Tracing the Origins of Distributed Denial of Service Attacks

A.Peart
Senior Lecturer
amanda.peart@port.ac.uk
University of Portsmouth, UK

R.Raynsford.
Student
robert.raynsford@myport.ac.uk
University of Portsmouth, UK

P. Ross
Senior Lecturer
penny.ross@port.ac.uk
University of Portsmouth, UK

Keywords: Distributed Denial of Service, Security, IP spoofing, Denial of Service

Abstract

Distributed Denial of Service (DDoS) attacks, the cousin of Denial of Service (DoS), paralyse their target resource and on occasion inflict permanent damage, preventing it from serving its legitimate users. DoS (Denial of Service) has long been a method of cyber attack to render a host unavailable to its users through the use of various methods that either consume the victims resources or force it into a reset. Either way the target host is unable to serve it's legitimate users. More recently DDoS attacks have become popular, commonly in the form of SYN flooding and exploitation of the HTTP GET method. The majority of DDoS attacks make use of a bot-net, using a large group of unwillingly infected computers that can be unknowingly commanded to carry out a DoS attack on a specific target. IP spoofing commonly used in such DDoS attacks makes it difficult for attacks to be traced, this paper will look at the problems faced by victims of DDoS and proposes a new method of finding the origin of attack when the IP has been spoofed. The proposed method builds upon current techniques of tracing the attack back and uncovering the perpetrator's IP by reconstructing attacks paths and computationally comparing them to identify false positives in the trace. This in turn will provide a more accurate trace back path to the perpetrator with the aim to eliminate the DDoS promptly.

Introduction

Distributed Denial of Service (DDoS) attacks, the cousin of Denial of Service (DoS), paralyse their target resource and on occasion inflict permanent damage, preventing it from serving its legitimate users, (Landesman, 2010). One popular type of DDoS attack is carried out by utilising the TCP handshake to packet flood the host, sending constant malformed SYN(synchronise) requests, a simple but effective technique. Normally a client would send one SYN request to the host who would then return a SYN ACK with the client finally replying with an ACK and thus a connection is establish. This can be abused by sending a multitude of SYN requests to a host in which the malicious client can skip the final sending of ACK or spoof its IP to which the server replies using an incorrect address. The targeted host will wait a degree of time for the ACK to arrive, potentially using up resource's on the server which if enough malformed requests are made can render it unable to serve. This type of attack is thus called a 'resource depletion attack'.

Often used in DDoS is the Botnet (although not exclusively tied to DDoS attacks) this is a large number of computers all connected to the internet and in some way infected with malware. A single individual or master can take control of a proportion of these 'slave' system's resources without the owner's knowledge. Botnet's can potentially become huge distributed systems, one recent example is that of the 'Bredolab' botnet consisting of as many as 29 million infected computers (Constantin, 2010), botnets of this size or smaller provided a large amount of power in the hands of an individual. It should be noted however that a DDoS can still be performed without a botnet through a group willing to co-operate with one another to DoS a host. Such applications of DDoS attacks are; cyberwarfare, internet protests, attacking competing businesses, political gain, and personal vendettas to name but a few. To express the threat of DDoS some examples include the recent attack on Burma's connection to the wider web, ("Burma hit by", 2010) an attack on various Estonian websites including the Ministry of Finance in 2007 (" Bots Hammer", 2007) and the historical attack on Yahoo disabling the service for 3 hours in 2000 ("Yahoo attack", 2000).

DDoS Attacks

Such attacks can have severe consequences for the victim and although routes are available to avoid and withstand them, the ultimate deterrent would be to quell the source of the attack. The problem in finding the culprit of a DDoS attack is that it is distributed among many sources with each packet routed independently, the origin IP can be easily spoofed and no details on the path traversed by a packet are recorded, anyone behind a botnet would thus be hard to trace back too. Current Traceback methods to find the true sources of an attack when an IP is spoofed are reviewed. Such methods are 'Link Testing' (hop-by-hop): tests network links that carried the attack packets starting with the router closest to the origin, Logging: packets at key routers are logged then using data mining techniques information is extracted, ICMP trace-back: in addition to regular traffic information, details on a packets source, sender and authentication is received for 1 in every X number of packets, Packet marking: uses a packets IP header to place trace-back details of the routers it has passed through (Aljifri, 2003). The focus of this paper is the use of packet marking and the problems this entails, with a look at current solutions to the packet marking problems and solution.

Figure 1. *(Aljifri, 2003, p.28) "Packet marking. The router probabilistically marks packets as they travel through it (by inserting an indication of the router IP address). The marking process depends on the method adapted. "*

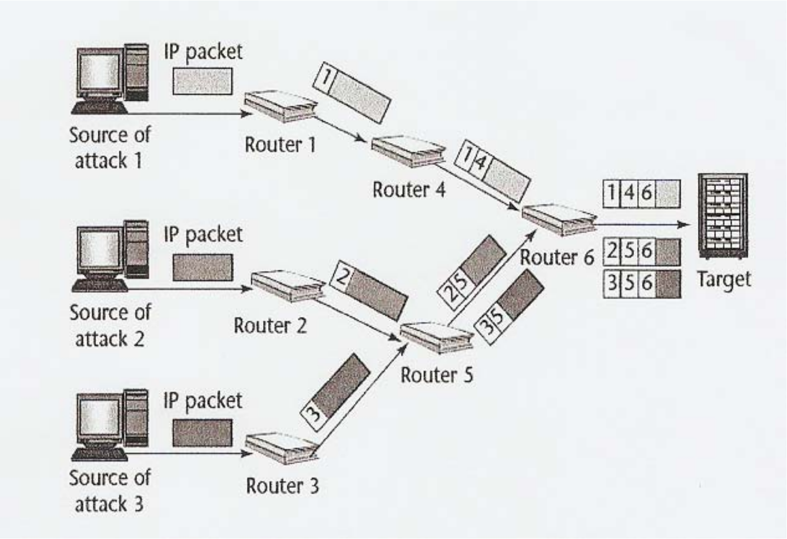

Figure 1 illustrates the basic concept of packet marking. As each packet passes through a router, the router marks the IP packet with trace-back data, upon reaching the victim of the attack the host will have enough information to re-construct an 'attack path' revealing the source of attack (Aljifri, 2003). Figure 1 demonstrates how a packet can be traced. The attack from source 1 passes through Router 1 (RI), R4 and R6 and at each stage of its journey it is being marked by these routers. Savage et al (2001) states that all marking algorithms have two components, marking procedure and path reconstruction with the former being performed by routers marking packets and the latter by the victim using the marked packets to attempt a reconstruction of the attack path. Savage et al (2001) also provides an example algorithm of the simplest form of packet marking in figure 2:

Figure 2. *(Savage et al, 2001, p.230) "Node append algorithm"*

```
Marking procedure at router R:
        for each packet w, append R to w

Path reconstruction procedure at victim v:
        for any packet w from attacker
        extract path (Ri .. Rj) from the suffix of w
```

The inherit problem of such an algorithm and one of the original problems of packet marking, occurs when the router addresses are stored in the 'options' field of the IP header. Due to the inability to predict the length of route taken by the packets you cannot make sure sufficient space will be available, as each hop increases the packet size and the chance of fragmentation occurring if the packet size breaches the MTU (Maximum Transmission Unit) size set out. Regardless if you were able to predict the number of hops on a route, attackers could still fill the header with fake data (Savage et al, 2001). The problem comes when trying to find a new approach to packet marking that maintains the accuracy of packet marking but without the huge amounts of fragmentation or large overheads.

Known Solutions

Savage et al, (2001), presents a new type of packet marking, **probabilistic packet marking** (PPM), which has since gained widespread attention due to its potential low cost (AIjifri, 2003). In brief the basic principle of PMM constitutes marking each packet with the probability of 1/25 significantly driving down overhead. In greater detail Savage et al (2001), proposed that routers mark the packets with their IP address (node sampling) or the edges of the attack path (edge sampling), using PPM. Node sampling gives each router a probability to mark packets with their IP, the aim is for the victim to receive enough packets marked by each router to reconstruct the attack path. An attack path is reconstructed by ordering the routers by the number of packets that they have marked, with the theory being the furthermost router has little probability of one of its marked packets making it untouched to the victim, thus the fewer marking received from a router the further down an attack path it must be. Figure 3 shows the node sampling algorithm:

Edge sampling still using a probabilistic approach records the start and end of a path (an edge) method. Each packet contains a field for the start and end addresses as well as a counter for the distance from start to end. The first router in a path that decides to mark the packet's start address with its own increments the distance to 0, the next router checks if the distance is >=0 it then sets the end address to its own and increments the distance by one. The victim will end up with the start and end addresses in the attack path along with the distance between them. Edge sampling improved upon node sampling in regards of lowering the router overhead, which compression techniques.

Figure 3. *(Savage et aI, 2001, p.230) "Node sampling can be improved upon further with algorithm"*

```
Marking procedure at router R:
    for each packet w
        let x be a random number from [0 .. 1)
        if x < p then,
            write R into w.node

Path reconstruction procedure at victim v:
    let NodeTbl be a table of tuples (node,count)
    for each packet w from attacker
        z := lookup w.node in NodeTbl
        if z != NIL then
            increment z.counl
        else
            insert tuple (w.node,l) in NodeTbl
        sort N odeTbl by count
        extract path (Ri .. Rj) from ordered node fields in N odeTbl
```

The drawbacks and problems facing PPM are those such as; false positives (when a router appears in the reconstructed attack path but not the real attack path), packets being marked that are not part of the attack or marking occurring without an attack present, large reconstruction time of the attack path and the large amount of attack packets needed to make an accurate path (Tupakula et al, 2006). One large problem relating to the context of DDoS, is when PPM is used against DDoS attacks due the difficulty in trying to group together multiple attack paths and the large amounts of fragments to sort through to structure an attack path, with false positives only adding to this (Savage et aI, 2001). Research by Song et al (2001) finds that while using a compression technique formulated by Savage et al (2001) for PPM, DDoS attacks from as little as 25 sources, can incur days of reconstruction to produce an attack graph.

An early solution to PPM, concerned with solving the high computation overhead in attack path reconstruction and the large number of false positives experienced when under attack by DDoS, is presented by Song et al (2001). The technique that Song et al (2001) purposes named 'Advanced Marking Scheme' (AMS) uses hash values to encode router IP's into packets, adapting on the previous compression methods for edge sampling designed by Savage et al (2001) which encoded router IP address into eight fragments. The first scheme advocated by Song et al uses the 16 bit ID field of a packet by using 5 bits to represent the distance and 11 bits for the paths 'edge', this still produces a significant amount of false positives when the number of attacker's rise above 60.

To improve upon the first scheme a second is devised in which two sets of independent hash functions are used, when tested in a simulation this method produced only 20 false positives when under attack by 2000 separate attackers and reproduced an attack path in only 100 second a vast improvement upon Savage's design. This method however has not stood the test of time, with the increase in the size of DDoS attacks like the earlier reported 29 million strong botnet, an attack of hundreds of thousands would be enough to create a significant number of false positives under the advanced marking scheme to threaten the legitimacy of an attack graph.

Yaar A, Perrig A and Song D (2005) worked on a new theory (FIT: Fast Internet Traceback) in an attempt to further decrease false positives, computational overhead, packets required for path construction and various other issues in AMS. Contrasting with AMS, FIT makes use of node sampling instead of edge sampling, allowing each attack path router to be recorded rather than just the edge's, the advantage being a reduction in false positives and less packets required for path reconstruction. The IP ID field is used again to mark packets, this time split into a 1 bit distance field, 2 bit fragment field and a 13 bit hash field, the 1 bit used for distance is unlike most PPM based schemes which commonly use 5 bits, this essentially allows 13 bits for the hash field thus reducing again the number of false positives. FIT also provides a tool for the creation of the up-stream router map, improving upon the accuracy of the commonly used Traceroute (known as tracert in Microsoft WIndows) tool which proves "inaccurate in the presence of asymmetric paths" (Yaar et al, 2005, p.1398) as the victim to potential attacker path traced by Traceroute can differ from the potential attacker to victim path. Previous instalments of PPM have faltered when confronted with legacy routers that do not increment the hop count resulting in numerous false positives, Yaar et al (2005) have provided a solution by calculating the distance from when the packet was last marked, setting the 5 least significant bits of a packets time to live (TTL) to a global constant and storing the 6th bit of the TTL in the distance field, the next FIT enabled router can then calculate the distance the packet has travelled.

In an attempt to improve upon the above scheme Akyuz et al (2009) puts forward the idea of marking packets with dynamic probabilities based on their distance from the source. Using TTL again to determine distance, packets far from the source will have a lower probability of being marked, opposing those closer will have a higher probability, this results in a fairer marking scheme evenly distributing the amount of packets marked per router. When tested in a simulation Akyuz et al (2009) found the number of packets required to reconstruct an attack path of 25 hops was 498, marginally less than the 565 required in a FIT implementation. However this scheme did not improve on the number of false positives generated, as in a simulation of 5000 attacker's this scheme had 11 false positives 4 more than when using FIT. Akyuz et al attributed this down to the fact that 5 bits are used, keep a counter of the distance, leaving less for the hash than the l3bits used by FIT.

One key issue that these solutions keep on suffering from is that of false positives, not only a problem when trying to reconstruct an attack path but also when carrying out legal action against attacker's as Tupakula et al (2006 p.ll8) claims "If the victim initiates legal proceedings with the evidence captured from traceback techniques, the attacker can argue that it can be a false positive."

Quality Assurance Check for the Attack Path

Missing from the current solutions is a way to validate the resulting attack path. This paper proposes a quality assurance check for the attack path, calculated using the FIT method in an attempt to find and reduce the few false positives it produces. The checking method would need to be a lightweight implementation such that it does not require large changes to be implemented and provides a simple method of attack path reconstruction.

A method which best fills this criteria is one by Thing et al (2007). The aim of the method titled 'Non-Intrusive IP Traceback for DDoS Attacks' is to provide a simple and efficient means to attack path reconstruction. Supported by evidence that end to end routes remain relatively stable this method conducts what is referred to as the 'learning process', while in this stage samples of the common flow of traffic between source and destination are cached in a white list 'caching device'. In the first onset of an attack the white list stops updating the cache and removes a set amount of previous entries to make up for the time it took to detect the attack (attack data would corrupt the white list). As explained by Thing et al (2007, p.371) "When an attacker spoofs a legitimate source address, the packet may pass through routers which are not on the normal source-destination routing path" this occurrence is used to reconstructed the attack path. To reconstruct the attack path cached data from the white list is compared to the collected attack data, the comparison reveals instances of when packets have diverged from their usual path onto paths not commonly used to carry the related source-destination IP addresses. From this we can devise the true source of spoofed IP packets.

Such a method as the one above meets the 'lightweight' criteria as it does not require routers to be changed or modified, instead as an alternative to installing tools on routers, monitoring devices can be placed along network paths, the white list caching devices can also provided relief for the victim from logging and computation tasks but most importantly changes do not need to be made to packets travelling through the network as they do not need to be marked. One concern could be that the marking of packets performed by the FIT method would conflict with the 'lightweight' method, however the markings are made in the identification field and the source/destination IP fields used in the 'lightweight' method are left unaltered.

Due to this simple nature it can be implemented alongside a complex form of trace-back such as FIT. At the end of trace-back both FIT and the 'lightweight' method produce an attack graph, showing the routers conversed by the attack packets, it is these two attack graphs that is to be investigated. By cross referencing the two attack graphs it would be possible to highlight the false positives encountered in both, these false positives are routers appearing in the attack path when in reality they do not belong there. False positives are produced differently in each method, in FIT they are a result of the hash based recording of router addresses being mismatch in the reconstruction process, where as the false positives in the 'lightweight' approach are usually caused by new legitimate traffic that has not yet been added to the white list, causing routers to flag-up that they are carrying unfamiliar packets.

The advantage of the differences in false positive creation for both of these methods is that the same false positives should not be generated both trace-back techniques, allowing for the victim to essentially 'weed' out the incorrect attack paths by picking up on discrepancies between the two reconstructed attack paths. When the falsified data is removed the victim should be left with the combination of the two attack paths and the correct sequence of routers traversed by the attacking packets.

Figure 4, *FIT generated attack path, A * are potential attackers, R * are routers, red meaning false positive; green is the correct attack path.*

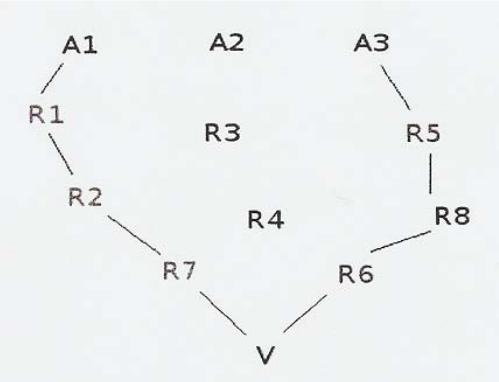

Figure 5, *attack path generated by the lightweight method, A * are potential attackers, R * are routers, red meaning false positive; green is the correct attack path.*

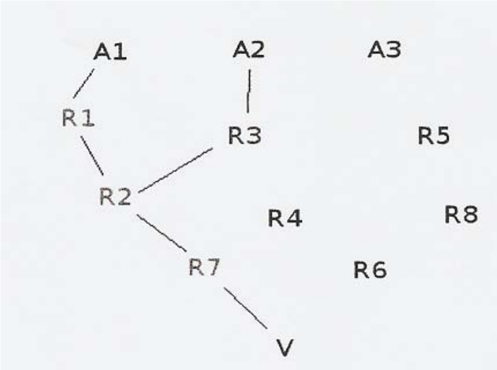

Figure 4 and 5 show how the false positives will stand out when a comparison is made. It is clear that the correct attack path is Rl>R2>R7 as it appears in both reconstructions thus R3, R6 and R5 are false positives.

Figure 6, *Pseudo code for comparing the methods to detect false positives*

```
dataCollection()
while (atkData is undetected)
    whitelistState="updating cache"    **No attack, whitelist continues to update cache
    If (atkData is dectected) then
        whitelistState="paused updating"
        whitelist.rollback
        while (atkData is dectected)
            blacklistState="collecting attack data"
            markedPackets = "recording packets marked by FIT"}
        pathReconstruction(blackList, markedPackets)

pathReconstruction(blackList, markedPackets)
    atkGraph1 = FIT's reconstructed attack path
    atkGraph2 = whiteList and blackList comparison of data
    comparison (atkGraph1, atkGraph2)

comparison(atkGraph1, atkGraph2)
    for (number of routers)
        If (router*.flag in atkGraph1 != router*.flag in atkGraph2) then
            router*="false positive"
            **Router is a false positive as it does not appear in both attack graphs
        else
            add router* to finalGraph
```

Figure 6, illustrates the algorithm to compare the attack path paths of both FIT and the lightweight method to detect false positive which will highlight the origins of the attack. False negatives, if one were to occur, meaning either method failed to pick up on a router carrying attack data. It would not make it through the comparison stage as it would be seen as a false positive and deleted from the attack graph, a potential drawback to this solution.

Conclusion

Looking at the potential risks DDoS attacks present to various parties, from governments to businesses, we can recognise the need for a viable method to trace DDoS attackers, but as described the problems of tracing back such attacks can be difficult due to IP spoofing. One popular attempt at DDoS trace-back called packet marking, in which packets moving from attacker to victim are marked with path information. The problem which such a method was it has a large computational overhead, and when reducing this it becomes difficult to maintain the accuracy in the tracing of attack packets to source. We have looked at the three main solutions to such a problem each building upon the previous method in both accuracy (fewer false positives) and overhead.

In this paper a unique solution to trace-back was presented, the basis of which was to take the Fast Internet Trace-back method in its entirety and combine it with a second method of trace-back, one which was simplistic and did not rely on packet marking. Such a simplistic and efficient method was Thing et al’s 'Non-Intrusive IP Traceback'. The two methods of trace-back can be allowed to run their course, with their resulting reconstructed attacks paths being compared against one another. When comparing attack paths any false positives should be easily spotted as they likely hood that both methods would produce the same one's is significantly reduced.

This combined approach to trace-back allows victims to reduce the number of false positives more so than when using the standard single trace-back approach. The 'lightweight' method is flexible enough that it can be combined with almost any other method, FIT was chosen as an example in this paper as it produced the least false positives of any other packet marking approaches. False negatives, when routers on the correct attack path are not detected, could potentially have an adverse effect on the purposed method as a router could appear in one of the attack paths to be compared but not on the other due to a false negative, resulting in the false negative being marked as a false positive thus being removed. Secondly although unlikely it is still possible for both attack graphs to produce the same false positive, resulting in this false positive making it into the final reconstructed attack path. In conclusion this method of trace-back has gone some way in solving the problem of false positives but still remains vulnerable to false negatives, work would need to be performed on this solution to allow for some kind of false negative detection.

References

AKyuz, T. Sogukpinar, L. (2009). Packet Marking with Distance Based Probabilities for IP Traceback [Electronic version]. NETCOM 2009, 433-438.
Aljifri, H. (2003). IP Traceback: A New Denial-of-Service Deterrent? [Online version]. IEEE / Security & Privacy, 8
Burma hit by massive net attack ahead of election. (2010, November 4th). Retrieved November 5th, 2010, from the BBC news website: http://www.bbc.co.uk/news/technology-11693214 .

Constantin, L. (2010, October 2nd). Suspect Bredolab Botnet Runner Arrested in Armenia.
Retrieved November 16th, 2010, from Softpedia:
http://news.softpedia.comlnews/Suspected-Bredolab- Runner-Arrested-in-Armenia-163068.shtml

Greenemeier, L. (2007, May 18th). Bots Hammer Estonia In Cyber Vendetta.
Retrieved November 5th, 2010, from InformationWeek:
http://www.informationweek.comlnews/internetl show Article. ihtml ?articleID= 199602023

Landesman, M (2005). What is a DDoS attack?, Retrieved November 5th, 2010, from
http://antivirus.about.com/od/whatisavirus/a/ddosattacks.htm

Savage, S. Wetherall, D. Karlin, A. Anderson, T. (2001). Network Support for IP Traceback [Electronic version]. IEE/ACM Transactions on Networking, 9(3), 226-237

Song, D. Perrig, A. (2001). Advanced and Authenticated Marking Schemes for IP Traceback [Electronic version]. INFOCOM 2001, 2, 878-886.

Thing, V. Sloman, M. Dulay, N. (2007). Non-Intrusive IP Traceback for DDoS Attacks [Electronic version]. ASIACSS '07,371-373.

Tupakula, U. Varadharajan, V. (2006). Analysis of trace back techniques [Electronic version]. ACSW Frontiers, 54, 115-124.

Yaar, A. Perrig, A. Song, D. (2005). FIT: fast Internet traceback [Electronic version]. INFOCOM 2005, 2, 1395-1406.

Yahoo attack exposes web weakness. (2000, February 9th). Retrieved November 5th, 2010, from the BBC news website:
http://news.bbc.co.uk/l/hi/sci/tech/635444.stm