Amplification and DRDoS Attack Defense – A Survey and New Perspectives

Fabrice J. Ryba*, Matthew Orlinski*, Matthias Wählisch*, Christian Rossow†, Thomas C. Schmidt‡
* Freie Universität Berlin, Berlin, Germany,
Email: {fabrice.ryba, m.orlinski, m.waehlisch}@fu-berlin.de
† Saarland University, Saarbruecken, Germany,
Email: crossow@mmci.uni-saarland.de
‡ HAW Hamburg, Hamburg, Germany,
Email: t.schmidt@haw-hamburg.de

***Abstract*—The severity of amplification attacks has grown in recent years with recent attacks involving several hundreds of Gbps attack volume. In this paper, we survey the threat of amplification attacks and compare a wide selection of different proposals for preventing, detecting, and filtering amplification attack traffic. Since source IP spoofing plays an important part in almost all of the amplification attacks surveyed, we also survey state-of-the-art of spoofing defenses as well as approaches to trace spoofing sources. This work acts as an introduction into many different types of amplification attacks and source IP address spoofing. By combining previous works into a single comprehensive discussion we hope to prevent redundant work and encourage others to find practical solutions for defending against future amplification attacks.**

I. Introduction

Denial of Service (DoS) attacks against networks attempt to make a system or an entire network unavailable for its intended purpose [1]. In some cases DoS attacks cause a complete loss of Internet connectivity for the victim [2]. The motivation for DoS attacks can be financial [3], [4], political [2], ideological [5], reputation [6], [7], to test new techniques and prepare for larger attacks, or purely for disruptive purposes [8].

A general overview of DoS attacks and defense strategies can already be found in the surveys by Mirkovic and Reiher [9], Douligeris and Mitrokotsa [10], and more recently Zargar et al. [11]. Instead, this paper focuses on a particular type of DoS attack called an *amplification attack* where the attacker seeks to maximise the volume of attack traffic sent to the victim whilst minimising the volume of traffic they send to trigger the attack [12]–[18]. We will not discuss other low traffic volume and high impact DoS attacks, e.g., Slowloris [19] or TCP SYN floods [20], unless their defenses are also applicable to amplification attacks.

Generally, the adversary in amplification attacks targets vulnerabilities in Internet protocols and services to amplify the amount of attack traffic. The ease of performing amplification attacks and greater understanding of their effect has led to an increase of attacks in recent years. This is supported, e.g., by the OpenDNS Security Lab that saw over 5,000 different amplification attacks every hour in 2014 [21], and by Rossow's survey of amplification vulnerabilities [22].

The most prominent form of amplification attack seen in recent years abuses the lack of endpoint verification in the Internet Protocol (IP) in order to trick third-party servers into sending large amounts of data to victims. That is, attackers use source address IP spoofing [23] to hide their identity and cause third-parties to send data to the victim as identified by the source address field of the IP packet. This is also sometimes called *reflection* because attackers "reflect" attack traffic off of benign services.

As well as reflection, attackers sometimes strive to maximize the attack bandwidth sent to the victim by using *amplification*. For example, many UDP-based protocols (such as DNS or NTP) that have a higher response payload size than the requests can be abused to amplify the reflected attack traffic. Consider DNS, while DNS lookup requests are typically rather small, the responses may be significantly larger, e.g., due to verbose information such as DNSSEC records.

By combining reflection and amplification attackers can generate attack traffic volumes which are significantly higher than their uplink bandwidth [2], [4], [8], [24]. Furthermore, not only single hosts may be affected by such attacks. Entire networks may struggle to cope with the extra bandwidth and processing demands [24].

Defending against this form of amplification attack is challenging for network operators for at least two reasons. Firstly, the attacker sends only a small amount of (triggering) packets which may be hard to distinguish from legitimate traffic in high volume uplinks. Secondly, the attack packets received by victims are reflected by benign third-parties such as open DNS resolvers thus hiding the attackers identity [25].

It is also common to see DoS attacks which utalise reflection and amplification referred to as Distributed Reflective Denial of Service (DRDoS) attacks [22], [26], DNS reply flooding [27], or simply as reflection attacks [28]. Section II will bring these works together by offering a more precise distinction between reflection and amplification, and discuss other application layers protocols than DNS which have been used in amplification attacks.

So far in this introduction we have only described one type of amplification attack. There is some overloading of the term "amplification" in the Internet security literature which is important to disambiguate. Therefore, in Section II we also briefly describe some other types of amplification attacks, such as the "Fork Loop" attack against VoIP networks [29], and older Smurf and Fraggle attacks [23].

The rest of this paper is organised as shown in Figure 1. In Section II, we present prominent incidents from the past and discuss amplification attacks in great detail. We then describe methods to prevent amplification attacks in Section III. Then special attention is paid to preventing reflection and IP spoofing in Section IV. We finish the discussion about the solution space in Section V by presenting approaches to detect and filter amplification attacks. Finally, we conclude in Section VI.

II. Amplification Attacks

This section will provide a detailed introduction into the different techniques for amplifying attack traffic. We start by giving a brief taxonomy and timeline of different types of amplificaiton attacks, and then we end the section by describing the techniques used in more detail.

A. Taxonomy and Historical Background

There are many ways of amplifying the volume of attack traffic in DoS attacks. In this paper we give special focus to amplification attacks which use UDP based amplifying protocols and reflection. However, it is important to be able to distinguist between the different types of amplificaiton attacks.

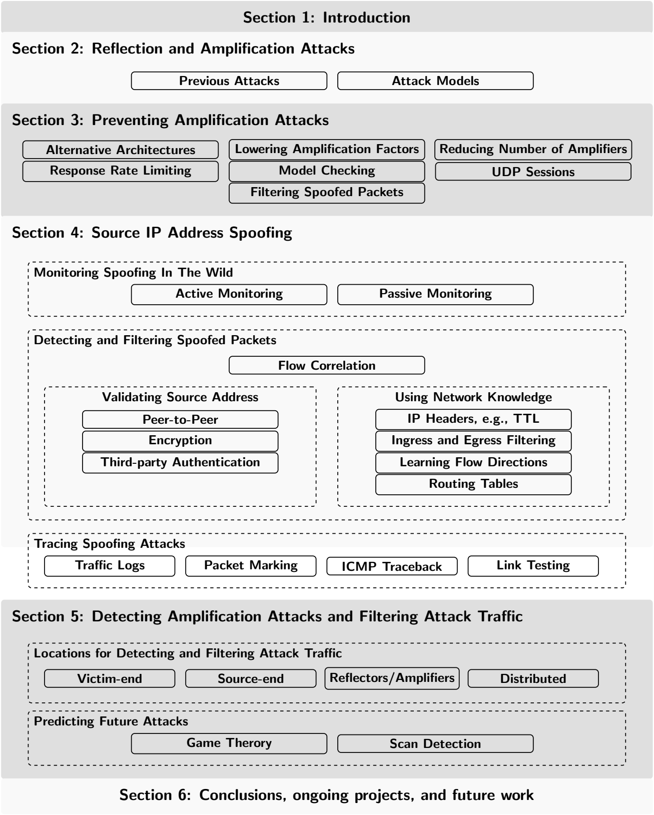
Section 1: Introduction
Section 2: Reflection and Amplification Attacks
Previous Attacks
Attack Models
Section 3: Preventing Amplification Attacks
Alternative Architectures
Lowering Amplification Factors
Reducing Number of Amplifiers
Response Rate Limiting
Model Checking
UDP Sessions
Filtering Spoofed Packets
Section 4: Source IP Address Spoofing
Monitoring Spoofing In The Wild
Active Monitoring
Passive Monitoring
Detecting and Filtering Spoofed Packets
Flow Correlation
Validating Source Address
Peer-to-Peer
Encryption
Third-party Authentication
Using Network Knowledge
IP Headers, e.g., TTL
Ingress and Egress Filtering
Learning Flow Directions
Routing Tables
Tracing Spoofing Attacks
Traffic Logs
Packet Marking
ICMP Traceback
Link Testing
Section 5: Detecting Amplification Attacks and Filtering Attack Traffic
Locations for Detecting and Filtering Attack Traffic
Victim-end
Source-end
Reflectors/Amplifiers
Distributed
Predicting Future Attacks
Game Theory
Scan Detection
Section 6: Conclusions, ongoing projects, and future work

Fig. 1. How we have categorised the literature.

1) DNS Servers as Amplifiers: In 1999, AusCERT [30] warned about amplification attacks where attackers trick DNS servers into sending large amounts of data to spoofed IP addresses [23]. In 2000, the CIAC warned of Distributed Denial of Service (DDoS) attacks where attackers can send spoofed IP packets from many different locations [31].

Today, DNS is widely used in amplification attacks because a 60 byte request from an attacker can easily generate a 512 byte response [2], [8], [12]. Furthermore, DNS servers which support Extension mechanisms for DNS (EDNS) [32] can generate responses which are nearly 100 times larger than requests [22]. Some attackers even ensure large responses by creating domains specifically to use in amplification attacks [21].

Any public DNS server which is configured to respond to requests for hosts outside of their domain can be used in amplification attacks. DNS servers that are configured to respond to all such requests are known as open DNS resolvers. According to the Open DNS Resolver Project approximately 25 million open DNS resolvers currently "pose a significant threat" to the Internet [25]. It is also not difficult to find enough open DNS resolvers to use

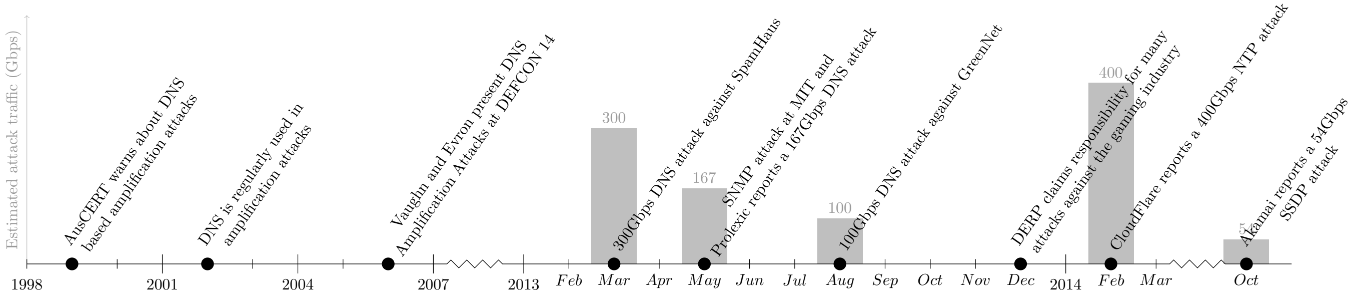

Fig. 2. Timeline of amplification attacks and related events.

in amplification attacks. Rossow showed that it currently only takes 90 seconds to acquire a list of 100,000 open DNS resolvers [22].

2) Other UDP Based Services: The DNS based amplification attacks just described use source IP address spoofing to trick third-parties into attacking a victim [23].

Source IP address spoofing also makes it very difficult for the victim to track attacker because all attack traffic they receive will come from third-parties. UDP is also used by attackers as the transport layer protocol because it is stateless and has no built in mechanism to verifying the source IP address of packets.

Recently, another UDP based protocol (SSDP) has been used with source IP address spoofing in amplification attacks. The United States Computer Emergency Readiness Team (US-CERT) first issued a warning about SSDP in January 2014 [33], and in October 2014 it was used to generate 54Gbps of traffic in a single attack [34].

Other well documented DoS attacks that used UDP and source IP address spoofing are the MIT SNMP based attack [35], and the CloudFlare attack which reached a peak of nearly 400Gbps using 4,529 NTP servers [24]. The huge volume of traffic generated in the CloudFlare attack was caused by a command in NTP called "monlist", which returns a list of the previous 600 IP addresses to access the server. Monlist does not affect the core functionality of NTP and should be disabled by upgrading to version 4.2.7 and above [36].

3) Smurf and Fraggle Attacks: As well as misusing UDP based services, there are many other ways of amplifying attack traffic volume. For example, prior to 1998 attackers were sending ICMP requests to the broadcast address of networks whilst spoofing the source IP address of victims. The routers receiving the ICMP requests broadcast the requests to all of the hosts on their network. These attacks are known as Smurf attacks [38], [39], and in Smurf attacks the routers which broadcast the requests act as amplifiers because they increase the volume of attack traffic.

End hosts which respond to requests after they are broadcast may be amplifiers as well. For example, in Fraggle attacks an attacker may send CharGen requests to the broadcast address of many networks at the same time [40]. End hosts which receive CharGen requests may respond with much more data than they receive, thus the end hosts in Fraggle attacks are amplifiers as well as the routers which broadcast the request.

4) Fork Loops: There is also another category of amplification attacks which we have not yet covered. In 2009 Shankesi et al. [29] described an amplification attack as being where ”the number of messages on the network can amplify to essentially an arbitrary large number” [14]. It is important to note however that this definition doesn't take into account the size difference between request and response packets, which is an important factor in the UDP based amplification attacks we have already mentioned.

The amplification attacks described by Shankesi et al. are caused by "fork loops" which are used to attack VoIP networks running SIP [29]. Fork loops are situations where requests are sent between SIP proxies indefinitely and at least one extra request is generated every iteration [14]. Using this type of amplification attack it is possible to cause 2 SIP proxies to exchange up to 2^{70} duplicate requests consuming their resources and preventing them from responding to legitimate requests [29].

B. Responding to a Growing Threat

Today, Smurf, Fraggle, and Fork Loop attacks have largely been countered by better network configuration and patches to SIP respectively. However, attacks involving reflection and UDP based amplification are growing in number.

Reference	Year	Amplification Protocol	Peak Traffic	Attacker	Description
MIT [35]	2013	SNMP	Unknown	Unknown	Spoofed requests were directed to SNMP enabled devices on the MIT network which attacked devices outside the network.
GreenNet [2]	2013	DNS	100 Gbps	Unknown	Attackers brought down GreenNet, a hosting company who's customers including Privacy International and Zimbabwe Human Rights Forum.
Prolexic [4]	2013	DNS	167Gbps	Unknown	The attack targeted a real-time financial exchange platform.
Spamhaus [8]	2013	DNS	300Gbps	Cyberbunk	Over 30,000 DNS servers were involved in the amplification attack. Attackers also used SYN floods and prefix hijacking to disrupt the Internet connectivity of Spamhaus and Project Honeypot.
PhantomL0rd [7]	2013	NTP	100Gbps	A hacker group named DERP claimed responsibility on Twitter.	Attackers targeted game servers and streaming sites. Attacks against the games industry are becoming increasingly popular [37].
CloudFlare [24]	2014	NTP	400Gbps	Unknown	Latency increased all over Europe following attack using 4,529 NTP servers. French hosting firm OVH also claimed to have seen an attack over 350Gbps at the same time.
Akamai [34]	2014	SSDP	54Gbps	Unknown	Many of the devices used in this attack were home-based UPnP devices.

TABLE I
RECENT LARGE SCALE AMPLIFICATION ATTACKS AND THEIR PROPERTIES

Akamai detected 9 DoS attacks involving NTP, CharGen, and SSDP which peaked over 100 Gbps in the last 3 months of 2014. This is three times more than in the same period in 2013 [37].

Figure 2 and Table I illustrate that the bandwidth used in amplification attacks is also growing. In March 2013 an amplification attack was launched against Spamhaus [8]. The DNS based attack reached an estimated peak of about 300Gbps and was the biggest DoS attack ever recorded [41]. Nearly a year later and an even bigger attack reportedly reached a peak of 400Gbps by using NTP [24].

The attack traffic generated during the Spamhaus attack came from over 30,000 different DNS servers. In order to guard against this and future attacks, they deployed anycast routing and a distributed Content Delivery Network (CDN) [8]. Anycast involves BGP routers simultaneously advertising the same destination IP address so that attack traffic can be absorbed by many different data centers. However, relying on anycast and CDNs may be a short term solution, and it can only prevent amplification attacks when amplifiers and anycast routes are evenly distributed around the Internet.

The CloudFlare attack came 1 month after a "monlist" warning issued by US-CERT [36], and during a 13-week period in which Kührer et al. reported a 92% drop in the number of vulnerable NTP servers [17]. As a consequence we may need to reevaluate how we respond to threats as they are discovered. For example, alerting system administrators about protocol vulnerabilities may prompt attackers to abuse the identified amplification vectors.

C. Problem Description

In this section we will provide high level descriptions of the attacks and concepts seen so far. We will start by discussing the amplification factor of different attacks and go on to describe the building blocks of the most common amplification attacks.

1) Amplification Factor: The amplification factor of an attack ($X(P)$) for a protocol (P) is the ratio of the total number of bytes in the amplified traffic and the bytes sent by the attacker as shown in Equation 1.

$$X(P) = \frac{number\, of\, response\, bytes(P)}{number\, of\, request\, bytes(P)} \quad (1)$$

The wording used for $X(P)$ in Equation 1 is different from the one used in [22] because response packets for the same protocol can be different sizes. Nevertheless it is useful to look at the amplification factors provided by Rossow [22] because they offer us a convenient way of comparing protocols and can help with prioritization. For example, Rossow found that NTP had the highest amplification factor all of the protocols he tested. He found that if attacks carefully select the NTP servers to maximize their attack they can achieve an amplification factor of 4670, which when compared to 64 for open DNS resolvers shows us how urgent it is to update existing NTP infrastructure.

2) Reflection: Many of the amplification attacks in the past were possible because the attacker can spoof the address of the victim causing traffic to be "reflected" by third-party servers, e.g., by open DNS resolvers [42] (see Section II-A). Figure 3 shows a reflection attack where an attacker (M) changes the source address in the request it sends to the reflector (R). Not knowing that the source address was spoofed, the reflector sends its response to the final destination (D, i.e., the victim).

The key advantages of reflection attacks for the attacker are: (a) attackers hide their identity from D because all traffic received by D comes from third-parties; (b) attackers can trigger attacks coming from different geographic or topological regions of the Internet; and (c) attackers do not receive responses and therefore do not risk using up their available download bandwidth.

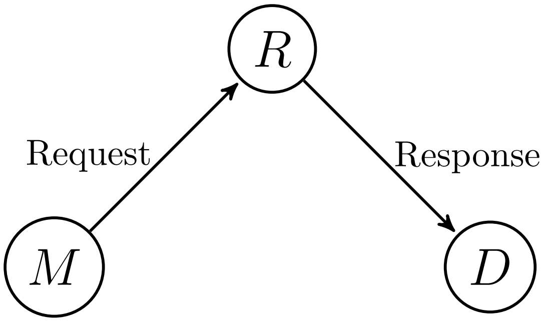

Fig. 3. Reflection attack. An attacker (M) sends a spoofed request to a reflector (R) which responds to the final destination (D).

One example of a DoS attack which uses reflection is a TCP SYN/ACK attack, where an attacker sends spoofed SYN requests to a reflector so that the reflector sends SYN/ACK packets to the victim [43].

As well as the final destination as indicated by the spoofed source IP address, it is also possible for reflectors and those that use them legitimately to be victims of reflection attacks. For example, in SYN flooding attacks [27], [44]–[48], the attacker floods a reflector with spoofed SYN requests and changes the source IP address of packets so they do not receive replies. The aim of SYN flooding attacks is not to attack the final destination indicated by the spoofed IP address, but to consume enough resources at the reflector to make it unresponsive to legitimate traffic. The adversary can also attack multi-homed networks using reflectors. For example, if the reflector is connected to the Internet by at least two uplinks, and the link between the reflector and the final destination offers significantly lower bandwidth compared to the link between the attacker and reflector. Then by carefully selecting a spoofed source address which is reachable via the low bandwidth connection, the attacker can implement a DoS attack on this link.

It is worth mentioning again, however, that not all of the amplification attacks described in this paper use reflection, e.g., fork loop based attacks do not [14].

3) Amplification: A key factor in amplification attacks is the ability of attackers to trigger responses which are significantly larger than the requests. Figure 4 describes a hypothetical scenario without reflection where an attacker sends a request to an amplifier (A) which responds with a higher volume of data than it receives. In this scenario the amplifier, and the network between the attacker and the amplifier, are the victim because they cannot cope with the amplified traffic. Figure 4 can be used to describe attacks against mobile devices which typically have lower upload than download bandwidth, or attacks where attackers can change their IP address often and do not receive the returning amplified traffic. The main advantage of such an attack is resource saving for the adversary.

4) Reflection and Amplification Together: All of the amplification attacks seen in Table I combine reflection and amplification, a high level overview of these attacks is given in Figure 5. In this model an attacker (M) sends requests using the amplification protocol (P) to the amplifiers (A) but also uses reflection so that the amplifiers respond to the victim (V).

The volume of attack traffic which can be sent to V depends on the accumulated available bandwidth between M and $A_1...A_n$, the accumulated available bandwidth

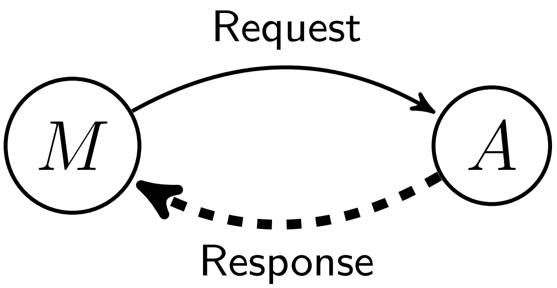

Fig. 4. Simple amplification attack. An attacker (M) sends a request to an amplifier (A) which responds with more data than it received. The response line is dashed in this instance to indicate that the attacker may not receive the response.

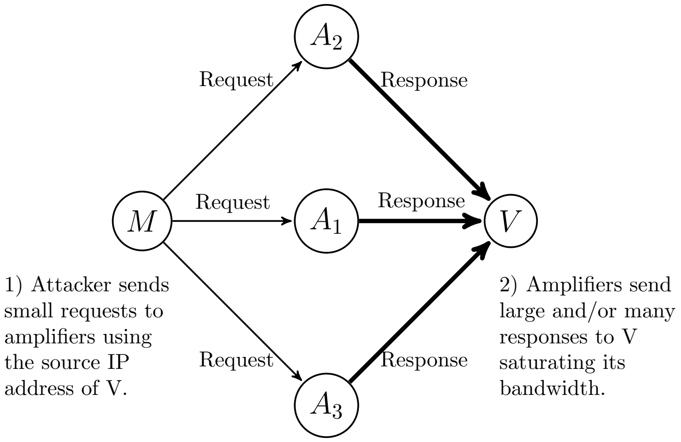
A_2
Request
Response
M
Request
A_1
Response
V
1) Attacker sends small requests to amplifiers using the source IP address of V.
Request
Response
2) Amplifiers send large and/or many responses to V saturating its bandwidth.
A_3

Fig. 5. An attacker (M) sends spoofed requests to many amplifiers (A) causing amplified responses to be sent to the victim (V).

between $A_1 ... A_n$ and V, and the amplification factor of P. Depending on the success of the attack, the volume of traffic sent to the victim can result in a complete loss of connectivity for the victim and their network.

5) Amplification Attacks Using a Botnet: A single attacker may not have sufficient uplink bandwidth to send requests to many amplifiers at the same time. Attackers can target more amplifiers, and further obscure the origin of attacks, by using a botnet [15], [49].

Botnets are compromised computers (bots) which can be remotely controlled by attackers. Different types of controllers have been identified, including IRC-based controllers such as Trinity v3 [50] and Knight [51]), and web-based controllers such as BlackEnergy [52]. Bots typically operate synchronized, meaning that a central entity (the *botmaster*) can inject commands to an entire botnet to initiate a large scale attack involving many bots.

The owners of bot-infected systems are unaware that their machines are being used in DDoS attacks. Alternatively, in crowd-sourced DDoS botnets, participants willingly install attacking software on their machines, e.g., the Low Orbit Ion Cannon (LOIC) software [53]. Such volunteer botnets were used in *Operation Payback* to attack global companies, such as MasterCard, Visa, and PayPal [5].

Figure 6 shows one model of an amplification attack using a botnet. In this example the attacker uses a controller (C) to send instructions to the botnet (B). The bots then send spoofed requests to amplifiers at similar times, such that the bandwidth of $B_1 ... B_n$ accumulates.

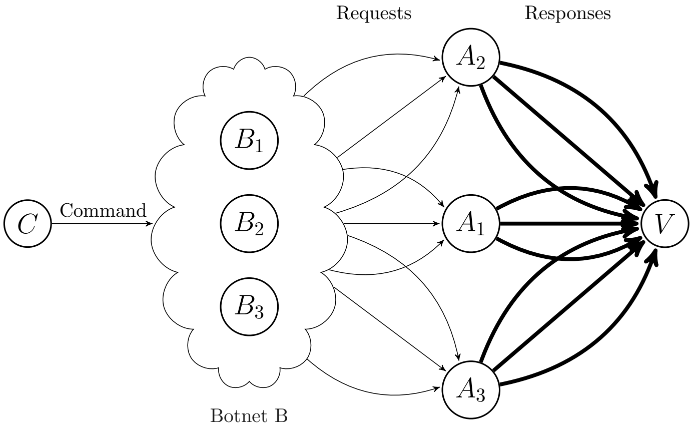

Fig. 6. An attacker uses a controller (C) to instruct a botnet (B) to simultaneously send spoofed requests to many amplifiers (A), causing amplified responses to be sent to the victim (V).

6) Fraggle and Smurf Attacks: Fraggle and Smurf also use reflection and amplification, but their attacks look slightly different from the model in Figures 5 and 6. Figure 7 shows a Smurf attack where M sends a request to the broadcast address of network (N), A acts as the amplifier by duplicating the request, and the reflectors (R) send responses to the victim V. Fraggle attacks are similar to Smurf attacks but the requests are amplified even further by the end hosts that receive them.

7) Fork Loops: Fork loops are different to the reflection and amplification attacks because they rely on messages recursively being sent between SIP proxies [14]. Yet fork loops have also been referred to as amplification attacks because they provide a way for an attacker to amplify the number of requests they send to SIP proxies.

Figure 8 shows one example of what can happen when two SIP proxies (L_1 and L_2) are not configured correctly and do not detect loops. It depicts the scenario described in [29] where a request for the user $a@L_1$ is sent to the proxy L_1. In this example, L_1 has two options for reaching $a@L_1$ which are both located at L_2. This causes L_1 to fork the request and send two copies to $a@L_2$ and $b@L_2$ which also both fork the request to $a@L_1$ and

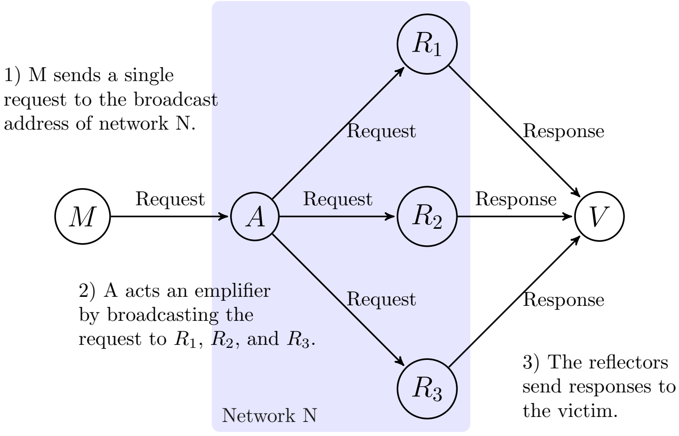
1) M sends a single request to the broadcast address of network N.
M
Request
A
Request
Request
Request
R_1
R_2
R_3
Response
Response
Response
V
2) A acts an emplifier by broadcasting the request to R_1, R_2, and R_3.
3) The reflectors send responses to the victim.
Network N

Fig. 7. Smurf attack. An attacker (M) sends a spoofed request to the broadcast address of network (N). The amplifier (A) broadcasts the request to many reflectors (R) which respond to the victim (V).

$b@L_1$. The scenario is caused by misconfiguration and can potentially result in 2^n duplicate requests created where n is the number of iterations.

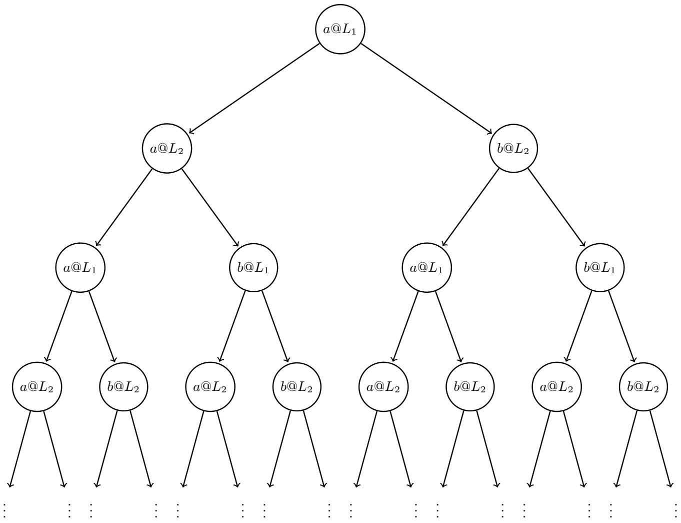

Fig. 8. Fork Loops. In this example taken from [29] a single request spawns 2^n duplicates where n is the number of iterations.

D. Discussion

In this section we have described many different types of amplification attacks. It is clear from our descriptions that the attacks seen in Table I are very different from Fraggle and Fork Loop attacks.

Therefore, this survey will focus on the types of amplification attacks seen in Table I and as described in Figures 5 and 6. The major advantages of amplification attacks from an adversarial point of view are:

- Attack traffic can be amplified beyond the attacker's upload bandwidth.
- The attacker remains anonymous by using reflection (IP address spoofing).
- The attack traffic at reflectors and amplifiers can be very difficult to distinguish from legitimate traffic.
- Using this "force over technique" [37] attack, the attacker exploits vulnerabilities in the Internet and the finite resources of the victim, and does not need to compromise the systems involved with malware prior to the attack.

III. Preventing Amplification Attack

In this section, we will describe preventative measures for the amplification attacks. For example, we discuss preventative measures which amplifiers can use before responding to requests (e.g. response rate limiting), and more radical approaches such as alternative content delivery systems [54]. We will discuss non-preventive (i.e., reactive) countermeasures, such as detection and filtering, in Section V.

A. Alternative Internet Architectures

Alternative Internet architectures such as Capability Based Architectures (CBA) [55], [56] or Content-Centric Networking (CCN) [54] may remove existing DoS vectors. For example, in CBA, senders must first obtain "permission to send" from the destination before sending packets. Anderson et al. [55] proposed that verification points around the network ensure that the source has permission to send data to the destination. To obtain permission, Anderson et al. suggested that the source uses Request-To-Send (RTS) packets, which are routed between RTS servers to ensure the victim is not flooded.

However, the problem with the CBA system from Anderson et al. [55], and the more recent TVA [56], is that every packet must include some kind of extra "capability" information. Maintaining, exchanging, and checking capabilities creates a lot of extra overhead for networks, and the increased packet size may also increase fragmentation.

CCN will remove the possibility of DNS based amplification attacks simply because DNS is no longer required. In CCN, ISPs can deploy content routers to cache data which many users request. However, CCN uses the request/response paradigm in the form of "Interest" and "Data" packets, and it is currently unknown how resilient CCN would be to amplification attacks or new forms of DoS attacks.

B. Lowering Amplification Factors

Alternative, to remain compatible with the current Internet design, is fixing the protocols that are vulnerable to amplification. One such approach is to lower the amplification factors, e.g., by increasing the size of the requests in vulnerable protocols. This would reduce the amplification factor, but also increase the traffic across the Internet, which is not desirable. Moreover, it is not reasonable to suggest that all protocols should be amplification free. For example, it is quite reasonable to expect that request packets for large files are smaller or fewer than the response.

A more practical approach for lowering the amplification factors may be to disable the redundant services that amplifiers offer. For example, Rossow showed that the Network Time Protocol (NTP) has the highest amplification factor out of the 14 UDP protocols he tested [22]. Kührer et al. achieved a reduction of the amplification factor of NTP by disabling the NTP services with the highest request/response factors [17]. Since the services with the highest request/response factors do not affect the core capabilities of NTP, these extra services can be disabled while still enabling time synchronization.

Next to NTP, many other protocols (e.g. DNS, CharGen, SNMP) can be used for amplification attacks. Some DNS providers have recently started deprecating the DNS ANY request in response to the growing threat of amplification attacks. One provider, CloudFlare, now responds with the RCODE 4 "Not Implemented"[1]. However, identifying and disabling services with high request/response factors takes time, and may result in the loss of functionality for benign applications.

C. Reducing the Number of Amplifiers

Many amplifiers were used in the larger amplification attacks. For example, over 30,000 separate open DNS resolvers were used in the Spamhaus attack (See Table I). A number of projects listed in Section VI are attempting to bring misconfigured amplifiers to the attention of system administrators so that they can be patched or configured correctly. However, this relies on widespread cooperation between administrators who may not have the knowledge or the resources to update their servers.

Another interesting point to make here is does the Internet need so many public services such as DNS servers? Perhaps more cooperation between network providers can lower the impact of amplification attacks by lowering the number of amplifiers required to operate the Internet.

D. Response Rate Limiting

Moreover, amplifiers could also be configured to respond to a limited number of request from each IP or network address within a given time frame. For example, support for response rate limiting is easily applied to DNS when using newer versions of BIND. However, rate response rate limiting protects against abuse of a single amplifier and an attacker may simply choose to use multiple amplifiers at a low request rate. It only takes an attacker a short amount of time to look for a sufficient number of different amplifiers (less than a minute in some case [22]).

E. Model Checking

Model checking can also be used to reveal protocols which suffer from amplification attacks due to misconfiguration or bad design [14]. By modeling the protocol using rewriting logic [57], a set of system states can be generated and checked for amplification by comparing the cost of servicing requests against a predefined threshold[2]. This approach can be used to detect flaws in protocols where the amplification factor at a single point might not seem significantly high, and it has been used to describe forking loops in SIP [14].

However, the model checking as used in [14] relies on a breadth first search in a graph of system states to look for signs of amplification. This offers many challenges, the most difficult of which may be that it requires a lot of time and space to construct the graph of states for larger and more complex systems.

F. Sessions for UDP

Many amplification attacks rely on stateless communication via the UDP protocol in order to send large amounts of data to spoofed IP addresses. Those amplification attacks are made more difficult if UDP-based protocols required sessions to be opened (similar to the TCP three-way handshake) before large amounts of response data can be transmitted. To counter this, one can include session information in UDP packets. For example, clients using the Steam query protocol have to request a 4-byte challenge before they can request large amounts of information about a game server [58]. The client has to append the challenge response to future requests. However, attackers may be able to use responses from

[1] https://blog.cloudflare.com/deprecating-dns-any-meta-query-type/

[2] The cost in [14] is the number of duplicate packets generated by SIP proxies

session initialisation exchanges for amplification attacks so long as they can open new sessions for the same victim with many amplifiers, and can also eavesdrop on the victims Internet traffic to see responses.

Another downside to session hardening UDP protocols is that it increases the packet sizes and the number of packets sent, which conflicts in particular with mobile regimes. It may also add latency to the initial request to open a sessions. The most challenging aspect of session hardening existing UDP-based services is compatibility with existing clients. In order to prevent amplification attacks session support would have to be universal, and upgrading may cause problems with legacy systems.

G. Differentiating Bots From Humans

Another technique which would prevent attackers from using certain services in amplification attacks is to use a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [59]. For example, DNS requests are often caused by a user looking for a website. If such DNS requests required that a user completes a CAPTCHA, then it would be harder for attackers to use open DNS resolvers for attacks. However, this may delay a users access to information and complicate legitimate automated services which use DNS. With this in mind, Oikonomou [60] attempted to model human behavior so that servers can automatically distinguish between requests from bots and requests from humans.

There is another problem with only replying to requests from humans, and that is that many legitimate automated processes need to make requests for information over the Internet. These processes are prevented from accessing services secured by a CAPTCHA or other bot detection mechanisms.

H. Filtering Spoofed Packets

In most amplification attacks the attacker sends requests with the spoofed IP address of the victim. Filtering spoofed packets is considered one of the most effective countermeasures against amplification and other DoS attacks [17], [61]–[63]. However, in 2005 the MIT ANA Spoofer Project showed that around 25% of ASes allowed spoofed IP packets to be sent out of their network, and the ability of attackers to launch amplification attacks shows that it is still a problem today. Recent estimates state that it is still possible to send spoofed IP packets from about 20% of the Internet [61]. The next section will discuss spoofing in more detail and discuss some of the methods proposed to defend against it.

IV. Source IP Address Spoofing

An inherent requirement for amplification attacks in practice is IP address spoofing. UDP and IP have no built-in mechanism to determine if the source address is spoofed, so amplifiers reply to the spoofed address instead of the original sender. Using spoofing for malicious purposes was first discussed in 1989 [23], and a detailed analyses of the problem was carried out by Heberlein and Bishop in 1996 [64], and again by Dunigan in 2001 [65]. Since then there have been many other surveys related to spoofing which we have summarized in Table II. These surveys can be broadly categorized as either focusing on spoofed packet detection/filtering [66], [67], or tracing the attacker [68]–[71].

Chen and Yeung [44] also define three common IP spoofing techniques which all have different purposes, random spoofing, subnet spoofing, and fixed spoofing. In *random spoofing*, the attacker randomly generates source addresses for the attacking packets. This technique is often used in TCP SYN flood attacks [20] where the destination of replies are not relevant to the success of the attack but the attacker still wants to obscure their identity. In *subnet spoofing* the attacker will choose a source address in its own subnet, which can help to avoid some countermeasures such as ingress filtering, but limits the number of possible victims. *Fixed spoofing* is the form of spoofing which is most commonly used in amplification attacks. This is where the attacker chooses the IP address of a single victim to be the source address on the spoofed packets, meaning that replies will be sent to the victim and not the attacker(s).

We will start this section by discussing the techniques used to monitor spoofing and to map networks which do not guard against it. Later we will (a) survey the contributions listed in Table II in a single concise discussion, and (b) significantly extending the discussion to include new approaches. This will involve categorizing spoofing defenses as being for either packet filtering or traceback, as shown in Figure 9.

A. Monitoring Spoofing In The Wild

There are two techniques which are being used to look at spoofing on the Internet. *Active monitoring* attempts to send spoofed packets and monitor the replies, while *passive monitoring* requires analyses of Internet traffic, e.g, NetFlow data. In the following two subsections we examples of both approaches.

1) Active Monitoring: The MIT Spoofer Project [61] measures how susceptible the Internet is to IP spoofing. To achieve a high coverage they have created and

Survey Name	Papers Covered
Detecting Spoofed Packets [72]	[73]–[75]
IP Spoofing Defense: An Introduction [66]	[76]–[91]
On the State of IP Spoofing Defense [67]	[72], [76], [78], [79], [84], [85], [88], [89], [92]–[109]
DDoS: Survey of Traceback Methods [69]	[74], [79], [93], [110]–[120]
A Survey of IP Traceback Mechanisms to Overcome Denial-of-service Attacks [70]	[113], [116], [121]–[124]
A Survey on Packet Marking and Logging [71]	[93], [113], [114], [116], [118], [121], [123]–[131]
On IP Traceback [68]	[27], [73], [107], [110]–[114], [116], [132], [133]

TABLE II
SURVEY PAPERS FOR FILTERING SPOOFED IP PACKETS AND DETECTING ATTACKERS.

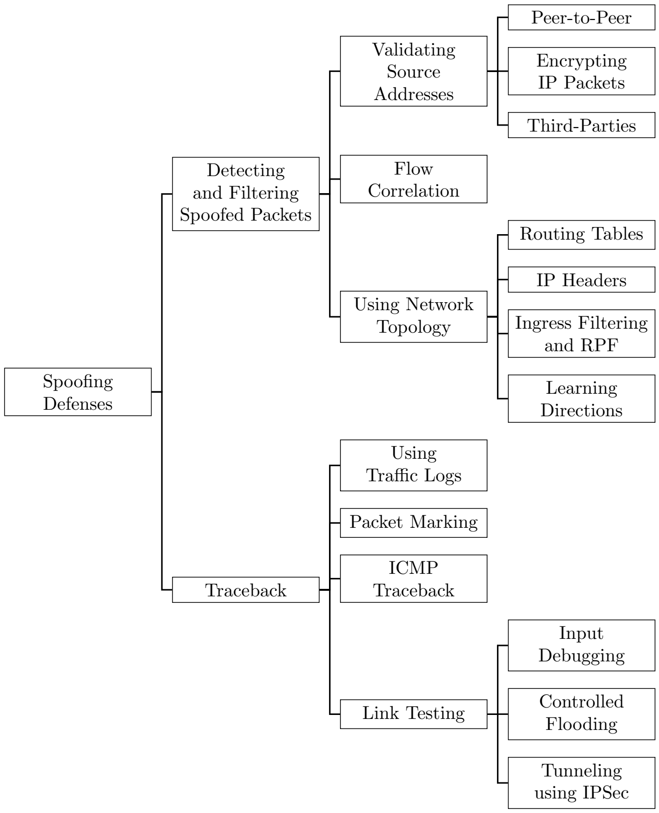

Fig. 9. A categorization of source IP address spoofing defenses.

distributed an application which volunteers can install. The application sends several spoofed UDP packets to a server controlled by the project. The application then connects to the server using legitimate means to determine which packets were received or lost. The results of the MIT Spoofer Project have shown that about 25% of ASs are currently vulnerable to IP spoofing. Daily statistics are also available on their website [134].

Kührer et al. [17] leverage the fact that some DNS proxies do not correctly change the IP addresses when forwarding DNS packets to test if their AS allows IP spoofing. They found 2,063 ASes that allow spoofed traffic by detecting if DNS replies have a different source IP addresses than the destination of the initial request.

The disadvantage of their approach is that it relies on networks which have open DNS resolvers that do not change the source IP address on responses. However, their approach does not depend on the distribution of volunteers like the Spoofer Project, and the researchers were largely free to choose which networks to investigate due to the many available open DNS resolvers.

2) Passive Monitoring: An orthogonal question to see who is spoofing, is to measure how much Internet traffic is actually being spoofed. The UCSD Network Telescope from CAIDA [62] is attempting to answer this question by monitoring traffic which is sent to darknets. A darknet is an address-space which is routable but where no traffic is expected to be sent, i.e., the addresses are not allocated. Monitoring traffic to darknets can be used to spot attacks, scans of the Internet, and misconfiguration of machines.

To explain how a darknet can be used to detect spoofed packets we will briefly describe the steps of a TCP-SYN flooding attack [20]. When an attacker is using spoofing for a TCP-SYN flooding attack and randomly chooses a source address for the spoofed packets, there is a chance that the address is in the address space of the darknet. As a consequence, the victim of the attack will response to the spoofed addresses, and the SYN/ACK packets from the victim will appear as traffic to the darknet. These SYN/ACK packets are known as backscatter, and since the darknets are supposed to cause no traffic, backscatter is assumed to be the result of spoofing once misconfiguration has been ruled out.

B. Validating the Source IP Address

There is no built in mechanism in the Internet Protocol which can prevent spoofing. Its function is simply to provide best effort delivery of datagrams [135]. Nevertheless, over the next two subsections we will discuss various techniques which can be used to filter spoofed packets. In this section we will concentrate on methods which use some form of authentication to validate packets as they arrive and filter those which are not validated. In the next section we will look at more probabilistic methods.

1) Peer-to-Peer: There are many peer-to-peer methods with which peers can attempt to validate the source IP address of packets. For example, probes can be used to provoke responses from the source address to see if it sent the packet [47]. However, packet loss means that some probes may be lost, and other unpredictable behaviour in the Internet means that probing cannot guarantee that the packet was spoofed.

Additional information embedded in packets and which can be validated using a peer-to-peer network has been used to authenticate the source IP address [78], [87], [100], [106], [136], [137]. This information can be unique between two peers, and may also have a time limit associated with it. The literature also uses different terminology to refer this data. Some papers use the word "cookie" [100], [101], [138], whilst others use the word "key" if the data is used in cryptographic functions [78], [106], [139].

Whilst the basic principle of using peer-to-peer methods to validate the source IP address is the same, all of the proposals have their own strengths and weaknesses. For example, large "puzzles" sent by servers may themselves be used in amplification attacks [101], and there are additional processing and space overheads associated with generating and stamping "passports" that are appended to packets [106]. Some of these overheads can be lowered, for example, if only routers at the edge of networks add data to, and validate, packets. However, this implies cooperation between network operators to install new functionality at the edges of their networks when the benefits to them may not outweigh the extra costs.

Lu et al. [136] proposed Multi-purpose Anti-Spoofing Key (MASK) where packets are tagged with MASK labels which are unique between the source and destination routers. At first, MASK labels are exchanged using a TCP connection. Then, MASK labels can be included in subsequent packets by overwriting the ID field in the IP header. However, MASK needs to be evaluated in more realistic scenarios. Overwriting the ID field means that datagrams can not be fragmented. It is also not clear how big MASK labels are, and if they will fit in the 16-bit ID field. Furthermore, MASK appears to be susceptible to eavesdropping attacks as MASK labels are not exchanged securely and are also sent in plain text in subsequent packets.

Another example is the Source Address Validation Architecture with Host Identity Protocol (SAVAH) [139], which can filter spoofed packets at the edge of local networks. It requires that all end-hosts and gateway routers support the Host Identity Protocol (HIP) [140], [141]. SAVAH filters spoofed packets by requiring senders replace the source IP address of packets with a hash. The hash is created using the contents of each packet and a key agreed previously when the sender joined the network. This is cryptographically stronger than [136] so long as the key was exchange securely. However, this approach degrades performance of the network rapidly due to the processing overheads of hashing every packet.

2) Encryption: An overall more secure approach than appending validation data to packets is to encrypt all IP packets. Using encryption between peers would not only prevent most spoofing attacks, but it also has the added benefit of all Internet traffic being secure by default [97]. One suitable mechanism by Gilad and Herzberg provides relatively lightweight encrypted communication [48].

However, the computational overheads and extra traffic caused by encrypting all IP traffic is currently impractical, and would also conflict with IoT scenarios using constrained devices. Nevertheless, encryption can be used in select cases such as communication with amplifiers.

3) Third-party Authentication: Sometimes it is not desirable for routers and end-points to use a peer-to-peer network, or negotiate keys and encrypt packets because of the additional processing and latency overheads. Using extra third-parties which are usually not involved in packet forwarding offers a way of authenticating packets which can potentially lower the extra processing demands placed on routers and end-points [46], [142]–[144].

For example, Gonzalez et al. [143] proposed that routers should send copies of packets to a "judge" (the trusted third-party). The judge should know the IP addresses serviced by the routers in order to make a decision whether or not the packet is spoofed, and the judge can also query routers in the absence of data.

Noureldien et al. [46], [144] suggested that authentication servers in local networks can be used to determine if TCP-SYN packets are spoofed. They propose that all packets that exit a network are inspected by an authentication server. An authentication server can also check incoming packets with the authentication server at the source network. This approach can lower processing overheads for some hosts in the network but it does not address concerns about additional latency and traffic volume.

C. Flow Correlation

These methods are mostly aimed at detecting TCP-SYN flooding attacks by observing unbalanced SYN and SYN-ACK packet flows [44], [45], [116], [124], [145], [146]. Even though they cannot help defend against UDP based amplification attacker, we have included them in this section on spoofing for completeness.

D. Filtering Spoofed Packets Using Network Topology

Almost all of the filtering methods seen so far come with a high cost in terms of additional latency, traffic volume, and/or processing at routers. In this section we will discuss methods which can lower the overheads by using what information is already known about the structure of the Internet. For example, some IP addresses are not meant to be used outside of local networks. If a packet enters a network with a private IP address (e.g., 192.168.1.1) then this packet should not be routed, as Network Address Translation (NAT) [147] at the source network should have changed the source address of the packet.

More generic methods of detecting spoofed packets, as discussed next, are probabilistic because the Internet is both dynamic and complex. However, some of these approaches also include a mechanism to determine if a packet was actually spoofed following detection.

1) Using IP Headers: One way with which spoofed packets can be filtered without using additional bandwidth is to analyze information present in the IP headers. If the legitimate values for a source IP address is known, then spoofed packets which do not have the correct information can be identified and filtered.

For example, spoofed packets can be filtered if their hop count does not match what is known about the topology of upstream routers. This information can be estimated using the Time To Live (TTL) field in the IP header [88], [102], [148]. However, there may be many paths between the source and destination of spoofed packets. One interesting idea is to use Ant Colony Optimisation (ACO) algorithms to find these paths [149], and to collect TTL values which can be used to filter spoofed packets.

It is even possible to filter spoofed packets by detecting what operating system a remote host is using, i.e., by estimating the initial TTL of packets [94], [96]. However, filtering spoofed packets based on the TTL and other fields does not guard against spoofing attacks where the attacker can manipulate the values in the IP header.

2) Ingress Filtering and RPF: Filtering IP packets with spoofed addresses can be achieved using knowledge of the IP addresses allocated by the upstream or downstream networks [76], [103], [150]. *Ingress* filtering filters incoming packets. In contrast, *egress* filtering filters packets which are exiting the network.

Ingress filtering (BCP 38) is sometimes implemented at the periphery of the Internet to stop packets with spoofed addresses being routed by edge routers [76]. However, not all network operators implement BCP 38. Ingress Access Lists are also typically maintained manually, and having outdated or misconfigured lists can prevent legitimate or allow spoofed traffic [105].

Unicast Reverse Path Forwarding (uRPF) is an implementation of BCP 38 and uses forward routing information to filter spoofed packets [86]. Incoming packets are checked against a routers Forwarding Information Base (FIB) to ensure that packets are only forwarded if they come from the interface which is on the router's best path to the source address. uRPF also has a "loose mode" where it can check the source addresses packets without taking the interface into account. This allowed uRPF to be used on routers with multiple links to multiple ISPs [86].

3) Learning Directions: Routers can record information from incoming packets in order to filter traffic which comes from unexpected directions [77], [81]–[83], [92], [109]. The Source Address Validity Enforcement (SAVE) protocol [107] is one example where routers can learn the expected incoming interface for a source address in order to authenticate subsequent packets.

The approach proposed by Wu et al. [92] is called Source Address Validation Architecture (SAVA). To prevent IP spoofing in a local network, SAVA routers bind the source IP address and MAC address to a specific switch port. The next step is to filter packets at the Intra-AS level. This is done by associating source addresses with the incoming interface. The last step, filtering spoofed packets at inter-AS level, involves adjacent ASes exchanging address blocks so that packets from other ASes can be filtered.

However, these methods do not perform well during legitimate changes to the Internet's structure or when there are multiple paths between two routers [77]. To counter this problem these methods can consider multiple valid interfaces per source address at the cost of lower filtering accuracy. Mirkovic et al. [77] proposed dropping some TCP packets which originate from new interfaces. Under normal circumstances these packets should be retransmitted and the operation of the Internet is not severely compromised. However, this approach does not help with combating amplification attacks which are perpetrated using UDP or where the attacker retransmits attack packets.

4) BGP and Routing Tables: BGP is responsible exchanging routing information between ASes. As a result, many papers have proposed that BGP can also be used to help filter spoofed packets [79], [80], [108], e.g., by adding anti-spoofing information to BGP update messages [84] using packet marking techniques such as Pi which we talk about in Section IV-E2.

One way to use BGP routing tables to filter spoofed packets is to use Distributed Packet Filtering (DPF) [79], and its extensions Inter-Domain Packet Filters (IDPF) [80], [108] and Extended IDPF [137]. Using IDPF a router examines BGP routing tables to check if it is on the optimal path between the packet's source and destination. However, this approach suffers from the same security flaws as BGP because there is no built-in mechanism to check the integrity and source of BGP messages [151]. To improve the security of BGP and thus IDPF, Dawakhar et al. have suggested using Credible BGP [152].

E. Tracing the Attacker

Next to detecting and filtering spoofed IP packets, an orthogonal challenge is to trace the actual source of spoofed traffic [68]. Such tracing is useful, not only to stop attacks as they are happening by filtering the attack traffic at the source (thus saving the resources at upstream networks), but also so that the perpetrators can be prosecuted and further attacks prevented.

Tracing spoofed packets is difficult because of limited access to external routers and the the high overheads involved. The existing traceback vary in their objective and where they are applied. For example, an administrator of a local network might want to know which MAC address or interface is being used to send spoofed packets [153], whereas an attack victim might want to know which network the attack originated from.

In amplification attacks, the victim receives attack traffic from amplifiers. Thus, the traffic facing the victim is not spoofed. Instead, spoofing detection and tracebacks needs to be performed at the amplifiers, possibly with the help of honeypots [154].

It is also important to realize that the traceback can only be used to identify the source of spoofed packets. Detecting the *individuals* involved requires many out-band techniques such as monitoring the chat networks used by cyber criminals. Furthermore, tracing becomes inherently difficult in case of distributed attack sources, such as DDoS botnets.

1) Using Traffic Logs: Tracing spoofed packets back to the attacker can be done by analyzing traffic flow data collected by routers [116], [124], [146]. With log-based traceback there is always the potential to trace a single packet [145], and to trace attackers who use reflection. However, these methods often require that traffic data is either collected or stored by routers [110]. This comes with many practical difficulties such as additional hardware costs, as was the case with the original proposal by Snoeren et al. [116], which requires Source Path Isolation Engine (SPIE) routers in place of existing routers. One solution to replacing existing routers is to use tap devices which eavesdrop on traffic [155]. Finally, one can lower the amount of storage required to trace attackers by using hashes [124].

2) Packet Marking: The idea of packet marking is to record the path which packets follow into the packets [74]. Early packet marking schemes appended the IP addresses of routers into packets [74], [122]. However, the four addition bytes needed to represent an IPv4 address and the large number of hops quickly adds to the size of packets. Song and Perrig suggested reducing the amount of space required by encoding addresses using hashing and a map of upstream routers [114]. Yaar et al. discussed ways to lower the space needed by using shorter identifiers than IP addresses [85]. However, appending data unnecessarily to packets is still undesirable because it can increase fragmentation. To address this problem some packet marking schemes also suggested overloading IP header fields [74], [89], [91], [93], [114] and only encoding the addresses of edge routers [90].

Savage et al. introduced Probabilistic Packet Marking (PPM) to reduce size and processing overheads of packet marking even further [74], [113][3]. PPM can be used to identify network path(s) traversed during an attack so long as the victim receives sufficient marked packets. Dong et al. [157] designed a single-bit-per-packet scheme. However, Adler [95] noted that there is a trade-off between the number of bits allocated to PPM and the number of packets that must be received by the victim.

Duwairi et al [158] proposed a hybrid PPM and data storage method called Distributed Linked-List Traceback (DLLT). The "store, mark, and forward" operation used by DLLT requires a "marking field" in each packet big enough to store a single IP address, and a data

[3] Shokri et al. [156] had a similar idea 6 years later and called it Dynamic Marking.

structure on each router in which to store information about forwarded packets. Any router which marks a packet must first store the address in the marking field (if there is one) along with a packet identifier, then the router is free to write its address in the marking field before forwarding the packet. A linked list is inherently created using this operation because the address in the marking field points to the previous router, which has a record of where it received the packet from, and so on. However, its difficult to trace attacks with multiple attack sources [130]. To counter this, Song and Perrig looked at the security of markings and proposed using time-released key chains to authenticate them [114].

Li et al. [26] proposed an efficient packet marking scheme specifically for reflection attacks called Authenticated Deterministic Packet Marking (ADPM), which is an extension of the approach proposed by Belenky et al. [159]. ADPM assumes that all routers are capable of matching request and response packets and that routers adjacent to reflectors cooperate to copy packet markings from the request packets to the replies.

Dean et al. phrase packet marking as a polynomial reconstruction problem and encodes path information as points on polynomials [93]. Chen and Lee also extended their work to target reflection attacks [160]. They did this by requiring reflectors copy packet markings from the request packets to the replies. This enables the victim to trace paths to the attacker using less than 4,100 packets, but it may be possible to combine logging at routers and PPM to achieve traceback with a smaller number of marked packets [123]

A drawback of most packet marking schemes is that they rely on routers to maintain and update the markers. In case routers drop or do not extend the marks, packet marking becomes significantly less effective.

3) ICMP Traceback: Bellovin et al. proposed iTrace where routers send ICMP messages to the destination address of packets they forward so that the destination can see the path which the packet followed [112]. ICMP traceback can also be probabilistic in that the chance of a router generating an ICMP message is small, thus lowering overheads in a similar way to PPM [132], [161], [162].

Mankin et al. improved iTrace with "intention-driven" traceback after they noticed that the chance of a router picking an attack packet close to the attacker is much smaller than closer to the victim for certain DoS attacks [132]. Wang and Schulzrinne also proposed a new ICMP message which can be used in ICMP traceback for reflection attacks [163].

4) Link Testing: Link testing is the systematic testing of intermediary links between routers, such as implemented via *input debugging* [110], DECIDUOUS [73], or *controlled flooding* [111], as described next.

Input debugging attempts to determine from which upstream router an attack is coming from by recursively generating attack signatures to distinguish malicious packets from normal traffic. However, attack signatures that only match malicious traffic are difficult to create. Good attack signatures match only attack traffic and a only little to no legitimate traffic [110].

The Intruder Detection and Isolation Protocol (IDIP) [75] describes an architecture which also uses attack signatures and can be used for link testing and attack mitigation. In IDIP a single Discovery Coordinator sends trace messages to other IDIP enabled devices. The trace messages contain a description of the event, i.e., an attack signature, using the Common Intrusion Specification Language (CISL) [164]. IDIP devices which receive trace messages reply with report packets. The Discovery Coordinator can then decide what action to take, including instructing other IDIP devices to block attack traffic for a short amount of time [75].

Another link testing method uses IPSec [165]–[167] to trace the network from where a spoofed packet originated. For example, the DECIDUOUS implementation [73] creates secure tunnels between itself and upstream routers until it finds the network from where the attack is coming from. However, DECIDUOUS needs to have a complete overview of the network topology to choose routers with which to tunnel. Furthermore, DECIDUOUS has high processing and traffic overheads caused by setting up and tearing down secure tunnels.

Controlled Flooding is a link testing method which floods each upstream router with packets. This technique needs the knowledge of the topology of the Internet. Thus, a map of the Internet is obtained before link testing [111]. The purpose of flooding upstreaming routers is to determine if the attack traffic affected. If the attack traffic is disturbed significantly when sending traffic to a router then it is assumed that this particular router is one forwarding attack traffic.

F. Discussion

We have surveyed approaches to defend against spoofing. However, there are also more theoretical works in this area. For example, given a graph of the network topology and a set of firewall rules for each node, Santiraveewan and Permpoontanalarp proposed a methodology which can check to see if spoofing is possible [168].

With all of these defenses, it may be confusing as to why spoofing is still a problem. There are many practical and economic reasons. Many of the proposals incur extra latency, traffic volume, computation, and/or storage overheads. Others are expensive to implement, as they require new hardware or staff to maintain them. Finally, it is not clear how well the methods will perform with only limited deployment [79], and what rewards they offer to early adopters [84].

We have seen that it is to difficult to prevent spoofing or trace the attackers in reflection/amplification attacks without considerable investment and cooperation by the different network operators. Therefore, we have to look for other ways of detecting and mitigating amplification attacks as soon as possible. The next section will address this topic in more detail.

V. Detecting and Mitigating Amplification Attacks

The preventative measures described in Sections III and IV are effective defences against amplification attacks. Nevertheless, we have seen that they cannot always be relied upon. So the defences described in this section are to be used when preventative measures fail.

Defending against amplification attacks involves *attack detection* and *attack mitigation*, e.g. packet filtering or black holing. Attack detection is easier to perform near to the victim because traffic flows resemble a funnel with many attackers sending packets to a single location [11]. In contrast, attack mitigation is best performed closer to the source to save the resources of the victim and upstream networks.

Chang [27] explained that DoS defenses can be located at four areas: (i) the victim's network, (ii) the victim's ISP network, (iii) further upstream networks, and (iv) attack source networks. Zargar et al. [11] classifies DoS defenses into two categories: (i) network/transport-level, and (ii) application-level. They also sub-categorized DoS defenses based on their deployment location: (i) source, (ii) destination, (iii) network, and (iv) distributed/hybrid.

However, in this survey we are concerned specifically with amplification attacks, and we observed that it is also possible to defend against amplification attacks at the amplifiers. Therefore, we have have split amplification attack defenses into the following location categories described in Figure 10: (i) at the victim's network, (ii) at the source network(s), (iii) a single unspecified location in cases where the authors have not been clear about the deployment location, (iv) distributed, and (v) at the amplifier(s). In cases where the amplifier itself is the target of an amplification attack, defenses will be limited to source-based and amplifier-based defenses.

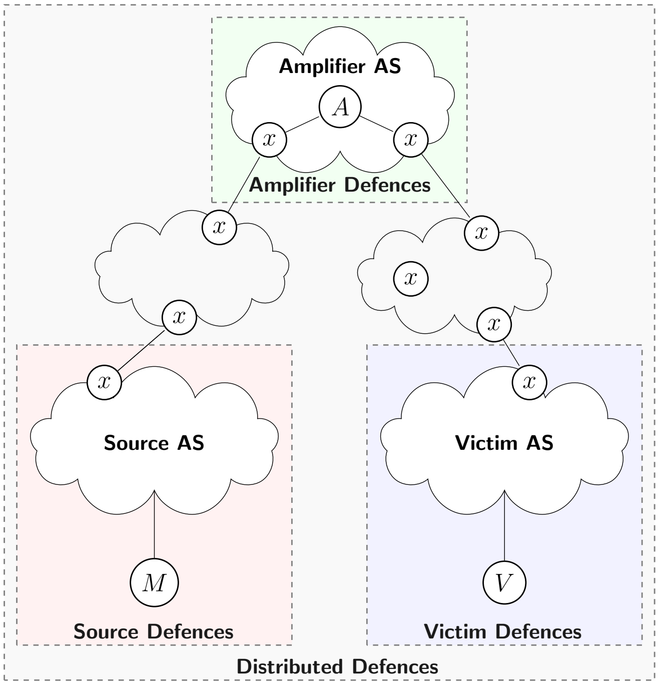

Fig. 10. Locations for defending against amplification attacks. Routers are marked x.

It is also important to note that not all of the DoS defenses surveyed by Chang and Zargar et al. are applicable to amplification attacks. The most devastating attacks we described in Section II originated from multiple sources, and combined both traffic amplification and reflection. Therefore, in this section, we will discuss the defenses in Table III which include new proposals that specifically target amplification attacks, but we will also include some older proposals which can part-defend against amplification attacks, e.g., defenses against reflection.

A. Victim-end Defenses

Despite attack detection and packet filtering being separate objectives, many proposals attempt to do both near the victim.

Kambourakis et al. [13], [170] proposed a DNS Amplification Attacks Detector (DAAD) which logs outgoing DNS requests and incoming responses in lookup tables at the victim's network. Responses are matched to outgoing requests and responses that don't match requests are flagged as suspicious. The authors also propose that the DAAD should be able to alter firewall rules to filter packets. However, amplification attacks do

Method and References	Description	Location	Time	
			Before	During
Attack pattern correlation [169]	Tries to detect reflection attacks by correlating the arrival rate of packets at the victim with known attacks.	Victim		✓
Missing request packets [13], [170]–[173]	Attempt to match incoming responses to previously sent requests.	Victim		✓
History-based IP Filtering (HIP) [174]	Attempts to filter packets during periods of high congestion which come from previously unseen addresses.	Victim		✓
Comparing incoming and outgoing packets at source networks (D-WARD) [175], [176]	Uses protocol specific knowledge to predict the number of expected responses, e.g., replies to ICMP requests. Can also rate limit attacks at the source.	Source		✓
Comparing the volume of incoming and outgoing traffic [177], [178]	MULTOPS [177] and TOPS [178]	Unspecified		✓
Detection with Information Sharing (DIS) [179]	Aimed at TCP based reflection attacks, but reflectors collaborate to improve the accuracy of attack detection. Similar collaboration could potentially improve attack detection at amplifiers.	Reflectors / Amplifiers		✓
PacketScore [180]	Routers score packets, and communicate with a central server for a global view and filter packets with a high score.	Distributed		✓
Repetitive packet sizes [28]	Monitoring DNS packet sizes as well as the number of packets across multiple routers. Suspicion is based on thresholds calculated using real data from the GEANT network.	Distributed		✓
Comparing payload bytes of request and response packets [22]	Analysed NetFlow data to detect pair-flows where the response payloads are larger than the request payloads (a similar idea to that which is used in MULTOPS and TOPS). Limited to 14 UDP protocols.	Distributed		✓
Comparing payload bytes of request and response packets, plus compare packet sizes of requests [18]	Combines techniques from Rossow [22] (comparing request and response sizes) and Huistra [28] (similar packet sizes between requests).	Distributed		✓
Aggregate-based Congestion Control (ACC) [181]	Looks for high bandwidth aggregates (collections of packets sharing the same destination address). Uses pushback (also used in [182]) to instruct upstream routers to rate-limit flows.	Distributed		✓
Spoofing protection for amplifiers [138]	Specific to amplification attacks. This method detects spoofed packets which do not have the correct session key provided by the amplifier for the source address.	Distributed		✓
AD and PAD [183], and TRACK [184]	A congested victim instructs upstream routers to use packet marking in order to trace the attack and then requests they filter the traffic. These methods may need to be updated for amplification attacks to take reflection into account.	Distributed		✓
Game theory to predict attacks [185]	It is possible to study the motivation of attackers and the security of the Internet to predict attacks.	n/a	✓	
Darknets to detect scans [22], [186]–[188]	Monitoring darknet traffic can detect random scans for amplifiers such as those seen before the Spamhaus attack [188].	Distributed	✓	
Honeypots to detect scans and attacks [22]	Honeypots can take the place of amplifiers and are a powerful way of detecting scans and amplification attacks. They also require careful configuration so they do not cause harm to the victim.	Amplifiers	✓	✓

TABLE III

METHODS FOR DETECTING AND PREDICTING AMPLIFICATION ATTACKS. SOME OF THESE PROPOSALS ALSO INCLUDE NOVEL TECHNIQUES WHICH AIM TO IDENTIFY AND FILTER ATTACK PACKETS.

not only misuse DNS as we described in Section II. So a more general detection method is desirable.

Tsunoda et al. [172] proposed a general method for matching request and response packets at a single point between the amplifier and the victim, e.g., at the victim's gateway router. They later extended their work with mathematical analysis and additional experimental results [173]. However, in order to correctly identify the responses for each request, a router needs to record the source and destination addresses, protocol field, application headers, etc. for every request it forwards. This requires memory-intense management operations. A partial solution to this problem is to use Bloom filters to efficiently store requests and check them against incoming response packets [171].

History-based IP Filtering (HIP) [174] attempts to filter packets during periods of high congestion which come from previously unseen addresses. It does this by adding addresses to a table during normal operation, and uses a sliding window to remove expired addresses.

Wei et al. [169] proposed using traffic pattern correlation to look for patterns which are out of the ordinary, or which match known attack patterns. The advantages of using traffic pattern correlation and HIP for attack detection, are that they are lightweight and protocol independent. However, methods to detect attacks based on previous behavior can struggle to distinguish between attacks and unexpected bursts in legitimate traffic (flash crowds) [189]. Furthermore, an attacker that knows these defenses can "train" the victim before the attack [174].

Ultimately, detecting attacks at the victim's network may be ineffective because bandwidth may have already become saturated, or the volume of traffic is too much to process. This means that automated responses for filtering packets or alerting upstream routers to throttle traffic may not function correctly. In order to avoid this problem, another possible location for detecting and filtering amplification attacks is at the source network.

B. Source-end Defenses

Mirkovic et al. [175], [176] proposed D-WARD specifically to detect DoS attacks which involve spoofed packets or high traffic volumes. They monitor both inbound and outbound traffic flows at the source network, and compare them with flow models derived from normal traffic. The authors also note that an attacker will not reduce its outbound traffic even when notified of congestion at the victim, so D-WARD can rate limit suspected source addresses at upstream routers.

However, source-end detection and filtering techniques are not an effective defense against amplification attacks because: (i) the source of the attack can be distributed, meaning that the solutions need to be deployed on all networks, (ii) it can be difficult to differentiate between legitimate and attack traffic if each attacker only sends a few requests, and (iii) there is no immediate incentive for network providers to deploy source-end detection and filtering techniques because it is unclear what benefit it offers their customers.

C. Distributed Defenses

Victim-end and source-end defenses tend to be designed to run in isolation on a single machine, e.g., at a border router between networks. However, detecting and filtering amplification attacks at a single point in the Internet is problematic because of asymmetric routing, and the fact that attacks are distributed. This means that in order for isolated defenses to be effective, all inbound and outbound paths in the Internet should be symmetrical, and all networks should support the new defenses.

Amplification attacks are distributed threats against many potential victims, and as a result they require distributed defenses. Some distributed defenses also can detect and filter attacks after a victim has been taken offline by the attack and cannot use automated methods which use the affected connections to alert upstream routers [18], [22], [28], [180]. It is even possible to detect and filter DoS attacks with limited deployment of a distributed protocol alongside legacy equipment/protocols [190].

Kim et al. [180] proposed PacketScore which uses a DDoS Control Server (DCS) to collate reports from routers across the Internet. Routers with special reporting capabilities, called Detecting-Differentiating-Discarding Routers (3D-R), mark suspicious packets with a "score" which reflects how likely they are to be involved in an attack. The 3D-Rs then filter packets based on a variable score threshold provided by the DCS, which varies the threshold depending on the current load at the victim and the score distribution of attacking packets.

Mahajan et al. proposed attack detection at victims in terms of "aggregate" flows, and proposed using pushback mechanisms to filter flows at upstream networks whenever a link experiences sustained severe congestion [181]. In pushback, the congested router asks upstream routers which are involved in the flow aggregate to rate limit traffic flows [181], [182].

Chen and Park [183] proposed an Attack Diagnostic (AD) system in which DoS attacks are detected near to the victim, and packet filtering is executed close to the attacker. AD also combines some familiar techniques; packet marking (Section IV-E2) is used for traceback, and an approach similar to pushback is used to alert the source networks. Similarly, TRACK combines packing marking with packet filtering [184]. However, AD and Track are currently not suitable defenses for amplification attacks because their traceback methods will only trace back to the amplifier, but not to the actual attacker.

It is also possible to detect amplification attacks by analyzing traffic flow data collected by multiple routers. Usually these methods require data from locations between the source and the amplifier, as well as data from between the amplifier and the victim [18], [22], [28]. However, it is not required to have data from between the attacker and amplifier simply to detect DDoS attacks. Yu et al. [189] proposed using flow correlation to detect DDoS attacks, but unlike the correlating method proposed by Wei et al. [169], Yu's method can distinguish attacks from flash crowds. This was possible because it has access to data collected by multiple routers in a "community network".

Huistra showed that amplification attacks can be detected specifically by monitoring DNS packet sizes as well as the number of packets across multiple routers [28]. Huistra's method is split into two parts. The first (which can also be used to detect attackers) focuses on detecting IP addresses generating a suspicious number of similar sized DNS requests. Suspicion is based on thresholds calculated using real data from the GEANT network [28]. The second part can be used to detect the victim, and looks at the number of large DNS packets sent to a single IP address.

Rossow's analysis and detection of amplification attacks is similar to Huistra's but is not limited to a single protocol [22]. Instead, he focused on 14 UDP protocols which allow amplification. By analyzing traffic samples taken from routers belonging to a single ISP, Rossow was able to detect 15 real life amplification attacks against multiple victims by comparing sent and received bytes.

Based on the work by Rossow, Böttger et al. [18] developed a protocol-agnostic approach to detect amplification attacks. They introduce the following detection criteria, which are independent of a specific network service, but base on the assumption that attackers reuse attack requests: (i) similar packet size among all requests (and responses), (ii) similar payload among all requests (and responses), (iii) an increased amount of ICMP unreachable messages, and (iv) incorrect TTL values. They found that the first two criteria are most suitable for amplification detection.

D. Defenses At Reflectors/Amplifiers

Rossow and Huistra also showed that it is possible to detect open DNS resolvers that have been abused in amplification attacks by analyzing traffic data collected by routers in a similar way as mentioned in the previous subsection, but from an amplifiers perspective [22], [28].

Another approach for filtering spoofed packets, which was specifically designed to defend against amplification attacks, requires that session tokens are sent along with requests to amplifiers [138]. This requires that the first time a request is received from a new address, the amplifier replies with a unique session token which the sender should include in all future requests.

Including the same session token in all requests has low processing overheads but adds to the size of each packet. Furthermore, it is prone to eavesdropping if session tokens are not encrypted. If the attacker can eavesdrop on the victims Internet traffic then they can also agree new session tokens with amplifiers the victim has not yet contacted.

The work of Peng et al. [179] focused on TCP based reflection attacks but it used some interesting mechanics which might also improve the accuracy of detection at amplifiers. Their method monitored the cause of packets instead of just counting them. In their proposal, reflectors monitor incoming TCP RST packets and monitor those which have a corresponding SYN/ACK state for the outgoing connection. RST packets indicate that the reflector received a spoofed SYN packet from an attacker and sent a SYN/ACK to the victim, at which point the victim has replied with a RST packet. Peng et al. suggested that collaboration between reflectors can be used to improve the accuracy of attack detection. Once an attack has been detected, their system sends a warning to all other participating reflectors instructing them to stop the attack.

E. Defenses at Single Unspecified Locations

Not all DoS attack defenses are designed with a particular location in mind, or the location was not explicitly specified. This is problematic because the challenges vary among locations.

MUlti-Level Tree for On-line Packet Statistics (MULTOPS) [177] is one example where a router somewhere on the Internet is used to compare the volume of incoming and outgoing traffic to look for DoS attacks.

However, if MULTOPS is used on central routers which route packets for many different addresses, then the tree data structure it relies on will consume a significant amount of resources. In fact, MULTOPS itself can be targeted by memory exhaustion attack [177], [178].

Tabulated On-line Packet Statistics (TOPS) [178] improves on the accuracy of MULTOPS by taking into account the protocol being used, and also improves efficiency by using fixed length tables instead of trees. This makes TOPS suitable for use on busy links [11].

Both MULTOPS and TOPS assume that legitimate incoming and outgoing traffic is proportional [178]. However, this is not always the case, e.g., when streaming video. Attackers may also try to counteract MULTOPS and TOPS by trying to balance incoming and outgoing data volumes during an attack [11].

F. Predicting Future Attacks

It is desirable to know about DoS attacks before they happen. To this end, Fultz et al. [185] proposed using a game theoretical approach to predict DDoS attacks. They found that attackers only launch attacks if defenders have not invested in adequate protection or if penalties, e.g, monetary costs and risks of being caught are low.

It is also possible to detect scans for amplifiers which may act as a warning for future attacks. For example, darknets (also known as blackholes) are unused IP address spaces on the Internet. Since the IP addresses in darknets are supposed to be unused, any traffic to or from a darknet is a sign of either scanning, misconfiguration, or malicious intent [191].

Fachkha et al. [186]–[188] looked at 1.44TB of traffic data for a /13 address block of darknet IPs [4], and found 134 separate incidents which they claimed may have been amplification attacks.

However, Fachkha et al. may have detected scans for open DNS resolvers rather than actual attacks. To explore this idea we can look at Tables 7 and 8 in [188] to see the incidents they detected. Specifically, they detected 2 large events (February 19^{th} and March 18^{th}) with very high packets per minute. March 18^{th} is interesting because it is the same day as the Spamhaus attack [8]. However, this looks like a horizontal IP scan because it targets 50,257 separate IPs with 1 packet each, which is indicative of a horizontal scan rather than an attack.

We can also take a closer look at the Spamhaus incident by using Figure 6 of [188], which shows the packets per hour going to the Darknet monitored by Farsight Security:

- The first spike we are interested in is at 337 hours, which is on the 14^{th} of March. This is shortly before the Spamhaus attack which was reported by CloudFlare as taking place on March 18^{th} [8].
- The second interesting data spike at 385-409 hours is also before the attack.
- The graph for the 18^{th} of March (the day of the attack) is one of the quietest periods shown.
- The next spike they detect is at 517 hours, which is 3 days after the attack.

By referencing what we know about the Spamhaus attack [8], we suggest that the information presented by Fachkha et al. [188] shows horizontal scans for open DNS resolvers which may be related to the Spamhaus attack. Furthermore, attackers are unlikely to waste bandwidth sending speculative requests to unknown IPs when they can try to maximise the impact of an attack by only sending requests to known amplifiers. This supported by the quiet period observed by Fachkha et al. on March 18^{th}.

Finally, to support defenses and monitor attacks, one can set up amplification honeypots that emulate protocols that are vulnerable to amplification abuse. Honeypots can be used to log scanning activity and report attacks that abuse them. In addition, Krämer et al. show that honeypots can be used to derive signatures of attack traffic, such as domains abused in DNS requests [**?**], which is useful for detecting future attacks.

G. Discussion

Most of the defenses described in this section focus on amplification attack detection and filtering at the victim or by using distributed algorithms. Amplification attack defenses at the source networks are more desirable, but it is unlikely that defences specific to a single attack, e.g., amplification attacks, will be implemented in the short-term and on a large scale.

One contribution of this paper is to survey the defenses currently available for amplifiers and to make some suggestions for future directions. We found that this area has had relatively little attention compared to defenses at routers, but by adopting one of the approaches detailed in Section III it may be possible for amplifiers to prevent amplification attacks from happening in the first place.

VI. Conclusions

This paper discusses the current state of the art of research proposals for detecting, preventing, and tracing

[4] The data was provided to Fachkha et al. by Farsight Security https://www.farsightsecurity.com/.

TABLE IV
INITIATIVES TO IDENTIFY AMPLIFIERS

Reference	Description
Open Resolver Scanning Project [192]	Scans for open DNS resolvers and provide an automated system to notify affected networks.
Open DNS Resolver Project [25]	Scans for open resolvers and allows to query for such resolvers in a certain IP address range.
Measurement Factory [193]	Maintains a list of DNS servers that are known to serve as globally accessible open resolvers.
Open NTP Project [194]	Scans for NTP servers in IPv4 which can be used in an amplification attack.

amplification attacks. As part of this, we have also surveyed defenses against source IP address spoofing, which is essential for amplification attacks.

We have concluded that preventing source IP spoofing is the effective way of defending against amplification attacks. However, spoofed packets can still be sent from large parts of the Internet. So we need effective methods to detect and filter DoS attack packets as close to the source as possible.

Another subject for future work is to assess all of the defenses we have covered against one another using a wider list of empirical criteria, e.g., monetary cost, accuracy, memory overheads, levels of inter-AS cooperation, etc. Zargar et al. [11] surveyed the metrics used to asses DoS defenses and discussed their categorization in some of these terms. However, a more detailed study on individual defenses is crucial because all of the proposals we surveyed come with their own strengths and weaknesses. Some of which may not be obvious, and it is important for system administrators to be able to weigh up their options in a concise and meaningful way.

Promising amplification attack defences appear to be distributed DoS detection and filtering algorithms, which can mitigate attacks even when the victim has gone offline and is unable to take counter-measures itself. However, these methods require more collaboration between network providers than is currently the case [11].

More research is required to close the attack vectors which are being used in amplification attacks, e.g, closing open DNS resolvers, patching NTP servers, and promoting ingress filtering. A number of ongoing initiatives aim at tackling these issues, as summarized in Table IV.

REFERENCES

[1] M. Handley, E. Rescorla, and IAB, "Internet Denial-of-Service Considerations," RFC 4732 (Informational), Internet Engineering Task Force, December 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4732.txt

[2] T. Brewster, "Cyber Attacks Strike Zimbabweans Around Controversial Election," August 2013. [Online]. Available: http://www.techweekeurope.co.uk/workspace/zimbabwe-election-cyber-attacks-123938

[3] J. Leyden, "US Credit Card Firm Fights DDoS Attack," September 2004. [Online]. Available: http://www.theregister.co.uk/2004/09/23/authorize_ddos_attack/

[4] Prolexic, "Prolexic Stops Largest-Ever DNS Reflection DDoS Attack ," May 2013. [Online]. Available: http://www.tinyurl.com/prolexic-167gbit

[5] A. Pras, A. Sperotto, G. Moura, I. Drago, R. Barbosa, R. Sadre, R. Schmidt, and R. Hofstede, "Attacks by Anonymous WikiLeaks Proponents not Anonymous," 2010.

[6] M. Calce and C. Silverman, *Mafiaboy: How I Cracked the Internet and Why It's Still Broken*. Viking, 2008.

[7] D. Takahashi, "Hackers Attack Dota 2 and League of Legends Servers in Quest For One Game Livestreamer," December 2013. [Online]. Available: http://tinyurl.com/PhantomL0rd-drdos

[8] I. M. Prince; CloudFlare, "The DDoS That Almost Broke the Internet," March 2013. [Online]. Available: http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

[9] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[10] C. Douligeris and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643 – 666, 2004.

[11] S. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2046–2069, 2013.

[12] R. Vaughn and G. Evron, "DNS Amplification Attacks," March 2006. [Online]. Available: http://crt.io/DNS-Amplification-Attacks.pdf

[13] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "A Fair Solution to DNS Amplification Attacks," 2nd International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007), pp. 38–47, 2007.

[14] R. Shankesi, M. AlTurki, R. Sasse, C. A. Gunter, and J. Meseguer, "Model-checking DoS Amplification for VoIP Session Initiation," in *Computer Security–ESORICS*. Springer, 2009, pp. 390–405.

[15] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "DNS Amplification Attack Revisited," *Comput. Secur.*, vol. 39, pp. 475–485, Nov. 2013.

[16] X. Ye and Y. Ye, "A Practical Mechanism to Counteract DNS Amplification DDoS Attacks," *Journal of Computational Information Systems*, vol. 9, pp. 265–272, 2013.

[17] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *Proc. of the 23rd USENIX Security Symposium*. USENIX Assoc., 2014.

[18] T. Böttger, L. Braun, O. Gasser, F. von Eye, H. Reiser, and G. Carle, "DoS Amplification Attacks – Protocol-Agnostic Detection of Service Abuse in Amplifier Networks," in *7th*

International Workshop on Traffic Monitoring and Analysis (TMA), vol. 9053. Springer-Verlag, 2015, pp. 205–218.

[19] R. R. Hansen, "Slowloris," 2009. [Online]. Available: http://en.wikipedia.org/wiki/Slowloris_(software)

[20] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," RFC 4987 (Informational), Internet Engineering Task Force, August 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4987.txt

[21] D. Cornell, "DNS Amplification Attacks," March 2014. [Online]. Available: https://labs.opendns.com/2014/03/17/dns-amplification-attacks/

[22] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proc. of NDSS*. Internet Society, 2014.

[23] S. M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 2, pp. 32–48, Apr. 1989.

[24] M. Prince, "Technical Details Behind a 400Gbps NTP Amplification DDoS Attack," Feb 2014. [Online]. Available: http://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack

[25] "Open Resolver Project," April 2015. [Online]. Available: http://openresolverproject.org

[26] Y. Li, Q. Wang, F. Yang, and S. Su, "Traceback DRDoS Attacks," *Journal of Information & Computational Science*, vol. 8, pp. 94–111, 2011.

[27] R. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *Communications Magazine, IEEE*, vol. 40, no. 10, pp. 42–51, Oct 2002.

[28] D. Huistra, "Detecting Reflection Attacks in DNS Flows," in *19th Twente Student Conference on IT*, February 2013.

[29] R. Sparks, "Addressing an Amplification Vulnerability in Forking Proxies," 2006. [Online]. Available: https://tools.ietf.org/html/draft-ietf-sip-fork-loop-fix-00

[30] AusCERT, "Domain Name System (DNS) Denial of Service (DoS) Attacks," August 1999. [Online]. Available: http://www.securityfocus.com/advisories/1727

[31] P. J. Criscuolo, "Distributed denial of service - trin00, tribe flood network, tribe flood network 2000," Technical Report CIAC-2319, Department of Energy-CIAC (Computer Incident Advisory Capability), Tech. Rep., 2000.

[32] J. Damas, M. Graff, and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))," IETF, RFC 6891, April 2013.

[33] US-CERT, "UDP-based Amplification Attacks," 2014. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA14-017A

[34] Akamai, "SSDP Reflection DDOS Attacks," 2014. [Online]. Available: http://www.prolexic.com/kcresources/attack-report/attack_report_q214/Prolexic-Q22014-Global-Attack-Report-A4.pdf

[35] MIT, "2013-05-16 SNMP Amplification Attack," May 2013. [Online]. Available: http://tinyurl.com/MIT-drdos

[36] US-CERT, "NTP Amplification Attacks Using CVE-2013-5211," 2014. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA14-013A

[37] Akamai, "The State of the Internet [security]," September 2014. [Online]. Available: http://www.stateoftheinternet.com/resources-web-security-2014-q4-internet-security-report.html

[38] CERT Advisory, "Smurf IP Denial of Service Attacks," 1998. [Online]. Available: http://www.cert.org/advisories/CA-1998-01.html

[39] S. Kumar, "Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet," in *Proceedings of the Second International Conference on Internet Monitoring and Protection*. IEEE, 2007.

[40] A. Householder, A. Manion, L. Pesante, G. Weaver, and R. Thomas, "Managing the Threat of Denial-of-service Attacks," 2001. [Online]. Available: http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=52481

[41] T. Brewster, "Prolexic CEO: Biggest Cyber Attack Ever Was Built on Lies," April 2013. [Online]. Available: http://tinyurl.com/pr4j42d

[42] T. Rozekrans, M. Mekking, and J. de Koning, "Defending Against DNS Reflection Amplification Attacks," *University of Amsterdam, Tech. Rep.*, February 2013.

[43] V. Paxson, "An Analysis of Using Reflectors for Distributed Denial-of-service Attacks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, pp. 38–47, Jul. 2001.

[44] W. Chen and D.-Y. Yeung, "Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing," in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*. IEEE, April 2006.

[45] S. H. D. Nashat, Xiaohong Jiang, "Detecting SYN Flooding Agents under Any Type of IP Spoofing," in *IEEE International Conference on e-Business Engineering*. IEEE, October 2008, pp. 499–505.

[46] M. O. H. N. A. Noureldien, "Block Spoofed Packets at Source (BSPS): A Method for Detecting and Preventing all Types of Spoofed Source IP Packets and SYN Flooding Packets at Source: A Theoretical Framework," in *Second International Conference on the Applications of Digital Information and Web Technologies*. IEEE, Aug 2009, pp. 579–583.

[47] L. Kavisankar and C. Chellappan, "A Mitigation Model for TCP SYN Flooding with IP spoofing," in *International Conference on Recent Trends in Information Technology (ICRTIT)*. IEEE, June 2011, pp. 251–256.

[48] Y. Gilad and A. Herzberg, "LOT: A Defense Against IP Spoofing and Flooding Attacks," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 2, pp. 6:1–6:30, Jul. 2012.

[49] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection," in *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2009.

[50] B. Hancock, "Trinity v3, a DDoS Tool, Hits the Streets," *Computers & Security*, vol. 19, no. 7, p. 574, 2000.

[51] Bysin, "Knight," 2001. [Online]. Available: https://packetstormsecurity.com/distributed/knight.c

[52] J. Nazario, "BlackEnergy DDoS Bot Analysis," 2007. [Online]. Available: http://atlas-public.ec2.arbor.net/docs/BlackEnergy+DDoS+Bot+Analysis.pdf

[53] Praetox, "LOIC - A Network Stress Testing Application." [Online]. Available: http://sourceforge.net/projects/loic/

[54] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. ACM, 2009, pp. 1–12.

[55] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet Denial-of-service with Capabilities," *Computer Communication Review*, vol. 34, no. 1, pp. 39–44, 2004.

[56] X. Yang, D. Wetherall, and T. Anderson, "TVA: A DoS-limiting Network Architecture," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 6, pp. 1267–1280, 2008.

[57] J. Meseguer, “Rewriting Logic and Maude: A Wide-Spectrum Semantic Framework for Object-Based Distributed Systems,” in *Formal Methods for Open Object-Based Distributed Systems IV*. Springer US, 2000, vol. 49, pp. 89–117.

[58] Valve, “Steam Master Server Query Protocol,” 2015. [Online]. Available: https://developer.valvesoftware.com/wiki/Master_Server_Query_Protocol

[59] S. Kandula, D. Katabi, M. Jacob, and A. Berger, “Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds,” in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation – Volume 2*. USENIX Association, 2005, pp. 287–300.

[60] G. Oikonomou and J. Mirkovic, “Modeling human behavior for defense against flash-crowd attacks,” in *IEEE International Conference on Communications. ICC’09*. IEEE, 2009, pp. 1–6.

[61] R. Beverly and S. Bauer, “The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet,” *USENIX SRUTI: Steps to Reducing Unwanted Traffic on the Internet Workshop*, pp. 53–59, Jul. 2005.

[62] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring Internet denial-of-service activity,” *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006.

[63] J. Mauch, “SNMP DDoS: The Vulnerability You Might Not Know You Have,” http://seclists.org/nanog/2013/Aug/132, August 2013.

[64] L. T. Heberlein and M. Bishop, “Attack Class: Address Spoofing,” in *Proc. of the 19th National Information Systems Security Conference*, 1996, pp. 371–377.

[65] T. Dunigan, “Backtracking Spoofed Packets,” 2001. [Online]. Available: http://www.epm.ornl.gov/~dunigan/oci/back.ps

[66] L. Soon, M. Othman, and N. I. Udzir, “IP Spoofing Defense: An Introduction,” International Conference on Computing & Informatics (ICOCI09), 2009.

[67] T. Ehrenkranz and J. Li, “On the State of IP Spoofing Defense,” *ACM Trans. Internet Technol.*, vol. 9, no. 2, pp. 1–29, May 2009.

[68] A. Belenky and N. Ansari, “On ip traceback,” *Communications Magazine, IEEE*, vol. 41, no. 7, pp. 142–153, 2003.

[69] A. John and T. Sivakumar, “DDoS: Survey of Traceback Methods,” *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 241–245, May 2009.

[70] S. Vincent and J. I. J. Raja, “A Survey of IP Traceback Mechanisms to Overcome Denial-of-service Attacks,” Proceedings of the 12th International Conference on Networking, VLSI and Signal Processing, pp. 93–98, 2010.

[71] R. Jain and P. A. Meshram, “A Survey on Packet Marking and Logging,” *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 4, no. 3, pp. 426–429, 2013.

[72] S. J. Templeton and K. E. Levitt, “Detecting Spoofed Packets,” in *DARPA Information Survivability Conference and Exposition, 2003*, April 2003, pp. 164–175.

[73] H.-Y. Chang, R. Narayan, S. F. Wu, B. Vetter, X. Wang, M. Brown, J. Yuill, C. Sargor, Y. F. Jou, and F. Gong, “DecIdUouS: Decentralized Source Identification for Network-Based Intrusions,” in *Integrated Network Management*. IEEE, 1999, pp. 701–714.

[74] S. Savage, D. Wetherall, A. R. Karlin, and T. E. Anderson, “Practical Network Support for IP Traceback,” in *Proc. of ACM SIGCOMM*. ACM, 2000, pp. 295–306.

[75] D. Schnackenberg, K. Djahandari, and D. Sterne, “Infrastructure for Intrusion Detection and Response,” in *DARPA Information Survivability Conference and Exposition*, vol. 2, 2000, pp. 3–11.

[76] P. Ferguson and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” IETF, RFC 2827, May 2000.

[77] J. Mirkovic, N. Jevtic, and P. Reiher, “A Practical IP Spoofing Defense Through Route-Based Fltering,” University of Delaware, CIS department, Technical Report, CIS-TR, 2006.

[78] A. Bremler-barr and H. Levy, “Spoofing Prevention Method,” in *Proc. IEEE INFOCOM*, 2005, pp. 536–547.

[79] K. Park and H. Lee, “On the Effectiveness of Route-based Packet Filtering for Distributed DoS Attack Prevention in Power-law Internets,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 15–26, Aug. 2001.

[80] Z. D. X. Yuan and J. Chandrashekar, “Controlling IP Spoofing through Interdomain Packet Filters,” in *Dependable and Secure Computing, IEEE Transactions on*, vol. 5. IEEE, March 2008, pp. 22–36.

[81] T. Ohtsuka, F. Nakamura, Y. Sekiya, and Y. Wakahara, “Proposal and Efficient Implementation of Detecting and Filtering Method for IP Spoofed Packets,” in *International Conference on Information and Communication Technology*. IEEE, March 2007, pp. 327–330.

[82] J. Li, J. Mirkovic, T. Ehrenkranz, M. Wang, P. Reiher, and L. Zhang, “Learning the Valid Incoming Direction of IP Packets,” *Computer Networks*, vol. 52, no. 2, pp. 399–417, 2008.

[83] C. A. Shue, M. Gupta, and M. P. Davy, “Packet Forwarding with Source Verification,” *Comput. Netw.*, vol. 52, no. 8, pp. 1567–1582, Jun. 2008.

[84] H. Lee, M. Kwon, G. Hasker, and A. Perrig, “BASE: An Incrementally Deployable Mechanism for Viable IP Spoofing Prevention,” in *Proc. of the 2nd ACM ASIACCS*. ACM, 2007, pp. 20–31.

[85] A. Yaar, A. Perrig, and D. X. Song, “Pi: A Path Identification Mechanism to Defend against DDoS Attack,” in *Proc. of IEEE Symposium on Security and Privacy*. IEEE, 2003.

[86] Cisco Systems, “Unicast Reverse Path Forwarding Enhancements For The Internet Service Provider — Internet Service Provider Network Edge,” Cisco, White Paper, 2005.

[87] G.-F. Lv and Z.-G. Sun, “Towards Spoofing Prevention Based on Hierarchical Coordination Model,” in *Proc. of Workshop on High Performance Switching and Routing (HPSR’07)*. IEEE, June 2007, pp. 1–6.

[88] H. Wang, C. Jin, and K. G. Shin, “Defense Against Spoofed IP Traffic Using Hop-count Filtering,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 40–53, Feb. 2007.

[89] A. Yaar, A. Perrig, and D. Song, “StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1853–1863, 2006.

[90] Z. Gao and N. Ansari, “A Practical and Robust Inter-domain Marking Scheme for IP Traceback,” *Computer Networks*, vol. 51, no. 3, pp. 732–750, 2007.

[91] R. Chen, J.-M. Park, and R. Marchany, “A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 5, pp. 577–588, May 2007.

[92] J. Wu, G. Ren, and X. Li, “Source Address Validation: Architecture and Protocol Design,” in *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*. IEEE, 2007, pp. 276–283.

[93] D. Dean, M. K. Franklin, and A. Stubblefield, "An Algebraic Approach to IP traceback," *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 2, pp. 119–137, 2002.

[94] G. Taleck, "Ambiguity Resolution via Passive OS Fingerprinting," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer, 2003, vol. 2820, pp. 192–206.

[95] M. Adler, "Trade-offs in Probabilistic Packet Marking for IP Traceback," *J. ACM*, vol. 52, no. 2, pp. 217–244, Mar. 2005.

[96] Fyodor, "Remote OS Detection," 2006. [Online]. Available: http://nmap.org/book/osdetect.html

[97] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Standard), Internet Engineering Task Force, December 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4301.txt

[98] R. Beverly, "A Robust Classifier for Passive TCP/IP Fingerprinting," in *Proc. of 5th International Workshop on Passive and Active Network Measurement (PAM)*. Heidelberg Berlin: Springer, 2004, pp. 158–167.

[99] G. Taleck, "Ambiguity Resolution via Passive OS Fingerprinting," in *Proc. of RAID*. Springer, 2003, pp. 192–206.

[100] B. Berntein, "SYN Cookies," 1996. [Online]. Available: http://cr.yp.to/syncookies.html

[101] W. Feng, E. Kaiser, W. Feng, and A. Luu, "Design and implementation of network puzzles," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, March 2005, pp. 2372–2382 vol. 4.

[102] C. Jin, H. Wang, and K. G. Shin, "Hop-count Filtering: An Effective Defense Against Spoofed DDoS Traffic," in *Proc. of the 10th ACM CCS*. ACM, 2003, pp. 30–41.

[103] T. Killalea, "Recommended Internet Service Provider Security Services and Procedures," RFC 3013 (Best Current Practice), Internet Engineering Task Force, Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc3013.txt

[104] F. Baker, "Requirements for IP Version 4 Routers," RFC 1812 (Proposed Standard), Internet Engineering Task Force, June 1995, updated by RFC 2644. [Online]. Available: http://www.ietf.org/rfc/rfc1812.txt

[105] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," RFC 3704 (Best Current Practice), Internet Engineering Task Force, March 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3704.txt

[106] X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and Adoptable Source Authentication," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, 2008, pp. 365–378.

[107] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "SAVE: Source address validity enforcement protocol," in *IEEE INFOCOM*, 2002, pp. 1557–1566.

[108] Z. Duan, X. Yuan, and J. Chandrashekar, "Constructing Inter-Domain Packet Filters to Control IP Spoofing Based on BGP Updates," in *Proc. of IEEE INFOCOM*, 2006, pp. 1–12.

[109] T. Ehrenkranz and J. Li, "An Incrementally Deployable Protocol for Learning the Valid Icoming Direction of IP Packets," University of Oregon, Tech. Rep., October 2007.

[110] R. Stone, "Centertrack: An IP Overlay Network for Tracking DoS Floods," in *Proc. of the 9th conference on USENIX Security Symposium*. USENIX Association, 2000, pp. 15–15.

[111] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," in *Proc. of the 14th Conference on Systems Administration (LISA 2000)*. USENIX, December 2000, pp. 319–327.

[112] S. Bellovin and M. Leech, "ICMP Traceback Messages," February 2000. [Online]. Available: https://tools.ietf.org/html/draft-ietf-itrace-00

[113] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 226–237, Jun. 2001.

[114] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in *Proc. of IEEE INFOCOM*, vol. 2, 2001, pp. 878–886.

[115] A. Belenky and N. Ansari, "Accommodating Fragmentation in Deterministic Packet Marking for IP Traceback," in *Proceedings of IEEE Global Telecommunications Conference*, 2003, pp. 1374–1378.

[116] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-packet IP traceback." *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 721–734, 2002.

[117] Y. Xiang, W. Zhou, and M. Guo, "Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 4, pp. 567–580, April 2009.

[118] L. Zhang and Y. Guan, "TOPO: A Topology-aware Single Packet Attack Traceback Scheme," in *Securecomm and Workshops*. IEEE Press, September 2006, pp. 1–10.

[119] S. Matsuda, T. Baba, A. Hayakawa, and T. Nakamura, "Design and Implementation of Unauthorized Access Tracing System," in *Proceedings of the 2002 Symposium on Applications and the Internet*. IEEE Computer Society, 2002, pp. 74–81.

[120] H. A. Alwis, R. C. Doss, P. S. Hewage, and M. U. Chowdhury, "Topology Based Packet Marking for IP Traceback," in *Proc. of the Australian Telecommunication Networks and Applications Conference (ATNAC)*. University of Melbourne, 2006, pp. 224–228.

[121] C. Gong and K. Sarac, "A More Practical Approach for Single-Packet IP Traceback using Packet Logging and Marking ," pp. 1310–1324, 2008.

[122] T. W. Doeppner, P. N. Klein, and A. Koyfman, "Using Router Stamping to Identify the Source of IP Packets," in *Proc. of the 7th ACM CCS*. ACM, 2000, pp. 184–189.

[123] B. Al-Duwairi and M. Govindarasu, "Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 5, pp. 403–418, May 2006.

[124] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based IP Traceback," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 3–14, Aug. 2001.

[125] D. Yan, Y. Wang, S. Su, and F. Yang, "A Precise and Practical IP Traceback Technique Based on Packet Marking and Logging." *J. Inf. Sci. Eng.*, vol. 28, no. 3, pp. 453–470, 2012.

[126] A. Belenky and N. Ansari, "IP Traceback With Deterministic Packet Marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, April 2003.

[127] C. Gong and K. Sarac, "Toward a Practical Packet Marking Approach for IP Traceback," *International Journal of Network Security*, vol. 8, no. 3, pp. 271–281, 2009.

[128] M. T. Goodrich, "Probabilistic Packet Marking for Large-scale IP Traceback," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 15–24, Feb. 2008.

[129] X.-J. Wang and X.-Y. Wang, “Topology-assisted Deterministic Packet Marking for IP Traceback,” *The Journal of China Universities of Posts and Telecommunications*, vol. 17, pp. 116–121, April 2010.

[130] K. Park and H. Lee, “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack,” pp. 338–347, 2001.

[131] T. Baba and S. Matsuda, “Tracing Network Attacks to Their Sources,” *IEEE Internet Computing*, vol. 6, no. 2, pp. 20–26, Mar. 2002.

[132] A. Mankin, D. Massey, C.-L. Wu, S. F. Wu, and L. Zhang, “On Design and Evaluation of ”Intention-Driven” ICMP Traceback,” in *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*. IEEE, 2001, pp. 159–165.

[133] S. Lee and C. Shields, “Challenges to Automated Attack Traceback,” *IT Professional*, vol. 4, no. 3, pp. 12–18, May 2002.

[134] R. Beverly, “Spoofer project: State of ip spoofing,” November 2014. [Online]. Available: http://spoofer.cmand.org/summary.php

[135] J. Postel, “Internet Protocol,” RFC 791 (Internet Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1349, 2474. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[136] X. Lu, G. L, P. Zhu, and Y. Chen, “MASK: An Efficient Mechanism to Extend Inter-domain IP Spoofing Preventions,” pp. 1745–1760, November 2008.

[137] G. Velmayil and S. Pannirselvam, “Detection and Removal of IP Spoofing through Extended-Inter Domain Packet Filter Architecture,” *International Journal of Computer Applications*, vol. 49, no. 17, pp. 37–43, July 2012.

[138] F. Guo, J. Chen, and T. cker Chiueh, “Spoof Detection for Preventing DoS Attacks against DNS Servers,” in *Proc. of ICDCS*. IEEE, 2006.

[139] D. Kuptsov and A. Gurtov, “SAVAH: Source Address Validation with Host Identity Protocol,” in *Proc. of the First International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MobiSec’09*. Springer, 2009, pp. 190–201.

[140] A. Gurtov, *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. John Wiley & Sons, 2008, vol. 21.

[141] R. Moskowitz and P. Nikander, “Host Identity Protocol (HIP) Architecture,” RFC 4423 (Informational), Internet Engineering Task Force, may 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4423.txt

[142] H. Ishibashi, N. Yamai, K. Abe, and T. Matsuura, “A Protection Method Against Unauthorized Access and Address Spoofing for Open Network Access Systems,” in *IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, vol. 1. IEEE, 2001, pp. 10–13.

[143] J. M. Gonzalez, M. Anwar, and J. B. Joshi, “A Trust-based Approach Against IP-spoofing Attacks,” in *Ninth Annual Conference on Privacy, Security and Trust (PST 2011)*. IEEE, July 2011, pp. 63–70.

[144] M. O. H. Noureldien A. Noureldien, “Block Spoofed Packets at Source (BSPS): A Method for Detecting and Preventing All Types of Spoofed Source IP Packets and SYN Flooding Packets at Source: A Theoretical Framework,” *International Journal of Networks and Communications*, vol. 2, no. 3, pp. 33–37, 2012.

[145] T. Korkmaz, C. Gong, K. Sara, and S. G. Dykes, “Single Packet IP Traceback in AS-level Partial Deployment Scenario,” *IJSN*, pp. 95–108, 2007.

[146] M. Sung, J. Xu, J. Li, and L. Li, “Large-scale IP Traceback in High-speed Internet: Practical Techniques and Information-theoretic Foundation,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1253–1266, Dec. 2008.

[147] K. Egevang and P. Francis, “The IP Network Address Translator (NAT),” RFC 1631 (Informational), Internet Engineering Task Force, May 1994, obsoleted by RFC 3022. [Online]. Available: http://www.ietf.org/rfc/rfc1631.txt

[148] K. Ali, M. Zulkernine, and H. S. Hassanein, “Packet Filtering Based on Source Router Marking and Hop-Count,” in *LCN*. IEEE Computer Society, 2007, pp. 1061–1068.

[149] N. Arumugam and C. Venkatesh, “A Dynamic Method to Detect IP Spoofing on Data Network Using Ant Algorithm,” *IOSR Journal of Engineering (IOSRJEN)*, vol. 2, october 2012.

[150] P. Ferguson and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” RFC 2267 (Informational), Internet Engineering Task Force, January 1998, obsoleted by RFC 2827. [Online]. Available: http://www.ietf.org/rfc/rfc2267.txt

[151] K. Butler, T. Farley, P. Mcdaniel, and J. Rexford, “A survey of BGP security issues and solutions,” *AT&T Labs Research*, 2008.

[152] J. Israr, M. Guennoun, and H. T. Mouftah, “Mitigating IP Spoofing by Validating BGP Routes Updates,” *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 9, no. 5, pp. 71–76, May 2009.

[153] C. L. Abad and R. I. Bonilla, “An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks,” in *27th International Conference on Distributed Computing Systems Workshops*. IEEE, 2007, pp. 60–60.

[154] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, “AmpPot: Monitoring and Defending Amplification DDoS Attacks,” in *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions and Defenses*, November 2015.

[155] W. Strayer, C. Jones, F. Tchakountio, A. Snoeren, B. Schwartz, R. Clements, M. Condell, and C. Partridge, “Traceback of single IP packets using SPIE,” in *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, vol. 2, April 2003, pp. 266–270.

[156] R. Shokri, A. Varshovi, H. Mohammadi, and N. Yazdani, “DDPM: Dynamic Deterministic Packet Marking for IP Traceback,” in *14th IEEE International Conference on Networks*, vol. 2. IEEE, September 2006, pp. 1–6.

[157] Q. Dong, M. Adler, S. Banerjee, and K. Hirata, “Efficient Probabilistic Packet Marking,” in *13th IEEE ICNP*. IEEE, November 2005.

[158] B. Duwairi, A. Chakrabarti, and G. Manimaran, “An Efficient Probabilistic Packet Marking Scheme for IP Traceback,” pp. 1263–1269, November 2004.

[159] A. Belenky and N. Ansari, “On Deterministic Packet Marking,” *Comput. Netw.*, vol. 51, no. 10, pp. 2677–2700, Jul. 2007.

[160] Z. Chen and M.-C. Lee, “An IP Traceback Technique Against Denial-of-service Attacks,” in *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, Dec 2003, pp. 96–104.

[161] H. C. J. Lee, V. L. L. Thing, Y. Xu, and M. Ma, “ICMP Traceback with Cumulative Path, an Efficient Solution for IP Traceback,” in *Proc. of Information and Communications Security, 5th International Conference, ICICS*, vol. 2836. Springer, 2003, pp. 124–135.

[162] H.-W. Lee, M.-G. Kang, and C.-W. Choi, "PTrace: Pushback/SVM Based ICMP Traceback Mechanism against DDoS Attack," in *Computational Science and Its Applications - ICCSA 2004*, vol. 3043. Springer, 2004, pp. 302–309.

[163] B.-T. Wang and H. Schulzrinne, "An IP Traceback Mechanism for Reflective DoS Attacks," in *Canadian Conference on Electrical and Computer Engineering*, vol. 2, May 2004, pp. 901–904.

[164] R. Feiertag, C. Kahn, P. Porras, D. Schnackenberg, S. Staniford-Chen, and B. Tung, "A Common Intrusion Specification Language (CISL)," 1999.

[165] R. Atkinson, "Security Architecture for the Internet Protocol," IETF, RFC 1825, August 1995.

[166] S. Kent, "IP Authentication Header," IETF, RFC 4302, December 2005.

[167] ——, "IP Encapsulating Security Payload (ESP)," IETF, RFC 4303, December 2005.

[168] V. Santiraveewan and Y. Permpoontanalarp, "A Graph-based Methodology for Analyzing IP Spoofing Attack," in *Proc. of the 18th International Conference on Advanced Information Networking and Applications*, ser. AINA '04. IEEE Computer Society, 2004.

[169] W. Wei, F. Chen, Y. Xia, and G. Jin, "A Rank Correlation Based Detection against Distributed Reflection DoS Attacks," *Communications Letters, IEEE*, vol. 17, no. 1, pp. 173–175, January 2013.

[170] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "Detecting DNS Amplification Attacks," in *Workshop on Critical Information Infrastructures Security (CRITIS)*, vol. 5141. Springer, 2008, pp. 185–196.

[171] S. Di Paola and D. Lombardo, "Protecting Against DNS Reflection Attacks with Bloom Filters," in *Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. DIMVA'11. Springer-Verlag, 2011, pp. 1–16.

[172] H. Tsunoda, Y. Nemoto, K. Ohta, and A. Yamamoto, "A Simple Response Packet Confirmation Method for DRDoS Detection," in *Proc. of the 8th International Conference on Advanced Communication Technology (ICACT 2006)*, vol. 3. IEEE Press, February 2006.

[173] H. Tsunoda, K. Ohta, A. Yamamoto, N. Ansari, Y. Waizumi, and Y. Nemoto, "Detecting DRDoS Attacks by a Simple Response Packet Confirmation Mechanism," *Comput. Commun.*, vol. 31, no. 14, pp. 3299–3306, Sep. 2008.

[174] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from Distributed Denial of Service Attacks Using History-based IP Filtering," in *Communications, 2003. ICC'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 482–486.

[175] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," in *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on.* IEEE, 2002, pp. 312–321.

[176] ——, "Source-end DDoS Defense," in *Second IEEE International Symposium on Network Computing and Applications. NCA 2003.* IEEE, 2003, pp. 171–178.

[177] T. M. Gil and M. Poletto, "MULTOPS: A Data-structure for Bandwidth Attack Detection," in *USENIX Security Symposium*, 2001.

[178] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami, "An Efficient Filter for Denial-of-Service Bandwidth Attacks," in *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, vol. 3. IEEE, 2003, pp. 1353–1357.

[179] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting Reflector Attacks by Sharing Beliefs," in *Proc. of IEEE GLOBECOM*, 2003, pp. 1358–1362.

[180] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: A Statistics-Based Packet Filtering Scheme Against Distributed Denial-of-Service Attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 3, no. 2, pp. 141–155, 2006.

[181] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling High Bandwidth Aggregates in the Network," *Computer Communication Review*, vol. 32, no. 3, pp. 62–73, 2002.

[182] D. K. Yau, J. Lui, F. Liang, and Y. Yam, "Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router Throttles," *Transactions on Networking (TON)*, vol. 13, no. 1, pp. 29–42, 2005.

[183] R. Chen and J.-M. Park, "Attack Diagnosis: Throttling Distributed Denial-of-Service Attacks Close to the Attack Sources," in *Proceedings. 14th International Conference on Computer Communications and Networks.* IEEE, 2005, pp. 275–280.

[184] R. Chen, J.-M. Park, and R. Marchany, "TRACK: A Novel Approach for Defending Against Distributed Denial-of-Service Attacks," *Technical Report TR ECEO6-02. Dept. of Electrical and Computer Engineering, Virginia Tech*, 2006.

[185] N. Fultz and J. Grossklags, "Blue Versus Red: Towards a Model of Distributed Security Attacks," in *Financial Cryptography and Data Security.* Springer, 2009, pp. 167–183.

[186] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Towards a Forecasting Model for Distributed Denial of Service Activities," in *12th IEEE International Symposium on Network Computing and Applications (NCA)*, August 2013, pp. 110–117.

[187] ——, "Fingerprinting Internet DNS Amplification DDoS Activities," in *6th International Conference on New Technologies, Mobility and Security (NTMS)*, March 2014, pp. 1–5.

[188] ——, "Inferring Distributed Reflection Denial of Service Attacks from Darknet," *Computer Communications*, vol. 62, pp. 59 – 71, 2015.

[189] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 1073–1080, 2012.

[190] J. Mirkovic, M. Robinson, and P. Reiher, "Alliance Formation for DDoS Defense," in *Proceedings of the 2003 workshop on New security paradigms.* ACM, 2003, pp. 11–18.

[191] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks," in *Proceedings of the 2014 Conference on Internet Measurement Conference.* ACM, 2014, pp. 435–448.

[192] The Shadowserver Foundation, April 2015. [Online]. Available: https://dnsscan.shadowserver.org/

[193] The Measurement Factory, April 2015. [Online]. Available: http://dns.measurement-factory.com

[194] Network Time Foundation, April 2015. [Online]. Available: http://openntpproject.org/