Deliverable D6.1.1.2 < Technical report on technologies to protect against DDoS attacks >

Mikko Särelä

ICT SHOK Future Internet Programme
(ICT SHOK FI)

Phase 2: 1.6.2009 – 31.12.2010

Tivit, Yritysten tutkimus- ja kehittämisrahoitus, Päätös 516/09, 29.5.2009, Dnro 560/31/09

TKK, Tutkimusrahoituspäätös 40212/09, 29.5.2009, Dnro 925/31/09

www.futureinternet.fi

www.tivit.fi

This work was supported by TEKES as part of the Future Internet programme of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

Executive summary

Title: Technical report on technologies to protect against DDoS attacks

Distributed denial of service attacks are a major problem in today's Internet. This report surveys the existing work in mitigating denial of service attacks Most existing proposals utilize traffic or packet classification combined with proactive or reactive filtering. The classification can be based on e.g. markets, traffic analysis, or explicit information from the end points. Additionally, replication of functionality to increase resistance to unwanted traffic and hiding have been proposed and in specific instances also successfully deployed, e.g. DNS root servers.

The report also discusses the problems related to deployment of such technologies in the Internet ecosystem. The main findings are that within the Internet's valley free policy routing model, in which each AS pays its provider and receives compensation from its customers both for incoming and outgoing traffic, there may be no monetary incentives for operators in the downstream of the flow to monitor and to block denial of service traffic, unless the technology deployment is accompanied by changes in the business contracts.

Contact info: Mikko Särelä, mikko.sarela@ericsson.com

1 Introduction

The Internet was designed as an open network in which everyone can send packets to anyone. The design has been a great success largely due to the fact that anyone can create an innovative use or application that uses it. Unfortunately, the Internet also supports malicious innovative uses such as distributed denial of service (DDoS) attacks, virus, Internet worms, spam, and phishing.

One reason for this is that the architecture makes an implicit assumption that a host connected to the Internet trusts and wants to receive traffic from anyone who wishes to send to it. This means that a great number of hosts can send traffic towards a destination and overwhelm its capacity to process the data or the capacity of its incoming network link bandwidth.

In a distributed denial of service attack a large number of computers connected to the Internet start flooding the victim with data packets simultaneously. Typically, such attacks are accomplished by an attacker controlling a large number of bots, i.e. ordinary Internet connected computers, which have been hijacked earlier with malware. Such attacks utilise the distributed nature of the Internet and are especially hard to defend against. The Internet was built to allow new innovative uses of the network. Distributed denial of service attacks are an example of such a new unanticipated innovation (albeit a malicious one) that happened in the Internet.

The aim of DDoS attack is to create congestion in a chosen part of the network, often in the link between the target and the rest of the Internet, which effectively denies access to the target resources from the Internet (and Internet resources from the target). One could compare this to a large group of truckers or tractor drivers congregating in all the highways towards a major city, thus blocking access to it. With the exception that a single attacker could hijack the automated driving computers of all those cars and instruct them to do it from the other side of the world.

DDoS attacks are a common and relatively well studied problem. The purpose of this document is to survey existing work. A great number of mitigation methods have been proposed in the scientific literature. These can be divided into proof-to-send schemes, reactive filtering, replication, and some miscellaneous schemes. The mitigation schemes will be covered in more detail in Chapter 3.

A successful attack increases the amount of traffic the victim receives to a level of high congestion. One way of describing this is to consider the target as having a fixed capacity per time unit to receive traffic. Each malicious packet increases the total rate of data sent to the destination and once the rate exceeds the capacity of the target to handle all requests, some requests are dropped. The more data is sent, the more packets will be dropped, and above a certain threshold enough legitimate packets will be dropped that basically no communications with the victim.

Currently, the markets for distributed denial of service mitigation are, practically, non-existent, even though, clearly, the problem causes large costs on a number of entities in the Internet. Clearly, the costs imposed on the victims mean that there is demand for such service. It is, thus, our belief that the problems are in the supply side of the market. In this work, we analyze the Internet market structure and how it affects the creation of a security market for distributed denial of service attack mitigation.

2 DDoS Value Chain Analysis

Understanding deployability of a technology depends on understanding the business network in which it is to be deployed and the parties potentially benefiting from the change. In the case of distributed denial of service attacks, the relevant business network is the network of interconnected autonomous systems, commonly referred to as ASes, comprising the Internet.

ASes have two main types of interconnection policies, transit contracts and settlement free peering. In a transit contract, the transit customer buys access to the rest of the Internet from the transit provider. In a peering contract, two ASes, typically of roughly the same size, choose to interconnect their networks without monetary compensation in either direction. As the value of the network grows supra-linearly with the size [13] of the network, both ASes can benefit from this inter-connection, even without monetary compensation.

Having a functioning security market for distributed denial of service attack mitigation means that edge networks and transit ASes have a choice of buying protection from their transit providers and that they can monitor the performance of the provider. In this work, we concentrate on the problems that the economic structure of the Internet creates for deploying architecture mitigating the distributed denial of service attacks.

The rest of this paper is organised so that the next section describes existing architectural work on distributed denial of service attacks mitigation. We, then, continue to analyse the business structure of the interconnection market between autonomous systems that 'creates' the Internet. Finally, we provide an initial model and analysis of the system.

Background

Distributed denial of service attacks consist of a large number (thousands to millions) of hosts sending traffic, potentially masquerading as legitimate requests, to the victim, and basically causing so much congestion that little if any legitimate traffic can get through. Such attacks are commonplace in the Internet and used against governments, the attacks on Estonia [34] being possibly the best known example, content providers, corporations, and individual DSL/ cable users.[40]

Ballani et al. [9] note that mechanisms deployed purely at the victim are not useful against distributed denial of service attacks. Thus, existing architectural mechanisms mitigating distributed denial of service attacks can be divided into proactive and reactive, and connection based and filtering based approaches.

The purpose of this work is to understand the effects and requirements that arise from the business network of the Internet to the deployability of technologies designed for mitigating distributed denial of service attacks. Our hypothesis is that the pressures towards distributed denial of service attack resistance originate from edges and (aside from legislation[1]) only competition from the edges inward can push such technologies into the core of the Internet.

[1] We shall not consider regulatory aspects in this report.

The business structure of the Internet

The Internet consists of a large number of interconnected autonomous systems (AS). Each of these systems interconnects with one or more other ASes. The routing in the Internet is divided into intra-domain routing, handled purely by the AS in question and inter-domain routing. Our concern here is on understanding the inter-domain business network, as that is the business environment in which technologies against distributed denial of service attacks needs to be deployed in.

There are three main types of interconnection policies/contracts [42], which 'regulate' the BGP4 [47, 48] routing system. It is these policies that define the inter-domain topology [23, 57] and, thus, the inter-domain business network. The three main types of contract are transit agreements, peering agreements, and sibling relationships. Networks interconnect with each other, because it gives their customers access to the customers of the other networks, i.e. it is this interconnecting that forms the internet. Due to policies, the inter-connection graph is different for each end point [20, 67]. This property of the internetwork both simplifies routing and makes its analysis more complex.

In a transit agreement, a transit customer buys access and interconnection service to the global Internet from a transit provider [19]. The transit provider agrees to advertise the all the IP addresses that reside in the customer network to the rest of the Internet. At the same time, the transit customer receives routes to all Internet destinations via the transit provider.

In a peering relationship, two ASes, typically of roughly the same size, choose to interconnect their networks and their customers networks, typically without monetary compensation, allowing their and their transit customer networks to inter-connect [15]. However, it must be noted that peers do not, typically, advertise the rest of the Internet via the peering link. They only advertise their own network and their customers networks via it. There are also paid peering contracts, in which the other AS pays for the interconnection service to the other.

In a sibling relationship, a network owner has chosen to partition its own network to two (or more) autonomous systems, which have a sibling relationship. However, we leave such relationships out of our analysis making the simplifying assumption that networks owned by a single entity will operate to maximise the total value of the network rather than act in competition between each other.

The motivation to interconnect autonomous systems comes from the increased value that comes from interconnecting the users in the interconnected networks.

The peering and transit contractual relationships form a peering hierarchy. At the top of the hierarchy, there are a number of ASes that do not buy transit from anyone. They all peer with each other and peer or sell transit to others. These are called Tier 1 ASes [41]. Tier 2 ASes are ASes that both buy and sell transit from and to at least one AS. They may also peer with a number of other ASes. Content providers are autonomous systems that do not sell transit.

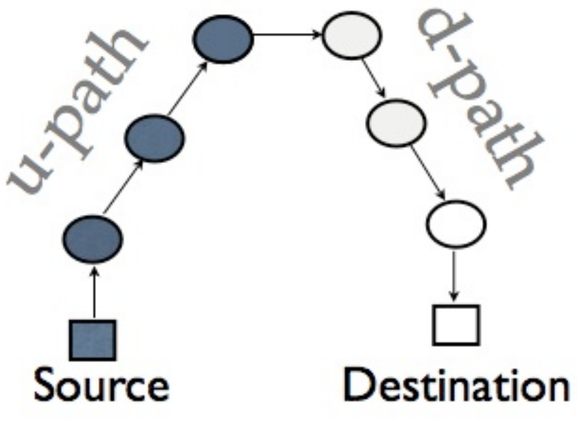

Figure 1. Valley free path

There is a crucial distinction between types of peering. In case of two Tier 1s, should they de-peer, there will be no interconnection between their and their customer networks. See the battle between Cogent and Sprint as a recent example [17, 21][2] . Tier 1's peer by necessity, there is no Internet without Tier 1 peering. On the other hand, when a Tier 2 or a content provider peers, it can always rely on its transit provider as a backup should the peering relationship prove problematic. Thus, for Tier 2s and content providers peering is a cost saving solution (and in some cases it may reduce latency and service level and thus improve customer satisfaction).

As shown in Figure 1, the peering hierarchy and policies utilised in it can be described as a valley free routing model [23], in which the route is divided into three parts.

Uphill path is the part of the path when the packet is delivered upstream through zero or more transit providers. At some point, the path may go through zero or one peer-to-peer links. And finally, it traverses the downhill path in which the packet is delivered downstream through zero or more transit links to transit customers until it arrives to its destination.

We assume the valley free model for the analysis.

Analysis

The communication capacity and its cost has large fixed investment costs and relatively low marginal costs, as long as the existing capacity is not exceeded. In the limit of the capacity, the network starts experiencing congestion, which results in lowered quality of service for customers. Such increase will then prompt, at some point, a need for additional investment into the communications infrastructure. In other words, the cost of traffic, at the margin, should be modelled as a step function.

[2] The authors take no sides in this or any such battles, we merely note that such de-peering causes the two sides to disconnect from each other.

The cost of routing packets through an autonomous system can be divided into congestion cost and investment cost. These costs are substitutes for each other, i.e. an operator may reduce congestion by increasing capacity, or reduce investment costs by accepting higher rates of congestion.

The cost of routing packets through an internetwork can be divided into intra-domain costs and interconnection costs. Using transit link involves both monetary compensation and possible cost of congestion. The exact form of monetary compensation depends on the contract between transit provider and customer, but is typically tied to peak rate using 95th percentile sampling technique [43], in which traffic is sampled over 5 minute intervals and the peak rate is defined to be the 95th percentile. Using a peering link involves investment costs in peering capacity as well as the possible cost of congestion.

Analysing costs and benefits reveals asymmetries in deployment incentives in the uphill and downhill portion, as shown in Figures 2 and 3. Starting the analysis from the edges, the victim clearly benefits from distributed denial of service attack prevention. Interestingly, also the owner of the attacking host may benefit from the prevention, after all most of the attacking hosts are hijacked from their owners using malware and spend both computational resources and network bandwidth.

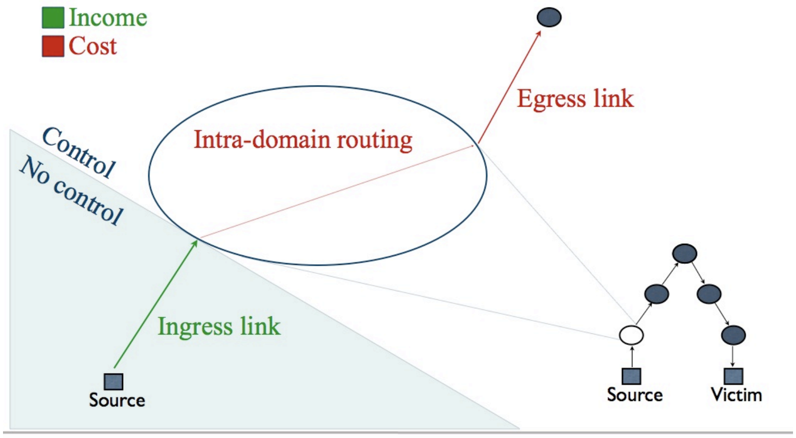

Figure 2 Upstream provider makes its money upfront and pays the costs when it forwards the packet. The upstream provider has a small, but positive, incentive to block unwanted traffic.

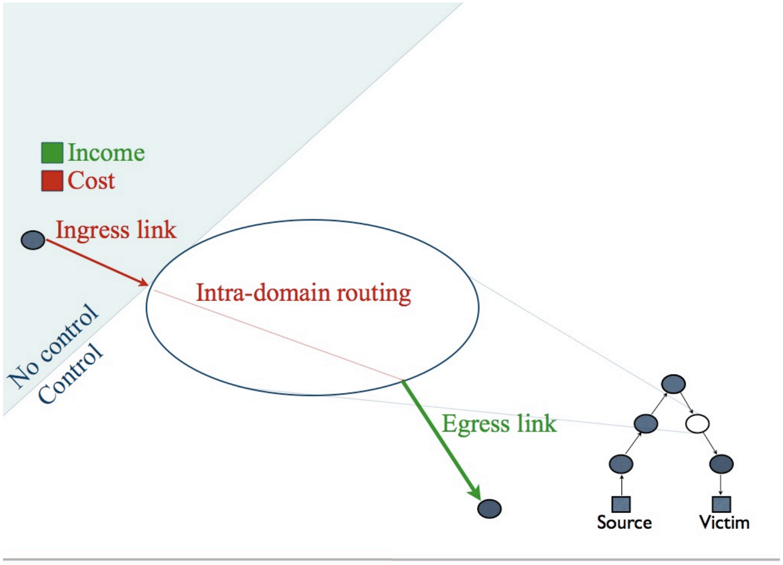

Figure 3. Downstream provider pays the costs first, and makes money only after it has delivered it to its customer. The downstream provider has a disincentive to block unwanted traffic.

On AS level the utility a network operator receives from delivering attack packets depends on the investment costs that the attack traffic causes in its own network and potential peering/ transit links and the type and amount of compensation it receives from its customers, and pays to its providers. There are three main types of compensation structures, the fixed monthly cost for most end users, per bit based, and the peak traffic based described above.

As the DSL/cable customers typically pay fixed fee per month, an edge AS serving the attacking end user hosts has only increased costs from delivering this traffic (unless it can send it directly to its transit customers), delivering it does not even improve its customer satisfaction. A transit AS in the uphill portion of the route will receive compensation from its transit customer for all packets the customer sends to its network, irrespective of whether they are delivered or not. It also has costs from its internal network usage and from using the peering or transit link, in the form of investment costs and monetary compensation, respectively.

In the other end, the AS serving the victim receives either fixed or traffic or peak traffic based compensation from the victim and pays for intra-network investment, and transit costs or peering investment. In the case of fixed monthly compensation, the above analysis applies. In the case of per bit based compensation, the AS receives compensation for each packet sent to from the victim, but also pays its provider higher transit costs. A distributed denial of service attack concentrates traffic in time and space, thus likely increasing the measured peak traffic for the duration of the attack. In the case of peak traffic based charging, which is the case for a transit AS in the downhill path, the AS pays monetary compensation for the packets it receives from its provider (and faces investment costs for the utilisation of the peering link) and receives compensation for the packets it delivers to its customers.

Our preliminary findings indicate that the AS serving the attacking hosts has the clearest and strongest direct incentive to reduce attack traffic. The direct incentives of the victim AS depend on the type of contract it has with the victim, they may be in favour of deployment or opposed to it, even if deployment cost of the resistance technology is not factored in to the analysis. However, one should remember that the link between the victim and the AS providing Internet service faces the most direct competition, thus increasing the deployment incentives, assuming that a single AS can mitigate the problem without others.

The incentives of the transit ASes have an interesting asymmetry. The upstream transit AS receives compensation when its customers send it the attack packets and has to pay for delivering them forward. If it could consistently drop the malicious packets, it could save money. The downstream transit AS, on the other hand, pays first either in the form of transit cost to its provider or in the form of higher peering link utilisation and receives monetary compensation, when delivering the packets to its customer. However, one should remember that all ASes act both as downhill and uphill ASes, depending on the direction of the traffic.

The authors believe that the primary incentives for deployment are in the edges and that competition for edge networks will prove important in driving distributed denial of service technologies up in the hierarchy. If this is the case, then it is not enough for the solution to work technically, it also has to provide required reliability and enforceability so that filtering bad traffic can be made a contractual obligation. Firstly, this means removing information asymmetries, i.e. that a transit customer needs to be able to monitor the performance of its transit provider. Secondly, the provider needs to be able to reliably implement the contractual obligations.

Summary

In this chapter, we have analysed the failure of a security market for distributed denial of service mitigation to appear. Based on our initial analysis, we hypothesise that the demand for such technology will be driven from the edges, to be precise primarily from the receiver and and secondarily from the sender end. At the same time, the incentive structure of the Internet is built in such a way that it may be harder to push such technologies to the AS hierarchy in the downhill path than in the uphill path. Additionally, we note that there is a clear danger of creating a lemon market. Unless customers can observe that their providers comply to the contractual obligations of distributed denial of service mitigation, it will be very hard to create working markets. Additionally, the providers need to be secure in their knowledge that they can reliably fulfil the contractual obligations and thus avoiding potential litigation.

3 DDoS Mitigation Technologies

There are a few basic methods which existing proposals utilise. These are replication, hiding, and traffic classification and filtering. The main focus in research community has been on traffic classification and filtering.

The application community has successfully deployed service replication for a number of individual services (see e.g. DNS root servers [], Google’s and Amazon’s distributed service []). This approach, while a solution to the problem, is costly and is affordable for those few providers that create enough revenue to justify the cost. Cloud computing is an aim to provide general purpose computing services in a widely distributed and replicated manner. They might, thus, provide an step forward in the denial of service resistant services (without help from the communication network).

The network based DDoS research has concentrated in classification techniques, though some work has also been done that concentrates on replication and hiding techniques. Packet classification methods strive for a way to order packets in such a way that good traffic gets service first. The main systems of classification are:

market based: such as quality of service agreements, congestion markers [12], puzzle auctions [61], proof-of-humanness [58]

cryptographic capabilities provided by the server [4]

target or network initiated reactive filtering [6]

Liu et al. [36] provide a comparison between capability and filtering approaches. The results indicate that neither capabilities nor filtering is better in fighting all kinds of attacks.

A prospective technology for classification and filtering has to answer the following questions:

- What elements in the network do the classification?

- What elements in the network do the filtering?

- What other infrastructure is needed for the system to work?

It is also worth keeping in mind that in aggregate the elements doing the classification and the filtering must be capable of doing it for the whole attack. In the rest of this chapter, each of these techniques is explored.

Taxonomy and miscellaneous

A taxonomy of DDoS attacks and mitigation technologies is presented in [39]. Wun et al. present a taxonomy of attacks in content based pub/sub systems [65]. Steps toward a DoS-resistant Internet architecture [25] presents a number of techniques some of which are commonly utilised: separate client and server address, non-global client addresses, RPF checking of server addresses, state setup bit as state setup packets are more risky than others, nonce exchange and puzzles for validation and pushing higher costs to initiator, middlewalls, multicast with source validation provides receiver control over which channels to receive.

Classification	In network	In packet
Proactive	Off by default	Capabilities
Reactive	Filtering	

Table 1. Classification methods

The technologies are formed of ’two components’: the boxes that are used for the deployment of the technology and the chosen method for squeezing the unwanted traffic out. The functionality used can reside, depending on the technology, in routing infrastructure, overlays, DSL boxes. Table 3.1 shows the categorization of different classification and filtering techniques and examples of techniques using the given combination. The technologies can be divided into proactive and reactive methods, on one hand. And on the other hand, to methods in which the necessary information for classification is stored in the packet and those in which it is stored in the network elements (such as routers or overlay boxes). Note, that reactive schemes require filtering information in the network, as the base case is to pass all traffic and without such enforcement nothing can force the attacker to comply with requests to add required filtering information.

Table 2 shows the main mechanisms used for mitigating DDoS attacks and lists some of their relevant properties. The values used described the referenced example in the category and may not be directly comparable.

	Pro/re active	In packet state	Forwarding state	Ref	Notes
Capabilities	P	64/ 64+N*80	~32MB/Gbps	TVA [68,69]	
Filtering	Both	-			
Pushback	R	-	32-	Pushback [28]	Hop-by-hop pro-pagation
Edge-to-edge	R	-	5s flow cache <100MB/Gbps link	StopIt [36]	Filtering at access router
End-to-end	R	-	-	AIP [3], Good Intentions [50]	Filtering at end hosts or smart NIC
Default off	P	0/N*32bits	~7MB per-formance tradeoff	Off by default [10]	Bloom filters
Hiding	P	-	TBD	Anycast,	
Replication	P	-	TBD	DONA [30]	

Table 2 shows the properties of various existing approaches for DDoS mitigation. N denotes the path length

Capabilities - Proof of Right to Send

Capabilities were first proposed by Anderson et al [4]. In a capability based DDoS resistance, each host must obtain a permission to send, i.e. a capability, from the server before allowed to send. This capability can be obtained in a variety of ways, depending on the actual details of the implementation. Typically, routers reserve a small amount of bandwidth (e.g. 5%) for capability requests sent to the server. The server then responds to those requests it wishes to give capabilities to. There are several different ways of constructing and delivering capabilities that have been proposed in the existing literature.

For example, Fastpass [63, 64] proposes a system in which the server can delegate the right to provide capabilities to third parties (who may utilise e.g. CAPTCHAs or other methods of access control before delivering the requested capability). This approach has the benefit that it allows the capability request service to be massively replicated and amortising cost among multiple services utilising it.

Anderson et al. [4] utilise a 64 bit hashed value along with a sequence number that is sent to the host via the request to send servers, which provides it to the verification point coupled with it. This hash and sequence number constitute the capability that verification points sitting near BGP routers store for each flow. To make getting new permissions easier, the hashed value is part of a hash chain. That way the server can send the next hash value to the sender directly, which utilises it along with incremented sequence number and the intermediate verification points can easily check their validity.

SIFF [66] proposed using a method for removing the need for per flow state [8] in the routers verifying the capabilities. In the system, each router stamps a few bits to the capability request. These bits are the ”last z bits” of the output of a keyed hash function with the following parameters as input: the IP address of the interface at which the packet arrived at the current router, the last-hop routers outgoing interface IP address 3, and the source and destination IP addresses of the packet being forwarded”[66]. These bits concatenated form the capability that the server gives to host requesting the capability. Thus, the sender has to include exactly those values in the future IP packets that it sends making it cryptographically hard to utilise it outside the given path.

TVA [68, 69] also utilises routers for the construction of capabilities. Each router, when receiving a capability request, attaches a pre-capability to the packet. The pre-capability contains 8 bits timestamp and 56 bits hash of src IP, dst IP, in iface, time, and secret. The server then calculates the actual capabilities from these by hashing the pre-capability, N, and T. N represents the number of bits the host is allowed to send and T the time the capability is valid. TVA also utilises hierarchical fair queueing to reduce the effects of a single bad AS with many attackers for hosts in other ASes and tries to add accountability for individual ASes in that way.

Phalanx [18] combines capabilities with an overlay. It utilises a set of mail-boxes through which traffic must traverse to be delivered to the recipient. The ISP builds a filtering ring around its perimeter that blocks traffic that does not comply with this requirement. The mailboxes receive traffic from the client and the recipient requests these packets explicitly from the mailboxes. In addition, it utilises multiple paths to reduce effects of an attack on a single mailbox.

Denial of capability attacks

Argyraki et al. argue that capability based systems have a flaw called denial of capability [7] (DoC). A solution to the problem cannot be based on capabilities and, thus, will require a different solution that could, they argue, be put to general use and render capabilities unnecessary. The counter claim is that capabilities can reduce the scope of the problem and leave the designer with a problem that is more tractable.

Puzzles can be used for levelling the playing field in getting capabilities [61]. They can be outsourced [62]. Portcullis [44] shows a system that mitigates DoC attacks and has a theoretical proof that no system can improve on the bounds of Portcullis. The case for public work [16] proposes a public work function that can be verified by anyone.

Puzzle examples

An early example of a puzzle is a requirement for a client to find a string X such that the first m bits of h(Ns , Nc , X) are zeros, where Ns is a nonce from server and Nc nonce from client. In here, m is called the puzzle difficulty and can be set by the client, though a protocol where it was set by the server can also easily be constructed. This kind of puzzle requires and consumes mainly processing power of the client and is easy for the server to check. The assumption is that the most efficient way of finding such hash is to go through a large number of strings X. [44] argues that memory based puzzles are more fair across a variety of different edge devices than computational puzzles, because the differences between computational capabilities (and available bandwidth) are much larger than the differences between memory.

Filtering

Filtering approaches are divided into reactive and pro-active mechanisms. A reactive approach installs in-network filters on-demand that (hopefully) remove the offending traffic while minimising damage to legitimate traffic. Much of the early work on filtering concentrated on finding out where the attacking packets come from. Because of source address spoofing, this problem is an important sub-problem for many filtering approaches. There are many proposed methods in the early literature. Here, we only briefly mention traceback [49, 53, 54] and Passport [35].

Reactive filtering

There are two basic methods for reactive filtering, which we shall call pushback and edge-to-edge. In both, the first step is to identify malicious flows. Once the flows have been identified, the routers can block them and either forward the blocking request to the previous hop routers as is done in Pushback [28], or if the actual origin of the flow is known the request to block can be sent directly to the source AS (which can then block the flow and ask the misbehaving originator to stop). I will call the latter approach edge-to-edge approach. It is utilised at least by AITF [6, 5], Edge-to-edge filtering architecture [27], and StopIt [36].

Taming IP packet flooding attacks [33] argues that hosts should have the ability to control their incoming traffic flows with precision. When under pressure, a host knows the traffic’s order of importance better than elements in the network and may wish to cut off specific flows or degrade all flows equally, or do use some other means of reducing load. The paper propose a router based and an overlay based system that enables receiver to control which traffic network forwards to it and which it drops.

dFence [38] utilises BGP within a single AS to introduce DDoS protecting middle boxes in a way that does not require changes in the servers, routers, nor in the clients.

The ”evil bit” [52] proposed by Simon et al. is a reactive filtering approach, which allows neighbouring ASes to organise into areas, which all agree to filter out traffic that receiver does not want and are willing to enforce source addresses that are tied to the customer records in the operator’s customer management software. Each AS in a group agree to configure their border routers to set the evil bit in all packets coming from source not in the trusted group. This usage enables classification and treatment of packets differently depending on whether they have traversed any untrusted regions or not.

Flow-cookies [14] utilise a network middle box with greater bandwidth and traffic filtering based on IP address black list from target. It also discusses the trust boundaries (and has an explicit discussion about the boundary created by a peering link as opposed to a transit - provider customer link).

It utilises a rather clever technique of making itself invisible to other boxes. It is an extension of SYN-cookies. The cookie box in the network does the initial TCP handshake and then hands the connection over to the server. For all TCP packets that carry an ACK flag, the cookie box checks the validity of the flow cookie (which is a keyed hash of the connection 4-tuple). The system is backwards compatible, as it induces a legacy client to include this hash by using TCP timestamp option. The timestamp works as follows: the sender puts a timestamp into a packet and the receiver then echoes it back. They have verified that the approach works in practice.

The pushback approach pushes filtering requests upstream hop-by-hop. Such approach has incentive problems when it tries to pass over AS boundaries. This can be corrected by changing contracts; however, the processes of changing contract can be more difficult and costly than one might assume. If the flow comes from an upstream AS, it means that the AS in question may be paying money to receive that traffic to its upstream provider and only recoups the money by charging its customer for it. If a filtering solution blocks the stream inside it, then it ends up paying but not receiving.

The easy solution of changing the contract in such a way that the customer is required to pay for the blocked traffic is problematic as it cannot observe the amount of traffic its provider blocks. If, on the other hand, the payment is by blocked flow or some other aggregate, the provider takes risks that the costs from the flow are greater than the payment it receives (which will complicate the contractual negotiations and potentially drive up the prices from efficient level).

The Edge-to-edge approach requires deployment at both ends of the communicating flows and some method to reliably find the domain level path or the origin of the flow. As source addresses themselves can be spoofed, they cannot be directly utilised.

The filtering of bad traffic can either be done by blocking it completely or by reducing its share to a fair share of the network bandwidth or to only allow it to pass as long as it does not cause congestion (or some other schema). The design must also explain what to do, if the AS in the sender's end does not co-operate. AITF [5] proposes that such ASes should be blocked totally by default to increase incentives for the ASes to co-operate. However, if the receiving AS has resources, it can do filtering on more fine grained level.

The source AS can also ask the originating host to stop sending the flow in question and thus offload some of its resource needs to the violating host. It can keep the filter in its cache (i.e. slow and cheap memory such as DRAM) and assume that the source host complies. When it receives a filtering request, it will first check if such filter is in its cache to see if the filter already exists. If it does, then it categorises the source as a non-responsive malicious host and blocks it totally. However, if the host lies behind a NAT, then the block may cause collateral damage among the hosts using the same NAT address. Nevertheless, such a block forms a powerful incentive for the operator of the network to comply with such requests (and the ability to comply could be incorporated into the NAT box/firewall, which would minimise the need for end hosts to upgrade).

The edge-to-edge filtering approaches need to decide what to do with traffic from legacy networks that do not support the architecture (as they cannot by definition respond to block requests). The basic main are to block those completely (which is rather harsh), or to block individual flows per recipient requests or give them lower priority than traffic from those who have upgraded to the filtering approach.

The findings from AITF and StopIt are that it takes hundreds of seconds for current reactive filtering schemes to setup the required filters in the network once an attack has begun. This means that it can effectively cut all existing TCP communications due to timeouts. In the meanwhile, the destination is blocked. Additionally, due to memory size constraints in filtering devices and the need to make filters temporary, the attacker may be able to cause problems periodically.

End-to-end filtering is a special case of edge-to-edge filtering as it happens directly between end hosts, or their network equipment: Holding the Internet accountable [3] suggests an Internet architecture, which utilises addresses that are composed of AS based and end host based part, each of which is a flat public key cryptography based identifier. This prevents source address spoofing and enables them to construct a simple end-to-end filtering scheme (based on shut-off messages and smart NICs).

Leveraging good intentions [50] utilises the fact that most hosts participating in an attack are actually owned by well meaning owners and used for bad purposes by an outsider who has gained control of the machine from afar. Thus, the actual owners are, at least, not opposed to the idea that an end-to-end protocol stops an outsider from using their machine for an attack. The paper proposes a separate end-to-end control protocol. Both hosts can use it to signal the other to stop sending for a while.

Proactive filtering

Off by default [10] proposes an architecture in which by default routers block packets unless there is a filter that approves sender to send to the receiver in each router along the path. To make the system scalable, the filters in the routers are based on bloom filters [11]. Bloom filters are probabilistic so that false positives are possible. In this case, it means that a single router may forward packets not approved, but never drop a packet that has been approved. The approach, however, suffers from long connection setup times, as it takes tens of seconds for the filters to propagate in an Internet sized network.

Puzzle auctions can be used for filtering request and levelling playing field between attackers and legitimate senders [61]. Another similar method is DDoS defense by offense[59, 60] in which the server encourages all users to ’speak up’ i.e. use more bandwidth, thus increasing the share of resources that good clients have. In effect, this approach uses the fair queueing used by today’s routers as a filtering method where congestion in the network occurs.

One problem with such method comes, if the server is under-provisioned compared to the number of real customers there are. This would cause the thinner to send constant speak-up replies to clients and would permanently raise the amount of traffic going through the network.

The puzzle auctions utilises a first price auction mechanism, in which each sender sends additional bytes on a side channel and when the auctioneer hands out another slot for the server, the one with the highest number of bytes wins and his account is set to zero. Would the system have better properties, if the price enacted was the second price instead? And similarly, if the server was running multiple processors and thus several access prizes were auctioned simultaneously, would this change the situation? At least, it seems that it could make the auction more fair, but might have problems with attackers being capable of building large 'balance' in their account over time.

Packet symmetry can also be utilised for flow classification [31]. The basic idea is that legitimate traffic seems to be bidirectional and have certain amount of symmetry to it. Thus, one could use this to classify flows into good and bad and then filter out the bad traffic, which is highly skewed in one direction. It is based on the idea that it is much easier for a receiver to indicate the traffic it likes than to indicate the traffic it does not like. "We argue that while end-to-end design is vital to maximise freedom to innovate, the network must enforce a higher degree of mutual consent between communicating hosts" [31]. Packet symmetry embeds this concept of mutual consent into the network. They propose that the filter would be placed as close to the sender as possible, preferably to the NIC (or potentially to the local ISP).

Should one somehow make a distinction between the initiator and responder in the network? Such distinctions have served well in protocol design. After all, I think, no good design would have the initiator pouring huge amount of traffic to a responder who does not indicate a wish to receive; whereas the opposite might hold true, as the fact of initiating is an indication of willingness to communicate. However, unless source address spoofing is prevented, attackers could forge symmetry by sending each other packets utilising the attack target's source address. Note that if the symmetry filter is deployed in the NIC, it can also do source address validation.

Replication and diffusion

Replication and diffusion is still relatively little researched area and it contains more questions than answers. Some services, such as the root DNS servers, already use such methods.

Internet sites can be ordered according to the amount of traffic they invite, either during normal operations or during their peak traffic (excluding DDoS attack traffic). Some sites have such large traffic finger print that even today's DDoS attacks seem to be unable to increase their load enough to take them offline. Google is probably a prime example of such. For such sites, investment in sufficient bandwidth, memory, and computation will likely be enough to allow them to withstand large scale DDoS attacks today. Other, smaller sites, are not as lucky.

Services for which the average traffic is relatively small, but the peak is high face poor performance during peak, high costs on average. One answer is to find ways of aggregating, i.e. pooling, the server resources with other similar services that do not have coinciding traffic peaks. An important question is how to orchestrate a set of services with differing peaks, so that an attacker cannot manipulate them to coincide with his attack?

Data can be replicated with relative ease. DONA [30] shows that it can even be done reactively when demand for certain piece of data increases. The biggest hurdle to such approaches in today's Internet is that the only method we have of recognising/naming the data is tied to the name of the entity servicing it (i.e. utilising URLs which first refer to a DNS name and then to a file system path or a locally understandable database 'query').

Example things and tasks that have been suggested for replication. These tend to be things that can be separately deployed in a manner that serves a large number of different Internet services:

- Filtering (overlay or routers)

- Data
- CAPTCHAs
- User authentication

Overlays

SOS: Secure Overlay Service [29]. Mayday [2] generalises on SOS. They propose an architecture, in which a filtering ring around the server blocks all traffic destined to it except for those coming from authorised nodes in the overlay. The verification is done using simple non-cryptographic methods, such as source address or destination address filtering, destination port filtering, etc.

FONET [32] is similar to SOS/Mayday. It utilises a technique called p2cast between overlay nodes and from the overlay to the protected server. P2cast is a unicast technique similar to PIMSSM (protocol independent source specific multicast) except it only allows unicast traffic. This seems to reduce the chances of successful attacks against the overlay compared to SOS.

(Secure) Internet Indirection Infrastructure i3 [56, 1] proposes an overlay which enables communicating hosts to keep their IP addresses hidden. The overlay utilises a distributed hash table called Chord.

Stavrou and Keromytis propose Stateless Multipath Overlays [55], which extends overlay based filtering approaches with multipath capability. This enables the overlay to withstand relatively large targeted attacks with small performance degradation for end-to-end traffic.

OverDoSe [51] hides protected nodes at IP level, allowing only overlay nodes to communicate with the target. The overlay is placed at the ISP in strategic locations to enable filtering all traffic to destination that does not pass the overlay.

Hi3 [24] proposes separating control plane from data plane using Host Identity Protocol [] and utilizing an overlay for connection setup.

There are also a few anycast based solutions: Firebreak [22] and stateful anycast [26]. Stateful anycast extends the traditional anycast to support stateful sessions. In other words, it ensures that a session, once initiated, will continue to be routed to the same host (and not one of the other hosts using the same anycast address). This way it is possible to deploy a set of boxes in the network protecting the target from DDoS attacks. For example, one can make the targets IP addresses unreachable by anyone else but the ISP proxies that utilise anycast addresses.

Congestion

DDoS attacks can be seen as congestion problem and thus methods used for congestion control may also mitigate DDoS attacks. Decongestion control [45] is a method in which hosts send at full rate and vary the coding of the data in packets as a response to congestion, instead of varying sending rate. Game based approach to congestion control is propose by Lukyanenko et al. [37]

Re-feedback (Re-ECN) [12] proposes to reuse explicit congestion notification [46] in a way that allows an explicit market in congestion in a single round trip timeframe. It gives many possibilities for creating a congestion market. In between AS traffic, the operators could charge not only based on volume, but also based on actual congestion caused and in the end user market an operator could, for example, use congestion pricing or sell different flat price plans each with a congestion allowance. The allowance would determine the number of congestion markers that the end user could place in its packets per time unit (e.g. second).

4 Open problems

There are still many open issues for most of the approaches suggested in the literature. These include compatibility of suggested protocols with network reality, e.g. middle-boxes such as NAT, firewalls, and layer 4 (TCP) middleboxes. The main question, however, is a question of deployment. Are there features in the current network system: inter-domain routing, intra-domain routing, etc. that can be utilised for DDoS mitigation, or will new systems need to be deployed? If new systems are deployed, they can be deployed either as part of the routing infrastructure, middle-boxes, or overlays (which are essentially a distributed group of middle-boxes working in concert).

Bibliography

[1] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. Towards a more functional and secure network infrastructure. 2003.

[2] D. Andersen. Mayday: Distributed filtering for internet services. USITS: 4th USENIX Symposium on Internet Technologies and Systems, 2003.

[3] D. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Holding the internet accountable. Hotnets-VI, Nov 2007.

[4] T. Anderson, T. Roscoe, and D. Wetherall. Preventing internet denial- of-service with capabilities. Hotnets II, pages 39–44, 2004.

[5] K. Argyraki and D. Cheriton. Scalable network-layer defense against internet bandwidth-flooding attacks. ACM/IEEE ToN.

[6] K. Argyraki and D. Cheriton. Active internet traffic filtering: Real-time response to denial-of-service attacks. Usenix, 2005.

[7] K. Argyraki and D. Cheriton. Network capabilities: The good, the bad and the ugly. ACM HotNets-IV, Jan 2005.

[8] T. Aura, P. Nikander, and J. Leiwo. DOS-resistant authentication with client puzzles. Lecture Notes in Computer Science, 2133(170-177):3–10, 2001.

[9] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker. ”Off by default!”. In In Proc. of the 4th ACM Workshop on Hot Topics in Networks (HotNets), 2005.

[10] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker. Off by default. ACM HotNets 2004, Aug 2005.

[11] B. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. Communications of ACM, 13, 1970.

[12] B. Briscoe, A. Jacquet, C. D. Cairano-Gilfedder, A. Salvatori, A. Soppera, and M. Koyabe. Policing congestion response in an internetwork using re-feedback. SIGCOMM '05, pages 277–288, Jun 2005.

[13] B. Briscoe, A. Odlyzko, and B. Tilly. Metcalfe's Law is Wrong. IEEE Spectrum, 3, 2006.

[14] M. Casado, P. Cao, A. Akella, and N. Provos. Flow-cookies: Using bandwidth amplification to defend against ddos flooding attacks. 14th IEEE International Workshop on Quality of Service, 2006. IWQoS 2006, pages 286–287, 2006.

[15] H. Chang, S. Jamin, and W. Willinger. To Peer or not to Peer: Modeling the Evolution of the Internets AS-level Topology. In IEEE Infocom, pages 1–12, 2006.

[16] W. chang Feng and E. Kaiser. The case for public work. IEEE Global Internet Symposium, pages 43–48, 2007.

[17] http://www.earthtimes.org/articles/show/sprint-nextel-severs-its- internet-connection-to-cogent-communications,603138.shtml.

[18] C. Dixon and T. Anderson. Phalanx: Withstanding multimillion-node botnets. Usenix NSDI, 2008.

[19] P. Faratin, D. Clark, P. Gilmore, S. Bauer, A. Berger, and W. Lehr. Complexity of internet interconnections: Technology, incentives and implications for policy. The 35th Research Conference on Communication, Information and Internet Policy (TPRC), 2007.

[20] J. Feigenbaum and M. Schapira. Incentive-compatible interdomain routing. ACM New York, NY, USA, 2006.

[21] Telecom knockout. http://www.forbes.com/technology/forbes/2008/1013/064.html, September 2008.

[22] P. Francis. Firebreak: An ip perimeter defense architecture. Hotnets Reject, 2004.

[23] L. Gao. On inferring autonomous system relationships in the Internet. Networking, IEEE/ACM Transactions on, 9(6):733–745, Dec 2001.

[24] A. Gurtov, D. Korzun, A. Lukyanenko, and P. Nikander. Hi3: An efficient and secure networking architecture for mobile hosts. Computer Communications, 31:2457–2467, Mar 2008.

[25] M. Handley and A. Greenhalgh. Steps towards a dos-resistant internet architecture. Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, pages 49–56, 2004.

[26] R. E. Hansen. Stateful anycast for ddos mitigation. CSAIL, TR-2007-035, Jun 2007.

[27] F. Huici and M. Handley. An edge-to-edge filtering architecture against dos. 2007.

[28] J. Ioannidis and S. Bellovin. Implementing pushback: Router-based defense against ddos attacks. In Proceedings of NDSS, 2002.

[29] A. Keromytis, V. Misra, and D. Rubenstein. Sos: secure overlay services. SIGCOMM, 32 (4):61–72, 2002.

[30] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pages 181–192, 2007.

[31] C. Kreibich, A. Warfield, J. Crowcroft, S. Hand, and I. Pratt. Using packet symmetry to curtail malicious traffic. Hotnets-IV, 2005.

[32] J. Kurian and K. Sarac. Fonet: A federated overlay network for dos defense in the internet. Proceedings of Global Internet Symposium, 2006.

[33] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica. Taming ip packet flooding attacks. Sigcomm Computer Communication Review, 34(1):45–50, 2004.

[34] M. Lesk. The New Front Line: Estonia under Cyberassault. IEEE SECURITY & PRIVACY, pages 76–79, 2007.

[35] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and Adoptable Source Authentication. In USENIX/ACM NSDI, 2008.

[36] X. Liu, X. Yang, and Y. Lu. To filter or to authorize: network-layer dos defense against multimillion-node botnets. Proceedings of the ACM SIGCOMM 2008 conference on Data communication, pages 195–206, 2008.

[37] A. Lukyanenko and A. Gurtov. Towards behavioral control in multiplayer network games. GameComm’08, page 7, Jun 2008.

[38] A. Mahimkar, J. Dange, V. Shmatikov, H. Vin, and Y. Zhang. dfence: Transparent network-based denial of service mitigation. NSDI’07, 2007.

[39] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. Sigcomm Computer Communication Review, 34(2):39–53, 2004.

[40] D. Moore, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. In Proceedings of the 10th Usenix Security Symposium, pages 9–22, 2001.

[41] W. Norton. Internet Service Providers and Peering. In Proceedings of NANOG, volume 19, pages 1–17, 2001.

[42] W. Norton. A Business Case for ISP Peering. White Paper, February, 2002.

[43] W. Norton. A Business Case for ISP Peering. White Paper, February, 2002.

[44] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y. Hu. Portcullis: Protecting connection setup from denial-of-capability attacks. 2007.

[45] B. Raghavan and A. C. Snoeren. Decongestion control. Hotnets-V, Nov 2006.

[46] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP, 2001.

[47] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771 (Draft Standard), Mar. 1995. Obsoleted by RFC 4271.

[48] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), Jan. 2006.

[49] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In ACM SIGCOMM, 2000.

[50] M. Shaw. Leveraging good intentions to reduce unwanted network traffic. Proc. USENIX Steps to Reduce Unwanted Traffic on the Internet workshop, 2006.

[51] E. Shi, I. Stoica, D. Andersen, and A. Perrig. Overdose: A generic ddos protection service using an overlay network. Technical report, CMU-CS-06-114, Carnegie Mellon University, Feb 2006

[52] D. Simon, S. Agarwal, and D. Maltz. As-based accountability as a cost-effective ddos defense.

[53] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, and W. Strayer. Hash-Based IP Traceback. In ACM SIGCOMM, 2001.

[54] D. Song and A. Perrig. Advance and Authenticated Marking Schemes for IP Traceback. In IEEE Infocom, 2001.

[55] A. Stavrou and A. Keromytis. Countering dos attacks with stateless multipath overlays. 12th ACM conference on Computer and communications security, pages 249–259, 2005.

[56] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. Proceedings of the 2002 SIGCOMM conference, 32(4):73–86, 2002.

[57] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2, 2001.

[58] L. von Ahn, M. Blum, N. J. Hopper, and L. J. CAPTCHA: Using Hard AI Problems For Security. In EUROCRYPT03, May 2003.

[59] M. Walfish, H. Balakrishnan, D. Karger, and S. Shenker. Dos: Fighting fire with fire. 4th ACM Workshop on Hot Topics in Networks (HotNets), 2005.

[60] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker. Ddos defense by offense. conference on Applications, technologies, architectures, and protocols for computer communications, pages 303–314, 2006.

[61] X. Wang and M. Reiter. Defending against denial-of-service attacks with puzzle auctions. In proceedings of Symposium on Security and Privacy, 2003, pages 78–92, 2003.

[62] B. Waters, J. A. Halderman, A. Juels, and E. W. Felten. New client puzzle outsourcing techniques for dos resistance. 11th ACM conference on Computer and communications security, pages 246–256, 2004.

[63] D. Wendlandt, D. Andersen, and A. Perrig. Fastpass: Providing first packet delivery. Technical report CMU cylab, 2006.

[64] D. Wendlant, D. Andersen, and A. Perrig. Bypassing network flooding attacks using fastpass.

[65] A. Wun, A. Cheung, and H.-A. Jacobsen. A taxonomy for denial of service attacks in content-based publish/subscribe systems. Inaugural International Conference on Distributed Event-Based Systems, 233, Jun 2007.

[66] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. Security and Privacy, pages 130–143, 2004.

[67] X. Yang, D. Clark, and A. Berger. NIRA: A New Inter-Domain Routing Architecture. IEEE ACM TRANSACTIONS ON NETWORKING, 15(4):775, 2007.

[68] X. Yang, D. Wetherall, and T. Anderson. A dos-limiting network architecture. SIGCOMM, 2005.

[69] X. Yang, D. Wetherall, and T. Anderson. Tva: A dos-limiting network architecture. IEEE/ ACM Transactions on Networking, 16(6):1267–1280, 2008.