin/bash
while [ 1 ]
do
./owdmSend -a 10.2.1.2 -p 80 -C 9999999999 -c 1400 -t 99999999999
done
```

It was not necessary to run multiple instances of mtools. One instance filed up 100Mbit/s of traffic easily. This flow of traffic was very consistent, and it did not matter of the targeted host was up or down.

Mtools was not created to do any harm. This could be seen by the way a targeted hosts handles the traffic. With the ping flood option less traffic was generated, but rendered the targeted machine useless. Mtools generated 100Mbit of traffic, but the targeted machine was as fast as always, in didnt had any significant cpu usage increase.

7.3 Measurement tools

All the used machinery was equipped with activity leds. These leds came in handy when we wanted to know to which direction the DDoS traffic flowed in case of a diversion.

But blinking leds are hard to display in a report, and also are not very accurate. To obtain accuracy, we used two measurement tools; Cacti [18], and the *PHP current traffic tool*. Both had their own advantages on specific areas.

7.3.1 Cacti

Cacti[18] is the successor of MRTG, and is a very popular tool to generate a graphical representation of network traffic via the SNMP protocol FootnoteSimple Management Protocol. Examples of graphs generated with cacti can be seen in Appendix C through G.

We installed cacti on a Linux debian machine, and connected this machine to the management network of the test setup. Via this management network, the management interface of the Internet Exchange switch could be reached. Cacti then polls the traffic counters of the switch at an interval of 5 minutes.

Cacti gives a good impression of the traffic flows flowing over the Internet Exchange over a long period of time. A short fluctuation in traffic can hardly be noticed in the graphs since the poll interval in 5 minute. Cacti was very useful in explaining how a DDoS is diverted, because it can be clearly seen that a traffic flow is diverted to another host over a long period of time.

However, it was harder to see when the traffic started to flow to another interface (you had to wait for 5 minutes before it could be noticed).

For this specific issue, we use the PHP current traffic tool.

7.3.2 PHP current traffic tool

The PHP current traffic tool[8], is a tool which was designed by NL-ix. It gives them an impression on current traffic flows in their network. When requesting the PHP page, the tool will query the Internet Exchange switch via SNMP, and retrieve the current Mbit/s counters.

We altered this tool a bit, so it could run stand alone on the same Linux machine was running cacti. With this tool, sudden traffic increased could be noticed easily. For historical data, we revert to the cacti graphs.

[8] An impression of this tool can be seen at http://www.nl-ix.net/traffic.php

8 Tests performed

Once the test setup was build and all the equipment was configured, we could start testing if BGP diversion would work. We started out with basic tests, which are known to work in both theory and in practice. This was very useful for getting familiar with BGP in general, and diverting traffic with BGP specifically.

During this project, the following test categories where applied.

- Diversion; divert network to other host by use of more specific routing.
- Next-hop-self; divert traffic to other host, with the bgp next-hop-self parameter.
- Nullrouting the traffic; dropping the incoming traffic.
- Community; let peers drop traffic based on given community.

8.1 Diversion

The first test we performed was diverting traffic, based on the more specific rules of BGP. This means that announcing a network that is smaller (e.g. /26)will be preferred above a more general announcement(/24).

In our test setup, we tried the following:
We initiated a DDoS attack from AS1, with destination 10.2.1.2 (AS2). AS2 announces his network 10.2.1.0/24 to its peers. Next we got AS4 to announce 10.2.1.2/32. This route is more specific, which means that all peers (including AS1) will target traffic to 10.2.1.2 to AS4. This will result in the fact that the DDoS attack is diverted away from AS2, and is now directed to AS4.

Announcing the more specific route with AS4:

```
AS4:˜# telnet localhost 2605
Password: <password>
AS4-bgpd> enable
AS4-bgpd# configure terminal
AS4-bgpd(config)# router bgp 4
AS4-bgpd(config-router)# network 10.2.1.2/32
```

Using a more specific route to route traffic to a specific destination is common practice in bgp (e.g. the use of a default route), so it wasnt a big surprise that this construction worked. However, it was a very good exercise to get familiar with BGP.

What we did notice, was that peering sessions could go down, since there was no bandwidth left to communicate with its peers due to the DDoS attack. This means that the payload traffic should be separated from bgp traffic, or that the diversion mechanism should be supplied with very high bandwidth connection. A good solution for this problem, is the usage of the BGP *next-hop-self* parameter.

8.2 Next-hop-self

Within BGP it is possible to adjust parameters that accompany the announcement of a route. One of these parameters is next-hop-self. The next-hop-self parameter, indicates that the announced route can be reached via the supplied

IP-address. Normally the next-hop-self is the IP-address of the router itself. In special situations, the next-hop-self parameter can be altered.

The next-hop-self parameter proofed to be a very good tool to separate the bgp-traffic from the payload traffic. The next-hop-self parameter was altered to an IP-address within the Internet Exchange range (in our case 192.168.50.5). The alternation of this parameter forces the peers of AS4 to designate all the payload traffic to the next-hop-self IP-address, but it still directly communicates BGP with AS4.

8.3 Null-routing traffic

With the next-hop-self parameter, we could divert the payload traffic away from de BGP diversion mechanism. But after the diversion, the traffic needed to be null routed. Initially we thought this wasnt a big issue, but during the test we found out that this was a bigger issue than expected.

The biggest problem is, that a null route has to be able to receive an unlimited amount of traffic (until the interface is physically full), and still needs to reply on ARP[9]-requests from other hosts.

8.3.1 Normal interface

When testing the next-hop-self setup, we set the next-hop-self parameter to the IP-address of a Linux machine, which then could act as a null route interface where all the diverted data could flow to.This setup worked fine until we discovered that if the connection to this null route interface was 100% full, it wouldnt be able to send ARP-replies for its IP-address anymore.

If the host isnt able to send ARP-replies anymore, this would result in the fact that the sender host cant resolve the IP-address of the null route interface anymore, and will broadcast ARP-request packets to all the routers connected to the Internet Exchange. Combined with the fact that there is a traffic flow(destined to the specific MAC-address from machines that still have an ARP entry for this MAC-address), which the switch doesnt have a destination for in its CAM-table. The latter situation could cause a broadcast storm, because the switch broadcasts all traffic where no CAM-entry is available.

Somehow, the ARP request/reply packets have to be separated from the payload traffic. The solution might be the miss-use of ARP-proxy in combination with statically configuring MAC-addresses on a switch port. This technique is a further discussed in 8.3.3.

8.3.2 Null routing on router

Null routing on the router itself (for use with communities) did not succeed due to unknown reasons. It looks like a specific issue with specific Foundry hardware used. We did find some examples for Cisco and Juniper equipment, which indicates that is should be possible to null-route traffic on a router. Also carriers are using null-routing on their core-routers, so it is a technique that is definitely used before.

The specific issue with the foundry routers is reported to foundry America, and they are investigating this issue at this moment.

As a temporary work around we used the following construction:

[9] Address Resolution Protocol is used to match layer 2 and layer3 addresses

- create a UTP connector with send and receive wires connected to each other;
- plug the UTP connector into an unused switch port;
- add the port to an unused VLAN;
- add a virtual interface to the VLAN;
- assign the virtual interface with an IP-address (e.g. 1.2.3.1/30);
- make a static ARP entry for 1.2.3.2;
- null route the traffic to 1.2.3.2.

In the above situation, the traffic will get dropped at wire speed (in this case 100Mbit/s). There will not be any ARP-requests, since it is statically configured. The major advantage of this solution is that the packets are dropped using hardware ASICS. Whereas most other solutions will drop using the router CPU, which could result in a very high cpu load.

8.3.3 ARP-proxy

One way to seperate BGP traffic from the payload traffic could be the miss-use of proxy arp. This came up as a wild idea, but due to time restictions we where not ably to fully test it.

The arp-proxy solution works as follow:

- equip a machine with ARP-proxy software. This machine replies on ARP-requests for the null-route IP-address with the MAC-address 01:02:03:04:05:06:07;
- statically configure the MAC-address on a unused port;
- use a null-route plug (see previous paragraph) in this interface to bring it up.

With this situation, all the traffic destined to the null-route interface will be dropped at layer2, but ARP-replies still be answered by the ARP-proxy software which is running on another switch port.

8.4 Communities

Another test that we performed was the usage of communities to ask peers to drop traffic. In this case, the advertisement of a network is attached with a pre-defined community.

The receiving side (transit provider, or peering partners) can act upon this community. In this case, he would drop the traffic.

Attaching a route with a community is very simple. This is done via a *route-map*, within this route-map we specified the community ffff:0005. Every route that is announced by AS4, will be attached with this community.

The receiving part is a bit harder. Also on this site, a route-map has to be created. and This route-map has to be applied on the incoming routes of AS4. After that, the route-map needs a 'if/then' like contstruction to null-route the traffic. As can be read in paragraph 8.3.2 we where unable to configure this kind of route-map on the foundry router due to unknown reasons. The routers we used to test this construction where needed unexpectedly, so we where unable to show any configuration examples of these route-maps.

9 Future work

In this section we will discuss possible future work for this project.

9.0.1 Traffic learning and measuring

In the ideal situation, the BGP DDoS diversion mechanism is tightly coupled with a statistical network traffic measurement system. The traffic system actively monitors the traffic. A mechanism has to be designed which is able to detect abnormal traffic patterns. When an abnormal pattern is detected the owner of the specific monitored item can be notified in several ways, letting him decide what to do with the abnormal discovery. In the most ultimate situation, the process is completely automated and the BGP DDoS diversion mechanism will react to a DDoS attack without human interference.

9.0.2 Writing a RFC

To ensure that the BGP DDoS prevention diversion project becomes a standard, writing a RFC (Request For Change (also known as an Internet Draft)) would be preferable. A draft document has to be handed in at a specific department of the IETF[16]. Having a standardized project ensures you that your project will be accepted more easily within the worldwide Internet community.

10 Conclusion

During this project, we learned a lot of specific issues of the BGP protocol. During our education we had the chance to experiment a bit with this protocol, but using it with real life equipment makes the internals of this protocol more clear. We started the project with a literature study, during this period we had the time to get familiar with the different aspects of BGP and diverting with this protocol.

Looking at the different analyzed DDoS diversion mechanisms, BGP diversion is one of the most effective solutions to deal with DDoS attacks and its side effects. The diversion mechanism can be implemented at several levels in the internet topology, making it a very flexible and clear solution for diverting unwanted traffic. Unfortunately the diversion mechanism is not a full solution to defend against a DDoS attack. Traffic can be diverted, but the DDoS attack itself is still effective since the host is not reachable for the rest of the internet anymore. Another issue is that the internet community has to embrace the project, to make it a success, since its most effective if everyone uses it

References

[1] RFC 971, Internet Protocol http://www.ietf.org/rfc/rfc0791.txt

[2] Cisco Traffic Anomaly Detectors, http://www.cisco.com/en/US/products/ps5887/index.html

[3] Riverhead Guard, http://www.riverhead.com/pr/guard.html

[4] Border Gateway Protocol, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/bgp.htm#1020548

[5] SYN Flood attack, http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm

[6] Building Reliable Networks using the Border Gateway Protocol, Iljitsch Beijnum, ISBN: 0-596-00254-8

[7] The Netherlands Internet Exchange, http://www.nl-ix.net

[8] Open peering Initiative, http://www.openpeering.nl

[9] Internet Engineering Task Force, http://www.ietf.org/

[10] Proxy ARP, Cisco Systems, http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094adb.shtml#howdoesproxyarpwork

[11] Zebra routing software, http://www.zebra.org

[12] Quagga routing software, http://www.quagga.net

[13] TCPDump, http://www.tcpdump.org/

[14] Réseaux IP Européens, http://www.ripe.net

[15] Internet Assigned Numbers Authority, www.iana.org

[16] Internet Engineering Task Force, www.ietf.org

[17] Foundry Networks, www.foundrynet.com

[18] Cacti network grapher tool, www.cacti.net

[19] Mtools networking tools, http://mtools.linux.lu/?no-to-this-CONstitution

[20] Ruben Valke, http://www.os3.nl/~ruben/rp2.html, mailto:ruben@os3.nl

[21] , Wouter Borremans, http://www.os3.nl/~wborremans/rp2.html, mailto:wborremans@os3.nl

A DDoS traffic flow

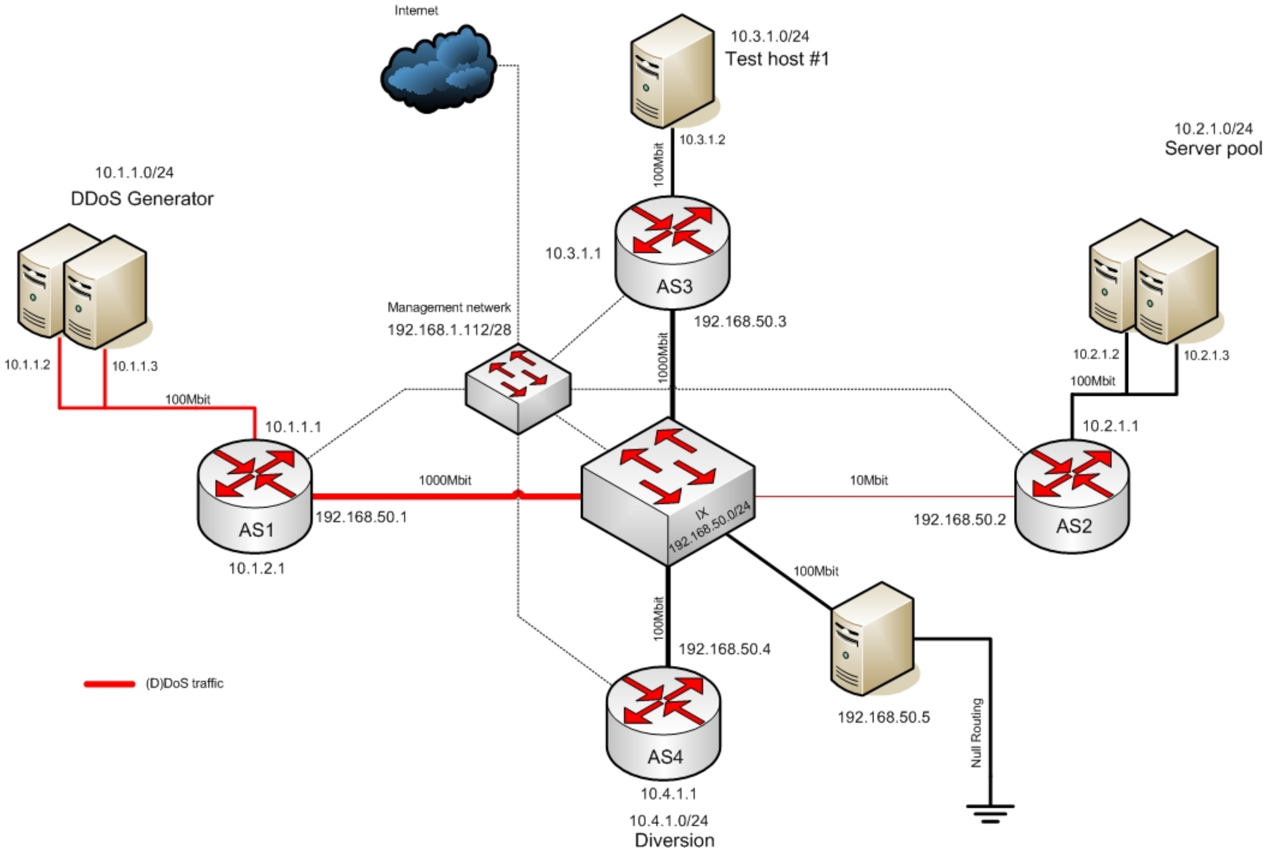

Figure 5: Test setup (D)DoS traffic flow

In figure (5) you can see the traffic flow (represented in red) which is traveling from AS1 to AS2. Because the bandwidth of the connection between the Internet exchange and AS2 is very limited (10Mbit) the line to which AS2 is reachable will completely be flooded with useless traffic resulting in AS2 becoming completely unreachable. The test carried out is a very realistic one, often seen in similar situations on real life Internet exchanges.

B Diverted (D)DoS traffic flow

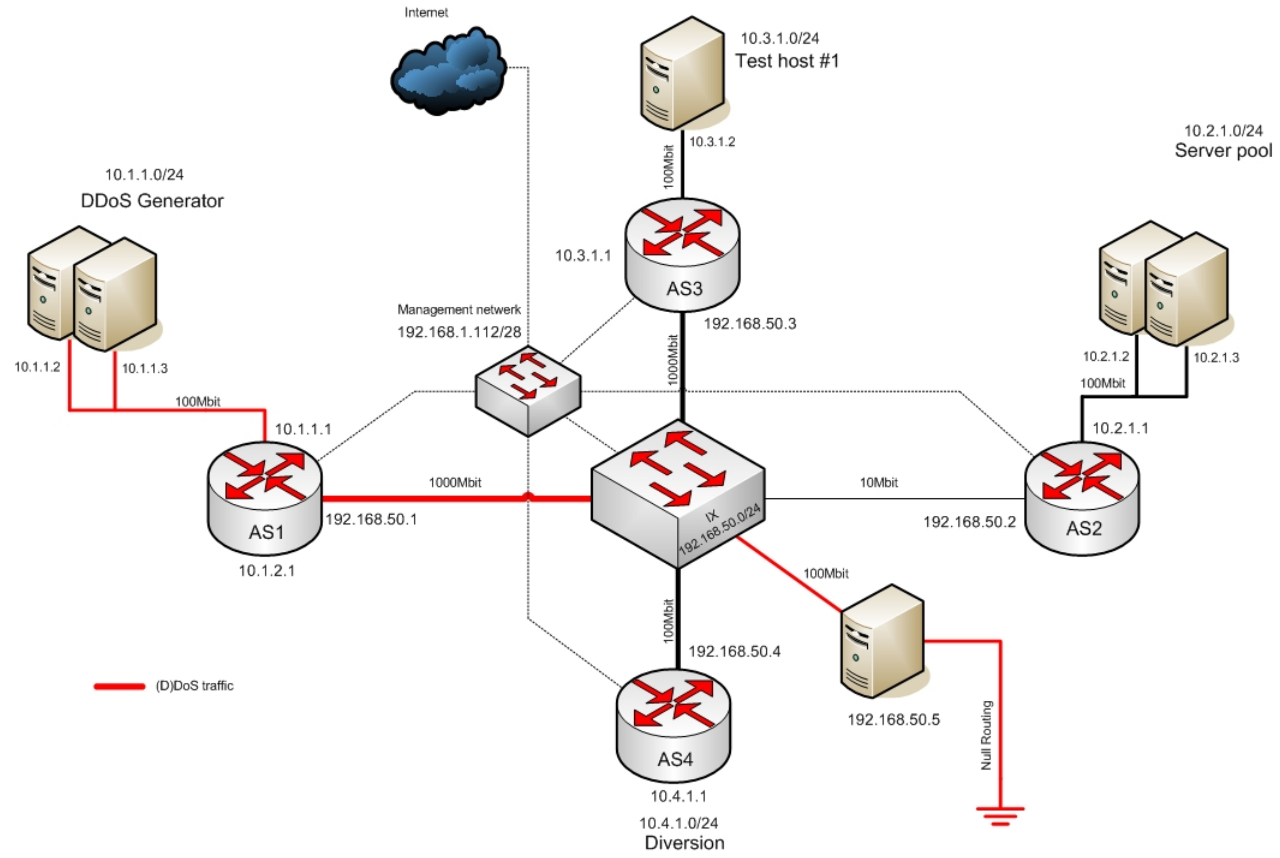

Figure 6: Test setup (D)DoS diverted traffic flow

In figure (6) you can see the traffic flow once it is diverted to the null route interface implemented within AS4. The router that represents AS4 will start announcing 10.2.1.2/32 (the attacked host) with the *next-hop-self* attribute pointing to 192.168.50.5 which is the null route interface. The (D)DoS traffic will directly be diverted away from AS2 resulting in the entire AS to become reachable again. The host with 10.2.1.2 as its IP-address will still be unreachable. When the (D)DoS attack is over, the route to 10.2.1.2/32 must be taken out of the routing table of AS4 and has to be announced by AS2 itself from that point on.

C Traffic graph (D)DoS generator (AS1)

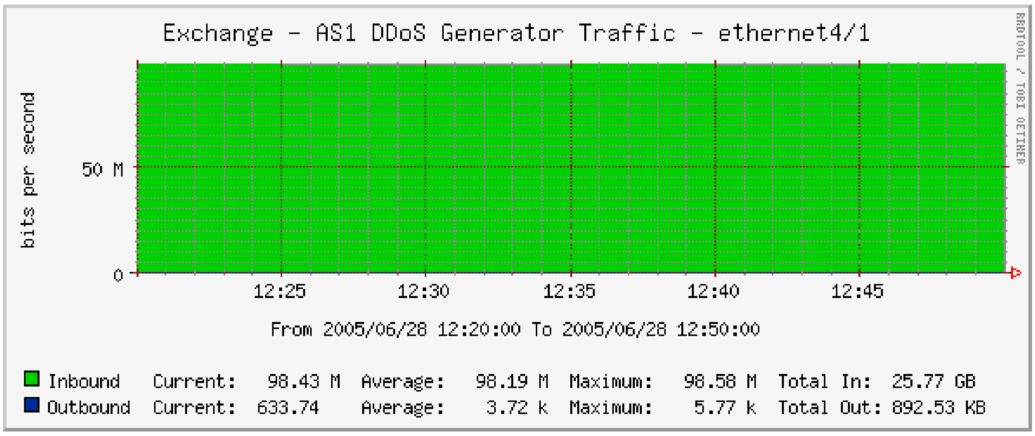

Figure 7: (D)DoS generator traffic (AS1)

In figure (7) you can see the traffic generated by the (D)DoS generator originating from AS1 as can be seen in figure (4). The graph itself is a specifically selected part over a timespan of 30 minutes to point out the amount of (D)DoS traffic which is nearly 100Mbit.

D Traffic graph (D)DoS target (AS2)

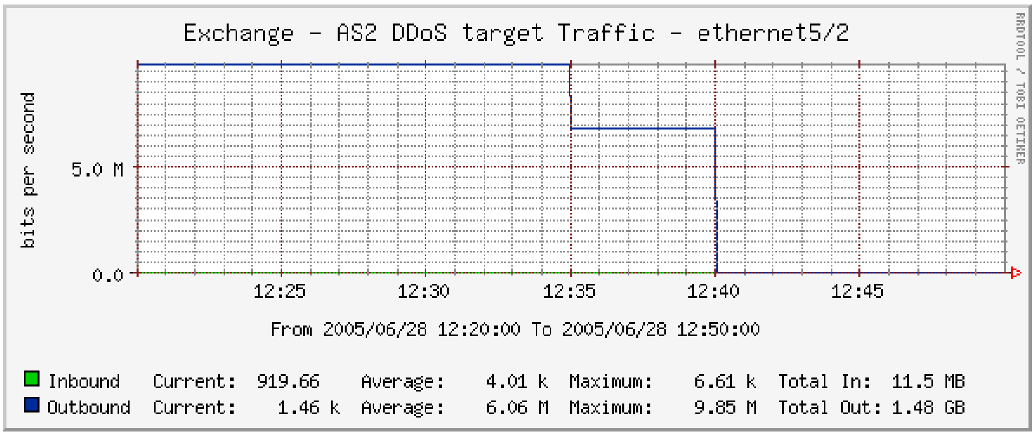

Figure 8: (D)DoS target traffic (AS2)

The graph in figure (8) shows the traffic flow between the Internet Exchange and AS2 as can be seen in figure (4). The total bandwidth of the line monitored is 10Mbit, the graph shows the line is completely filled up with useless traffic generated at AS1. In the beginning no spare bandwidth was available on the line at that specific time resulting in AS2 to become unreachable. At about 12:35 the (D)DoS attack stops and the amount of traffic dramatically drops. Since our traffic monitoring system[18] polls the router interfaces once in the 5 minutes, the graph show an average value of the measured traffic. By analyzing the graph, it appears that the traffic drops in phases, in practice the traffic drops immediately to almost zero.

E Traffic graph (D)DoS NullRoute interface (AS4)

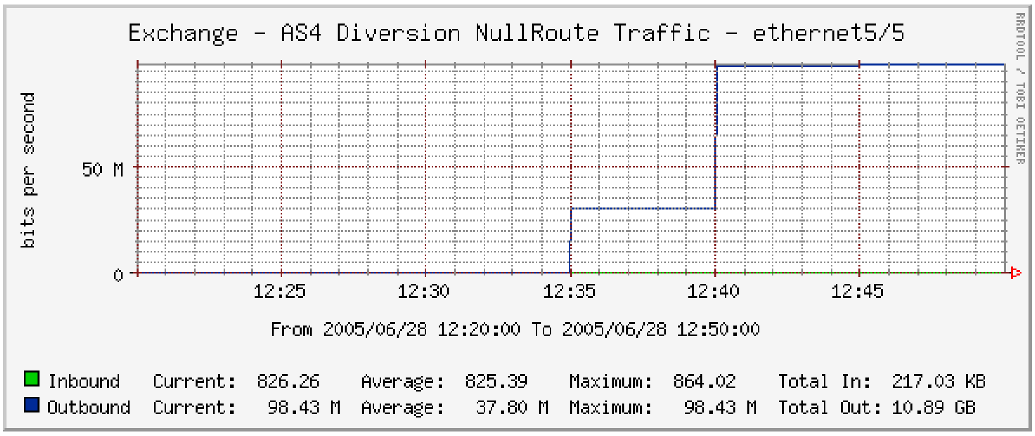

Figure 9: (D)DoS NullRoute interface traffic (AS4)

The graph in figure (9) shows the traffic at the null route interface at the BGP Diversion system implemented within AS4 as can be seen in figure (4). When we started using the BGP (D)DoS diversion mechanism to divert the (D)DoS traffic originating from AS1, you can see that at 12:35 the traffic dramatically increases to 100Mbit. This increase clearly indicates that the orginal traffic which was traveling from AS1 to AS2 is now diverted to the null route interface implemented at AS4 resulting in AS2 to become reachable again.

F Traffic graph (D)DoS diversion host(AS4)

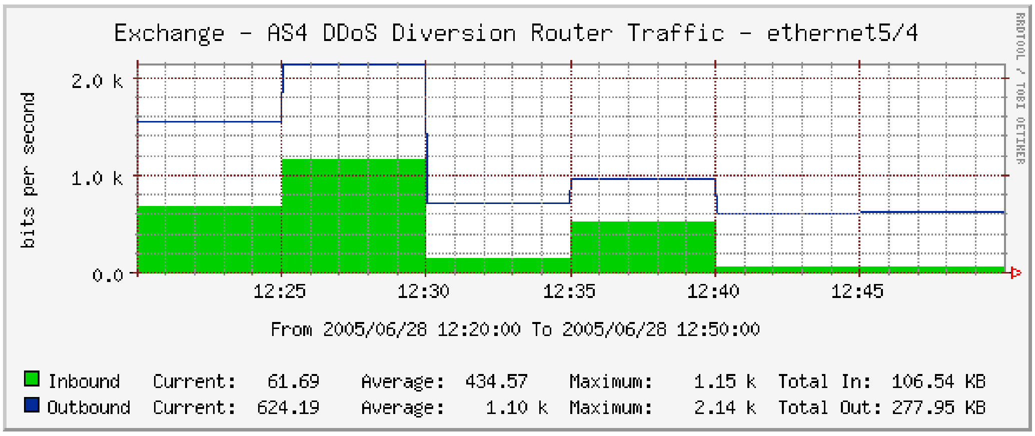

Figure 10: Traffic graph (D)DoS diversion host (AS4)

The graph in figure (10) shows another interface of the BGP (D)DoS mechanism implemented at AS4 as you can see in figure (4). This interface is used to act as an interface to speak the BGP protocol over. During the actual diversion of the (D)DoS attack at 12:35, nothing remarkable can be detected. The router itself still can be reached by other hosts within the test network, due to the traffic that is leaded to an address different from the router interface.

G Traffic graph (D)DoS diversion host(AS4)

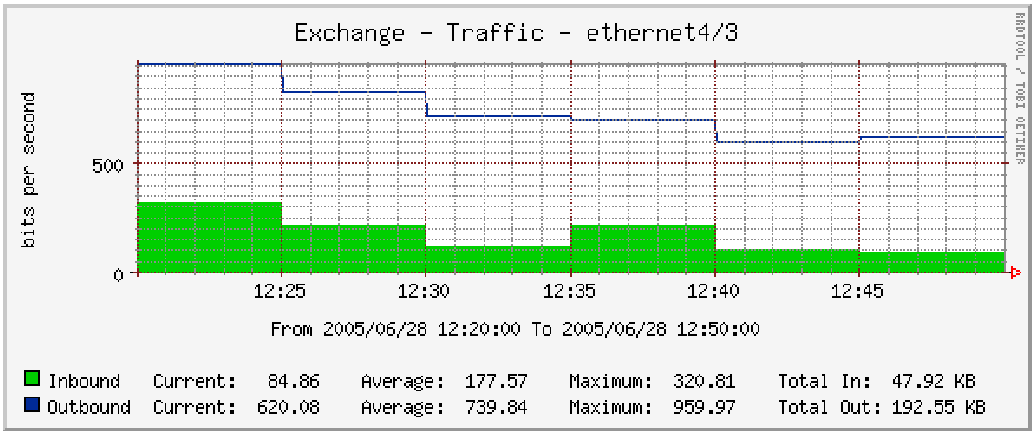

Figure 11: Traffic graph test host in AS3

The graph in figure (11) shows the traffic flow of a test host which we used to test the connectivity from AS3 to the attacked host in AS2 or to test connectivity from a host in AS3 to a host in AS4 (right through the (D)DoS traffic flow of AS1 to AS2). You can find AS3 in figure (4).

H Photo impression of test setup

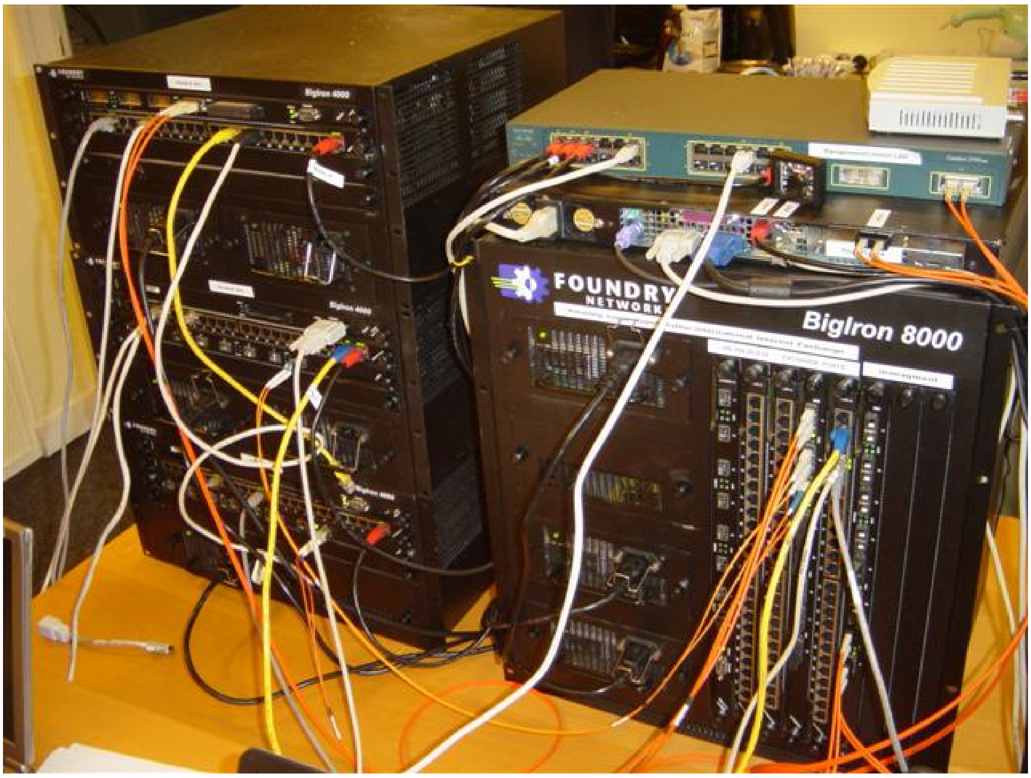

Figure 12: Test setup

In the figure above you can see the infrastructure we used to carry out our tests. On the left you can see three Foundry BigIron 4000[17] layer 2/3 devices. From bottom to top AS3, AS2, AS1 (See the logical scheme in figure (4)) The BigIron 8000 was used to interconnect the routers to each other, and thereby acting as an Internet exchange. On top of the BigIron you can see a normal server acting as a router using Zebra[11]. The Cisco Catalyst 2950 is acting as a management network switch, used to access the Internet and provide access to the telnet interfaces of the routers. The small white device is a Acer hub which we used to provide a link to test a null routing interface on a router.

I Router configuration #1 (AS1)

```
!
ver 07.6.04eT53
!
module 1 bi-4-port-gig-m4-management-module
module 2 bi-24-port-copper-module
!
global-protocol-vlan
!
!
vlan 1 name DEFAULT-VLAN by port
 router-interface ve 1
!
vlan 2 name Exchange by port
 untagged ethe 1/4
 router-interface ve 2
!
vlan 3 name managent by port
 untagged ethe 2/24
 router-interface ve 3
!
!
hostname AS1
no route-only
!
interface ve 1
 ip address 10.1.1.1 255.255.255.0
!
interface ve 2
 ip address 192.168.50.1 255.255.255.0
!
interface ve 3
 ip address 192.168.1.121 255.255.255.240
!
!
router bgp
 local-as 1
 neighbor 192.168.50.2 remote-as 2
 neighbor 192.168.50.2 soft-reconfiguration inbound
 neighbor 192.168.50.3 remote-as 3
 neighbor 192.168.50.3 soft-reconfiguration inbound
 neighbor 192.168.50.4 remote-as 4
 neighbor 192.168.50.4 soft-reconfiguration inbound
 network 10.1.1.0 255.255.255.0
!
!
!
!
end
```

J Router configuration #2 (AS2)

```
!
ver 07.6.04eT53
!
module 1 bi-0-port-m4-management-module
module 2 bi-24-port-copper-module
module 3 bi-8-port-gig-module
module 4 bi-8-port-gig-module
!
global-protocol-vlan
!
!
vlan 1 name DEFAULT-VLAN by port
 router-interface ve 1
!
vlan 2 name exchange by port
 untagged ethe 2/23 ethe 3/8
 router-interface ve 2
!
vlan 10 name management by port
 untagged ethe 2/24
 router-interface ve 3
!
!
hostname AS2
route-only
!
interface ethernet 2/23 // Interface to Exchange
 speed-duplex 10-half
!
interface ve 1
 ip address 10.2.1.1 255.255.255.0
!
interface ve 2
 ip address 192.168.50.2 255.255.255.0
!
interface ve 3
 ip address 192.168.1.122 255.255.255.240
!
!
router bgp
 local-as 2
 neighbor 192.168.50.1 remote-as 1
 neighbor 192.168.50.1 soft-reconfiguration inbound
 neighbor 192.168.50.3 remote-as 3
 neighbor 192.168.50.3 soft-reconfiguration inbound
 neighbor 192.168.50.4 remote-as 4
 neighbor 192.168.50.4 soft-reconfiguration inbound
 network 10.2.1.0 255.255.255.0
!
end
```

K Router configuration #3 (AS3)

```
ver 07.5.04T53
!
module 1 bi-0-port-m4-management-module
module 2 bi-24-port-copper-module
module 3 bi-8-port-gig-module
!
global-protocol-vlan
!
!
vlan 1 name DEFAULT-VLAN by port
 router-interface ve 1
!
vlan 2 name exchange by port
 untagged ethe 2/23 ethe 3/8
 router-interface ve 2
!
vlan 10 name managent by port
 untagged ethe 2/24
 router-interface ve 3
!
!
hostname AS3
route-only
password-change any
interface ve 1
 ip address 10.3.1.1 255.255.255.0
!
interface ve 2
 ip address 192.168.50.3 255.255.255.0
!
interface ve 3
 ip address 192.168.1.123 255.255.255.240
!
!
!
router bgp
 local-as 3
 neighbor 192.168.50.1 remote-as 1
 neighbor 192.168.50.2 remote-as 2
 neighbor 192.168.50.4 remote-as 4
 network 10.3.1.0 255.255.255.0
!
!
!
!
end
```

L Router configuration #4 (AS4) BGPD

```
!
! Zebra configuration saved from vty
!   2005/06/28 15:07:25
!
hostname AS4-BGPD
password zebra
log file /var/log/zebra/bgpd.log
!
router bgp 4
 network 10.2.1.2/32 // Adress of the diverted machine
 neighbor 192.168.50.1 remote-as 1
 neighbor 192.168.50.1 route-map blackhole in
 neighbor 192.168.50.1 route-map blackhole out
 neighbor 192.168.50.2 remote-as 2
 neighbor 192.168.50.2 route-map blackhole in
 neighbor 192.168.50.2 route-map blackhole out
 neighbor 192.168.50.3 remote-as 3
 neighbor 192.168.50.3 route-map blackhole in
 neighbor 192.168.50.3 route-map blackhole out
!
! # Route-map to route all traffic to 192.168.50.5
route-map blackhole permit 10
 set ip next-hop 192.168.50.5
!
line vty
!
```

M Router configuration #4 (AS4) Zebra

```
!
! Zebra configuration saved from vty
!   2005/06/22 18:40:47
!
hostname Router
password zebra
log file /var/log/zebra/zebra.log
!
!
line vty
!
```

N Exchange switch configuration

```
!
ver 07.6.05hT51
!
module 1 bi-8-port-gig-m4-management-module
module 2 bi-24-port-copper-module
module 3 bi-24-port-copper-module
module 4 bi-8-port-gig-module
module 5 bi-24-port-copper-module
module 6 bi-8-port-gig-module
!
hostname Exchange-switch
!
vlan 1 name DEFAULT-VLAN by port
!
vlan 10 name mangement by port
 untagged ethe 5/24
!
!
ip address 192.168.1.120 255.255.255.240
!
interface ethernet 5/2 // Port to AS2
 speed-duplex 10-half
!
!
end
```