Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks

Johannes Krupp
CISPA, Saarland University
Saarland Informatics Campus

Michael Backes
CISPA, Saarland University &
MPI-SWS
Saarland Informatics Campus

Christian Rossow
CISPA, Saarland University
Saarland Informatics Campus

ABSTRACT

Amplification DDoS attacks have gained popularity and become a serious threat to Internet participants. However, little is known about where these attacks originate, and revealing the attack sources is a non-trivial problem due to the spoofed nature of the traffic. In this paper, we present novel techniques to uncover the infrastructures behind amplification DDoS attacks. We follow a two-step approach to tackle this challenge: First, we develop a methodology to impose a fingerprint on scanners that perform the reconnaissance for amplification attacks that allows us to link subsequent attacks back to the scanner. Our methodology attributes over 58% of attacks to a scanner with a confidence of over 99.9%. Second, we use Time-to-Live-based trilateration techniques to map scanners to the actual infrastructures launching the attacks. Using this technique, we identify 34 networks as being the source for amplification attacks at 98% certainty.

1. INTRODUCTION

Amplification attacks [26] have become one of the most popular and dangerous classes of distributed denial-of-service (DDoS) attacks nowadays. By spoofing the source IP address of requests sent to open Internet services (such as DNS or NTP servers), attackers can amplify traffic and disguise their identity at the same time. Incidents in the recent past have demonstrated that amplification attacks can cause attack bandwidths in the range of several hundreds of Gbit/s [25, 32]. Amplification attacks are not only a problem in terms of bandwidth, but also in terms of frequency and global scale: During a five-month measurement period in 2015, Kührer et al. monitored over 1.5 million such attacks (10k per day) that targeted victims in 192 countries.

Unfortunately, due to the IP-spoofing nature of amplification attacks, the true origin of the attacks remains hidden. Consequently, victims do not know whom to contact to stop the attacks, nor can they file legal complaints against attack originators. Even worse, from the victim's perspective, the third-party reflectors may appear to be the attack origin, giving false attribution hints. Despite the need for effective mechanisms to trace back the origin of amplification attacks, we still lack usable mechanisms. While there are attempts to identify spoofing-enabled networks [4, 21], the coverage of such active probes is limited, and without further evidence of abuses, the identified parties feel little social pressure to ban spoofing from their networks (e.g., using BCP38 [24]).

In this work, we tackle this problem and aim to attribute amplification attacks back to the infrastructures that have caused them. Whereas application-layer DoS attacks can be attributed to the origin due to the nature of the TCP handshake, finding the source of amplification attacks is inherently more difficult, given that attackers (i) spoof the IP addresses and (ii) use reflectors to diversify traffic sources. IP traceback and similar packet marking schemes have long been the *de facto* standard proposal to detect the origin of spoofed traffic [31, 11, 36, 41, 15, 12, 14], but none of these designs were ever deployed at scale to allow for global attack attribution. Still, decades after spoofing attacks were discovered, tracking the origin of attack traffic remains an adhoc process that requires coordination between many ISPs scattered around the world. The outcome of such tedious manual attribution processes (if any) comes hours or days after the traffic originated.

We follow a two-step process to establish an attribution process that identifies the infrastructures operated by attackers to *prepare* and *launch* amplification attacks. In a first step, we aim to link the *reconnaissance* and the *attack* phases by tracking which scan for amplifiers has resulted in which attacks. We leverage the fact that scans cannot forge their source IP address and thus learn about the scanning infrastructures, despite the fact that attack traffic is spoofed. Our key idea is to offer each scanner a different set of potential honeypots that it can abuse. This way, we implicitly encode a secret identifier to the set of honeypots that any subsequent attack will use, which varies per scan source. In a second step, we test if the scan infrastructure is also used to actually *launch* (and not just to prepare) the attacks. We follow the observation that similar traffic sources should have similar "distances" (in terms of hops) to globally-distributed sensors. Using trilateration, we can link scanners to attack origins based on hop counts.

With these proposals, we provide a practically-usable attribution methodology for amplification attacks. Our framework fulfills important goals: (i) Our method can work in real-time; that is, we can attribute attacks on the fly without noticable delay between attack start and attribution outcome. (ii) The attribution does not require any coopera-

CCS'16, October 24 - 28, 2016, Vienna, Austria

ISBN 978-1-4503-4139-4/16/10. . . $15.00

DOI: http://dx.doi.org/10.1145/2976749.2978293

tion between ISPs, and thus solves one of the main practical problems of existing solutions like IP traceback. (iii) Our method gives probabilistic guarantees that show if—and at what confidence level—the attribution outcome is correct.

We have deployed our attribution methodology on a snapshot of 1,351,852 amplification attacks monitored by honeypots during 23 weeks in 2015 and 2016. Our findings show that we can identify the scanners that were used during the reconnaissance phase of 58% of all attacks in our data set. Further analyses show that only 20 scanners are responsible for nearly 50% of the attacks. Using our hop-based trilateration process, we reveal that 22% of the attacks were actually launched from scan infrastructures, for which we have perfect IP, network and geographical attribution information. We report on the distribution of attack sources and reveal black sheep networks that cause massive spoofing attacks.

To summarize, our contributions are as follows:

- We present a novel honeypot-based technique, *selective response*, that enables us to assign a fingerprint to scanners during the reconnaissance for amplification DDoS attacks and give confidence guarantees for subsequent attribution.
- We evaluate our technique on a set of 1,531,852 attacks recorded by our honeypot, of which we can link 785,285 back to their corresponding scanner with a confidence of 99.9% or higher.
- We leverage the TTL field of the IPv4 header to compare the location of scanners to origins of attacks, after evaluating our methodology on data collected by RIPE Atlas probes.
- We find that for 22% of all attacks, the scanner linked to the attacks is also the source of the attack with 95% confidence.

The remainder of this paper is structured as follows. In Section 2, we define our threat model, discuss the ethical implications of our work, and describe the data used in this paper in Section 3. Section 4 introduces a novel honeypot-based technique to assign identifiers to systems that scan for amplifiers. We evaluate this technique in Section 5. In Section 6, we measure if the infrastructure used to scan for amplifiers is identical to the infrastructure used to launch amplification DDoS attacks. After reviewing our initial assumptions in Section 7 and providing an overview of related work in Section 8, we conclude with Section 9.

2. BACKGROUND

In this section, we will define the threat model this paper considers and discuss the ethical implications of our work.

2.1 Threat Model

The focus of this paper is on amplification DDoS attacks. Before defining our threat model, we first give a short description of this type of attack. The goal of an amplification attack is to render a system or network unusable by flooding the target's network with a huge amount of traffic, eventually leading to network congestion. To this end, an attacker can leverage amplification vectors in various network protocols by which Internet-facing servers (such as DNS or NTP) will send many packets towards the target. Our threat model is therefore comprised of at least three parties: The *attacker*, the *victim*, and a set of *amplifiers*.

In an attack, the *attacker* will send requests carrying a spoofed IP header to innocent servers (*amplifiers*). These amplifiers will then (unknowingly) direct their responses towards the *victim*, given that the victim's IP is specified as the request source in the spoofed header. Thereby, an amplifier will be acting as a "reflector", effectively hiding the attacker's IP address from the victim. Due to various amplification vectors in the service implementations, the size of the responses will be multiple times larger than the initial request sent by the attacker. This leads to a bandwith *amplification*, as the incoming bandwidth of traffic at the victim's system will be much higher than the one sent by the attacker.

Several protocols have been identified to be amplification-prone [26], with amplification factors (ratio between response and request size) ranging from 5 to 4000, and misconfigured systems and support for legacy options lead to a plethora of potential amplifiers. Finding these amplifiers is a vital step in attack reconnaissance, and is typically performed by scanning, i.e., sending requests to every address in a given range, and recording the answers. Therefore, we include this additional fourth party of a *scanner* to our threat model.

While attackers could potentially leverage botnets to launch amplification attacks, previous work documented that the vast majority of amplification attacks stem from a single origin [20], which also coincides with our findings. In the following, we will thus assume that attackers use only a single system to launch their attacks.

We will further assume that scanners do not spoof their source addresses when performing a scan. While techniques to perform scans using spoofed addresses are known for TCP (e.g. "idle scan" [27]), no similar techniques are known for UDP. Since all known amplification DDoS attacks are UDP-based, this is a valid assumption.

2.2 Ethical Considerations

The data sets used in this paper were collected leveraging AmpPot [20], a honeypot for DDoS amplification, which works by emulating a server for vulnerable protocols and thereby becoming one of the amplifiers used in attacks. Deployment of such a honeypot pose a challenge from an ethical point of view: By design, an amplification honeypot will also act as an amplifier in an actual attack and thus send unwanted traffic towards DDoS victims.

We argue that the contribution towards attack traffic by our honeypot is negligible and only incurs minimal harm to the victim's system. We did not modify the thresholds chosen by the authors of AmpPot, by which the honeypot will answer at most three requests per attack. Although we deployed our honeypot to listen on 48 IPs, as discussed later, at most 24 of those IPs will send replies towards a victim's system. Therefore, our honeypot will reply to at most 72 packets in total, i.e., a few kilobytes at most. Taking into account that these attacks usually flood a victim's system with traffic in the order of several Gbit/s, we conclude that the contribution of our honeypot is negligible.

In addition, we offered attack victims a method to opt out from our measurements. During the course of our experiments, we received three complaints that we immediately answered describing our experimental setup, but none of the complainers asked to opt out. We refer the interested reader to a more detailed ethical discussion in [20].

Finally, please note that our (non-US) legislation and university system does not require nor offer IRB approvals, and hence, we also could not request such an approval.

3. DATASET

Our data was collected using AmpPot [20] by Krämer et al., a honeypot for DDoS amplfication attacks. AmpPot emulates a server offering seven UDP-based protocols which are known to be abused, namely `QOTD`, `CharGen`, `DNS`, `NTP`, `RIPv1`, `MSSQL`, and `SSDP`. For incoming packets, AmpPot will record all header fields as well as some protocol-specific information from the packet's payload. Due to the vast amount of traffic in a DDoS attack, a sampling approach is employed: Once a source exceeds 100 packets within one hour, packets from this source will only be recorded with a probability of $1/100$. In order not to contribute to DDoS amplification attacks and to keep the harm on DDoS victims minimal, AmpPot will stop sending responses after the third packet for sources exceeding 1 packet per minute. We use the same conservative definition of an *attack* as Krämer et al., who define an attack as a stream of at least 100 consecutive packets from the same source to the same port without gaps longer than one hour. Further details may be found in [20]. We leveraged AmpPot in two ways:

First, to attribute attacks to scanners, we extended AmpPot in that it only *selectively* replies to requests. The basic idea behind this is that every scanner will see a different set of honeypots, which will become a distinctive feature for attribution. We provide an in-depth description of this technique in Section 4. We deployed our modified AmpPot version on Nov. 25th, 2015.

Second, the authors of AmpPot granted us access to data they collected from 11 honeypots, which were deployed in late 2014 and have been operated since then. Combining their data sets with ours allowed us to examine whether scans and attacks were launched from the same infrastructure by comparing TTL values (cf. Section 6).

We base our results on data collected between November 25th 2015 and May 1st 2015 (exclusive). Within this time our modified AmpPot version observed 1,351,852 attacks, 1,254,102 of which were also recorded by the secondary data sets contributed by the AmpPot authors.

4. SELECTIVE RESPONSE

In this section, we describe the idea that our honeypots selectively respond depending on the scanner origin—a fundamental technique that allows for attack attribution.

4.1 Intuition

Launching amplification attacks requires prior knowledge of a set of servers that can be abused as amplifiers during the attack. Finding such servers is commonly achieved through scanning, i.e., sending a query to every IP in a certain range, and recording which IPs send back a reply. Since nowadays scanning the entire IPv4 address space is feasible in a reasonable amount of time even from a single machine [13, 16], we assumed that in most cases the chosen amplifier set was based on the scan result(s) from a single scan system. Note that we will verify this assumption in later analyses.

The main goal of this work is to correlate scan events with amplification attacks. We therefore follow the idea that every scanner will find a different (ideally unique) subset of our deployed honeypots. We influence the scan result in such a way that we can re-identify the scanner once its scan result (the set of amplifiers) is used in subsequent amplification attacks. Our approach ensures that within a network segment under our control, every scanner finds a *different* set of potential amplifiers. That is, we launch honeypots on all IP addresses on that segment, but only selectively respond to a scanner-derived and therefore unique subset of IP addresses. If our assumption on single-source scans was correct, this would mean that attacks based on different scans would also use different amplifier sets.

4.2 Implementation

Technically, we implemented the selective response scheme as follows. We fix a fraction α of the network that responds to a scan, so that every scanner that performs a full scan on the network of size N would see replies from $\alpha \cdot N$ hosts. In order to maximize the number of possible combinations $\binom{N}{\alpha N}$, we set $\alpha = 1/2$, i.e., respond with exactly half of the honeypots, and remain "quiet" with the other half.

To select the αN IP addresses from the network, we compute a hash over the source IP address (i.e., an identifier for the scanner), the protocol, the base of the network, and a secret key (string). We added the secret key such that an attacker cannot precompute the set of responding honeypots based on our (otherwise deterministic) hash function. The resulting hash is then used to derive a permutation of the N IP addresses in the network. From this permutation, the first αN addresses are selected as responding honeypots.

Our selective response honeypot uses three `/28` networks, which gives a total of 48 static IPs distributed over three networks with 16 IP addresses each. Due to the split over three distinct `/28` networks, we decided to perform the selection *per subnet*. Separating the networks into three independent ranges has the advantage that we can even attribute scanners that scanned only one of the networks. Although this separation reduces the number of possible combinations from $\binom{48}{24} \approx 3.2 \times 10^{13}$ to $\binom{16}{8}^3 \approx 2.1 \times 10^{12}$, it is still two orders of magnitude larger than the number of IPv4 addresses, i.e., it is very likely that every scanner will be assigned a unique set of 24 amplifiers.

5. ATTRIBUTING ATTACKS TO SCANS

5.1 Methodology

The basic idea of our attribution is simple yet effective. For every attack we monitor, we inspect the set of honeypots that were abused for this attack. Typically, attackers leverage multiple amplifiers at the same time, and often also multiple of our honeypots are abused for the same attack. Figure 1 shows the cumulated percentage of attacks (y-axis) that have an attack set of at least a given size (x-axis). Over 95% of all attacks use at least four honeypots; 80% use at least 10 of our honeypots simultaneously.

Remember that the amplifier set greatly depends on the scan prior to the attack, for which our selective response scheme has introduced artificial entropy. From the set of honeypots abused in an attack, we therefore aim to derive which scanner has discovered these very same honeypots.

Technically, for every IP of our honeypot, we maintain a set of all sources that discovered this IP as an amplifier, i.e., sent a packet to this IP *and* got back a response. We denote those sources as being *aware* of the corresponding IP. Since we cannot perfectly distinguish attacks from scans, we will consider *every* source that contacted our honeypot, which explicitly includes victims of attacks.

Upon attributing an attack, we first extract the set of IPs used as amplifiers in this attack. Conjecturing that the

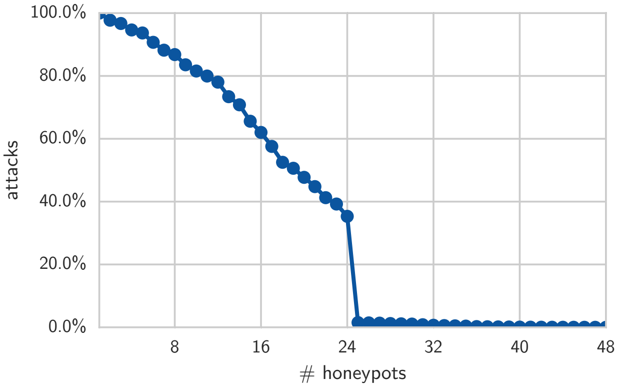

Figure 1: Percentages of attacks (y-axis) that use at least the given number of honeypots (x-axis)

honeypot IP	aware sources
10.0.0.1	{169.254.0.10, 192.168.2.100, 198.18.3.24}
10.0.0.2	{169.254.0.10, 172.16.5.27, 198.18.3.24}
10.0.0.3	{192.168.2.100, 172.16.5.27, 198.18.3.24}
10.0.0.4	{169.254.0.10, 172.16.5.27, 192.168.2.100}

Table 1: Example honeypots and *aware* scanners

scanner behind this attack must have scanned all of these IPs and received a reply, it must be contained in the set of aware sources for each of them. We can therefore find the scanner by building the intersection of these sets.

Since neither maintaining the list of aware sources nor computing a set intersection is computationally expensive, our methodology can also be applied in real-time, i.e., once an attack is detected, the result of the attribution can be obtained without any noticeable delay.

Consider the toy example in Table 1 that lists the set of aware sources for 4 honeypots. Assume that we observe an attack using honeypot IPs 10.0.0.1, 10.0.0.3, and 10.0.0.4. We can then find the potential scanner in the following way: As 10.0.0.1 is contained in the attack set, the scanner should be one of {169.254.0.10, 192.168.2.100, 198.18.3.24}. Since 10.0.0.3 is contained as well, we can narrow this down to {192.168.2.100, 198.18.3.24}, because 169.254.0.10 is *not* aware of 10.0.0.3. We can likewise exclude 198.18.3.24, as it is not aware of 10.0.0.4. This leaves only {192.168.2.100} as a potential scanner behind the attack.

Mapping scanners this way can result in three cases:

1. **Zero candidates** If the set of candidates is empty, then no single scanner was aware of this set of amplifiers. We will call such attacks *non-attributable*. This can occur for multiple reasons:

   Firstly, it could be that the attack is based on data from multiple scans, in which case the combined amplifier set is likely distinct from the sets found by other scanners. This is especially true for attacks that use more than $\alpha N = 8$ IP addresses per honeypot subnet, as a single scanner can only find up to eight amplifiers in each `/28` network.

   Secondly, due to the threshold of 100 packets that determines an attack, a scan can also be mistaken for an attack. If a scanner scans all of our 48 IPs, it can easily exceed the threshold by sending a little over 2 packets per IP. This happens, e.g., for scanners that start with a full scan (i.e., a single packet per IP address) and then verify each responding IP address by sending additional packets.

   Thirdly, because AmpPot considers an attack to have ended only if the packet-rate drops below 100/hour for one hour, two attacks targeting the same source in quick succession can be aggregated into a single attack.

   Although we could neither observe nor refute the first scenario, we have observed both the second and third scenarios in our data.

2. **Exactly one candidate.** If the set of candidates contains exactly one candidate, only a single scanner is aware of this set of amplifiers. We will consider this as a potential attribution. However, since we cannot exclude that the set of amplifiers was chosen by other means (e.g., combining data from multiple scans), we compute a *confidence* for this attribution. The computed confidence gives an indication of how likely it is that this attribution is correct. We give a detailed explanation of the confidence in Section 5.2.

3. **More than one candidate.** If the set of candidates contains multiple candidates, multiple scanners are aware of this set of amplifiers. We will call such attacks *non-unique*. This case occurs if the set of chosen amplifiers is relatively small and multiple scanners got responses from those IPs during their scans.

   In this case, we will refine the candidate set by finding scanner-to-victim relations in the set of candidates. That is, if both A and B are contained in the set of candidates, but have previously observed an attack against B which we could attribute to A, we will remove B from the set of candidates. This is based on the assumption that victims of DDoS attacks will most likely not act as scanners for DDoS attacks themselves, and even if they did, the set of amplifiers found would still be based on A's scan.

5.2 Confidence

Even if an attack can uniquely be attributed to an attack (case 2), it is unclear how confident this mapping is. In an ideal world, a scanner would scan all of our 48 IPs and subsequent attacks would use the full reply set of 24 IPs. Since the full reply set is unique per scanner, this would allow for a perfect attribution. However, in practice, several things impede this ideal-world assumption. For example, scanners might not query all 48 IP addresses. Even if they did, attackers could select a random subset of the found IP addresses to use in attacks. Worse, attackers might not base their attacks on the results of only a single scanner, but rather combine scan results from multiple sources. This raises the question whether our attribution is actually robust under such real-world conditions.

Our approach to answer this question is to define a confidence that expresses how likely is that our attribution result is actually correct, based on the following two sets. We will call the set of IP addresses that were queried by a scanner the *query set* Q, and refer to the set of IP addresses that replied as the *reply set* R. Since the reply set is determined using a hash function, which we assume to generate uniformly distributed values, we can consider the distribution of reply sets to be uniform as well.

We then analyze the probability with which we would falsely accuse a scanner of being responsible for an attack. The intuition behind this is as follows: Assuming that a given attack was *not* based on the reply set of a single scanner, what is the probability that any of the scanners still matches this attack by chance? That is, what is the probability we falsely accuse a scanner? If this probability is sufficiently small, we can conclude that—if we can attribute this attack to a scanner—this attribution is correct.

Formally, assume an attack that uses the IPs $A = A_1 \cup A_2 \cup A_3$, where A_i is the set of IPs from the ith subnet. We are now interested in the probability that a scanner that scanned a superset of A also gets replies from all IPs in A, i.e., $\Pr[A \subseteq R \mid A \subseteq Q]$. In each /28 subnet, the reply set the scanner observes is a subset of one out of $\binom{16}{8}$ sets. Out of these, $\binom{16-|A_i|}{8-|A_i|}$ are supersets of A_i (since the A_i IPs from the attack are fixed, a scanner could potentially receive responses from $8-|A_i|$ out of the remaining $16-|A_i|$). Thus, assuming a uniform distribution of reply sets, it holds that

$$\Pr[A_i \subseteq R \mid A \subseteq Q] = \frac{\binom{16-|A_i|}{8-|A_i|}}{\binom{16}{8}}$$

Therefore, the total probability that a scanner that scanned a superset of A also got replies from all IPs in A is

$$p = \Pr[A \subseteq R \mid A \subseteq Q] = \frac{\binom{16-|A_1|}{8-|A_1|}}{\binom{16}{8}} \cdot \frac{\binom{16-|A_2|}{8-|A_2|}}{\binom{16}{8}} \cdot \frac{\binom{16-|A_3|}{8-|A_3|}}{\binom{16}{8}}$$

From this individual probability for a single scanner we can now derive a probability for any scanner in our dataset. The probability that *any* of the S scanners that scanned a superset of A got replies from all IPs in A follows the "at-least-once" semantics and is

$$1 - (1-p)^S$$

Put differently, if we can attribute the attack to a scanner, our confidence that this attribution is correct is

$$(1-p)^S$$

For example, assume an attack that uses 5 IPs from the first, 4 IPs from the second, and 6 IPs from the third subnet respectively, i.e., $|A_1| = 5$, $|A_2| = 4$, $|A_3| = 6$. A single scanner then has probability

$$p = \frac{\binom{16-5}{8-5}}{\binom{16}{8}} \cdot \frac{\binom{16-4}{8-4}}{\binom{16}{8}} \cdot \frac{\binom{16-6}{8-6}}{\binom{16}{8}} = \frac{165 \cdot 495 \cdot 45}{12,870^3}$$
$$= \frac{3,675,375}{2,131,746,903,000} \approx 0.0001742\%$$

of receiving responses from this precise attack set during its scan. If at the time of the attack we had had contact with 200 scanners that scanned our entire network, the probability that *any* of them found this precise attack set is thus

$$1 - (1-p)^S = 1 - \left(1 - \frac{165 \cdot 495 \cdot 45}{12,870^3}\right)^{200}$$
$$\approx 0.03448\%.$$

Consequently, if we find a scanner that matches this attack, in 99.966% of all cases this does *not* happen by chance, and hence for such an attack we have a confidence of 99.966% that our attribution is correct.

Obviously, a larger set A will lead to a smaller probability, implying a higher confidence.

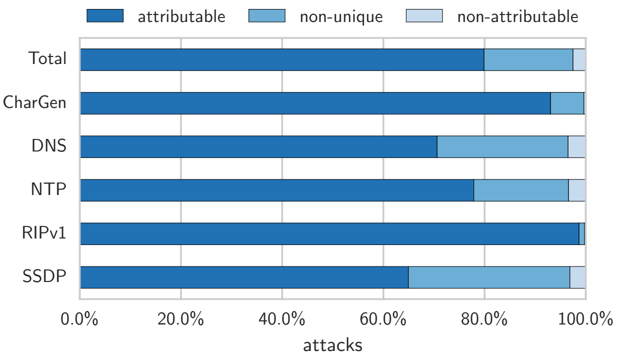

Figure 2: Attribution results per protocol

5.3 Experimental Results

We will now turn to the results of our attribution process using the dataset described in Section 3.

Figure 2 shows the percentages of attacks that were marked as *attributable*, *non-unique*, and *non-attributable*, respectively. Percentages are given both overall and per protocol. The absolute numbers for each category are given in Table 2, as well as attribution results for different levels of confidence. Since the QOTD and MSSQL protocols only account for a negligible number of attacks (1368, ≈ 0.1%), we will omit these protocols in the following.

5.3.1 Attributable Attacks

Most notably, out of the 1,351,852 attacks that we recorded at our honeypot, 785,285 (58.09%) could be attributed to a single scanner with a confidence of 99.9% or higher. This means that the chance that the attack was *not* based on the attributed scanner is less than 1 in 1,000. In fact, 643,956 attacks (47.64%) even have a confidence of 99.999% or higher, i.e., the chance of a false attribution is less than 1 in 100,000.

Surprisingly, our results are not homogeneous among different protocols. This can be seen in Figure 4, which depicts the fraction of attacks that could be attributed for various levels of confidence. While 74.70% of all CharGen attacks could be attributed with a confidence of 99.9% or higher, this holds for only 10.20% of the SSDP-based attacks. This discrepancy stems from the fact that the number of honeypot IPs used strongly varies between protocols, as can be seen in Figure 3, which shows the distribution of honeypot IPs used for all protocols. SSDP attacks only use 9.38 IPs on average, whereas CharGen attacks use 20.66. Consequently, SSDP also experiences a higher percentage of attacks marked as non-unique. These discrepancies can be explained by the global number of amplifiers available on the internet. For example, Rossow found 3,704,000 servers vulnerable to be used as SSDP amplifiers, in contrast to only 89,000 for CharGen [26]. In other words, our honeypots are less likely to be abused as amplifiers for SSDP-based attacks due to the abundance of available alternative SSDP amplifiers.

5.3.2 Non-Attributable Attacks

Only 34,058 attacks (2.52%) were considered to be *non-*

	QOTD	CharGen	DNS	NTP	RIPv1	MSSQL	SSDP	Total	
non-attributable	0	1 440	11 428	18 665	40	25	2 460	34 058	(2.52%)
non-unique	155	24 982	84 491	102 635	191	272	25 046	237 772	(17.59%)
attributable	53	353 300	230 626	426 962	17 253	863	50 965	1 080 022	(79.89%)
conf. $> 99\%$	53	294 913	208 696	342 852	15 163	784	13 989	876 450	(64.83%)
conf. $> 99.9\%$	53	283 665	201 555	279 467	11 928	610	8 007	785 285	(58.09%)
conf. $> 99.99\%$	53	280 514	179 033	233 914	10 395	536	6 260	710 705	(52.57%)
conf. $> 99.999\%$	52	274 617	140 055	214 329	8 441	535	5 927	643 956	(47.64%)
Sum	208	379 722	326 545	548 262	17 484	1 160	78 471	1 351 852	(100.00%)

Table 2: Attribution results and confidence breakdown

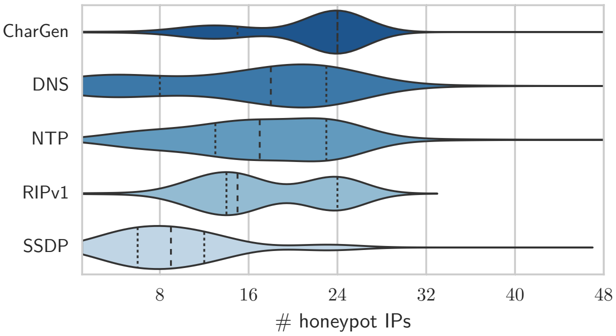

Figure 3: Number of abused honeypots per protocol

attributable. This indicates that our initial assumption, i.e., that most attackers use the result of only a single scanner, is true, as otherwise we would expect to see a much higher number of attacks without a matching scanner. However, of these few non-attributable attacks, more than 60% use more than 24 IPs, the maximum number of amplifiers a single scanner could have possibly found.

For these attacks that use more than 24 IPs, we can further analyze whether they are aggressive scans that exceeded the conservative threshold and are therefore counted as attacks, or whether they are based on the result of multiple scanners. Towards this goal we have to answer the following question: What is the probability of finding x distinct IPs when combining the results of y scans? Intuitively, while it is possible to receive answers from all 48 honeypots with just two scans, it is very unlikely, as this would mean that the second scanner received answers from *exactly* those 24 honeypots that did not answer the first scanner. That is, the likelihood for finding a larger number of x IPs for multiple scanners y increases with y and decreases with x. Formally, this can be modeled as an instance of the collector's problem with group drawings [38]. In our case, we find that for 80% of the attacks using more than 24 IPs, the corresponding attack sets can be found with a probability of over 60% by combining the results of only two scans. This explains the non-attributable cases in our data set. Rather than being aggressive scans, we conclude that most non-attributable attacks have combined data of multiple scanners.

5.3.3 *Non-Unique Attacks*

Finally, for 237,772 attacks (17.59%), our method found more than one potential scanner, i.e., multiple scanners got replies from the corresponding amplifier set, labeling those attacks as *non-unique*. Intuitively, this can only happen for attacks that use a relatively small amplifier set. Indeed, the average amplifier set size of non-unique attacks is 6.25, i.e., about a fourth of the full response set a scanner could find. In addition, more than 12.88% of the non-unique attacks have abused a single honeypot only.

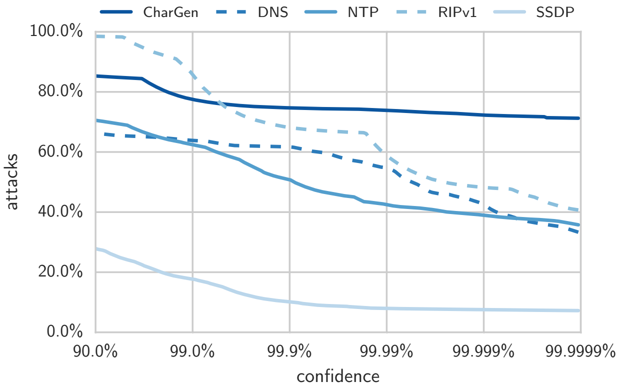

Figure 4: Percentage of attacks that could be attributed (y-axis) vs. level of confidence (x-axis)

5.4 Improving Attribution Confidence

While we attributed a substantial fraction of all attacks to their scanners with reasonable confidence, for roughly 40% we either found multiple potential scanners or could only attribute the attack to a scan with low confidence. A question that naturally arises is whether this is a inherent limitation of our methodology, or whether it can be alleviated by choosing different parameters, e.g., adjusting the response ratio or leveraging a larger network segment.

To this end, we analyze the influence of the network size and response ratio on the probability that the response set of a scanner is a superset of the attack set. Let N be the size of the network, α the response ratio, and A the attack set. Similar to Section 5.2, the probability that a scanner received replies from all IPs in the attack set is

$$p = \Pr\left[A \subseteq R \mid A \subseteq Q\right] = \frac{\binom{N-|A|}{\alpha N-|A|}}{\binom{N}{\alpha N}}.$$

Since the confidence is computed as $(1-p)^S$, in order to improve the attribution confidence, this probability should be as low as possible.

Interestingly, increasing the network size alone does not reduce this probability:

$$\lim_{N\to\infty} \frac{\binom{N-|A|}{\alpha N-|A|}}{\binom{N}{\alpha N}} = \alpha^{|A|}.$$

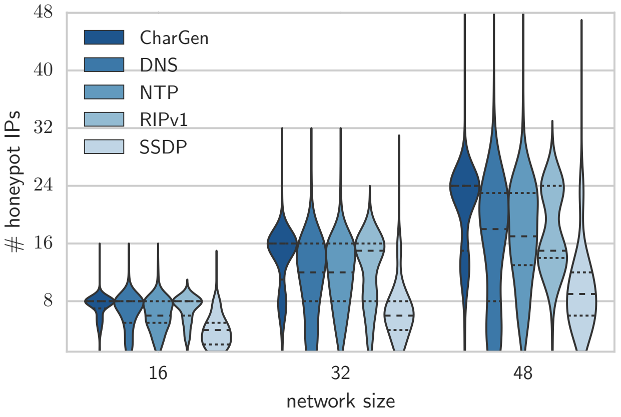

Figure 5: Honeypots used vs. network size

Furthermore, this seems to imply that α should be chosen to be very small. However, this is only true if $|A|$ was indepent of N and α. Obviously, the choice of α limits the number of IPs a single scanner can find by $|A| \leq \alpha N$ for single scanners.

We therefore analyzed the impact of our network size on the number of chosen IPs in attacks, i.e., the relation between $|A|$ and αN. To this end, we computed the distribution of attack sizes, simulating different network sizes by restricting our dataset to subnets.

We exploited the fact that our data was collected over three /28 networks by performing this analysis three times: using data from just a single subnet (16 IPs, 8 responses), using data from two subnets (32 IPs, 16 responses), and using data from all three subnets (48 IPs, 24 responses). Restricting the data on a subnet level ensures that the response ratio remains constant, e.g., in the case of 16 IPs, all scanners that scan the entire network will see 8 responses. If we had restricted the data to 16 *random* IPs, some scanners might have received 0 responses, while others might have received 16.

Figure 5 shows that the size of the attack sets correlates with the network size. Although our data is too sparse to make strong claims, data suggests that the relation between $|A|$ and αN is linear, with different slopes per protocol.

Assuming a linear relation $|A| = \beta\alpha N$, where β is the protocol-specific slope, we could further investigate the choice of α: We can rewrite the probability from above as

$$\Pr\left[A \subseteq R \mid A \subseteq Q\right] = \frac{\binom{(1-\beta\alpha)N}{(1-\beta)\alpha N}}{\binom{N}{\alpha N}} = \frac{((1-\beta\alpha)N)!(\alpha N)!}{((1-\beta)\alpha N)!N!},$$

which, for $\beta \in (0,1)$ and fixed N, has a global minimum at

$$\alpha = \left((1-\beta)^{1-1/\beta} + \beta\right)^{-1}.$$

Interestingly, the optimal response ratio is independent of the network size, and dependent only on β. Figure 6 shows the optimal value of α for $\beta \in (0,1)$. Counter to intuition, to improve confidence in the case of small attack sets, i.e., small β, one should *also* choose a lower response ratio α. In other words, the gain in confidence by reducing the response ratio α outweighs the gain obtained by increasing the size of attack sets $|A|$ due to α. Furthermore, we find that the above probability is dominated by the term $N!$ in the denominator, and thus increasing the network size N also leads to a dramatic increase in confidence.

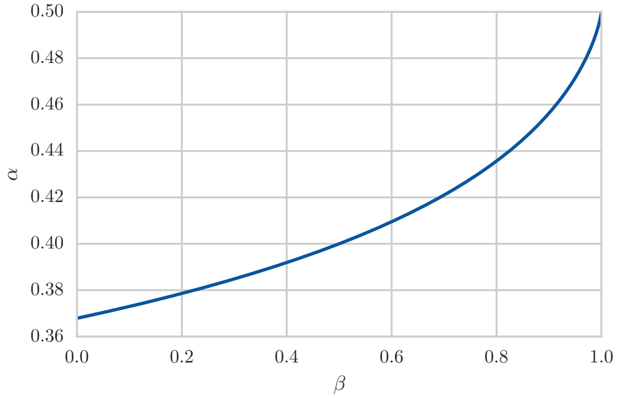

Figure 6: Optimal response ratio α for β

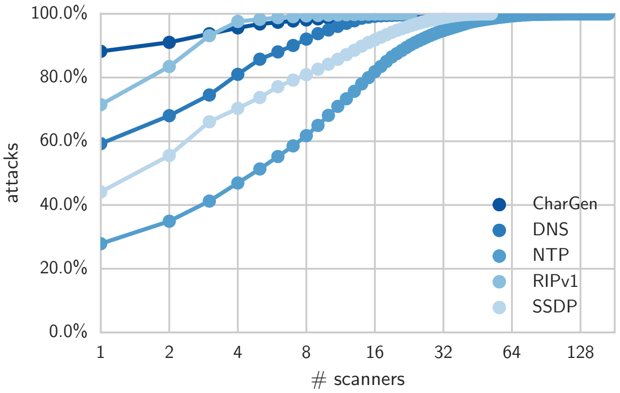

Figure 7: Percentage of attributed attacks (y-axis) vs. number of scanners (x-axis, log scale)

5.5 A Closer Look at Scanners

After uncovering the scanners providing the reconnaissance behind the attacks, we analyzed the scanners we found in more detail. To this end, we will focus on the 785,285 attacks we could link back to scanners with at least 99.9% confidence. Unless stated otherwise, percentages given in this subsection will be relative to this set of attacks.

5.5.1 Attacks vs. Scanners

Interestingly, the 785,285 attacks are based on just 286 different scanners. Furthermore, the number of linked attacks strongly varies per scanner. Figure 7 depicts the cumulative distribution function (CDF) over those attacks against scanners. As can be seen, a small number of scanners provided the amplifier sets for the majority of attacks. In the case of `NTP`, 90% of the attacks are based on the scans of less than 20 scanners. For `CharGen`, almost the same fraction of attacks is based only on the amplifier set found by a single scanner. This means that a single scanner provided the data for more than 13% of *all* attacks that our honeypot recorded.

Surprisingly, the cross-protocol share of scanners is quite large, i.e., scanners search for amplifiers of multiple protocols. About a quarter of the scanners (26%) scanned and

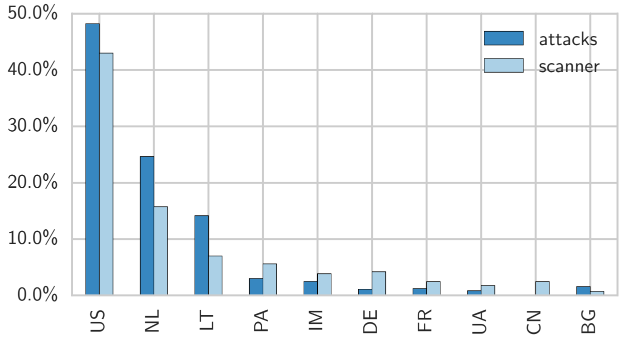

Figure 8: Percentage of hosted scanners and attributed attacks per country (top 10)

provided amplifier sets for two or more protocols, in a single case even five protocols.

5.5.2 Scanner Locations

To get a better understanding of the virtual and physical locations of scanners, we determined each scanner's autonomous system (AS), as well the country where the scanner's IP was registered. The geolocation was performed using the freely available GeoLite2-database by MaxMind [1]. We determined the autonomous system using the whois service run by Team Cymru [2]. If the latter returned no result we conducted a manual lookup by querying the respective regional Internet registry.

The 286 scanners we identified are located in 87 autonomous systems, in a long-tail distribution. The most prominent ten AS contain at least 10 scanners each, the top two even at least 25. Overall, the top 10 AS host 156 of the scanners (54.55%). This supports anecdotes that a small number of networks is reponsible for large parts of certain abuse types (here: scanning).

Even more surprisingly, the 286 scanners are distributed over only 30 different countries, again in a long-tail distribution. Figure 8 shows the percentage of scanners located in and the percentage of attacks attributed to scanners in the top 10 countries. 3/4 of all scanners are hosted in the US, the Netherlands, Lithuania, Panama, or Germany, with the vast majority of them being located in the US. Interestingly, scanners from the US, the Netherlands, and Lithuania have an above-average number of attacks attributed to them. In fact, over 87% of all attacks were attributed to scanners in those three countries.

6. MAPPING SCAN INFRASTRUCTURES TO ATTACK INFRASTRUCTURES

Mapping scans to attacks already allowed us to find one important part of the adversarial infrastructures, namely those systems that perform Internet-wide scans to prepare the attacks. In this section, we turn to the infrastructures that are actually used to *perform* the attacks. The main hypothesis that we would like to answer is the following: Are the systems used to perform scans also used to perform subsequent attacks? In fact, the technical requirements to launch attacks are very similar to those needed for Internet-scale scans. Both parts require a powerful network connection, and combining the infrastructure would certainly make attacks easier, as there is no need to exchange information. In the following, we therefore answer whether attackers actually reuse their scan infrastructure for launching attacks.

6.1 Methodology

In a DDoS amplification attack, the attacker sends out requests to a set of amplifiers, but spoofs the packet header to inject the victim's IP address. Therefore, from the amplifier's perspective, packets observed during an attack will only contain the IP address of the victim. Worse, the victim will only see that traffic is originating from amplifiers. Finding the actual packet source of amplification attack is thus a non-trivial problem—irrespective of the perspective.

To tackle this problem, we propose to combine our honeypot data with trilateration techniques to trace back the packet origin. To this end, we leverage the time-to-live (TTL) field in the packet header. When sending out packets, the sender chooses an initially high TTL value, and every hop along the route to the destination will decrease the TTL value by one. Thus, the difference between the initial TTL and the received TTL can be used to estimate the length of the route between the sender and the receiver. Having multiple globally-distributed vantage points to take TTL-based measurements between the source and the honeypots allows comparing two locations on the network following the concept of trilateration. In other words, if the locations of honeypots are wisely distributed, and if attacks abuse many honeypots at the same time, the honeypots allow measuring the path lengths between the packet origin and the various honeypot placements. This will help to approximate the packet origin: Our hypothesis is that packets from the same source will have equal (or at least similar) hop distances between the origin and our honeypots and that the initial TTL set by the sender is fixed. We back up this hypothesis with the observation that in 92% of all attacks no honeypot observed more than 3 distinct TTL values. If we thus compare the hop distances recorded at the honeypots, we can say if two packets originate in the same system by finding similar TTL distances—without relying on the IP addresses.

Formally, the TTL recorded at a receiver r is equal to the initial TTL set by the sender s minus the hop count distance $d_{s,r}$, i.e., $ttl_r^s = ttl_s - d_{r,s}$. Assuming that the sender uses a fixed intial TTL value, the location of a source s is relative to a set of n receivers $r_1, \ldots, r_n$ that can then be modeled as an n-dimensional vector capturing the distances d_{s,r_i} between the source and the receivers:

$$\vec{d}_{s,\vec{r}} = ttl_s \cdot \vec{1} - \vec{ttl}_{\vec{r}}^{s}$$

$$\begin{pmatrix} d_{s,r_1} \\ d_{s,r_2} \\ \vdots \\ d_{s,r_n} \end{pmatrix} = ttl_s \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} - \begin{pmatrix} ttl_{r_1}^s \\ ttl_{r_2}^s \\ \vdots \\ ttl_{r_n}^s \end{pmatrix}$$

To compare the location of two sources, we will use the $\ell 1$ distance (also known as the Manhattan distance or rectilinear distance). However, this is not trivial, as our model has so far assumed that the initial TTL set by the sender is known. While we have observed that most attacks indeed seem to be based on the maximum (255) as the initial TTL value, this is not a hard requirement, and might change in the future. Consequently, we do not assume a specific initial TTL value. In order to circumvent the missing initial TTL, we decided to "align" measurements. Consequently, the $\ell 1$-distance of two sources s_1 and s_2 is computed as

$$\left\| \vec{d}_{s_1,\vec{r}} - \vec{d}_{s_2,\vec{r}} \right\|_1 = \left\| \vec{ttl}^{s_2}_{\vec{r}} + \vec{ttl}^{s_1}_{\vec{r}} + (ttl_{s_1} - ttl_{s_2}) \cdot \vec{1} \right\|_1$$

and depends only on the *difference* between the initial TTL values. To "align" two measurements $\vec{ttl}^{s_2}_{\vec{r}}$, $\vec{ttl}^{s_1}_{\vec{r}}$, we find t (a value in the range $[-255, 255]$ that "shifts" the TTL values) that minimizes the following distance between two TTL vectors:

$$\left\| \vec{ttl}^{s_2}_{\vec{r}} + \vec{ttl}^{s_1}_{\vec{r}} + t \cdot \vec{1} \right\|_1 .$$

Intuitively, the more measurement points (i.e., honeypot locations) we have, the more accurately we can compute the TTL-wise distance between two sources. However, recall that while our honeypot listens on 48 IP addresses, all those IP addresses point to the same system and therefore likely have identical routes. Obviously, a single measurement point is not sufficient to perform true trilateration. Thankfully, the authors of AmpPot granted us access to their dataset. They operate 20 honeypots which are located in multiple continents, and therefore should observe different routes. Combining their dataset with ours gave us up to 21 measurement points, yielding an entropy that is sufficiently high to perform trilateration.

6.2 RIPE Atlas Probes

The TTL distance should approximate whether packets stem from the same source, in that "small" distances hint at similar packet sources. However, given Internet route changes and load balancing, it is unclear what distance we need to tolerate to spot same-origin packets. To validate whether our TTL metric is indeed meaningful and does not create false positives, we use a ground truth dataset. To this end, we leverage the Atlas project by RIPE [3]. In Atlas, volunteers host *probes*, small devices used to carry out measurements on the Internet such as "ping" or "traceroute". Measurements can be performed by anyone in exchange for a certain amount of credits, which can in turn be earned by hosting probes.

To establish our ground truth, we selected a random set of 200 probes and instructed them to send packets to the 11 most prominent honeypots. We instructed the honeypots to record the TTL values of the traffic coming from these probes. We were interested in the stability of routes at two time scales. To measure changes in the hop count in the order of minutes, we sent three packets at intervals of two minutes. To measure changes in the order of hours, we repeated this process five times at intervals of six hours. After excluding probes that only sent partial data (or no data at all), our dataset contained TTL values for 168 distinct sources for five measurements.

Our random selection of probes guaranteed creating a heterogeneous set in terms of probe locations. That is, we had probes with a large distance from each other, as well as clusters of probes from a small dense region, such as the Amsterdam area in the Netherlands. This was done to confirm the intuition that distant sources have very different routes, and to investigate whether different sources in the same proximity would be mistaken for one another—assuming that they share a large amount of routes.

To measure if our trilateration methodology would mistakenly flag two different sources as being the same, we computed the minimal $\ell 1$-distance between every pair of sources. Additionally, we also investigated the influence of the number of receivers on the resulting distance. Intuitively, given that distances sum up, a higher number of receivers could lead to a higher $\ell 1$ distance. To this end, we sampled random honeypot subsets of size $2, 3, \ldots, 11$ for each pair of sources and computed the minimal $\ell 1$ distance between the pair using the TTL values recorded by this subset.

From these distances we could then derive thresholds such that measurements with a distance below the threshold are likely to stem from the same source, while measurements with a distance above the threshold are more likely to stem from different sources. In order to measure the performance of a given threshold, we turned to two well-known measures from classification, namely the *true positive rate* (TPR), measuring the fraction of sources that could be correctly re-identified, and the *false positive rate* (FPR), measuring the fraction of sources falsely assumed to be identical. More formally, a TP means that a probe had two measurements with a distance below the threshold, while a FP corresponds to two measurements from two *different* probes that had a distance below the threshold. However, since every probe can be confused with every other probe, a global FPR is not applicable. Instead, we compute the FPR *per probe*.

We give example curves of the TPR, the average FPR, and the maximum FPR in Figure 9 for 7, 9, and 11 receivers, respectively. As expected, a smaller number of receivers increases the FPR. For example, using a threshold of 8 leads to a FPR of over 50% in the worst case when using just 7 receivers. Increasing the number of receivers to 11 decreases the FPR to below 5%. Furthermore, smaller thresholds decrease the FPR, but also lead to a loss of TPs.

This leads to the question of how the threshold should be chosen. Since we are mainly interested in learning if a scanner infrastructure is also used to launch attacks, we focus on the FPR. In a similar fashion to Section 5.2, we can fix a *confidence level* and derive a threshold for a given number of receivers: Since the FPR estimates the probability with which our method gives false accusations, the complementary probability of this corresponds to the level of confidence, i.e., the probability that the attribution is correct. Figure 10 states the thresholds per receiver set size for a confidence level of 95% and 98%, respectively. For example, two events observed by a set of 8 honeypots stem from the same source with 95% confidence if their TTL distance is below 4; to have a confidence of 98% their distance should be below 2.

6.3 Malicious Scanners

Knowing which thresholds to choose, we will now apply our methodology to our dataset of scanners. That is, by comparing the TTL vectors of a scan event and an attack, we now answer the question whether the infrastructure used to perform the scans is also used to launch attacks.

During an attack, packets are typically sent quasi-simultaneously to the honeypots. This is not necessarily true during scans, as a scanner may distribute its activities over a longer period of time. Therefore, to compare the TTL values between scanners and their attributed attacks, we computed the distance between the TTL values observed in the attack and the *chronologically closest* scan event for each honeypot. This minimizes the effects of potential route changes. To account for the fact that we might see small fluctuations of TTL values during an attack, we compare the scan against the

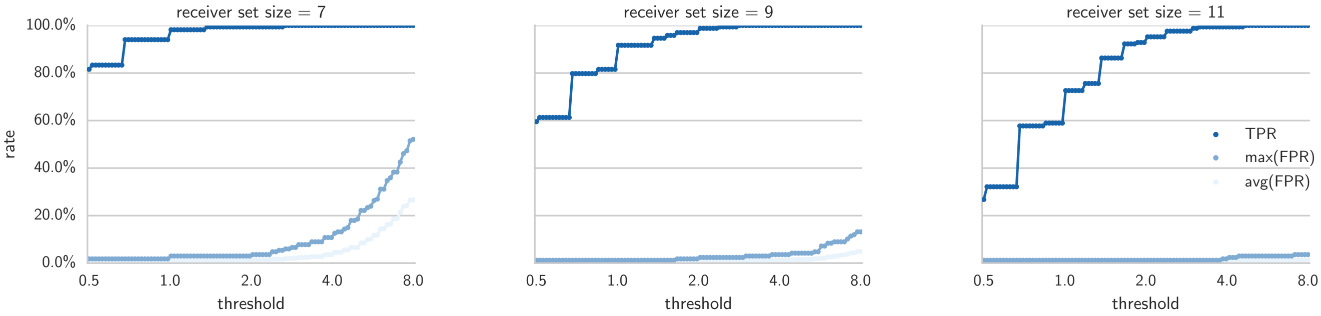

Figure 9: TPR, maximum FPR, and average FPR for receiver-set sizes 7, 9, and 11

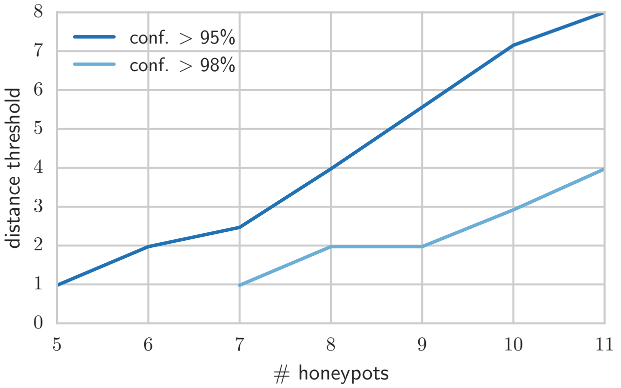

Figure 10: TTL distance thresholds (y-axis) depending on the receiver set sizes (x-axis)

average TTL value measured at a honeypot. Moreover, to stay consistent with the parameters of our Atlas-based measurements, we also limited the analysis to attacks that took place within 24 hours before or after a scan. Although this might sound like a strict limit, it excludes only six out of the 286 scanners (2%) we identified.

Under the assumption that the true distribution of scan and attack infrastructure is somewhat similar to the distribution of our Atlas probes[1], the thresholds found in the previous section are still applicable here. Using those thresholds we find that with confidence of 95% or more, 44 of the 286 scanners are presumably also launching attacks, 34 of them even with a confidence of 98% or higher. Furthermore, since these 34 scanners have been found responsible for 293,478 of the attacks with a confidence of 99.9% or higher, we conclude that our methodology successfully uncovered the true attack infrastructure behind over a third of the attacks for which we could find a scanner with 99.9% confidence, which equates to a fifth of *all* attacks observed at our honeypots.

7. DISCUSSION

Our novel methodologies help to identify infrastructures of current state-of-the-art attacks, which is significant progress in terms of finding the origin of amplification attacks. Having said this, we will now discuss some of the assumptions of our methodology and discuss how adversaries might be able to evade our attribution process in the future.

7.1 Single Scanner

As a primary assumption, we assumed that the amplifier set used in DDoS amplification attacks is scanned from a *single* public IP. This seemingly strong assumption is backed by two arguments. First, our results show a very small fraction (2.52%) of attacks for which no potential scanner could be found. Were attackers to compile their amplifier sets from multiple sources, we would expect a much higher number of attacks marked as *non-attributable* (see Section 5.3).

Secondly, attackers need to rescan for amplifiers at regular intervals due to amplifier IP address churn. Kührer et al. showed that typically less than half of the amplifiers are still reachable a week after the scan [21]. Periodic scans require a setup capable of scanning the entire IPv4 address space and suitable for long-term scan operation. Since launching large-scale scans violates most hosting providers' terms of service, we argue that maintaining such scanners incurs a non-negligible amount of work. Furthermore, when performing an Internet-wide scan, one would not expect to achieve a much different result when scanning from another source. All this combined leads us to the conclusion that attackers are not incentivized to maintain multiple scanners.

Having said this, combining the results of multiple scanners could evade our attribution in its current form. To tackle this problem, one could increase the network size N and reduce the response ratio α, such that our selective response scheme guarantees even higher entropy and also combinations of two or more scanners could be re-identified.

7.2 Initial TTL

When comparing scanner infrastructure to attack infrastructure, we assumed that the initial TTL set by the scanner and/or attacker was constant for all packets. This holds true for packets carrying non-spoofed headers, as the network stack of common operating systems will typically use a default value (usually one of 64/128/255, depending on the operating system). But even in the case of attack traffic we see fixed TTLs for the majority of attacks, in accordance with the observations made by Krämer et al. [20].

Unfortunately, attackers could evade our infrastructure comparison by randomizing their initial TTL values. However, we may be able to average the various TTL values to tolerate such randomizations, as the average survives randomization. While randomization is thus not an effective evasion technique, there are smarter ways our TTL-based methodology can be fooled, such as randomly choosing a single initial TTL per amplifier.

Worse, an attacker may try to provoke a *false* attribution result. Luckily, though, it is virtually impossible for an at-

[1] The Atlas probes were chosen at random from a globally-distributed set of systems. Lacking any ground truth on true attack sources, we had no sound methodology to further verify or invalidate this assumption.

tacker to choose its initial TTL values such that we would falsely accuse a scanner of also being an attack source, as this would require exact knowledge of the locations of and hop counts to all honeypots.

7.3 Honeypot-Driven Observations

Our selective response methodology assumes that attackers leverage sufficiently many amplifiers in an attack to allow for the attribution process. Related studies have shown that literally thousands of amplifiers are involved in attacks [29], supporting this assumption. However, an attacker may select amplifiers such that only a few honeypots see any attack traffic. For example, in our current deployment, an attacker could limit the number of amplifiers per subnet. Note that our methodology does not actually require that honeypots are located in the same subnet. The same scheme could be applied to honeypots scattered among various providers and in disjoint subnets, mitigating this problem.

Finally, attackers could aim to identify honeypots running AmpPot by its behavior and avoid their use. In its current operational mode, AmpPot can be identified, as it emulates certain protocols, usually even with a fixed response. However, as a countermeasure, one could configure AmpPot to run in a "proxy mode" that intermediates traffic between actual service implementations with amplification vulnerabilities (such as NTP servers). This would make the honeypot indistinguishable from other amplifiers. Note that such a stealth mode may also have implications on the rate-limiting functionality of honeypots, as limiting the traffic may potentially be another way to identify the honeypot.

8. RELATED WORK

In this section, we discuss existing works in the area of amplification attacks and traceback mechanisms. We refer to general surveys [23, 37] for an overview of DDoS in general.

Amplification DDoS and Booter Analysis: Our work was motivated by the research field that discusses and monitors amplification attacks. Rossow has identified amplification vulnerabilities in 14 UDP-based protocols [26]. Since then, several other researchers have analyzed booter services, which are suspected to be a major source for amplification attacks. For example, Santanna et al. analyzed 14 booter services and the attack types they offer [29]. They also investigated booter "infrastructures", but focused on the web front-ends of the booter services—which are decoupled from the booter attack infrastructure and much easier to replace than the infrastructure that we aimed to trace back. In another paper, Santanna et al. obtained and studied attack logs from 15 booter services [28], revealing insights into the distribution of attack victims and the booter website hosters. While fingerprinting the web front-end infrastructures (up to the CDN) is straightforward, we investigated the hidden infrastructures that are used to actually perform amplification attacks. Karami et al. document this clear separation between websites and back-end servers [18]. They study the ecosystem of three booter services, show the difficulty and costs for renting such back-end servers, and analyze two "spoofing-friendly" hosting providers. However, none of these studies have investigated ways to identify the scan and attack infrastructures. In addition, our work is not limited to booter services, and captures all attack sources.

Closest to our work, Kührer et al. have developed AmpPot, an amplification honeypot that can emulate Internet services to attract attacks. In their further analysis, the authors also describe early attempts to attribute the scanners that helped to prepare the attacks. However, their attribution is limited to the scan *software* (e.g., Zmap [13] or masscan), while our goal is to track the scan *origin*. Furthermore, we have significantly extended the AmpPot design with a novel idea of a probabilistic response scheme, which is a fundamental enhancement to foster tracebacks.

IP Traceback: Orthogonal to our work, IP traceback methods have been proposed to find the origin of IP packets [17]. One approach is to analyze flow telemetry collected by routers to trace the packet origin [35, 34, 39, 19]. However, these methods follow the assumption that ISPs collect and even share flow data—an unlikely scenario in practice. Another idea is to mark packets, i.e., record the path of packets as metadata in each IP packet [31, 11]. A naïve packet marking implementation could add the IP address of each router on the path to the IP packet (e.g., as an IP header option). This additional 4B overhead per hop can be reduced with better encoding schemes [36, 41, 15, 12, 14], by overloading unused IP header fields [31, 36, 10, 40, 7], using probabilistic packet marking schemes [31, 30, 33], or marking only the attack traffic [22, 6]. If deployed on a global scale, a combination of amplification honeypots and IP traceback could perfectly track the origin of spoofed traffic. However, the last two decades have demonstrated that traceback mechanisms are simply not adopted by providers, presumably as all schemes require changes to protocols or at least to router implementations. Furthermore, our methodology does not require cooperation between the providers, nor require changes to IP or router implementations.

Spoofing Detection: A few works tried to identify networks that—in general—allow for IP address spoofing [21, 4]. Our follow complementary goals in that we find network that actually *perform* spoofing for malicious purposes, and we propose first steps to attribute attacks back to networks.

9. CONCLUSION

Our novel methodology links scan infrastructures to amplification attacks, which is a major breakthrough in the process to identify the origin of amplification attacks, one of the major threats on the Internet. We found cases where the scan infrastructures are also used for attacks, i.e., we could even pinpoint the attacker down to the infrastructure that she used to perform the attacks. We have shared our findings with law enforcement agencies (in particular, Europol and the FBI) and a closed circle of tier-1 network providers that use our insights on an operational basis. Our output can be used as forensic evidence both in legal complaints and in ways to add social pressure against spoofing sources. Such pressure has led to successful interruptions of other malicious activities in the past, such as the significant drop of spam volume after the McColo shutdown in 2008 [8].

Admittedly, attackers may attempt to evade some parts of our methodology to fly under the radar. We acknowledge the possibilities for evasion, but still believe that our work is of great help to resolve the current situation of amplification sources. In the long run, we will have to seek more conceptual solutions to the problem, such as an Internet architectures that guarantee address authenticity by design (and thus prevent spoofing) [42], schemes that guarantee bandwidth reservations regardless of DDoS attacks [5], or efforts to close amplification vectors in Internet protocols [21, 9].

10. ACKNOWLEDGEMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA) (FKZ: 16KIS0656).

11. REFERENCES

[1] GeoLite2 Free Downloadable Databases. https://dev.maxmind.com/geoip/geoip2/geolite2/.

[2] IP to ASN mapping. https://www.team-cymru.org/IP-ASN-mapping.html.

[3] RIPE Atlas. https://atlas.ripe.net.

[4] The Spoofer Project. http://spoofer.cmand.org.

[5] Basescu, C., Reischuk, R. M., Szalachowski, P., Perrig, A., Zhang, Y., Hsiao, H.-C., Kubota, A., and Urakawa, J. SIBRA: Scalable Internet Bandwidth Reservation Architecture. In *NDSS '16*.

[6] Belenky, A., and Ansari, N. On Deterministic Packet Marking. *Comput. Netw. 51*, 10 (2007).

[7] Chen, R., Park, J.-M., and Marchany, R. A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks. *Parallel and Distributed Systems, IEEE Transactions on 18*, 5 (May 2007), 577–588.

[8] Clayton, R. How Much Did Shutting Down McColo Help? *CEAS '09*.

[9] Czyz, J., Kallitsis, M., Gharaibeh, M., Papadopoulos, C., Bailey, M., and Karir, M. Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks. In *ACM IMC '14*.

[10] Dean, D., Franklin, M. K., and Stubblefield, A. An Algebraic Approach to IP traceback. *ACM Trans. Inf. Syst. Secur. 5*, 2 (2002).

[11] Doeppner, T. W., Klein, P. N., and Koyfman, A. Using Router Stamping to Identify the Source of IP Packets. In *ACM CCS '00*.

[12] Dong, Q., Adler, M., Banerjee, S., and Hirata, K. Efficient Probabilistic Packet Marking. In *IEEE ICNP '05*.

[13] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-wide scanning and its security applications. In *USENIX Sec '13*.

[14] Duwairi, B., Chakrabarti, A., and Manimaran, G. An Efficient Probabilistic Packet Marking Scheme for IP Traceback, 2004.

[15] Gao, Z., and Ansari, N. A Practical and Robust Inter-domain Marking Scheme for IP Traceback. *Computer Networks 51*, 3 (2007).

[16] Graham, R. D. Masscan: Mass ip port scanner. *https://github.com/robertdavidgraham/masscan* (2014).

[17] John, A., and Sivakumar, T. DDoS: Survey of Traceback Methods. *International Journal of Recent Trends in Engineering 1*, 2 (2009).

[18] Karami, M., Park, Y., and McCoy, D. Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services. In *ACM WWW '16*.

[19] Korkmaz, T., Gong, C., SaraÃğ, K., and Dykes, S. G. Single Packet IP Traceback in AS-level Partial Deployment Scenario. *IJSN* (2007), 95–108.

[20] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K., and Rossow, C. Amppot: Monitoring and defending against amplification ddos attacks. In *RAID '15*.

[21] Kührer, M., Hupperich, T., Rossow, C., and Holz, T. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *USENIX Sec '14*.

[22] Li, Y., Wang, Q., Yang, F., and Su, S. Traceback DRDoS Attacks. *Journal of Information & Computational Science 8* (2011).

[23] Mirkovic, J., and Reiher, P. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Comput. Commun. Rev. 34*, 2 (2004).

[24] P. Ferguson, D. Senie. BCP 38 on Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. http://tools.ietf.org/html/bcp38, 2000.

[25] Prince, M. The DDoS That Almost Broke the Internet. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/, 2013.

[26] Rossow, C. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *NDSS '14* (2014).

[27] Salvatore Sanfilippo. New TCP Scan Method. http://seclists.org/bugtraq/1998/Dec/79.

[28] Santanna, J., Durban, R., Sperotto, A., and Pras, A. Inside Booters: An Analysis on Operational Databases. In *IFIP/IEEE IM '15* (2015).

[29] Santanna, J. J., van Rijswijk-Deij, R., Hofstede, R., Sperotto, A., Wierbosch, M., Granville, L. Z., and Pras, A. Booters - An Analysis of DDoS-As-a-Service Attacks. In *IFIP/IEEE IM '15*.

[30] Savage, S., Wetherall, D., Karlin, A., and Anderson, T. Network Support for IP Traceback. *IEEE/ACM Trans. Netw. 9*, 3 (2001).

[31] Savage, S., Wetherall, D., Karlin, A. R., and Anderson, T. E. Practical Network Support for IP Traceback. In *ACM SIGCOMM '00*.

[32] Schwarz, M. J. DDoS Attack Hits 400 Gbit/s, Breaks Record. http://www.darkreading.com/attacks-and-breaches/ddos-attack-hits-400-gbit-s-breaks-record/d/d-id/1113787, 2014.

[33] Shokri, R., Varshovi, A., Mohammadi, H., and Yazdani, N. DDPM: Dynamic Deterministic Packet Marking for IP Traceback. In *IEEE ICON '06*, vol. 2.

[34] Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Kent, S. T., and Strayer, W. T. Hash-based IP Traceback. *ACM SIGCOMM Comput. Commun. Rev. 31*, 4 (2001).

[35] Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Schwartz, B., Kent, S. T., and Strayer, W. T. Single-packet IP traceback. *IEEE/ACM Trans. Netw. 10*, 6 (2002).

[36] Song, D. X., and Perrig, A. Advanced and Authenticated Marking Schemes for IP Traceback. In *Proc. of IEEE INFOCOM* (2001), vol. 2.

[37] Specht, S. M., and Lee, R. B. Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures. In *International Workshop on Security in Parallel and Distributed Systems* (2004).

[38] Stadje, W. The collector's problem with group drawings. *Advances in Applied Probability 22*, 4 (1990).

[39] Sung, M., Xu, J., Li, J., and Li, L. Large-scale IP Traceback in High-speed Internet: Practical Techniques and Information-theoretic Foundation. *IEEE/ACM Trans. Netw. 16*, 6 (2008).

[40] Yaar, A., Perrig, A., and Song, D. StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense. *IEEE Journal on Selected Areas in Communications 24*, 10 (2006).

[41] Yaar, A., Perrig, A., and Song, D. X. Pi: A Path Identification Mechanism to Defend against DDoS Attack. In *IEEE S&P '03*.

[42] Zhang, X., Hsiao, H.-C., Hasker, G., Chan, H., Perrig, A., and Andersen, D. G. SCION: Scalability, Control, and Isolation on Next-Generation Networks. In *IEEE S&P '11*.