DDoS Protections for SMTP Servers

Michael Still mikal@stillhq.com
School of Computer Science
The Australian National University
ACT 0200 Australia

Eric C. McCreath ericm@cs.anu.edu.au
School of Computer Science
The Australian National University
ACT 0200 Australia

Abstract

Many businesses rely on email of some form for their day to day operation. This is especially true for product support organizations, who are largely unable to perform their role in the company if their in boxes are flooded with malicious email, or if important email is delayed because of the processing of attack traffic. Simple Message Transfer Protocol (SMTP) is the Internet protocol for the transmission of these emails. Denial of Service (DoS) attacks are deliberate attempts by an attacker to disrupt the normal operation of a service with the goal of stopping legitimate requests for the service from being processed. This disruption normally takes the form of large delays in responding to requests, dropped requests, and other service interruptions.

In this paper we explore the current state of research into Distributed Denial of Service (DDoS) attack detection, protection and mitigation for SMTP servers connected to the Internet. We find that whilst there has been significant research into DDoS protection and detection generally, much of it is not relevant to SMTP servers. During our survey we found only two papers directly addressing defending SMTP servers against such attacks.

Keywords: Distributed Denial of Service, email, Simple Mail Transfer Protocol, Survey Paper.

1. INTRODUCTION

Allman [4] states that spam costs US businesses $87 billion a year. It seems reasonable to assume that if a low level attack is costing that much, then a complete outage would impose an even greater burden on an enterprise. Interestingly, despite the importance of SMTP to modern business operations, little research appears to have been applied to how to protect SMTP from deliberate attack, apart from whatever protection may be derived from generic defenses.

SMTP is a unique protocol in terms of its needs of DDoS protection. This is largely because of the need to sync queued email to disk, so as to not lose email in the queue in the case of a system failure. In fact, SMTP is an unusually easy protocol to DDoS [8], requiring relatively small amounts of bandwidth to render inoperable. Also SMTP is of increasing importance to modern business operations, yet, approaches focused on DDoS protection for SMTP have not gained much attention. These factors make DDoS protection for SMTP an area of research interest and significance.

Denial of service attacks may be grouped into two main categories:

1. Attacks that exploit flaws in the implementation of the server system, normally in the form of misconfigurations [34, 23, 22, 31, 33, 36]. For example SYN flooding works on the assumption that the server's TCP implementation allocates memory for the TCP connection at the time that the SYN packet is received. The attacker therefore sends many SYN packets, but does not ACK connection establishment when the server offers it. The server therefore has this memory allocated until the TCP connection times out [12]. Modern operating systems either limit the number of connections per source, or use techniques such as SYN cookies to avoid allocating memory at the time of the SYN packet. These vulnerabilites may exist at the application layer as well as the operating system layer. For example, if you can send a request that causes the application server to crash, then you have denied access to that server until it can be restarted, either manually or automatically. Another example is a request that takes a disproportionally long time to respond to – for example, early versions of Microsoft's IIS web server would take extremely long times to parse certain malformed URLs [30].
2. An attack is simply a distributed attempt to consume all of a scarse resource [31, 33, 23, 36] such as CPU, network IO or disk IO. These attacks are termed Distributed Denial of Service (DDoS) attacks [22] as the flood traffic comes from many machines, and is not a single flow on the network [27]. When an attack targets a host's upstream network bandwidth specifically, then it is often termed a "bandwidth attack" [23, 36]. These flooding attacks are often not detected by traditional signature detection schemes [25], and are harder to defend against with simple address based filtering. Often these attacks use clients which forge their sender address, such forging if used is known as IP spoofing. These clients are known as zombies [19, 33], and are often poorly secured home machines on broadband connections [46]. A group of zombies under the control of a single hacker (or group of hackers) is known as a botnet. This flooding behavior is exacerbated by these attacking zombies ignoring TCP flow control mechanisms, whereas legitimate clients will reduce the size of their traffic flows – thus increasing the proportion of traffic which is malicious [36]. Worse, these attacks do not imply a mis-configuration on the part of the site administrator, and are much harder to defend against. The implementation of bandwidth attacks is based on the volume of requests, not the content of the requests [36].

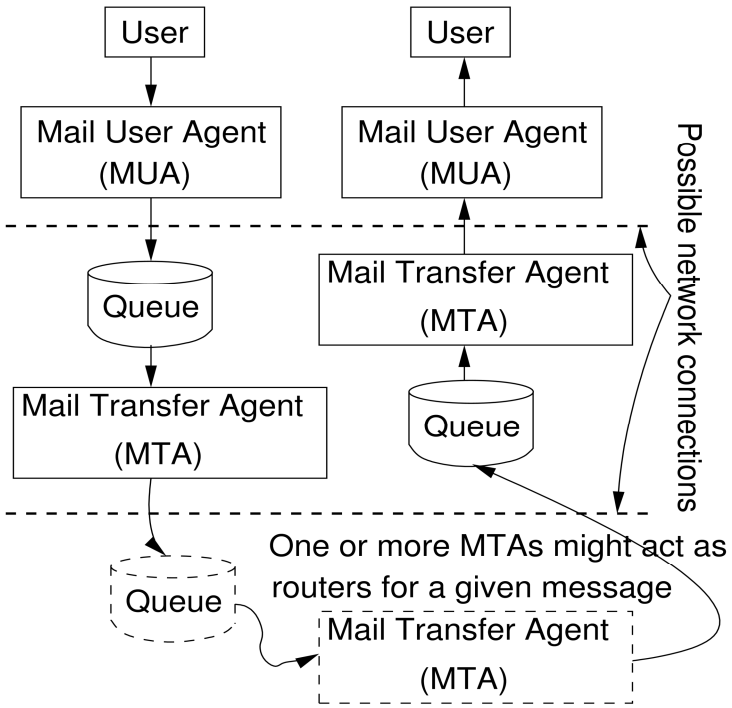

FIGURE 1: A common SMTP path

Both of these categories apply to SMTP servers. For example, a recent instance of a SMTP specific vulnerability in the implementation of server software is [14], which details a MX record parsing vulnerability in some Microsoft SMTP server implementations. However, this review focuses on the second form of attack, where a large number of requests are being made at any one time. This is because SMTP servers are unusually vulnerable to these distributed attacks because of the relatively low amount of bandwidth [8] required to saturate the available disk bandwidth of current servers.

Interestingly, the requests sent as part of DDoS are not necessarily malicious, they are just timed in such a way as to cause disruption to normal levels of service. For example, a large number of incoming emails following some sort of catastrophic event such as the 9/11 attacks on New York's World Trade Center can be characterized as a DDoS despite the intent not being malicious. The event causing the flood of emails also does not need to be catestrophic – large email newletter campaigns have also been known to cause SMTP servers to stop responding to requests in a reasonable amount of time [8].

2. SMTP ROUTING PRIMER

It is important to briefly introduce how an email is typically routed by SMTP servers, as this is important background to the DDoS protections discussed later in this review. This section is only a brief summary however, and reference to the relevant RFCs as well as Stevens' description in [39] are recommended for more detail. A common SMTP path is shown in Figure 1. In this example, the user creates an email using a Mail User Agent (MUA), and when they select the send action the mail is delivered to a Mail Transport Agent (MTA) on either the same machine as the MUA, or another machine.

This MTA routes the email to the destination Mail eXchanger (MX), possibly via a number of other MTAs depending on local configuration. The MX is simply another MTA, but it is listed in the Domain Name System (DNS) as being capable of delivering email for the destination domain name. At the MX, the email is sent through another variable length chain of MTAs until it reaches the MTA that can deliver mail to the recipient's mailbox on disk. This final MTA then uses a Mail

Delivery Agent (MDA) which may be built into the MTA or be a separate program such as procmail to actually write the message to disk. The mailbox is then checked periodically by another MUA, which displays the mail to the recipient. It is possible that there is a network connection between the mailbox and the destination MUA, or they might be on the same machine.

There are a few more aspects of this design that deserve more attention:

- This is an unusually complicated path. Most email will flow from a MUA to a local MTA, via one routing MTA (called a smart host) to the destination MX's MTA, and then to the recipient MUA. There is little research to support this assertion however.
- Every MTA along the delivery path is required to reliably add the email to its queue, as once the email is accepted, it is deleted from the sender's queue. This incurs costly disk syncs to ensure the data is queued reliability.
- It is possible to insert additional MTAs in the delivery path, which act much like proxies. This is commonly done to implement functionality such as virus and spam scanning. These checks can be extremely expensive to execute, this slows this MTA down futher.
- There are very few guarantees for how quickly an email will be delivered. This will depend on the number of MTAs in the mail's path, how busy they are, and how long the mail stays in each queue before being processed.

A successful SMTP DDoS needs only to cause congestion on the last provisioned portion of this path to cause an outage for the end user.

3. PRIOR DDoS RESEARCH

There has been extensive research into DDoS attacks. This section discusses this research in the context of SMTP servers specifically. We discuss: how common DDoS attacks are; existing methods for detecting attacks; and finally existing attack defenses. Unfortunately, not much of this research has examined SMTP specifically. The existing research specifically addressing SMTP servers that we could find was [8, 9]. We therefore comment on the specific implications of existing research on SMTP as appropriate.

3.1 How Common are DDoS Attacks?

CERT data indicates that security attacks overall are becoming much more common – so common in fact that CERT no longer reports individual incidents [36]. There is also existing research into the prevalence of DDoS attacks, which finds that the volume of attack traffic arriving at networks is significant. For example, Pang et al. [35] find that the Lawrence Berkeley National Laboratory experienced 8 million connection attempts to unused addresses in just one day. This was two thirds of the traffic received on that day. Moore et al. [33] found that the rate of attacks is relatively constant, although it has nearly tripled in the three years of sampling the paper covers. Clearly, scanning and attempted DDoS attacks are common on the modern Internet.

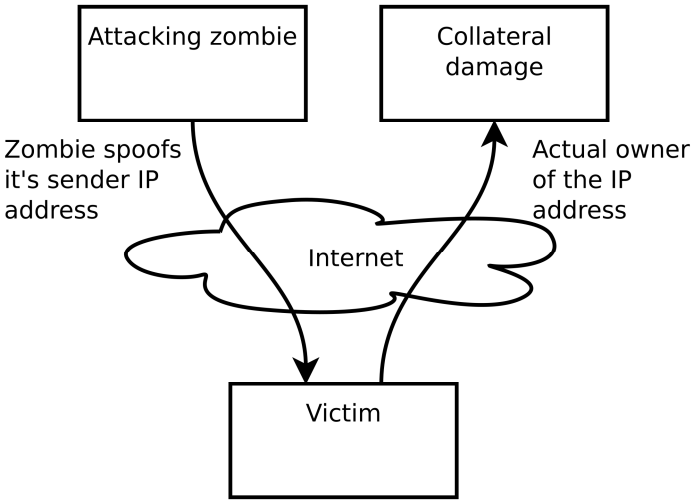

FIGURE 2: Conceptual overview of backscatter analysis

It should be noted that this research is largely dependent on either packet capture of some form[1] or backscatter analysis [33, 35].

Backscatter is essentially a form of network level "collateral damage". Figure 2 shows that when a zombie sends an attack packet with a forged sender IP address, it runs the risk of either using a real machine's IP address as the source address, or an address from an unallocated block of network addresses. When the server under attack attempts to reply to the attack packet, it will instead reply to the real owner of the address. This real owner can detect that an unsolicited packet has arrived because of the state of the IP connection at the time the packet arrives.

Backscatter analysis is the process of using these unsolicited packets to infer information about DDoS attacks occurring on the larger Internet. More discussion of this technique may be found in [33], although it should be noted that not all unsolicited traffic is an indication of a spoofed attack packet [35], for example network scanning and broadcast packets with all result in unsolicited traffic. The utility of this form of analysis is reduced by the decrease in the number of attacks using spoofing [28].

There are now several teams of researchers using large unallocated IP address blocks to analyze the backscatter from DDoS attacks which use address spoofing for example [36]. They refer to their packet capture setups as "network telescopes" because of the manner in which they amplify the signal from distant events.

These techniques however do not provide estimates for how commonly SMTP servers are attacked. SMTP servers are advertised in MX records for domain names, and are unlikely to be attacked based on simple scans for IP address space. In fact, random IP addresses are quite unlikely to run into a SMTP server, as shown by our recent surveys of SMTP servers on the Internet. Additionally, because SMTP traffic is not vulnerable to spoofing, backscatter analysis provides no assistance. We are unable to find any reference in existing research to the prevalence of SMTP DDoS attacks.

3.2 Attack Detection Methods

1 And therefore the requirement that the researchers have access to a network which either was attacked, was attacking, or provided transit to an attack.

It is important to successful DDoS mitigation that attack detection is both quick, and unlikely to incorrectly identify non-malicious traffic as an attack2 [36]. The system also needs to be able to implement an effective response that favours legitimate traffic once an attack is detected [32].

There are three main forms of DDoS detection discussed in the literature [31, 22, 25]:

1. Pattern detection – these techniques seek to find patterns in requests, and then determine if those patterns are associated with legitimate requests. Often these systems have predefined lists of signatures which indicate a common attack. These specific behaviours (such as executing a port scan) are considered indicative of malicious intent. This technique is widely deployed in the form of many Intrusion Detection Systems (IDS) such as snort [38]. Such schemes are possibly better implemented at a higher level of the network stack, where more information about the connection between the client and server is known (such as the user who is currently connected) [47].
2. Anomaly detection – a base line for "normal" traffic is generated and then used to identify possible attacks. These anomalies may be in the form of unusual traffic flows (for example a large amount of traffic to a machine which generally receives little), or a behavior (for example a failure to respect TCP flow control mechanisms for a TCP flow) [36]. This is hard to do on real networks, as traffic flows can be highly variable, whilst not being malicious. However, this approach holds the most promise for SMTP as anomalies would present themselves as unusual traffic flows, either in a larger than normal number of emails being delivered to one recipient, or a larger number of emails than usual coming from a limited number of clients [5]. Further research into this option is desirable. The baseline data for these anomaly detection systems is often at the packet level. There has been some discussion that moving these systems to the TCP layer would provide a more holistic view of flows and therefore improve the accuracy of attack detection [47]. Further moving this anomaly detection to the application layer would provide further benefits – such as knowledge of the specific users which are creating flows.
3. Third party detection – these are systems which do not perform any attack detection themselves, but act on instructions from an external source. This might be in the form of a commercial service, or a network wide traceback mechanism such as CenterTrack [5, 40].

3.3 Evaluation of Attack Detection Methods

Extensive research has been conducted into generic DDoS attack detection. However, there have been limited research into how to make these detection schemes scale well. One example of such research is [25], which investigates aggregation techniques as a method of improving performance. Many of these existing techniques, such as port scan detection, are currently implemented in the form of large vectors which do not scale to high data rates [25]. Aggregation of flows is an option for improving performance, but this can result in "behavioral aliasing" where either an aggregate falsely identifies non-malicious traffic as malicious, or an aggregate which fails to identify malicious traffic because it is masked by otherwise unrelated non-malicious traffic in the flow [25].

There is promise for these detection techniques for detecting attacks against SMTP servers, although there is currently little research into how to perform this detection. The only directly relevant research the authors have found during their review is [7]. Here, a very naive anomaly detection algorithm is used, with attack protection being triggered by overall processing queue length hitting a defined threshold (either queue overflow, or queue length meeting defined parameters). Once protection is triggered, attack traffic is identified by looking for network adresses with higher means than normal. This method is vulnerable to "traffic laundering" through constructs such as botnets, as individual network addresses can still be responsible for very small amounts of traffic, and the widely distributed. Traffic from identified sources is then discarded.

Whilst this implementation shows promise, it suffers from naive triffering and simplistic behaviour once triggered. We believe that attempting to cluster traffic using a variety of attributes would be a more accurate triggering mechanism, and possibly would be able to be used permanently, instead of only when in "attack mode". Future research into this area would be promising.

3.4 Attack Defenses

In this section we enumerate the various DDoS attack defenses discussed in the literature, and provide an evaluation of their effectiveness in the case of SMTP servers. Most of the existing evaluations assume that a solution to DDoS attacks should either be implemented at the source of the attack [44, 46], or be built into TCP/IP itself [31, 5, 25, 36]. Also some proactive approaches are possible [45].

DDoS attack defenses overall may be grouped into four categories:

1. Over provisioning – provide enough server capacity to handle the system peak load, plus a concurrent DDoS attack. This is a common technique, despite difficulty in predicting the largest DDoS attack which might occur. An example of this technique is Content Delivery Networks (CDN)s covered in Section 3.8. This is probably the most common approach employed by likely targets of SMTP DDoS, with many such organizations deploying large clusters of mail servers.
2. Routing controls – have attack traffic not routed to the server under attack. Examples include: some forms of overlay networks (Section 3.8); push back mechanisms (Section 3.7); various changes to core Internet protocols such as reworking how network addresses are allocated (Section 3.6). This approach holds significant promise for SMTP servers, and has been initially investigated by Bencsáth [9].
3. Currency proposals – DDoS attacks are premised on the assumption that clients are cheap and that servers are expensive. If this is made no longer true by making client connections more expensive, then many attackers will no longer be able to afford significant traffic levels. Currency does not have to be monetary – another commonly cited proposal is to use proof of expenditure of computational resources as currency – for example the computation of hashes. These are covered more in Section 3.5.
4. Authentication systems – such as whitelists[2]; blacklists[3]; and CAPTCHAs[4] .

3.5 Currency Proposals

One class of proposals to stop both spammers and DDoS attacks is to change the economic model used by the attackers. Both spammers and zombies operate on the assumption that clients are cheap, and that many may be used at once. There are currency proposals which aim to change this. Currency proposals include:

- System resources – teergruben-like systems [16] extend the length of possible spam SMTP connections dramatically, in an attempt to have spammers use capacity in their TCP stacks as payment for having sent the spam. Additionally, general tar pitting systems[5] are useful for rate limiting some forms of abusive sender [21].
- Expended effort – for example, proof that the sender has consumed a certain minimum number of CPU cycles in order to allow the delivery of this one email. Examples include Microsoft's Penny Black project [18, 17, 1, 2] and Hashcash[6]. Generally these schemes use the computation of hashes as proof of resource consumption.
- Money – finally, these are escrow proposals where actual money is held by a third party on the promise that the request from the client is not malicious. The escrow payment is released if the recipient agrees.

These proposals offer an interesting solution to DDoS attacks, as they make it more expensive to attempt to flood a server with traffic. However, these proposals suffer from the same practical limitations as ingress and egress filtering – to be effective they require a large scale deployment,

2 A list of users or servers always allowed to connect.
3 A list of users or servers never allowed to connect.
4 A simple character recognition puzzle used to separate machines from humans.
5 Algorithms which increasingly slow connections from systems which are deemed to be using more than their fair share of a finite resource such as server capacity.

which is difficult to achieve on the Internet. Additionally, computational time on zombie machines is effectively free, so expending resources is not a large burden in this case.

3.6 Address Allocation Changes

Handley and Greenhalgh [20], propose breaking the IP address space into "server" and "client" addresses. Clients would then be able to only initiate connections to servers, which would respond. Servers would not be allowed to initiate their own connections to clients, and clients would not be allowed to connect to other clients. They argue that this will stop zombies from receiving commands over the network. The authors also argue in favor of changes to the IP protocol to make it clearer when a session is still being setup.

The assertion that breaking the address space into client and server addresses would stop zombies from receiving commands ignores the possibility of the zombies polling a server for commands, which is common already because of the widespread use of Network Address Translation (NAT). This proposal also ignores the breakage of peer to peer applications that this proposal causes, although the authors address this by the suggestion of either providing two addresses to some machines, or building a network of proxies to forward on the connections from these client machines, which undermines the separation concept. Additionally in the SMTP case it is common for "client IPs" to contact servers. For example, roaming laptops and mobile phones often end email via remote authenticating SMTP servers to simplify configuraton.

3.7 Push Back Mechanisms

Push back is a mechanism in which routers upstream of the server under attack[6] are asked to start dropping packets to the server under attack [26, 27]. They address the failure of attacking zombie machines to respond correctly to TCP flow control mechanisms [36], as IP assumes that a client will respond to such requests with a reduced traffic level [26]. Lakshminarayanan et al. [26] proposes that push back be implemented by allowing hosts to add filtering rules to the router on the ISP's side of the network link offering transit to the attack, based on the assumption that the ISP is better provisioned to handle the level of traffic caused by the attack than the transit link is. This assumes that it is the network link to the server that is the resource being saturated during the attack. A proposed implementation is as shown in Figure 3.

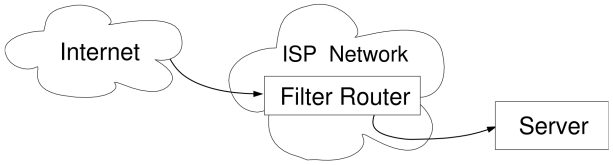

Figure 3: A remote traffic filtering implementation

In this case the server would push updates to the filtering rules to the router to protect the link between the server and the ISP's router from saturation. If the ISP was unwilling to allow customers to update filters on their routers, then the client could host their own proxy at the ISP, which implements the filtering required before sending legitimate traffic through the router to the server. The server under attack can then update the filter rules on the proxy without affecting the ISP's routing configuration.

Accordng to Bencsath's early research into their utility with SMTP [9], push back mechanisms show promise because they offer a means of controlling aggregate flows which are the result of

6 These routers are not necessarily well placed in the Internet's topology. The criteria is simply that they are between the attacking machines and the server under attack, and are better connected to the Internet than the server under attack. This is attractive because it is much more likely that these routers are under the same control as the server under attack, which means it is more likely that filtering rules can be implemented quickly compared with routers closer to the origin of the attack.

combining many low bandwidth flows into one larger flow. These smaller flows might all individually respect flow control mechanisms, but when combined still cause an overload condition [27]. Such mechanisms also offer assistance in the handling of unexpected traffic from "flash crowds" (a large unexpected burst of otherwise legitimate user traffic - sometimes referred to as the "Slashdot effect") [27]. These push back mechanisms are not a perfect solution to DDoS attacks, and in some cases can in fact make the situation worse [27].

3.8 Overlay Networks

There are DDoS attack protections that do not require the modification of core Internet protocols, forexample proposals that harness overlay networks to provide protection. Lakshminarayanan et al. [26] argue that the ability of a host to control the traffic sent to it is fundamental to the solution to the DDoS problem, especially as it is the end hosts who know the most about the traffic flows they are receiving. Whilst push back mechanisms go some way to offering control of the traffic sent to a server, Lakshminarayanan et al. are representative of the group of researchers who argue that further control is needed. Therefore, there are several proposals for proxy services which make the servers that provide appli cations which might be attacked anonymous. The idea is that you cannot attack something which you cannot find, and that the proxy network is so over-provisioned that it isn't vulnerable to realistic DDoS attacks. These proxy networks are a special case of an "overlay network".

More generally an overlay network is a network constructed on top of another network. There are a number of proposals[24, 26] which utilize an overlay network based on Internet Indirection Infrastructure (*i3*). In the *i3* network, a host registers an identifier, and packets requiring that host identifier are sent to *i3. i3* then looks up the identifier, and forwards the traffic onto the host. It is argued that because servers are not widely known to the Internet, they do not expose other ports than those required to implement the application to attack. Further, a server can stop traffic flow by simply unregistering its identifier (although this will affect legitimate users of the service as well). An i3 based proxy system also allows for the implementation of "next generation" IP services such as mobile clients, multicast and anycast.

As mentioned earlier, overlay networks are not always proxy based. An example of a non-proxy overlay network is VIPnet, proposed by Brustoloni in [11]. VIPnet implements a DDoS mitigation system by offering preferred routing to important users of a server in return for payments to the transit ISPs for the VIP user. Clearly this is not a generally applicable solution. Another non-proxy overlay network is CenterTrack [40], which uses an overlay equipped with IP traceback capable routers to perform network traceback on networks otherwise not capable of performing such analysis [36].

The goal of DDoS protection mechanisms is to minimize the harm to genuine users of an attack on the service. Two common ways to providing this harm minimization is to either stop the malicious requests from consuming resources on the server, or massively over provision the system [31] so that these malicious requests do not affect the requests of genuine users. For relatively static content which needs to be served globally, a common technique is to implement a Content Distribution Network (CDN), also known as Content Delivery Networks. CDNs are used to increase throughput for popular or vulnerable sites [37, 43, 10, 42, 41]. The most well known of these CDNs is run by a company called Akamai, and is composed of over 20,000 servers operating in 71 countries and 1,000 networks [41, 3]. Akamai deploys these servers onto ISP networks at no charge to the ISP. This is attractive to ISPs as it reduces their bandwidth expenses. CDNs are constructed from a set of geographically distributed proxies (also known as surrogates), which return results instead of the sites main servers. CDNs have often been compared to peer to peer (P2P) download networks [15, 37]. One notable difference is that CDNs are centrally controlled and managed, whereas P2P networks are not. CDNs have a number of advantages:

1. It moves the content closer to the user, thus reducing latency when fulfilling requests. This is because the TCP three way handshake happens over a much lower latency network path, thus improving TCP session setup speed.

2. It reduces peering and transit costs for ISPs by allowing them to reduce the number of times the same object must traverse their peering links, because the content is hosted at the ISP it need only be transferred over the ISP's peering links once.
3. It limits the region which is affected by a DoS attack. For example, if an attacker is on a network with a CDN proxy for the site they are attacking, then their requests will be responded to by that local proxy. Therefore, the only effect of the DoS attack is to reduce the speed of the site for other users of that local network, not all users of the site around the world.

CDNs are only useful for sites where the data to be added to the CDN is read only and where personalization mechanisms such as cookies are not allowed to reduce the cache-ability of objects from the site. This is especially true for sites which require cookies for all requests, even those where none is needed [10]. We are unaware of any CDN provider which currently supports distributing SMTP servers.

4. THE CURRENT STATE OF SMTP

What is the current state of SMTP servers on the Internet? These servers face several challenges, including consistent low level attacks from spammers, as well as email borne viruses and worms. DDoS protections for SMTP servers can be informed by previous work on these problems.

Unsolicited Commercial Email, also known as spam, may be characterized as illegitimate requests coming from many machines[7], however the request rate is low enough that it does not cause server outages and therefore cannot be characterized as a DDoS attack. Current estimates of spam rates indicate that up to 74.5% of emails sent are spam[29]. Whilst this is a significant percentage of the current SMTP traffic levels on the Internet, it has now been sustained for so long that it is considered part of the status quo and SMTP servers connected to the Internet are configured to handle the current spam workload.

Current spam detection techniques can be broken into two broad categories: content based techniques; and sender behavior based techniques. The content based techniques commonly used are [46]:

1. Email address filters – also known as origin-based filters [13]. These are simply lists of email senders or emails servers who are known spammers (a blacklist), known non-spammers (a whitelist) and possibly suspicious senders (a greylist).
2. Heuristic filters, including machine learning approaches, based on known spam features – for example words such as "viagra" [13].

Wong et al. [44] determine that outgoing email worms can be detected from the pattern of DNS requests that they make when sending their email. They propose implementing a mail worm watchdog on DNS servers to alert when worm email is being sent.

5. CONCLUSION

We have brought together some of the research on DDoS from the perspective of SMTP. This provides a useful starting point for research in this area. For there is significant scope and value of future research into: the state of SMTP transactions on the Internet, the vulnerability of SMTP servers to DDoS attacks, and the creation of defense approaches.

We believe that a viable approach to SMTP server DDoS protection is to deploy push back routers as intermediaries between the senders of email and the receiving server, as described in [7]. These servers could be deployed much like a Content Delivery Network, and therefore provide protection for more than one SMTP server at any given time. However, the push back routers should archive email which is categorized as having a high probability of being an attack,

7 Some of these machines in fact being zombies.

and this email should be processed by the recipient servers during non-peak periods where further analysis of the traffic is possible. Further work is also required on attack traffic detection.

6. REFERENCES

[1] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber. Bankable Postage for Network Services. In Proceedings of the 8th Asian Computing Science Conference. Springer-Verlag, 2003.

[2] M. Abadi, M. Burrows, M. Manasse, and T. Wobber. Moderately hard, memory-bound functions. ACM Transactions on Internet Technology (TOIT), 5(2):299–327, 2005.

[3] Akamai. Technology overview, 2007. Available from http://www.akamai.com/html/technology/index.htm, accessed on 5 July 2007.

[4] Eric Allman. Spam, Spam, Spam, Spam, Spam, the FTC, and Spam. Queue, 1(6):62–69, 2003.

[5] Tom Anderson, Timothy Roscoe, and David Wetherall. Preventing Internet denial-of-service with capabilities. SIGCOMM Comput. Commun. Rev., 34(1):39–44, 2004.

[6] Adam Back. Hashcash - A Denial of Service Counter-Measuree, 2002. Available from http://www.hashcash.org/papers/hashcash.pdf, accessed on 7 July 2007.

[7] Boldizsár Bencsáth. New Approaches to Mitigate Network Denial-of-Service Problems. PhD thesis, BME Informatikai Tudományok doktori iskola, 2009.

[8] Boldizsár Bencsáth and Miklós Aurél Rónai. Empirical analysis of denial of service attack against smtp servers. In Proceedings of The 2007 International Symposium on Collaborative Technologies and Systems, pages 72–79. IEEE, 2007.

[9] Boldizsár Bencsáth and István Vajda. Protection against ddos attacks based on traffic level measurements. In 2004 International Symposium on Collaborative Technologies and Systems, pages 22–28., San Diego, CA, USA, January 2004.

[10] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 522–533, New York, NY, USA, 2004.

[11] JosÃl' Brustoloni. Protecting electronic commerce from distributed denial-of-service attacks. In WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 553–561,New York, NY, USA, 2002.

[12] CERT. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, 1996. Available from http://www.cert.org/advisories/CA-1996-21.html, accessed on 4 October 2007.

[13] Duncan Cook, Jacky Hartnett, Kevin Manderson, and Joel Scanlan. Catching spam before it arrives: domain specific dynamic blacklists. In ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research, pages 193–202, Darlinghurst, Australia, Australia, 2006.

[14] Microsoft Corporation. Microsoft Security Bulletin MS10-024: Vulnerabilities in microsoft exchange and windows smtp service could allow denial of service (981832), April 2010.

[15] Shibsankar Das and Jussi Kangasharju. Evaluation of network impact of content distribution mechanisms. In InfoScale '06: Proceedings of the 1st international conference on Scalable information systems, page 35, New York, NY, USA, 2006.

[16] Lutz Donnerhacke. Teergurbing FAQ. Available from http://www.iks-jena.de/mitarb/lutz/usenet/teergrube.en.html, accessed on 12 November 2007.

[17] C. Dwork, A. Goldberg, and M. Naor. On memory-bound functions for fighting spam. Advances on Cryptology (CRYPTO 2003), Santa Barbara, CA, USA, August, 2003.

[18] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail. Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, pages 139–147, 1992.

[19] Hikmat Farhat. Protecting TCP services from denial of service attacks. In LSAD '06: Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, pages 155–160, New York, NY, USA, 2006.

[20] Mark Handley and Adam Greenhalgh. Steps towards a DoS-resistant internet architecture. In FDNA'04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, pages 49–56, New York, NY, USA, 2004.

[21] Tim Hunter, Paul Terry, and Alan Judge. Distributed Tarpitting: Impeding Spam Across Multiple Servers. In LISA '03: Proceedings of the 17th USENIX conference on System administration, pages 223–236, Berkeley, CA, USA, 2003.

[22] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. A framework for classifying denial of service attacks. In SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 99–110, New York, NY, USA, 2003.

[23] Frank Kargl, Joern Maier, and Michael Weber. Protecting web servers from distributed denial of service attacks. In WWW '01: Proceedings of the 10th international conference on World Wide Web, pages 514–524, New York, NY, USA, 2001.

[24] Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. SOS: secure overlay services. In SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pages 61–72, New York, NY, USA, 2002.

[25] Ramana Rao Kompella, Sumeet Singh, and George Varghese. On scalable attack detection in the network. In IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pages 187–200, New York, NY, USA, 2004.

[26] Karthik Lakshminarayanan, Daniel Adkins, Adrian Perrig, and Ion Stoica. Taming IP packet flooding attacks. SIGCOMM Comput. Commun. Rev., 34(1):45–50, 2004.

[27] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. Controlling high bandwidth aggregates in the network. SIGCOMM Comput. Commun. Rev., 32(3):62–73, 2002.

[28] Z. Morley Mao, Vyas Sekar, Oliver Spatscheck, Jacobus van der Merwe, and Rangarajan Vasudevan. Analyzing large DDoS attacks using multiple data sources. In LSAD '06: Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, pages 161–168, New York, NY, USA, 2006.

[29] MessageLabs. MessageLabs Intelligence. Available from http://www.messagelabs.com/mlireport/MLI_Report_October_2007.pdf, accessed on 12 November 2007.

[30] Microsoft. Microsoft Security Bulletin (MS00-030): Frequently Asked Questions. Available from http://www.microsoft.com/technet/security/bulletin/fq00-030.mspx, accessed on 12/11/2007.

[31] Jelena Mirkovic and Peter Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Comput. Commun. Rev., 34(2):39–53, 2004.

[32] Jelena Mirkovic, Max Robinson, and Peter Reiher. Alliance formation for DDoS defense. In NSPW '03: Proceedings of the 2003 workshop on New security paradigms, pages 11–18, New York, NY, USA, 2003.

[33] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. ACM Trans. Comput. Syst., 24(2):115–139, 2006.

[34] Judith M. Myerson. Identifying enterprise network vulnerabilities. Int. J. Netw. Manag., 12(3):135–144, 2002.

[35] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pages 27–40, New York, NY, USA, 2004.

[36] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv., 39(1):3, 2007.

[37] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. SIGOPS Oper. Syst. Rev., 36(SI):315–327, 2002.

[38] Snort Team. Website, 2007. Available from http://www.snort.org/, accessed on 1/12/2007.

[39] W. Richard Stevens. The Protocols (TCP/IP Illustrated, Volume 1). Addison-Wesley Professional, 1993.

[40] R. Stone. Centertrack: an IP overlay network for tracking dos floods. In Proc of the 9th conf. on USENIX Security Symposium - Volume 9, pages 15–15, Berkeley, CA, USA, 2000.

[41] Ao-Jan Su, David R. Choffnes, Aleksandar Kuzmanovic, and Fabián E. Bustamante. Drafting behind Akamai (travelocity-based detouring). In SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pages 435–446, New York, NY, USA, 2006.

[42] Chitra Venkatramani, Olivier Verscheure, Pascal Frossard, and Kang-Won Lee. Optimal proxy management for multimedia streaming in content distribution networks. In NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, pages 147–154, New York, NY, USA, 2002.

[43] Limin Wang, Vivek Pai, and Larry Peterson. The effectiveness of request redirection on CDN robustness. SIGOPS Oper. Syst. Rev., 36(SI):345–360, 2002.

[44] Cynthia Wong, Stan Bielski, Jonathan M. McCune, and Chenxi Wang. A study of mass-mailing worms. In WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode, pages 1–10, New York, NY, USA, 2004.

[45] Y. Xiang and W. Zhou. An Active Distributed Defense System to Protect Web Applications from DDoS Attacks. In The Sixth International Conference on Information Integration and Web Based Application & Services, 2004.

[46] Mengjun Xie, Heng Yin, and Haining Wang. An effective defense against email spam laundering. In CCS '06: Proceedings of the 13th ACM conference on Computer and communications security, pages 179–190, New York, NY, USA, 2006.

[47] Ying Xu and Roch GuÃl'rin. On the robustness of router-based denial-of-service (DoS) defense systems. SIGCOMM Comput. Commun. Rev., 35(3):47–60, 2005.