Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks

Harm Griffioen
harm.griffioen@hpi.de
Hasso Plattner Institute
Potsdam, Germany

Kris Oosthoek
k.oosthoek@tudelft.nl
Technische Universiteit Delft
Delft, The Netherlands

Paul van der Knaap
P.C.F.vanderKnaap@student.tudelft.nl
Technische Universiteit Delft
Delft, The Netherlands

Christian Doerr
christian.doerr@hpi.de
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

Amplification attacks generate an enormous flood of unwanted traffic towards a victim and are generated with the help of open, unsecured services, to which an adversary sends spoofed service requests that trigger large answer volumes to a victim. However, the actual execution of the packet flood is only one of the activities necessary for a successful attack. Adversaries need, for example, to develop attack tools, select open services to abuse, test them, and adapt the attacks if necessary, each of which can be implemented in myriad ways. Thus, to understand the entire ecosystem and how adversaries work, we need to look at the entire chain of activities.

This paper analyzes adversarial techniques, tactics, and procedures (TTPs) based on 549 honeypots deployed in 5 clouds that were rallied to participate in 13,479 attacks. Using a traffic shaping approach to prevent meaningful participation in DDoS activities while allowing short bursts of adversarial testing, we find that adversaries actively test for plausibility, packet loss, and amplification benefits of these servers, and show evidence of a "memory" of previously exploited servers among attackers. In practice, we demonstrate that even for commonplace amplification attacks, adversaries exhibit differences in how they work, and these behavioral differences allow us to cluster attacks to campaigns.

CCS CONCEPTS

• **Security and privacy → Denial-of-service attacks**.

ACM Reference Format:
Harm Griffioen, Kris Oosthoek, Paul van der Knaap, and Christian Doerr. 2021. Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3460120.3484747

1 INTRODUCTION

In order to impair the availability of Internet services, adversaries commonly deploy distributed denial-of-service (DDoS) attacks. Such attacks typically take one of two forms: (1) they target communication protocols, for example, by exhausting a maximum number of concurrent connections, a classic example of this being the TCP SYN flood; or (2) they target the connection itself by flooding the victim with large volumes of unwanted data so other data cannot come through anymore. As such packet floods would require the attacker to generate lots of traffic, these assaults are expensive for the adversary. For this reason, they are typically executed as reflection attacks, where the attacker forges a request on behalf of the victim to a third party by address spoofing and lets the answer be delivered to the impersonated victim. If the request is much smaller than the resulting response, the cost and effort for the adversary are minimal.

Amplification attacks are conceptually easy, straightforward to implement, and enable large attacks using minimal resources. The attacker only needs to find open services running on a connectionless protocol with an attractive amplification ratio. The importance of this attack vector to DDoS attacks has led to several research projects that looked into the ecosystem of these amplification attacks over the past decades. Notable contributions to the field have been made by Paxson [24] and Rossow [27], who created *amplification honeypots* to capture attacks in-the-wild, or Krämer et al. who ran 21 nodes with their AmpPot to discover previously unknown details on amplification attacks and their victims [17].

While several studies look at amplification attacks in the wild, little is known to date about the activity behind these attacks. In other words: how do adversaries plan, prepare, and execute these attacks? For example, when attackers collect lists of abusable services, do they indiscriminately use any servers they encounter, or do they carefully test, select and curate a portfolio of amplifiers? Do adversaries go for "textbook" attacks, or do they make efforts to understand and adjust their actions to cause the most damage?

To help answer these questions, we can look at DDoS attacks not just in the context of the resulting packet flood but aim to understand the entire chain of events that led to the execution and (successful) completion of the attack. As shown in figure 1, before packets flow towards the victim, adversaries will have to go through a series of planning and preparatory steps, understanding which attack vector to use, locating infrastructure that could be abused for amplification, selecting how the victim should be targeted, implementing

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8454-4/21/11.
https://doi.org/10.1145/3460120.3484747

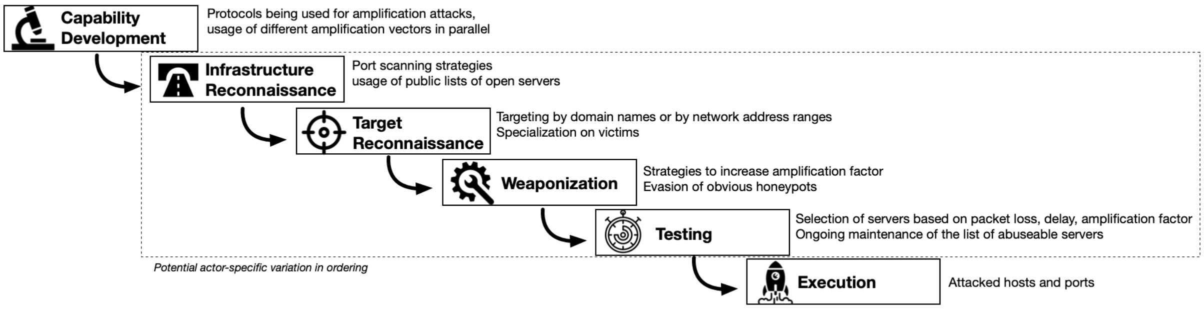

Capability Development
Protocols being used for amplification attacks, usage of different amplification vectors in parallel
Infrastructure Reconnaissance
Port scanning strategies usage of public lists of open servers
Target Reconnaissance
Targeting by domain names or by network address ranges Specialization on victims
Weaponization
Strategies to increase amplification factor Evasion of obvious honeypots
Testing
Selection of servers based on packet loss, delay, amplification factor Ongoing maintenance of the list of abuseable servers
Potential actor-specific variation in ordering
Execution
Attacked hosts and ports

Figure 1: To understand how adversaries work, we can separate DDoS attacks into separate phases and activities and investigate actor-specific techniques and commonalities in behavior between adversaries.

software and payloads to implement the attack, to eventual testing. Each of these steps can be implemented in various ways. By studying adversarial behavior, we gain better insights into the threat landscape of DDoS attacks assisting in developing better defenses.

In this paper, we perform an investigation into the different phases of DDoS attacks and in-depth analysis into the steps and techniques used by adversaries while developing, preparing, and executing such attacks. By deploying 549 honeypots in 5 public clouds around the world over a period of 3 months, we have collected a total of 605,181 reconnaissance and testing activities leading up to 13,479 attacks consisting of 720,995 individual attack flows on 8,315 victim systems. Through careful design of experimental conditions where we use a new traffic shaping approach to ensure we are not actively participating in these attacks, our study elicits adversarial behaviors by monitoring amplification attacks through an infrastructure an order of magnitude larger than previous work and provides significant new insights into the threat landscape that have been primarily investigated based on the attacks themselves. We show that a bulk of these 13,479 attacks can be linked together based on shared adversarial techniques, tactics, and procedures (TTPs). With this work, we make the following main contributions:

- We are the first to investigate the full ecosystem of amplification DDoS attacks and the impact of amplification power offered by a service on the level of abuse.
- We demonstrate a wide variety of ways how attackers maintain and curate their lists of abusable servers. We find that there is a "memory" on the Internet for some protocols that IPs once ran a service. Adversaries still return to these weeks after the service has stopped responding to requests.
- We show how sophisticated attackers optimize their bandwidth usage by conducting attacks in pulses rather than a constant packet stream, reducing the cost of attacks.
- We demonstrate that the number of honeypots needed to obtain sufficient attack coverage is much higher than previously shown in related work and that the threat landscape is more diverse than previously thought once we employ a large number of honeypots.
- We show that while the bulk of the threat landscape consists of unsophisticated actors with little capability, there are very knowledgeable adversaries who test services they abuse.

2 RELATED WORK

While DDoS attacks have historically been conducted using botnets as a means to amass enough firepower [5], the largest attacks to date were recorded as amplification attacks, with Google reporting attacks peaking at 2.5 Tbps [1]. The ease with which these amplification attacks create large traffic floods popularized this method, and researchers report a median of 1,930 attacks per day since 2014 [32]. Many works exist analyzing observed amplification attacks by setting up *amplification honeypots*. In 2001, Paxson conducted the first work on capturing these amplification attacks "in the wild" by setting up these fake amplification services [24]. In more recent work, Rossow [27] provides an in-depth study into attacks performed in 2014, analyzing captured traces on various protocols and evaluating open amplification services on the Internet. Subsequently Kührer et al. have performed significant work to reduce vulnerable NTP services and IP address spoofing [20], complemented by others that have focused on amplification using specific protocols, such as DNS [3, 8, 10, 21, 22, 31] or NTP [6, 25, 28, 30]. The largest study on amplification attacks to date has been performed by Thomas et al., who capture attack traces using a mean of 59.7 honeypots over 1010 days [32].

While the work on executed amplification attacks is abundant, research on the ecosystem behind these attacks is scarce. The first work to address this is from Krämer et al. [17], who deployed honeypots with different modes to capture differences in the behavior of adversaries using these systems. The authors used a fixed rate-limiting threshold to prevent their experiment from causing serious damage but identify the concern that this rate limit might lead to honeypots being detected as such by scanners. While work by Krämer et al. [17] has tried to establish how many honeypots are needed to capture a significant part of these attacks by setting up 21 honeypots in different geographical regions, the lack of understanding on how "smart" adversaries select their amplification servers can significantly bias the results as these attackers might test the devices and consequently stop using them. To further understand the ecosystem of amplification attacks, Krüpp et al. analyzed the reconnaissance activity performed as part of DDoS operations [18]. In another experiment, Krüpp et al. used a set of 11 honeypots to attribute their use to certain booter services [19]. While the authors could link usage of their honeypots to actual booter services, they only found their results representative for DNS and NTP as they missed portions of self-attacks for SSDP and CHARGEN. Other works have also investigated the use of booter services in order to gain more understanding of the ecosystem [12, 14, 15, 29]. However, only some of them have specifically explored the use of amplification DDoS honeypots in the context of these services, such as Kopp et al. [16] who have investigated spatial and temporal trends of DDoS attacks launched by booter services.

Contrary to honeypots aimed at compromisation attempts [33], it is currently unknown to which extent adversaries test and abandon amplification honeypots. While some works set initial steps into this direction by creating different configurations in the honeypots [17], none of the prior works has deployed a large enough network of honeypots to accurately capture these adversarial differences nor used a traffic shaping method to allow adversarial testing traffic. As prior works on DDoS attacks have mostly considered the execution and only look at a narrow set of the entire chain of attacks required for the attack to be successful, it is currently unknown to what extent adversaries test systems to use in their attack and whether they are actively weeding out known honeypots. The closest work to address the entire chain of steps is by Krupp et al. [18] who analyze the scanning behavior between different attacks but do not investigate other activities around the preparation or testing of an attack. Analyzing the entirety of these attacks will allow researchers to describe and discuss these attacks in more detail.

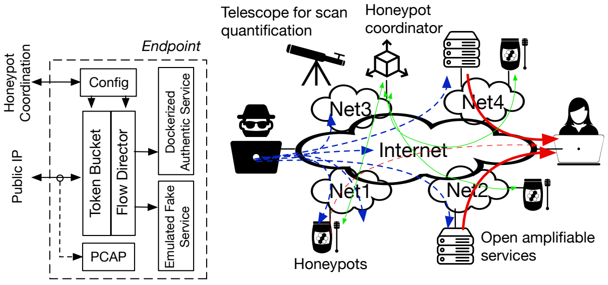
In this work, we go beyond adversarial scanning and set up various experiments using a configurable honeynet an order of magnitude larger than prior work. In a series of experiments, we not only investigate how adversaries discover and abuse amplifiers but study all steps adversaries take leading up to the DDoS attack itself. This allows us to study the sophistication of adversarial activities and a deeper assessment of the threat landscape than before.

3 DDOS ATTACK CHAIN AND SYSTEM SETUP

Previous studies discovered several methods used by adversaries to select amplification servers and victims. However, as discussed in the previous section, most research is limited to the attack itself and does not study the sequence of events before and during the attack nor investigates the adversarial characteristics behind the attack. In this study, we aim to address this gap through configurable, adaptive honeypots, which enable us to investigate how actors work towards attacks, how they prepare, what they look for in a service to abuse until the actual execution of the attack.

The DDoS Attack Chain. Amplification attacks typically do not suddenly emerge, but the onset of an attack on a victim results from a series of preparatory steps by the adversary that we can identify and measure. In this paper, we structure these activities along a sequence of six consecutive phases as depicted in figure 1, which allows for an effective discussion and thus a better understanding of how DDoS operations are realized. Although the concrete attack on a particular victim may be unique, adversaries can be assumed to recycle much of the supporting activities, such as the list of abusable servers or attack methods, which allows the recognition and linking of individual attacks, as well as to provide a better understanding of the ecosystem and an early warning system for upcoming activities.

As shown in figure 1, a pre-requisite for a DDoS attack is some *Capability Development* by the adversary. Depending on attacker sophistication, this can range from modifying standard scripts to tailor-made attack vectors. Next, the attacker would have to perform *Infrastructure Reconnaissance* to identify and collect systems that can be used as part of the attack. In terms of amplification attacks, the attacker would locate available amplifiers, but other attacks could recruit unsecured IoT devices or bots as a platform for the attack. The adversary then needs to know where to attack the victim. In *Target Reconnaissance* the fundamental details necessary for the attack, such as the victim's IP address, open ports, and vulnerabilities, are collected. During *Weaponization*, the output of the previous three phases is fused to craft the attack technique that will be used to perform the attack. To assess whether the developed weaponization is suitable to harm the target or even work with the identified infrastructure, some *Testing* may be required, before finally, the *Execution* commences, and is 'delivered' to the victim.



(a) Honeypot Design (b) System Architecture

Figure 2: Deployed honeypots maintain a control connection to a coordinator to perform effective rate-limiting.

System Overview. In order to observe these adversarial TTPs along the DDoS attack chain, we developed a framework for the operation and management of honeypots. The system aims to enable deployment and exposure of commonly used services towards the Internet and allow us to facilitate attacker-dependent, configurable responses to incoming requests dynamically.

Figure 2(a) shows an architectural diagram of the honeypot. In order to provide an authentic appearance, each honeypot runs a selection of containerized services, for example, *BIND* to service DNS requests or *ntpd* for NTP queries. Incoming requests from the Internet are either proxied to these instances or to an emulated service that returns (obviously) fake answers. This emulated service does not implement all features the protocol requires in order to test the interaction and reaction of adversaries to services. The Bandwidth Amplification Ratio (BAF) [27] of the honeypots are listed in Table 1. In addition to logging the interactions of adversary and backend services at the application layer, packets are also recorded at the link layer to discover reconnaissance actions like port scanning as well as implementation characteristics in the attack packets to potentially relate individual attacks and adversaries.

Our behavior differs depending on how the honeypots are contacted. As shown in figure 2(b), all honeypots are in constant communication with a central coordinator, which tracks the requests sent to all 549 honeypots. Suppose the incoming requests exhibit features of selected testing (e.g., only individual honeypots are triggered, short bursts instead of a packet flood are requested, requests are typical reconnaissance but no amplification packets, etc.). In that case, the coordinator instructs the honeypot to comply, if the requests show the characteristics of an attack, which we can detect due to our distributed setup, which will be explained in Section 5.2, the configuration is adjusted across *all* honeypots to reduce the data rate of all 549 servers towards the IP address to a maximum of 0.5 Mbps. This value was chosen to result in a minimal impact and ethical operation of the honeypot system. In several rounds of tests before the actual experiment, we verified whether adversaries would detect this traffic shaping and subsequently stopped using these systems. This was not the case. We used only honeypots at Digital Ocean for these tests and ran the experiments on different IP addresses. As we find no statistical differences between cloud locations, this verification did not bias our experimental setup.

Experimental Design. In order to discover how adversaries select their infrastructure and victims as well as perform their implementation and execution, we rolled out different configurations to parts of our honeynet. Specifically, we are interested in the influence of:

1. **Amplification Factor**: do adversaries search for and focus on servers with the highest amplification?
2. **Obvious Honeypots**: are adversaries monitoring and discarding servers if they do not fully implement the protocols or even explicitly announce to be a honeypot?
3. **Suspicious systems**: are adversaries actively investigating the system, and do they behave differently if an unbelievable number of open services is installed?
4. **Packet Loss**: do adversaries test and discard services if they are not constantly answering or drop requests?
5. **Packet Delay**: do adversaries select servers based on the response time and speed?

We have structured our system as follows: honeypots are split into four groups with low and high amplification ratios providing a correct response, and low and high amplification ratios providing a response that is obviously a honeypot by replying with a message such as *"This is a honeypot attack detector, usage is logged and rate-limited"*. So that size is not a confounding variable; these responses of the obvious honeypots are equally large as real responses. Additionally, we vary the number of services operating on the servers. To ensure that the results are not biased due to the location of the honeypots, we distributed groups equally over five different clouds and availability zones. Throughout the experiment, we alter the behavior of this setup, for example, by dropping packets or delaying the responses, to identify the reaction of the adversaries.

Ethical considerations. DDoS amplification honeypots run the risk of participating in attacks when being used by adversaries. In the past, honeypot systems were typically implemented to drop attack packets entirely [27], the same method is used for other works [13, 17, 32], however, this comes with the complication that the research community would not be able to observe the more sophisticated actors who would consequently abandon honeypots after testing. In order to balance the need to understand the threat landscape and actor actors while not irresponsibly participating in DDoS attacks, we implemented a token bucket as a flow shaper in front of the honeypots. Figure 3 visualizes the concept, where at a specific rate, tokens are being dropped into a bucket of maximum size. Every packet that leaves our amplification honeypots consumes a token, and if none are available, the packet is dropped. Token buckets have the advantage that they can be configured to throttle attack traffic aggressively but at the same time allow short temporary bursts of activity that we would see during testing. Adversaries would therefore consider the honeypot as a normal server. However, during an attack, we would not create a significant impact, leading to a loss of attack power for the adversary when using our system instead of a real amplification server. How we behave during testing and sustained attack is governed by the fill rate and bucket size. We chose a bucket depth for a peak of 25 packets per second, which we experimentally determined in our pre-study. This value was 20% above the maximum traffic we observed from adversaries during testing. Thus the honeypots would pass all testing procedures we observed. After the bucket is emptied, the *combined* attack force of our honeypot drops down to a maximum packet rate of 0.5 Mbps, which is less than $\frac{1}{160}$ of the average Internet speed [2]. Sending a fraction of the traffic that a *single* Internet user could send, we are not actively hampering victim systems. At the same time, we are in practice able to collect data on adversarial behavior in DDoS attacks.

Table 1: Response sizes of the honeypot setup in bytes and the Bandwidth Amplification Factor (BAF) [27] per protocol, the allocation over cloud providers and different experiments. Response contents are listed in Appendix A.

	Responses	Real small	Real large	Fake small	Fake large
RIP	Bytes	84	524	88	413
	Max BAF	3.5	21.8	3.7	17.2
CharGen	Bytes	94	1,406	94	1,450
	Max BAF	94	1,406	94	1,450
QOTD	Bytes	45-50	1,450	53	1,437
	Max BAF	45-50	1,450	53	1,437
SSDP	Bytes	272	430	277	429
	Max BAF	2.3	3.7	2.4	3.7
NTP	Bytes	Varies	Varies	Varies	347
	Max BAF	0	46+	0	43.4
DNS	Bytes	83	Varies	83	352
	Max BAF	1.6	1.6+	1.6	6.8
Cloud provider	**Total servers**	**Real small**	**Real large**	**Fake small**	**Fake large**
AWS	171	47	52	36	36
Azure	24	6	6	6	6
Digital Ocean	166	45	49	36	36
Google	124	32	34	29	29
OVH	64	16	16	16	16
Experiment	**Total servers**	**Real small**	**Real large**	**Fake small**	**Fake large**
Single protocol	120	30	30	30	30
Packet loss	60	15	15	15	15
Packet delay	60	15	15	15	15

Table 2: Timeline of data collection phases.

Phase	From	To
Passive (scanning baseline)	31-08-2019	07-09-2019
Active	07-09-2019	27-09-2019
Passive (attacker memory)	27-09-2019	30-11-2019

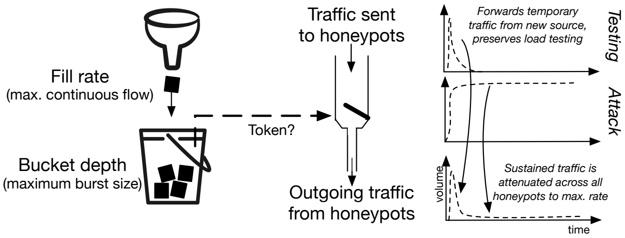

Fill rate
(max. continuous flow)
Bucket depth
(maximum burst size)
Token?
Traffic sent to honeypots
Outgoing traffic from honeypots
Forwards temporary traffic from new source, preserves load testing
Testing
Attack
Sustained traffic is attenuated across all honeypots to max. rate
volume
time

Figure 3: The token bucket limits the maximum outgoing flow, but preserves unseen temporary bursts and testing.

We have received approval from our IRB for the design, and additionally from our information security officer (ISO) whether the honeypots, rate limiting, and maximum data rates were responsible choices. No further requirements were put forth, and the experimental setup and data rates were confirmed as adequate choices.

4 DATASETS

Our 549 honeypots were distributed in the public clouds of the five largest cloud providers Google, Amazon, OVH, DigitalOcean, and Microsoft, and placed in availability zones in North America, Europe, Australia, and Asia between 31 August 2019 and 30 November 2019. The distribution of different servers over cloud providers and experiment groups is listed in Table 1. As we are interested to understand the entire chain of adversarial preparation and maintenance behavior and understand if adversaries work ad-hoc or build repositories of known servers, the honeypots were not activated immediately on the first day. Instead, for the first week, honeypots merely recorded all scanning activity without responding to establish a baseline. Subsequently, the honeypots actively answered requests for 20 days. Finally, we again switched to passively recording scan and attack packets that were directed towards us for another nine weeks (see Table 2) to understand the "memory" of the attack landscape, whether attackers use previously discovered amplifiers even though they are not active.

Aside from the application and link-layer traces from the honeypots, we used two additional datasets to further quantify how adversaries scan the Internet for open amplification services. While these datasets do not provide additional insights into amplification attacks, they do provide insights into the reconnaissance phases of an attack. First, a large network telescope of three partially populated /16 networks with 65,000 IP addresses providing a historical record of scan traffic for the past five years, and second, scan activity against the distributed telescope of the provider Greynoise. This allows us to differentiate whether attackers focus on select public clouds or perform broad scans across the entire Internet.

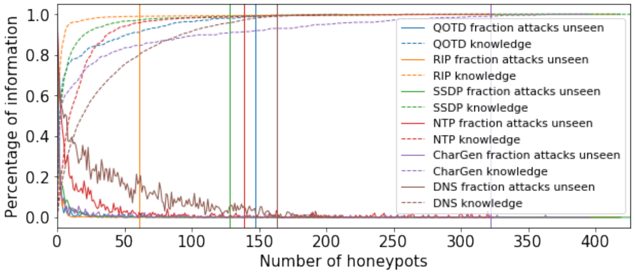


Figure 4: New information per extra honeypot. Dotted lines show the total number of attacks identified and regular lines show the extra information added per honeypot. Vertical lines indicate knowledge of 99% of attacks.

The number of honeypots needed to perform accurate measurements is much higher than previously believed. As we will show later, the threat landscape of actors performing amplification DDoS attacks is highly heterogeneous regarding techniques and resources applied. For instance, adversaries often do not exhaustively search the Internet for all open, amplifiable services. However, they are content with small sets of amplification servers, since on average, only 41 of our 549 honeypots were used in a given attack campaign. This means that to obtain a good overview of the ecosystem, it is necessary to operate many honeypots to capture small attacks and avoid bias towards adversaries conducting massive trawling through the entire Internet. Figure 4 shows the convergence of the spectrum of attacks seen as a function of honeypots in operation. Even in comparatively simple protocols such as NTP and quote-of-the-day (QOTD) where DDoS attacks presumably all look similar, the heterogeneity when looking at the entire sequence of attack steps is so large that as many as 150 honeypots are needed to capture 99% of actor behavior. This shows the constant evolution of the ecosystem, as previous work from 2015 identified that the majority of attacks were captured in 21 honeypots [17]. The honeynet size of more than 500 servers – an order of magnitude more than in previous studies – is thus necessary to provide a good understanding of the DDoS threat landscape.

5 AMPLIFICATION DDOS IN THE WILD

When we activated our honeypots after the week-long baseline, it took mere hours for the first adversaries to abuse our infrastructure in attacks. During its 20-day activation, we recorded 13,479 separate attack campaigns, targeting 8,315 unique source IPs located in 4,340 /24 subnets. Altogether, our honeypots collected 448 GB in amplification requests attributed to attacks, for which we generated on average 0.12 Mbps from our system towards a victim. In this section, we discuss these attacks using the model from figure 1.

5.1 Capability development

To perform an attack, an adversary first needs to develop a capability to execute the attack with a specific vector. In this study, we focus on six commonly abused protocols in amplification attacks [27]: NTP, DNS, SSDP, CharGen, RIP, and Quote-of-the-Day (QOTD). Overall, we find adversaries are abusing these long-established protocols with conceptually similar vectors. We however also see glimpses of innovation and capability development where attackers are not bound to a specific protocol and perform attacks using multiple protocols simultaneously. This however only occurs infrequently, as only 252 of the 13,479 attacks leverage multiple protocols in their attack.

Table 3: Number of scanning IPs per protocol.

Protocol	Residential	Hosting	Research
NTP	855 (50.1%)	941 (24.8%)	489 (25.1%)
DNS	317 (38.3%)	179 (9.3%)	658 (52.4%)
SSDP	341 (63.3%)	138 (9.0%)	425 (27.7%)
CharGen	63 (62.5%)	8 (5.3%)	51 (32.2%)
RIP	31 (73.1%)	2 (4.7%)	32 (22.2%)
QOTD	24 (50.5%)	2 (2.0%)	28 (47.5%)

Table 4: Distribution of connections per country.

Country	Residential	Hosting	Research
US	287 (28.8%)	361 (7.7%)	1377 (63.5%)
TR	500 (35.5%)	819 (64.5%)	0 (0%)
CN	291 (100%)	0 (0%)	0 (0%)
NL	90 (78.4%)	14 (5.4%)	92 (16.2%)
GB	43 (75.9%)	10 (18.3%)	108 (5.8%)
FR	49 (79.7%)	4 (6.0%)	32 (14.3%)
Other	377 (88.5%)	31 (7.3%)	18 (4.2%)

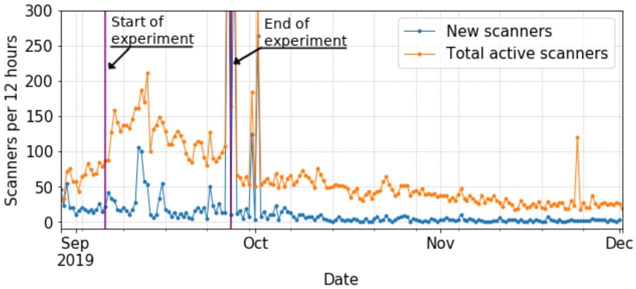


Figure 5: New/total active *non-research* scanners. After activation on Sep 7th, traffic from recurring scanners increased and only slowly declined after the shutdown on Sep 27.

5.2 Infrastructure reconnaissance

Before a server can be abused, an adversary needs to know about its existence. This is typically executed through port scanning, which our system can distinguish from attacks, as during a scan, only a few packets are sent to one honeypot, wherein an attack, many packets are sent from one IP address to multiple open services. To identify scanners, we apply our auxiliary telescope datasets and the honeypot servers that are only running a subset of services. If source IPs connect to dark IP addresses or honeypots where a service is not running, these requests are scanning and not part of attack usage. We experimentally derived that actors use up to 20 packets from the same source IP to test honeypots. In the following, we use this threshold to classify probes below this as scanning, while flows of more than 20 packets towards two or more of our honeypots are labeled as an attack.

Our honeypots report 3,650 distinct IP addresses sending scanning probes to our system, in which NTP, DNS, and SSDP are much more popular than the other protocols in terms of scanning activity. Based on the data in the telescopes, we can identify whether hosts scan the entire Internet or target the cloud providers in which our honeypots are located. Surprisingly, only 56% of all scans seem to target the Internet indiscriminately, and 44% was only observed on our honeypots in the cloud locations. Additionally, we find that 39% of IP addresses targeting only cloud instances are hitting only one cloud location, which we can largely attribute to scans originating from the same /24 subnets targeting their scans towards separate cloud locations spread across the netblock. The remainder of the scans targeting small parts of the network might originate from botnets segmenting their scanning activity or from attackers probing a single cloud provider. We observe a small fraction (6%) of all attacks only using amplification servers located in a single cloud, which we will show later, are not performed by sophisticated attackers.

Research scans are prevalent, and we need to account for these to avoid biases. Not all scanning traffic is malicious, as services such as Shodan, Censys, and Rapid7 scan the Internet for research purposes and identify themselves as such using hostnames such as *worker-01.sfj.corp.censys.io* and *census6.shodan.io* and originate from a known block of IP addresses. To obtain an accurate view of scanning with malicious intent, we manually classified IP addresses either as research, hosting provider, or residential, based on reverse DNS, BGP data, and cloud customer IP ranges. Table 3 shows the result per protocol. It is striking that overall, research-based scanning accounts for more than 30% of all scans and 37% of all IP addresses. For DNS, research-based scans even amount to over half of all scanning. While not common in scanning research, it appears that careful curation is needed.

When considering the geolocation of scanning IP addresses, most of the hosts are located in the US. However, as listed in table 4, the large majority of IP addresses scanning from the US belong to research institutions. In our analysis, we thus excluded research scans from our dataset and instructed our honeypots not to respond to these projects. By excluding scanning institutions such as Shodan and Censys [7, 23] our honeypots will not appear on their publicly available lists, meaning that attacks can only be a result of scanners from the other categories: residential and hosting IP addresses.

Responsive IPs make scanners come back twice as fast. As shown in figure 5, the number of IP addresses scanning our honeypots sharply increased after the services started to respond after a week of passive listening. Richter et al. [26] have investigated this by comparing scanning traffic in a CDN against a network telescope and find that the presence of active services would trigger adversaries to intensify their activities. By comparing the difference of our baseline and the active experiment, we find that this increase is primarily driven by intensified rescanning – IP addresses come back to a vulnerable server on average two times faster than when the scanned protocol is not present on the machine – and to a lesser degree on new IP addresses emerging. After the infrastructure returns to passive mode, previously connecting IP addresses slowly go back to the baseline. As we will show later, the attacker's "memory" of which services used to be available at an IP easily spans months.

Initial testing happens during scanning. During a scan, adversaries are already interested in the first quantification of the system. We have seen that even during the initial scan, the amplification supplied by the server is tested; adversaries would use the same packets that will eventually be used in attacks to immediately establish whether a server can be used later on based on the observed amplification power. We found this for all protocols except DNS, where scans primarily request the BIND version running.

5.3 Target reconnaissance

Compared to DDoS attacks that exploit a particular vulnerability, the amount of target reconnaissance activity is minimal in case of amplification attacks. Though an attacker can direct any kind of data to the victim to consume the bandwidth, we still observe slight nuances in how victims are targeted in practice.

Victimization has changed since previous studies. The majority of victims are located in the US and China, with 846 and 602 subnets being attacked, consistent with previous works and logical given IP address allocation. Jonker et al. [13] found attacks mostly follow Internet usage patterns with exceptions of, for example, Japan, Russia, and France. We however observe a different disproportional share of attacks on countries such as South Africa, Poland, or Kuwait. We find that this disproportionality mainly stems from different services located in these countries, such as hosting providers which are extensively attacked during our study. As our experiments only ran for a limited time due to the scale of our honeypot system, we cannot identify seasonality in the attacks, and we could therefore have a bias towards large attacks that happened during our measurements. However, as we identify significant changes in attack traffic than previous work, future work should identify whether there are trends in DDoS attacks or are largely spurred by opportunistic attacks.

Only 51% of all targeted IP addresses had a domain name pointed to it during the attack based on passive DNS and a database of daily active domain crawls. While we expect most DDoS attacks to be targeted against servers that would commonly have a domain name pointed to them, we find that there are also a large number of DDoS attacks on residential IP address space without domain names. As the domains are not only pointing towards web servers but also to, for example, Minecraft servers, we queried the Shodan API for active scanning data to find open ports on the victim devices. Shodan lists the open ports for 1,289 (9,6%) of the IP addresses that were attacked. While Jonker et al. [13] associate most UDP amplification attacks with online gaming, we find that only a small part of the attacks we have recorded are targeting game servers or ports used by multiplayer games.

Attacks towards large domains are rare. Load balancing interferes with an attacker's objective, as taking down a host leaves a service unimpaired, redirecting users to one of the hosts that are still online. Performing an attack on these services requires an attacker to attack all fronts, using the domain name resolution to resolve all IP addresses of this service. By identifying attacks on IP addresses hosting common domain names using passive DNS and active domain lookups, we can find attacks conducted on multiple IP addresses addressed by a single domain name. In total, we find 862 domain names for which more than one IP address was part of an attack and find that not only do attacks target multiple IP addresses for the primary domain but also multiple subdomains simultaneously. In an attack on the Discord service, providing chat rooms targeted at gaming communities, 61 servers were targeted, aiming to bring down a specific region indicated by domains such as *russia17.discord.gg*. Many of the attacks directed to multiple IP addresses hosting domains are targeting CDN providers such as *yunjiasu-cdn.net* or *googleusercontent.com*. As the attacks solely target the IP addresses where a domain is hosted and no other addresses belonging to the specific company, these attacks could only have been conducted using DNS lookups.

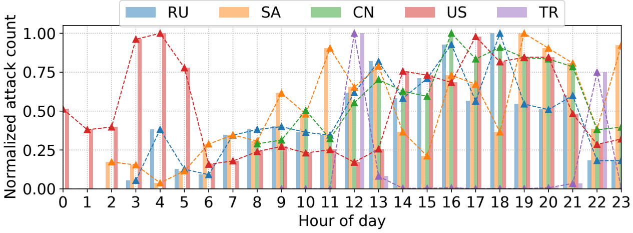


Figure 6: Local attack times for the top 5 victim countries. Attacks are mostly conducted in the afternoon and evening, except in the US where mainly data centers targeted at night.

Subnet-based attacks are increasingly rare. Attacking an enterprise can be daunting, as the resources or large organizations allow them to switch between different IP addresses to mitigate an attack rapidly. To avoid this, attackers choose not only to attack the IP addresses running a service but also to attack entire /24 subnets. These attacks do not target hosts but exhaust the router's capacity in front of these hosts, rendering all hosts unreachable. We find these attacks are increasingly rare as opposed to [32], as we observe 12 complete subnets being attacked with another 29 subnets being partially attacked during our entire study while Thomas et al. identify on average 5.39 of these attacks per day. All full subnet attacks we have identified are aimed at shared hosting providers, for which a single set of amplifiers is used for the entire attack across all IP addresses. This is implemented using a round-robin for the IP address of the subnet, which are all hit consecutively in the full-subnet attacks observed.

Attacks are scheduled to hit during prime time. While related work has not considered the time a DDoS attack is conducted from the victim's perspective, we find that DDoS attacks are aimed at a victim during times that there would be the most impact. Figure 6 shows the start of attacks in the local timezone of the victims for the top 5 attacked countries. We find that attacks occur mainly in the afternoon and evening, where many people would be using the attacked services. Based on open ports listed in Shodan, attacks on websites occur evenly during the afternoon and evening. Interestingly, we find that 83% of attacks on game servers occur after 6 PM as adversaries hit these servers at their busiest times.

5.4 Weaponization

After the reconnaissance phases, the adversary needs to create software and packet payloads to trigger the amplifiers. This might result in attacker-specific and tool-dependent implementations.

Packet content shows high homogeneity between attacks. In the first step of the analysis, we investigate the commands used to trigger the amplification, where we find as shown in table 5 very high homogeneity of what techniques adversaries use, even though we have observed 16,900 different payloads being directed to our honeypots. In NTP the monlist packet is present in almost

Table 5: Most popular requests per protocol.

Protocol	Command	% of attacks
NTP	Monlist	99.96
DNS	Root lookup domain	75.61
SSDP	ssdp:discover Host:239.255.255.250:1900	67.16
CharGen	0x01	71.07
RIP	Standard request	100
QOTD	0x01	72.66

all attacks, and similarly in RIP only one packet has been used for all attacks. Attacks using SSDP however show the use of multiple different strings, although the overall attack stays the same, where they all request *ssdp:discover* in the packet, albeit with different flags or different order in the fields sent. In QOTD however, 10% of the attacks were observed leaving the message *bigbo* in our honeypots, and another 8% of all attacks requested *getstatus*, which has no meaning in the quote of the day protocol. DNS attacks were mainly conducted with empty packets that did not request a server to perform a lookup and return some record, which results in a response containing a string of root servers, amplifying the DNS header approximately six times. Attackers could obtain much higher amplification rates when using servers enabling zone transfers, which allows attackers to make a DNS server pass a copy of its database to a victim. While we would expect adversaries to probe our systems on whether this is possible, none of the attackers has tried to identify non-restricted DNS servers.

Attackers care about the amplification power of a server - at least in some protocols. We placed honeypots with different amplification factors throughout our network to test the differences in how attackers located and used them. Intuitively, two things could happen when attackers find several servers with various amplification ratios: (1) The attacker is aware and consciously optimize based on amplification ratio, in which case we should see a selection bias towards high amplification machines, or (2) the attacker does not track this in which case we should see randomly sampling from the set of high and low amplification servers. Figure 7 shows the number of high and low amplification servers utilized in each attack, where each dot is an attack and the color indicates the protocol used. We configured an equal amount of machines to act as a high or low amplification server by letting them answer requests with a different response size. Suppose an adversary is not making any active selection. In that case, the same number of high and low amplification honeypots should statistically be picked, and the dot thus falls on the 45-degree line or in the blue shaded area for sampling from a hypergeometric distribution at 95%, 99%, and 99.9% confidence intervals.

As we see in the figure, there is active selection for “good” amplification servers in many attacks. These practices are differently pronounced depending on the protocol used in the attack. For example, RIP is never located outside the 95% confidence interval of non-selective behavior, while attacks using NTP and DNS are seldom located inside this interval. We see significantly different levels of heterogeneity for other protocols depending on the attack campaigns, with some actors randomly picking servers and more sophisticated attackers concentrating on high amplification servers only. It is interesting to note that no attacks use a disproportionate amount of low amplification servers, confirming the hypotheses that attackers either randomly sample or select higher amplification servers.

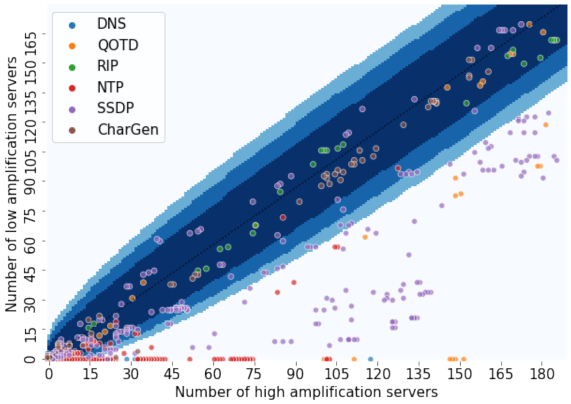


Figure 7: Selection of low and high amplification servers in attacks, showing a surprisingly little amount attackers actively select servers with higher amplification.

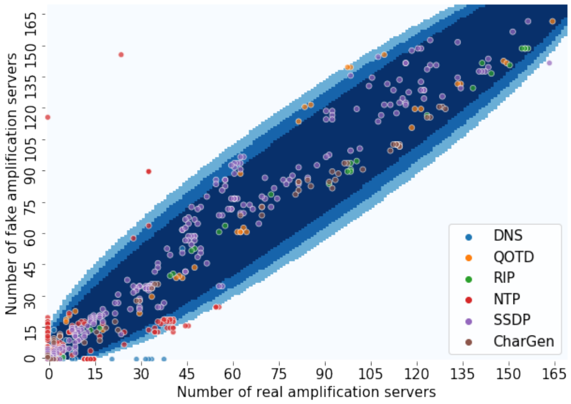


Figure 8: Attacks are largely indifferent about whether a service is real as long as it provides amplification.

Adversaries generally do not care whether a system is a honeypot. To investigate whether attackers only collect machines in an automated fashion or apply some level of post-processing or vetting of the discovered services, we allocated a new set of IPs to honeypots that would, as part of the response, clearly identify themselves as honeypots that are rate-limited. While this would require a human to look at the responses, we also made these honeypots behave non-protocol compliant, which meant that machine-based post-processing (if being done) should weed them out. To rule out a potential confounding based on the amplification as discussed in the previous section, the amplification ratio was identical to those of our real servers. Servers running fake responders were exclusively running fake services to prevent confounding.

Figure 8 shows the distribution of attacks between real and fake servers in an identical setup as figure 7, and we see that the bulk of the attackers is indifferent about the responses of the system. Only in DNS and NTP do we observe a handful of attacks being conducted without any fake services. However, in these rare situations, the adversaries did not show sensitivity to the honeypot identification string but expected a particular response to their query and would drop deviating servers from the list. Contrary to what one might think, adversaries do not make any effort to check the plausibility of servers, neither manually nor by scripts. Curiously, we also observed a series of NTP attacks that were *exclusively* using fake systems, the result of an adversary using an incorrect *Monlist* request packet that would be dropped by a regular NTP server implementation as an incorrect query.

Implausible amplifier setups do not matter to any adversary. Krämer et al. hypothesize that hosting multiple services on the same IP address might influence the behavior of an attacker when interacting with the device [17]. To establish whether such a difference exists, our honeypot system deployed 120 servers (24 per protocol) that are solely running one service. In contrast, the rest of the honeypots run all protocols in parallel, which is an implausible setup on a regular server. We do not find any statistically significant differences between those groups. Although anecdotal, we observe adversaries performing multi-vector attacks gladly using all services located at the same honeypot for their attack, even though in practice, the shared uplink would limit traffic at the amplifier.

5.5 Testing

After locating candidate systems for abuse, an adversary might opt to test amplifiers whether they are heavily rate-limited or drop packets and thus not useful in an attack. Actors would not observe these behaviors when scanning using a single packet but need to test hosts actively. Additionally, as services might be moved to other IPs or be shut down, it would make sense for adversaries to frequently test if the amplifiers are still online to not waste reflection potential by sending packets to a server that does not respond.

Only a few attackers test whether a system is rate-limited before performing an attack. Evidently, operators do not want their servers being abused in performing attacks, as this unnecessarily drains one's resources, and the victim might blame the amplifier for the packet flood. Operators would therefore minimize the risk for attacks, for example, by establishing rate limits on requests towards a service. To establish the extent to which adversaries actively test servers before using them in an attack, our honeypot system contains 60 servers per protocol that are actively dropping one-third, one-half, or two-thirds of all outgoing packets. While all servers will amplify incoming traffic when an attacker crafts the correct packet, it is clear that the amplification potential decreases dramatically when two-thirds of the packets are dropped. To ensure these servers will be found by a scan and not bias the setup, the first packet in a connection will always receive a reply. With this setup, only attackers who have probed the system with more than a single scan will know these differences. Figure 9 shows the distribution of attacks across these services and the probability of these server choices occurring at random using the confidence intervals explained in the previous section. The image shows that overall packet loss seems of no concern to adversaries. Only in SSDP do we see attackers making an effort to steer clear of servers with packet loss. As there is no significant deviation in how many times servers were scanned, these choices have to be made deliberately.

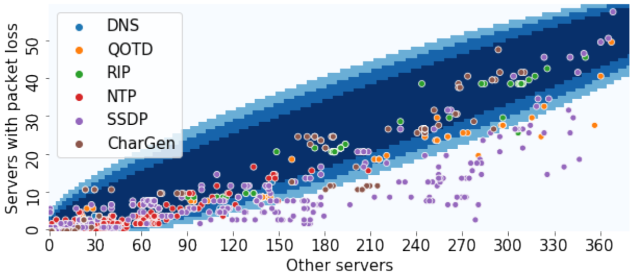


Figure 9: Usage of servers with packet loss in attacks, active selection only occurs in case of SSDP.

While many attacks using SSDP use a disproportionate amount of loss-free servers, we would expect adversaries to completely steer away from them when having the choice between different servers that perform better. However, almost no attacks are void of any lossy servers, raising the question of how adversaries make this selection. As the probability that packet loss will occur when contacting a host is in our case $1 - (1 - droprate)^{k-1}$ where k is the number of packets sent during one connection, some scans will not be enough to establish that some of our hosts never reply to a portion of the scans coming in. Assuming a scan rate of 5 packets per host, which we observe in 9% of scanning traffic, an adversary will have a 20% chance that packet loss is not observed when we drop $\frac{1}{3}$ of the packets, making the packet loss invisible to the adversary. Let us only consider the servers with $\frac{2}{3}$ packet loss, which would be visible 99% of the time when sending five packets towards this honeypot. We indeed find adversaries testing the infrastructure with 150 attacks in NTP and 96 attacks in SSDP that do not use any of these servers.

Adversaries do not care about server latency. Aside from dropping packets, also considerable packet delays might pose an obstacle for adversaries when launching attacks. We placed 60 servers that delayed the response by an extra delay of 500 ms to investigate a potential selection strategy for this. With regular service times below 1 ms and worst-case round-trip network delay from Europe to Asia being in the order of 300 ms, this delay should significantly stick out. A significant delay in sending responses could indicate a highly loaded or overloaded system, making it less attractive for an adversary as it might get overwhelmed during the attack. We do, however, not find any statistically significant differences that adversaries prefer non-delaying servers over these servers.

Sophisticated adversaries do not blindly trust their amplifier lists. Over time, results of the scans performed to find open amplifiers might not be accurate anymore due to the reconfiguration of servers or even the decommissioning of the machines. To not waste effort on non-responsive machines, an adversary should repeat scans and curate the list of used servers. To investigate the extent to which adversaries rescan the hosts to verify their status, we continued to collect all network traffic towards our honeypots after we concluded the active experimentation. In this passive phase, all traffic was sink-holed, and no replies were sent out anymore. Figure

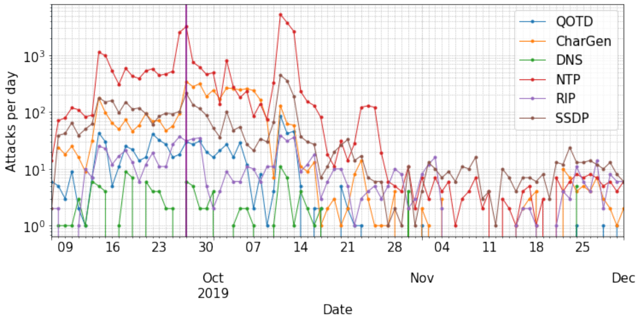


Figure 10: Incoming attacks during active replies and subsequent shutdown on September 27th.

10 shows the attack rate while we actively responded to requests and the rate after the shutdown of the amplifiers. In the baseline period before the amplifiers actively respond to requests, no attacks were measured. While we verified during the baseline establishment that the IPs were unknown to attackers and received no attack requests, reverting to this state and nullifying their use in amplification attacks did not cause the number of attacks conducted using the servers to go down drastically. On the contrary, attacks continued at roughly the same pace for all protocols for more than two weeks. Even after two months, attacks were still conducted using our infrastructure, which had been disabled for longer than it was ever enabled. We believe there can be two explanations for this behavior: (1) The attacks conducted with our servers originates from the same actors who have created lists of vulnerable servers and do not update these frequently, or (2) our servers have been listed in publicly available lists of amplification servers, and the people conducting attacks blindly trust these lists to be accurate and up to date. While we have not been able to find any notion of our servers being on amplifier lists, these lists might be shared inside closed communities. While many attacks are conducted after the system does not respond anymore, no large attacks are being conducted using the amplification servers. Additionally, after the servers are taken offline, the attacks conducted are only rarely using solely high amplification servers. The adversaries that are capable of performing large and lengthy DDoS attacks are thus updating their lists.

5.6 Execution

While a DDoS attack is a simple concept in which a victim's resources are drained, the ways in which these attacks are executed are diverse in practice. In this section, we will show various attack techniques used by adversaries captured in our honeypots.

Attack durations and modes differ between amplification protocols and targets. Amplification attacks sent to our honeypots had an average duration of 394 seconds, almost a factor of 5 longer than for TCP SYN floods, the other major type of DDoS attack vector [4, 11, 13]. The duration of attacks observed in our work falls between observed median durations in related work, which report a median of 658 [32] and 255 seconds [13]. Figure 11 shows how the durations even significantly differ per protocol, with QOTD having a median attack duration three times as high as NTP. In QOTD, we also see modes at 400 and 500 seconds that are not present in other attacks. The mode at 300 seconds is visible in QOTD, CharGen, NTP, and RIP but does not show in SSDP. The presence of attack durations and modes has earlier been identified by [4] for TCP SYN flooding attacks in 2017, where typical durations of 30 and 60 seconds were traced back to test attacks by booter services. For amplification DDoS attacks, these modes have been identified by [32]. In our study, we find that both the attacks take significantly longer and common modes are significantly higher opposed to TCP SYN flooding attacks, but both are lower compared to [32].

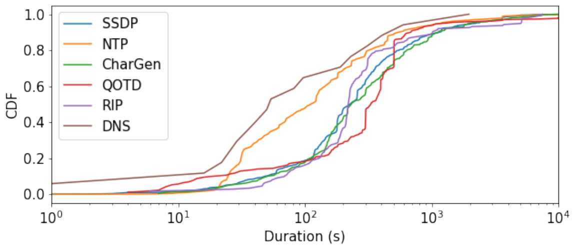


Figure 11: CDF for the duration of attacks per protocol.

Table 6: Source port usage of traffic coming into our system.

Protocol	Attacks	1. Single (%)	2. Static (%)	3. Rand (%)
CharGen	471	1	95	4
DNS	55	11	35	55
NTP	11,130	2	47	51
QOTD	264	6	94	0
RIP	184	2	98	1
SSDP	1375	30	68	1

Given the differences between protocols and the way attacks are run, it is natural to wonder whether these are used for different purposes. When we look at attacks on two different types of services, those on game hosts and web servers based on their open ports using Shodan, we find that the type of target influences the used attack vector and the duration of the attack. The attacks on game servers are solely conducted with NTP and SSDP, whereas attacks across all protocols attack web hosts. In terms of duration, game hosting servers are attacked more rigorously than web servers, with an average attack duration of 12 minutes. In contrast, web servers only deal with attacks that, on average, last a bit more than 5 minutes. Between the two groups, there is no difference in the selection of servers to be used in the attacks.

Attacked ports cannot be used in the protection against attacks - except when attackers are targeting TCP ports instead of UDP. Various controls exist to filter DDoS attacks [27]. One of these techniques is to filter based on ports. Like most other services, our honeypots run their services on default UDP ports, which would not provide an angle for the victim to filter out the amplified data. On the other hand, the attacker's spoofed requests need to originate from a source port, to which (at the victim's IP) the response will be sent. We distinguish three cases for the ports used to request data: (1) All requests towards honeypots are made from one port, indicating a crafted, injected packet. (2) The requests are made using different ports per honeypot. (3) For every request,

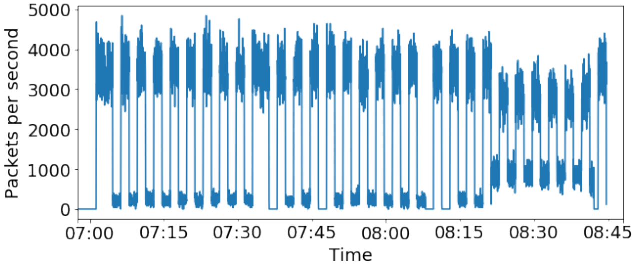


Figure 12: A pulsing attack using 40 honeypots.

the attacker generates a random port number, evenly distributing the traffic over all ports. All three cases show distinct implementation choices an attacker can make and could avoid packet-based filters that are placed as a countermeasure. Table 6 shows the distribution of source port usage in attacks, and we see a large part is implementing the frames using a static port per honeypot. While randomizing source ports is only slightly more complicated in implementation, most adversaries do not seem to choose this route, even though it can be more robust against packet level filtering [27]. Only in NTP and DNS a part of attacks is performed while randomizing ports across the entire port range.

This however raises another important operational aspect. An amplification attack towards a single port could be trivially blocked by packet filtering unless the filtering happens on the victim's premises. The flood is sufficient to congest the connection from the victim to the Internet. However, if the attack targets a service already running on the victim, the attack might be hard to distinguish from legitimate traffic as found by [13], who identify that most of the attacks are aimed at ports connected to online gaming. We do not find many attacks targeted at gaming ports and instead find that adversaries fail to take into account that protocols such as HTTP would run on *TCP* port 80 or 443 and not *UDP* port 80, which would not be whitelisted in a firewall.[1] Thus, the deliberate targeting of select ports would not have any more benefit than a packet flood towards a random destination port. From the 650 single port attacks, 364 send their attack towards port 80 on the victim, while another 20 target port 443. The only attacks aimed at a UDP service were 16 attacks on DNS port 53.

Attack pulses maximize the attack efficiency while minimizing the cost for the attacker. After a packet flood has disrupted a service, it might take a moment after the end of the attack for services to recover and clients to reconnect. This means that even after the flood itself has stopped, the effects might still be ongoing which can be used by an adversary to minimize the bandwidth required for a successful attack. We observed that several adversaries do not launch their attacks as continuous floods but instead adopt a pulsing behavior of floods followed by brief inactive periods. This would have the advantage of reducing the risk of "burning" the amplifiers due to continuous usage while still meeting the objective. Figure 12 shows this behavior for one such attack, with multiple waves of flooding. When we observe pulses, their behavior was highly coordinated, as we observe all honeypots used in the attack simultaneously receiving traffic and simultaneously stop receiving requests. We would expect sophisticated actors to perform these types of attacks, and we indeed find for all pulse attacks that they are exclusively conducted using only high amplification devices.

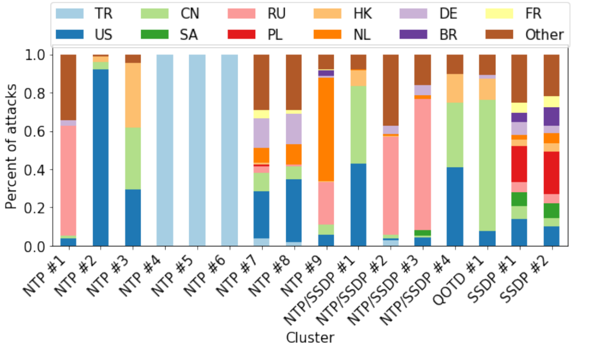


Figure 13: Attacks from clusters performing more than 100 attacks during our study.

6 DDOS ATTACK CAMPAIGNS

One of the most important arguments for investigating attacks along the DDoS attack chain presented in this paper is that adversaries will likely reuse components from a previous attack. Instead of finding infrastructure, testing it, and weaponizing every time a victim is being targeted, attackers would instead use the same set of servers, attack packets, etc., to perform multiple attacks.

Actors have a geographical focus. When multiple attacks use the same servers in our system, the probability of this occurring due to random selection of two different entities is negligible given the number of active honeypots. We use this as the first feature to cluster attacks, where we cluster attacks if these are using the same set of our amplifiers. We do not link attack instances if they have all honeypots of a particular kind in common – such as all NTP high amplification services –, as specific preferences of adversaries might result in multiple actors sharing these edge cases. From the 720,995 attack flows, we obtain 749 clusters of attacks, of which 351 attacks more than once. To verify if the same actor indeed performs the attacks inside a cluster, we use three criteria: (1) The attacks are performed using the same request packet. (2) The attacks use the same strategy in picking the source port. (3) The flows inside the cluster have the same characteristics in, for example, pulsing and intensity. Amazingly, all of the 351 clusters attacking more than once match these criteria and will therefore be considered in this section as being valid clusters. Figure 13 shows the distribution of targeted countries by the clusters attacking more than 100 separate IP addresses. We see actors show a significant degree of geographical specialization. While previous work locates the bulk of DDoS activity to the US and China [13], we find it much more nuanced from an actor perspective, which focuses and specializes on victims in a handful of countries. This shows that attacks are not randomly carried out but are aimed at an objective from the adversary; for example, NTP clusters 4, 5, and 6, which solely target

[1] A major service running on UDP would be QUIC on UDP port 80, however none of the hosts attacked on this port ran this protocol.

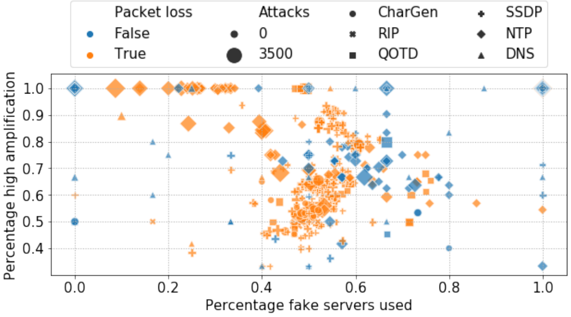


Figure 14: Scatterplot of attack clusters.

Turkey identify that the objective of these adversaries is different from other clusters. As discussed in section 5.3 attacks are often performed during the afternoon or evening at the location of the victim. As most attackers are biased towards certain countries, we would expect these actors to have distinct attack timings as well. From the 86 (25%) clusters performing attacks over multiple days, we indeed observe 68 (19%) showing a diurnal pattern, where the number of attacks drops significantly during certain hours. The remaining 18 (5%) clusters are not significantly lowering their activity in a day/night rhythm and might be an indication of automated booter services that are used by people in multiple countries.

Sophisticated actors reuse the same servers often. As all the individual attacks within a cluster use the same attack packet, with identical traffic dynamics and from the same (randomly chosen) source port, they can be attributed to the same setup by the adversary. Within clusters, all attacks are performed using the same command. For an attacker, this makes sense, as the attacker knows that the request sent to the server is amplified based on previous scan results. When we look at the volume of the attacks originating from a cluster and the way they select our honeypot infrastructure, we find an astonishing rich spectrum of behaviors. Figure 14 shows for each of the clusters the volume of attacks through the size of the market and on the x- and the y-axis the percentage of fake and high amplification honeypots that were used during the attacks. As clusters also showed the behavior of selecting the same set of amplification servers, these data points are not averages but static over time. Much of the clusters congregate in the middle of the graph, distributed around the expected values for the ratio of fake servers and high amplification devices an attacker would use based on a random selection, such as an indiscriminate scan of the Internet. Deviations to the left mean that the adversary is performing some post-processing to weed out suspicious devices, shifting to the top an active selection to maximize the attack volume. As we see in the graph, the more attacks an attacker performs, the stronger it will pre-select for high amplification servers and discard low-utility devices. The more attacks an attacker performs, the higher the likelihood to identify and discard unusually behaving servers from the attack list. Thus, the most extensive campaigns tend to be run by the more sophisticated adversaries. We indeed find a significant positive correlation ($p < .05$) between the number of attacks in a cluster and the amplification ratio, as well as a significant negative correlation ($p < .01$) with the number of fake servers used. There are however groups performing a large number of attacks using only fake high amplification servers. On inspection of these clusters, we find that these attackers are not using the correct protocol specification and do not receive replies from the real servers. However, the adversaries do select the servers that provide the highest amplification. Additionally, we find a significant negative correlation ($p < .01$) between the fake servers used in an attack and the amplification ratio, hinting that advanced adversaries are actively filtering out these obvious honeypots. This behavior is curiously only present in NTP and SSDP abuse, while adversaries targeting DNS, RIP, or QOTD do not seem to care how they achieve their objective. Surprisingly, the adversaries that perform highly targeted scans towards single cloud zones are not the sophisticated attackers as the adversaries actively removing poorly behaving amplification servers are also evaluating the entire ecosystem and use amplification servers hosted in multiple cloud locations.

Sophisticated attackers do not show advanced behavior across the entire spectrum but innovate only selectively. To better understand how adversaries work, the steps they take can be separated into different phases. Table 7 shows a mapping for seven clusters performing attacks using our amplification honeypots on the model shown in figure 1. Looking at the entire chain of activities rather than only the separate phases, we find that clusters performing longer-lasting attacks are more likely to use servers with a higher amplification factor, indicating that attackers capable of performing long-lasting DDoS attacks are making more effort to ensure their operation is successful. But there is no relation between the duration or amplification factors in attacks and whether or not the adversaries use our obvious honeypots. However, we find evidence that adversaries capable of performing large attacks are curating the set of amplifiers, whereas adversaries that only perform lower volume and duration attacks are using their amplifier lists for weeks without checking whether these systems are still online. Clusters performing long-lasting attacks with high amplification such as cluster #1 and #7, which perform attacks lasting over 2 hours, are continuously curating and updating their amplifier lists by continuously scanning the servers used in the attacks and are thus not active anymore after the active phase of our experiment concludes, as their continuous curation prevents them from using a non-responsive server in an attack. While we have seen many attacks being performed weeks after the infrastructure did not respond to attacks anymore, these attacks are almost exclusively under 10 minutes and do not care about the amplification ratio provided by a server. The actors capable of performing high-volume and high-duration attacks are thus test servers for their amplification ratio and weed out unresponsive servers before performing an attack.

7 DISCUSSION AND LIMITATIONS

Investigating the Tactics, Techniques, and Procedures used in DDoS attacks may change the way we organize our defenses. By looking at the entire DDoS attack chain with the model presented in this paper, we show that the ecosystem of DDoS amplification is more than a collection of individual attacks but that it is possible to correlate and link them based on solid behavioral features. The results show the presence of hundreds of actors with different sophistication, preferences for attack vectors and victims, and unique modus

Table 7: Groups of attacks modeled on DDoS phases and activities.

	Capability Development	**Infrastructure Reconnaissance**		**Target Reconnaissance**		**Weaponization**		**Testing**		**Execution**	
	Protocol	*Sources*	*Countries*	*Targets*	*Countries*	*Packet Content*	*Amp.*	*Fake serv.*	*Curation*	*Attacks*	*Max Duration*
1	CharGen	19	9	6	5	a	17.97	Yes	Within a day of every attack	9	7,891 seconds
2	DNS	2	2	2	2	Query: ufpa.br	11.60	No	Attacks within 2 hours of scan	2	283 seconds
3	NTP	5	2	8	8	monlist	15.82	No	Only initial scan	10	628 seconds
4	NTP	4	3	3	3	monlist + 10 bytes	5.40	No	Scan every attack	7	20 seconds
5	QOTD	2	2	5	3	single byte	16.12	Yes	One scan 2 days before attacks	5	3,606 seconds
6	QOTD	2	1	24	6	"Bigbo"	14.77	Yes	One scan, attacks for 10 days	26	3,640 seconds
7	QOTD	9	3	23	8	"getstatus"	17,34	No	Continuous scan	27	7,265 seconds

operandi. This suggests that instead of merely enduring and mitigating DDoS attacks, some degree of attribution for attacks is possible.

Although a significant share of DDoS appears as ‘dumb”, ‘script-kiddie”-driven activity, our study reveals the presence of sophisticated actors who investigate, inspect and measure services on the Internet and perform active selection to maximize their return-on-investment. This means that future research on DDoS and the Internet threat landscape has to account for such adversarial behavior and needs to advance in terms of techniques, as obvious decoys or servers not participating in test attacks that are currently being used in research would be discarded by sophisticated actors. This would result in a drastically biased perspective on the DDoS ecosystem. By using actual services and through momentary but ethical participation when rallied, we were the first to show this unexplored dimension of the threat landscape and demonstrate that ethical investigation of this behavior is feasible.

In this study, we deployed an order of magnitude more honeypots than any previous works, first to allow for systematic and statistically significant testing of different scenarios and configurations, but second because the heterogeneity of the ecosystem would otherwise lead to biased results from the overrepresentation of select actor groups. We also identify the need for rigid methods in separating malicious scanning activity from research scan activity to avoid large measurement biases. Using the technique from [27], we can show that 3-10 times as many honeypots are necessary to capture the richness of the threat landscape as were deployed in previous work, as shown in figure 4. To deal with such temporal biases and ecosystem heterogeneity, we advocate that future studies should deploy honeypots in the hundreds, not dozens, to be effective.

Accomplishing this deployment scale also has drawbacks, and in this study, we limit our experiments for monetary reasons to a significantly shorter online duration. While smaller honeypot studies operated for several months or even years [17, 27, 32], our system was deployed for three months, from which actively responding for three weeks. This limited-time might influence the number of adversaries using our systems and would not show those actors who enter the ecosystem only at longer spaced intervals. Finally, our study deployed honeypots within the IP ranges of cloud providers. Although many organizations are now moving their infrastructure to the cloud and adversaries should hence look for abusable systems there, it is possible that some actor groups specifically focus on network ranges owned by enterprises, which we would miss in such a study.

8 FUTURE WORK

Despite the heterogeneity of the overall ecosystem in the spectrum from script-kiddie to sophisticated actor, our results show that each attack – even though it might hit hundreds of amplifiers – is remarkably similar. This is understandable as the perpetrators are optimizing for the economies of scale, but it also opens up angles for mitigation. By identifying, automatically recognizing common tactics and techniques, and distributing them to defenders for detection [9], much of current distributed amplification DDoS could be mitigated. Our findings show that recent proposals such as BGP flowspec could be highly effective in practice by filtering incoming amplification requests and not only merely the resulting packet floods towards the victim. Given this similarity and the fact that essentially every perpetrator in our study horizontally scaled out to abuse many honeypots in the same attack, also collaborative identification of malicious flows – either at the level of networks or resolvers – could provide an angle towards reducing this attack vector if servers providing amplification would run a service that could not be disabled otherwise. Such an approach creates again interesting but solvable research problems to remediate the resulting privacy concerns. However, it could be a viable and scalable solution as incentives and costs are aligned for the operators of the abused services.

In this study, we focused on attacks measured by our deployed honeypots and cannot measure the attacks at the victim’s side. To understand the behavior of the victims and the actual attack sizes, future works should focus on a collaboration with victims, cloud service providers, or DDoS defense companies as an additional vantage point. This would allow for further analysis of attack “pulses” and give insights into the total number of amplifiers used in attacks.

9 CONCLUSIONS

We have analyzed adversarial techniques for amplification DDoS attacks using 549 honeypots running six amplification protocols that were rallied to 13,479 attacks over three weeks. We show that adversaries tend to select the servers with the highest amplification potential and that adversaries create tests to identify the servers that would create the largest impact. Additionally, we show the existence of “memory” of amplification services, which are abused long after the service is taken down. While the bulk of adversaries pursue simple techniques at a high level, we show the presence of highly advanced attacker groups, selectively picking servers and tactics to perform their attacks with the most firepower.

REFERENCES

[1] https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks, accessed at 2021-05-05.

[2] https://www.speedtest.net/global-index, accessed at 2021-05-05.

[3] Anagnostopoulos, M., Kambourakis, G., Kopanos, P., Louloudakis, G., and Gritzalis, S. DNS amplification attack revisited. *Computers & Security 39* (2013).

[4] Blenn, N., Ghiette, V., and Doerr, C. Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter. In *International Conference on Availability, Reliability and Security (ARES)* (2017).

[5] Büscher, A., and Holz, T. Tracking DDoS attacks: Insights into the business of disrupting the web. In *5th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 12)* (2012).

[6] 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 *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014).

[7] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 542–553.

[8] Fachkha, C., Bou-Harb, E., and Debbabi, M. Fingerprinting internet DNS amplification DDoS activities. In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)* (2014), IEEE.

[9] Griffioen, H., Booij, T. M., and Doerr, C. Quality evaluation of cyber threat intelligence feeds. In *International Conference on Applied Cryptography and Network Security (ACNS)* (2020).

[10] Griffioen, H., and Doerr, C. Taxonomy and adversarial strategies of random subdomain attacks. *International Conference on New Technologies, Mobility and Security* (2019).

[11] Griffioen, H., and Doerr, C. Quantifying TCP SYN DDoS Resilience: A Longitudinal Study of Internet Services. In *IFIP Networking* (2020).

[12] Hutchings, A., and Clayton, R. Exploring the provision of online booter services. *Deviant Behavior 37*, 10 (2016).

[13] Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A., and Dainotti, A. Millions of targets under attack: a macroscopic characterization of the DoS ecosystem. In *Proceedings of the 2017 Internet Measurement Conference* (2017).

[14] Karami, M., and McCoy, D. Rent to pwn: Analyzing commodity booter DDoS services. *Usenix login 38* (2013).

[15] Karami, M., Park, Y., and McCoy, D. Stress testing the booters: Understanding and undermining the business of DDoS services. In *Proceedings of the 25th International Conference on World Wide Web* (2016).

[16] Kopp, D., Wichtlhuber, M., Poese, I., Santanna, J., Hohlfeld, O., and Dietzel, C. DDoS Hide & Seek: On the Effectiveness of a Booter Services Takedown. In *Proceedings of the Internet Measurement Conference* (2019).

[17] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K., and Rossow, C. Amppot: Monitoring and defending against amplification DDoS attacks. In *International Symposium on Recent Advances in Intrusion Detection* (2015), Springer.

[18] Krupp, J., Backes, M., and Rossow, C. Identifying the scan and attack infrastructures behind amplification DDoS attacks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016).

[19] Krupp, J., Karami, M., Rossow, C., McCoy, D., and Backes, M. Linking amplification DDoS attacks to booter services. In *International Symposium on Research in Attacks, Intrusions, and Defenses* (2017), Springer.

[20] Kührer, M., Hupperich, T., Rossow, C., and Holz, T. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *23rd USENIX Security Symposium (USENIX Security 14)* (2014).

[21] MacFarland, D. C., Shue, C. A., and Kalafut, A. J. Characterizing optimal DNS amplification attacks and effective mitigation. In *International Conference on Passive and Active Network Measurement* (2015), Springer.

[22] MacFarland, D. C., Shue, C. A., and Kalafut, A. J. The best bang for the byte: Characterizing the potential of DNS amplification attacks. *Computer Networks* (2017).

[23] Matherly, J. Complete guide to shodan. *Shodan, LLC (2016-02-25) 1* (2015).

[24] Paxson, V. An analysis of using reflectors for distributed denial-of-service attacks. *ACM SIGCOMM Computer Communication Review 31*, 3 (2001).

[25] Prince, M. Technical details behind a 400Gbps NTP amplification DDoS attack. *Cloudflare, Inc 13* (2014).

[26] Richter, P., and Berger, A. Scanning the scanners: Sensing the Internet from a massively distributed network telescope. In *Proceedings of the Internet Measurement Conference* (2019).

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

[28] Rudman, L., and Irwin, B. Characterization and analysis of NTP amplification based DDoS attacks. In *2015 Information Security for South Africa (ISSA)* (2015), IEEE.

[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 *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (2015), IEEE.

[30] Sassani, B. A., Abarro, C., Pitton, I., Young, C., and Mehdipour, F. Analysis of NTP DRDoS attacks' performance effects and mitigation techniques. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)* (2016), IEEE.

[31] Sun, Z., Liu, B., and Hu, C. Method for effectively detecting and defending domain name server (DNS) amplification attacks, 2011.

[32] Thomas, D. R., Clayton, R., and Beresford, A. R. 1000 days of UDP amplification DDoS attacks. In *2017 APWG Symposium on Electronic Crime Research (eCrime)* (2017), IEEE.

[33] Vetterl, A., and Clayton, R. Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)* (2018).

A AMPLIFICATION FACTORS

This appendix contains the content of the experiment groups for all the six different services. For each of the services we include the Bandwidth Amplification Factor (BAF) introduced by Rossow [27] as well as the contents of the packets sent by the honeypots.

A.1 RIPv1

The Routing Information Protocol responds on correctly formatted packets. The BAF is calculated using the most frequently packet.
Real small - Responds with a small routing table consisting of 4 entries. The response has a size of 84 bytes and a BAF of 3.5.
Real large - Responds with a larger routing table consisting of 26 entries. The response has a size of 524 bytes and a BAF of 21.8.
Fake small - Responds with a fixed message of 88 bytes and a BAF of 3.7 that does not contain the header formats and cannot be parsed by a RIP protocol format parser: *THIS IS A HONEYPOT. YOUR IP IS LOGGED. DO NOT USE THIS. YOU NOW PARTICIPATE IN RESEARCH!*
Fake large - Responses are non-parsable by RIP parsers and are larger, consisting of 413 bytes and have a BAF of 17.2: *This is a honeypot! This is not a real server! You should not use this RIP server! Your IP is logged. We use this server to investigate who connects to it and what happens with anyone that is connecting! So it is really best you do not use this server! And if you do, please leave a cool message, since we log all the packets anyway ;) - Thanks a lot for participating in our research - Project Honeypot Research.*

A.2 CharGen

The Character Generation protocol discards any received input. For any request, it will return a random number of characters. The maximum amplification is thus achieved by sending a single byte.
Real small - This group responds with a string that is rotated by one byte after every received request. The size of the response is 94 bytes and has a BAF of 94: *!"#$%&'()*+,-./0123456789:;<=>?@ABCDEF GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg.*
Real large - Uses the same string as the small server, but rotates it 19 times. The data consists of 1406 bytes and the BAF is 1406.
Fake small - Sends a 94 byte static string indicating that the server is a honeypot with a BAF of 94: *THIS IS A HONEYPOT. YOUR IP IS LOGGED. DO NOT USE THIS. YOU ARE NOW PARTICIPATING IN RESEARCH!.*
Fake large - Returns a message with 1450 bytes and a BAF of 1450: *This is a honeypot system! This is not a real CHARGEN server! You should not be using this CHARGEN server! Your IP internet address is logged in a database. We use this server to investigate who connects*

to it and what happens with these quotes! So it is really best you do not use this server! And if you do, please leave a cool message, since we log all the packets anyway ;) We will publish any misuse of this service on the internet and detect your IP internet address. ***(Repeated 3x)*** *| Project Honeypot Research.*

A.3 QotD

Similar to CharGen, the QotD protocol does not care about user input and the response can be triggered by a single byte.
Real small - Returns a random quote with a size between 45-50 bytes. The BAF varies between 45-50. An example quote is for example: *To infinity.... and beyond! - Toy Story.*
Real large - Responds with large quotes averaging 1450 bytes and a BAF of 1450. Quotes are selections of Lorem Ipsum[2].
Fake small - Responds with a static string of 53 bytes and a BAF of 53 that indicates that the system is a honeypot: *This is a honeypot attack detector - Researchers.*
Fake large - Contains a single static string of 1437 bytes and a BAF of 1437: *This is a honeypot system! This is not a real QOTD server! You should not be using this QOTD server! Your IP internet address is logged in a database. We use this server to investigate who connects to it ans what happens with these quotes! So it is really best you do not use this server! And if you do, please leave a cool message, since we log all the packets anyway ;) We will publish any misuse of this service on the internet and detect your IP internet address.* ***(Repeated 3x)*** *- Project Honeypot Research*

A.4 SSDP

SSDP packets will only be sent on valid requests, except for the fake servers. While those can thus be triggered by sending only 1 byte, we calculate the BAF using the same packet as in the real case.
Real small - Responds one of the smallest possible responses on an M-SEARCH request with a size of 272 bytes and a BAF of 2.3. Fields in brackets are set with the correct values at time of the request:
*NOTIFY * HTTP/1.1*
HOST: 239.255.255.250:1900
DATE: {time}
CACHE-CONTROL: max-age = 1800
LOCATION: http://{server_ip}/rootDesc.xml
SERVER: UPnP/1.0
NTS: ssdp:alive
NT: upnp:rootdevice
USN: uuid:b4ca5004c5334bf4883046f2ee3e871a::upnp:rootdevice
Real large - Returns a response of 430 bytes with a BAF of 3.7. Fields in brackets are set with the correct values at time of the request: *NOTIFY * HTTP/1.1*
HOST: 239.255.255.250:1900
DATE: time
CACHE-CONTROL: max-age=60
LOCATION: http://server_ip:5000/rootDesc.xml
SERVER: OpenWRT/OpenWrt UPnP/1.1 MiniUPnPd/1.9
NT: upnp:rootdevice
USN: uuid:822db064-5a71-4375-ba79-20b582cd9309::upnp:rootdevice
NTS: ssdp:alive
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
01-NLS: timestamp
BOOTID.UPNP.ORG: timestamp
CONFIGID.UPNP.ORG: 1337
Fake small - Responds with a message of 277 bytes and a BAF of 2.4 that is not parse-able by the protocol: *This is a honeypot! This is not a real server! You should not use this SSDP server! Your IP is logged. We use this server to investigate who connects to it and what happens with these quotes! So it is really best you do not use this server! - Project Honeypot Research*
Fake large - Responds with a message of 429 bytes and a BAF of 3.7 that is not parse-able by the protocol: *This is a honeypot system! This is not a real SSDP server! You should not use this SSDP server! Simple Service Discovery Protocol part of UPnP Your IP network address is logged. We use this server to investigate who connects to it and what commands are transmitted to this server! So it is really best you do not use this server! And if you do, please leave a nice message ;) With Regards, - Project Honeypot Research*

A.5 NTP

The network time protocol is running in virtual machines that run the NTP software NTPD. The response sizes sent by the honeypots can vary based on the state of the NTP Deamon.
Real small - In the small server we are running an NTP server where the NTP amplification vulnerability is fixed. This means that there is no amplification factor for the adversary.
Real large - Uses a version of the NTP software that contains an amplification vulnerability in the *monlist* command. The BAF is set to at least 46 but can increase depending on the system state.
Fake small - The fake small server sends a non-expected answer to a *monlist* request, but does not provide amplification potential.
Fake large - Responds with an unexpected 347 byte long plain text message on a *monlist* request, providing a BAF of 43.4: *This is a honeypot! This is not a real server! You should not use this NTPD server! Your IP is logged. We use this server to investigate who connects to it and what happens with these quotes! So it is really best you do not use this server! And if you do, please leave a cool message, since we log all the packets anyway. Honeypot Research*

A.6 DNS

Similar to NTP, the Domain Name Servers are running on virtual machines, and queries are handled by the BIND DNS server.
Real small - Does not do recursion and only responds messages with a BAF of 1.6.
Real large - Enables recursion and thus responds to any query (regular open resolver). The minimum BAF is 1.6 but this increases based on the request to the resolver.
Fake small - Emulates a DNS server that only resolves queries for the *example.com* zone with a BAF of 1.6.
Fake large - Does not parse the incoming packet but responds with a 352 byte long message providing a BAF of 6.8 on a normal DNS query on the domain of Google: *This is a honeypot! This is not a real server! You should not use this QOTD server! Your IP is logged. We use this server to investigate who connects to it and what happens with these quotes! So it is really best you do not use this server! And if you do, please leave a cool message, since we log all the packets anyway ;) | Project Honeypot Research.*

[2] https://lipsum.com