1000 days of UDP amplification DDoS attacks

Daniel R. Thomas Richard Clayton Alastair R. Beresford

Computer Laboratory
University of Cambridge
Firstname.Lastname@cl.cam.ac.uk

***Abstract*—Distributed Denial of Service (DDoS) attacks employing reflected UDP amplification are regularly used to disrupt networks and systems. The amplification allows one rented server to generate significant volumes of data, while the reflection hides the identity of the attacker. Consequently this is an attractive, low risk, strategy for criminals bent on vandalism and extortion. To measure the uptake of this strategy we analyse the results of running a network of honeypot UDP reflectors (median size 65 nodes) from July 2014 onwards. We explore the life cycle of attacks that use our reflectors, from the scanning phase used to detect our honeypot machines, through to their use in attacks. We see a median of 1 450 malicious scanners per day across all UDP protocols, and have recorded details of 5.18 million subsequent attacks involving in excess of 3.31 trillion packets. Using a capture-recapture statistical technique, we estimate that our reflectors can see between 85.1% and 96.6% of UDP reflection attacks over our measurement period.**

I. Introduction

Denial of service is a generic term for attacks on computer systems that make them unavailable to their users. A common attack is to flood network links with specious traffic so that, in dealing with the excess, legitimate traffic is discarded. In a Distributed Denial of Service (DDoS) attack the abusive traffic arrives from many different computers at the same time, with each making a relatively small contribution to the attack.

In this paper we study Reflective UDP Amplification Attacks, a specific form of DDoS. UDP request packets are sent to a server, but the source IP address is spoofed to be that of the victim who receives the *reflected* replies. When the response is bigger than the request there is *amplification* and attackers can generate attack flows of many tens of Gb/s whilst themselves sending only 1Gb/s of traffic – something that is cheap and easy to arrange.

These attacks have, on occasion, been extremely disruptive and they are the subject of US CERT Alert TA14-017A, originally issued in January 2014 but revised in the light of ongoing activity with information about new types of attack in April 2016 [39].

Our approach to measuring this type of DDoS has been to deploy a small number of honeypot machines that mimic servers for commonly exploited UDP protocols. The attackers *scan* the Internet looking for reflectors and rope in our machines when they attack a victim. This allows us to estimate how many reflective UDP attacks occur each day and gives us an insight into the identity of the victims.

Our results are dominated by *booters / stressers* – websites offering DDoS for (very low-cost) hire used mainly by online games players to attack other online games players – but we see other types of victimisation as well.

We start by discussing earlier work on DDoS and explain our data collection system along with details of the scans and attacks we can measure. We also discuss the ethical issues we have addressed as we deployed our data collection system. We present the results of 1010 days of observations and the changes we have seen over time since July 2014. We explain how statistical techniques allow us to estimate not only how many reflective UDP amplification DDoS attacks occur each day, but also to estimate the number of potential attackers.

II. Related work

The first account of the use of DNS for reflective UDP amplification attacks (with a predicted amplification factor of 20) was in 1999 [34]. However, this method of attack did not become popular until much later [32]. In this paper we find that, since 2014, DNS reflective UDP amplification attacks are commonplace, with a median of 1 930 per day.

Paxson [26] gives an early account (2001) of the use of reflectors in DDoS attacks – but his focus is on the difficulties this poses for tracing back the attacker and there is no mention of amplification. In 2004 Mirkovic and Reiher [19] include reflection in their DDoS taxonomy but only mention smurf attacks [23] as an example.

Historically it has been believed that DDoS attacks were undertaken by botnets. For example, in 2012, Büscher and Holz, in their study of DirtJumper, stated that botnets were "the main mechanism" behind DDoS and describe traffic sent directly from the bots to victims [2]. However, in 2012 and 2013 numerous very high bandwidth DDoS attacks involved amplification, and since then reflective UDP amplification attacks have been the dominant form of DDoS, notwithstanding that since the summer of 2016 botnets, such as Mirai, which exploit insecure devices in the 'Internet of Things' have generated a number of very high bandwidth attacks [24].

When Rossow looked at real world DDoS in June 2013 he identified attacks involving the DNS, CHARGEN and SNMP protocols. He also surveyed which protocols, besides DNS, were capable of providing amplification [31]. This analysis inspired the measurement work described in this paper.

978-1-5386-2701-3/17/$31.00 ©2017 IEEE

While our work has focused on generic measurements of DDoS attacks, other work has focused on specific booter / stresser services: websites which provide DDoS for hire. In 2013, Karami and McCoy looked at the leaked database of a booter service, finding that it was responsible for around 900 attacks per day. Users playing online games purchased a dozen or so short attacks (10 minutes or less) per day, but some users bought attacks that went on for several days [11].

Santanna et al. analysed the attack traffic from 14 booters [32] and database dumps from 15 booters [33]. Karami et al. revisited the topic in 2016 [12] and, besides doing further analysis of leaked databases, they investigated how the requested DDoS attacks were performed. Generally, the front-end websites (through which payments are made and attacks requested) invoke back-end servers to generate the traffic towards the reflectors, spoofing the IP addresses of the victims so that they receive the amplified data. Recent analysis by Noroozian et al. determined that the bulk of the victims of UDP reflection attacks (62%) are users in access networks and only 26% are users of hosting providers [25].

Although compromised machines can be used as back-end servers it is more usual for machines to be hired for the purpose – a number of hosting providers advertise appropriate systems on underground forums. In the newer work, Karami et al. [12] also observed that there was considerable overlap between the amplifying reflectors used by different booters, but this overlap was higher for CHARGEN and NTP than for DNS and SSDP. Recent work by Krupp et al. [14] analyzed the TTL fields of spoofed packets sent during an attack and thereby determined the network location of the machines sending such packets. They found that a significant fraction of reflective UDP attacks are carried out by machines in the same network location as the scanner, suggesting that the common modus operandi is to use a single machine or collection of machines at one location rather than a larger botnet.

In 2015, Hutchings and Clayton contacted about 50 people operating booters and interviewed a quarter of them [9]. They found young males who believed that they were getting "easy money" and who had learnt their skills in online forums where "definitions favourable towards offending are shared".

Measuring DDoS attacks also has a long history. In 2001 Moore et al. [21] introduced *backscatter analysis* to enumerate DDoS attacks. They monitored packets arriving at unused IPv4 address space and identified responses that victims had generated when receiving traffic from forged source IP addresses. Their approach identified 12 805 attacks over 3 weeks. However, this approach fails to observe reflected attacks because in such a case the forged source IP address is that of the victim, rather than a random value. The same authors revised their paper in 2006 [22] to present 3 years of analysis with a peak DDoS rate of around 100 victim IPs per hour.

Czyz et al. [4] examined five datasets on NTP attacks dating from 2013 and 2014. They found that the volume of NTP attack traffic rose rapidly but fell after February 2014, possibly due to reconfiguration of NTP servers to prevent amplified reflection. Unfortunately there is no overlap between the period they considered and the period covered by the data described in this paper.

More recently, security companies have included DDoS in their reports about the state of Internet security. In particular, Akamai (who manufacture equipment for mitigating the effect of denial-of-service attacks) produce a quarterly report which describes numerous types of attack, but their data provides relative percentages rather than absolute figures [1].

The most similar work to this paper is by Krämer et al. [13] in which the authors use a /16 darknet to identify machines scanning for UDP reflectors. They use what they learnt to design and build a honeypot they called AmpPot. Their paper presents the results of running 21 AmpPot nodes between February and May 2015.

In contrast to prior work, our results are from a larger set of sensors, run over a considerably longer period of time. We look at both scanning and attacks and we present a robust method for using our results to estimate the total number of attacks that have occurred and illustrate how attacker strategies and techniques have evolved over time.

III. Data collection

We deployed our sensors on the public Internet. Sixteen of the sensors are on our own /28 subnet, with the rest mainly at low-cost hosting providers and a handful on consumer ISP connections. Since many scanners scan IP addresses at random, the sensors in the /28 subnet do not get scanned by an identical set of scanners and we observe different traffic volumes on different sensors. However, scanners that scan address space sequentially would discover all or none of these 16 sensors, so we do not consider them to be completely independent in later analysis. The diverse set of locations for the other sensors is intended to increase the likelihood that we respond to a scan of a limited part of the Internet, and to ensure that we still see activity even though some providers might drop traffic associated with DDoS events.

Over 1 010 days of data collection we have operated a mean of 59.7 sensors (median 65) with a 95% confidence interval of between 22 and 86 sensors. In the remainder of the paper we write "95% CI [22, 86]" as notation for this 95% confidence interval. A CDF of the number of sensors in operation per day is given in Appendix A-D.

Our sensors have two components that we developed from scratch: A program that emulates services exploited in reflective UDP amplification attacks, and a data collection module that records small numbers of raw packets and descriptive statistics of all UDP traffic. These components are described in the next two subsections.

A. Hopscotch: a UDP reflector

Hopscotch is a small (less than 3000 lines of C) program supporting multiple UDP services. Incoming traffic is checked for compliance with the relevant protocol, and if so, a valid response UDP packet is sent to the source IP of the request.

When we started our research in March 2014, we assessed which protocols were reported to be commonly abused for

Protocol	Port	Start	Reflection description
QOTD [29]	17	2014-03	An uplifting quotation of ∼250 ASCII characters is returned.
CHARGEN [28]	19	2014-03	A rolling pattern of 480 ASCII characters is returned.
DNS [20]	53	2014-03	The incoming packet is relayed to a real DNS server and the result returned. This avoids the complexity of implementing DNS and simplifies installation.
NTP [18]	123	2014-03	The incoming packet is relayed to a real NTP server and the result returned. If the incoming packet is a *monlist* request (mode 7, type 42) then the biggest possible fabricated response is sent, containing 600 IP addresses.
SSDP [38]	1900	2014-10	Only responds to `M-SEARCH` commands, sending a packet indicating the presence of a Wi-Fi device.
SQLMon [17]	1434	2015-02	Responds to relevant queries with a 580-byte list of fake database servers.
Portmap [35]	111	2016-01	Responds to v2 `DUMP` with ∼1K of data listing fake RPC services.
mDNS [3]	5353	2016-05	For simplicity, with no regard for the specification, treated just like DNS.

TABLE I: List of protocols supported by Hopscotch, their date of deployment, and key reflection capabilities.

DDoS and implemented: Quote Of The Day (QOTD), CHARGEN, DNS and NTP.

We later added further commonly abused protocols, including SSDP, MS SQL Monitor (SQLMon), Portmap and Multicast DNS (mDNS). Although some of these protocols were only intended to be used with multicast IPv4 addresses, many devices respond on standard IPv4 addresses, and hence so does Hopscotch.

Further details of the protocols supported by Hopscotch and details of the construction of the reflected packets can be found in Table I.

B. Sniffer: a packet collector

Sniffer is a small (around 1500 lines of C) program that uses libpcap [10] to inspect incoming UDP packets. The first fifteen raw packets received on the same port number for every source IP are recorded and the remaining packets are merely counted. Every five minutes a new raw packet file is created, and per port and per sender counts are written to a statistics file. These files are then delivered to a central server for analysis.

We decided not to record every packet because this would not have added very much to our understanding of DDoS attacks – the packets are invariably much the same and, where there are (not) variations in various identifier fields, inspecting 15 packets has proved sufficient.

The five minute period for rotating files and restarting counts is a trade-off between the detail of our measurements and the size and number of files created. When we started our work we were expecting to find long duration attacks that ran for many hours or days, so that although we know exactly when an attack starts we did not originally consider it a problem if we only knew to within 5 minutes when it finishes. However, to understand short attacks better, since August 2016, Sniffer has been modified to record the time of the last packet, as well as the first packet, precisely.

We also found that the clocks on the sensors could be unreliable since some cheap services failed to run NTP on their host and OpenVZ containers cannot set their own time. However, since December 2015 we track inaccuracies on our central server where data is delivered and correct the timestamps, and so this is not an issue after that point.

C. Ethical considerations

We were concerned that by deliberately providing UDP packet reflectors we would assist criminals in performing DDoS attacks. We addressed this by limiting the number of packets we reflect. We need to reflect the first few packets in order for the criminals' scanners to learn that our sensors were available for them to use. However, once we receive more than a handful of packets with the same source IP address we assume an attack is taking place, and we cease reflecting packets to that destination at all (that is 'Hopscotch' stops reflecting whilst 'sniffer' keeps on recording the details of the attack) until there has been no relevant activity for 30 minutes.

Furthermore, when any particular sensor identifies a victim this is reported without delay to a central server which promptly informs all the other sensors of the attack, so that they all immediately refuse to reflect any packets to the victim.

The result is that if our sensors are used by a criminal then after a very short period there will be rather *less* traffic delivered to the victim than if the sensors did not exist because some of the attack traffic generated by the criminal is being absorbed by our sensors rather than going to real reflectors and being reflected and amplified. Hence, as we run more sensors, benefit for victims of DDoS attacks is increased as we absorb more attack traffic.

There are a number of initiatives to try and reduce the number of UDP reflectors connected to the Internet. 'White-hat' scanners identify the reflectors and report them to the relevant ISP or hosting company who will then contact the owner of the machine. We were concerned that, by operating sensors, we might distract abuse handling teams from dealing with other customers who were inadvertently operating UDP reflectors. Accordingly we maintain an exclusion list of the white-hat scanners and never respond to their packets. We have also added a number of other scanners to this list, such as Shodan [16] and those operated by research teams at other Universities, because their scans distort our results and our responses will doubtless distort theirs.

Despite our efforts to be unnoticed, we have had six complaints about sensors in nearly three years of operation. On each occasion a victim had analysed packet captures from the start of a DDoS attack, found the sensor's IP address and

generated an automated complaint. In each case, we were able to deal with the issue by explaining the nature of our research.

We have also used leaked data from booter attack databases to check our results (§VI-E). These booters are no longer in operation and so there is no risk in revealing their names. The use of this leaked data is necessary as there is no other way of knowing the true behaviour of booters.

We followed our institution's ethical review procedure throughout. We carefully designed our experiments to operate ethically and we had no human subjects.

IV. Traffic classification

To perform a UDP reflection attack the criminal needs a list of reflectors for relevant protocols. They could use lists published by others or interrogate a search engine such as Shodan. Since Hopscotch ignores packets from known 'white-hat' scanners, this would be ineffective in locating our sensors. Therefore, we believe attackers actively scanned some or all of IPv4 address space to determine the location of our reflectors. Such a scan can take less than 45 minutes with ZMap [6], and as long ago as January 2014 ZMap was being utilised for 21.7% of the scans of more than 50% of IPv4 address space [5].

Once an attacker has located one or more of our reflectors, they must then send UDP packets with a (forged) source IP address set to that of the victim, for example by generating them on a rented host at a hosting company that is not BCP38/SAVE [7] compliant.

We wish to know the length of each attack, which we determine by identifying the first packet with a given port number and source IP address (the victim) which arrives at any of our sensors. We then look forward in time until there is a 15 minute gap without any further matching packets reaching any of our sensors.[1] This gives us times for the beginning and end of the attack. However, recall from §III-B that until August 2016 we did not record the time of the last packet of each type within the batch file. So prior to this date we overestimate the end time by up to 300 seconds (with an average excess of 150 seconds).

We expect the attacks to involve large numbers of packets whereas, to ensure they are quick, the scans will employ a small number of packets, probably just one per IP. This means that simply by counting the packets we can distinguish a scan from an attack. So, having identified the start and end times of an attack we then check to see if any sensor received more than 5 packets. If not then we classify the event as a scan.

This is a different definition of an attack from that used in other work, Amppot considers packets to be part of an attack if it observes 100 packets with the same source IP address with no gap of more than 3 600 seconds [13] and Noroozian et al. modify this to use a gap of 600 seconds [25]. We use a threshold of 5 packets *per sensor* with no gap of more than 900 seconds.

[1]The use of 15 minutes is justified in Appendix A-B.

Protocol	Mean	Median	5%	95%
CHARGEN	20.8	7	2	39
NTP	1 160	21	4	711
DNS	2 250	65	9	1 410
SIP	124	119	56	204
SNMP	9.73	8	1	21
SSDP	729	224	3	3 530
QOTD	1.47	1	1	2.85
Netis	424	55.5	1	1 180

TABLE II: Numbers of scanners per day for some popular UDP protocols, showing the mean and median values, along with the 5th and 95th percentiles.

There are two complications with this counting approach for distinguishing victims and scanners. The first is that some attackers send packets to randomly chosen IP addresses within a /24 (or bigger) prefix – reasoning that the volume will impact every host within that IP range and the randomness may make the attack harder to detect or block. We identify these 'prefix attacks' by assessing the number of victims within each address block allocation made by the Regional Internet Registries (RIRs) and if there are more than about 16 victims in a /24 (see Appendix A-C for the analysis details) then we conclude that the attack is not against individual hosts but record all the events as a single attack against the prefix. We discuss this type of attack further in §VI-B below.

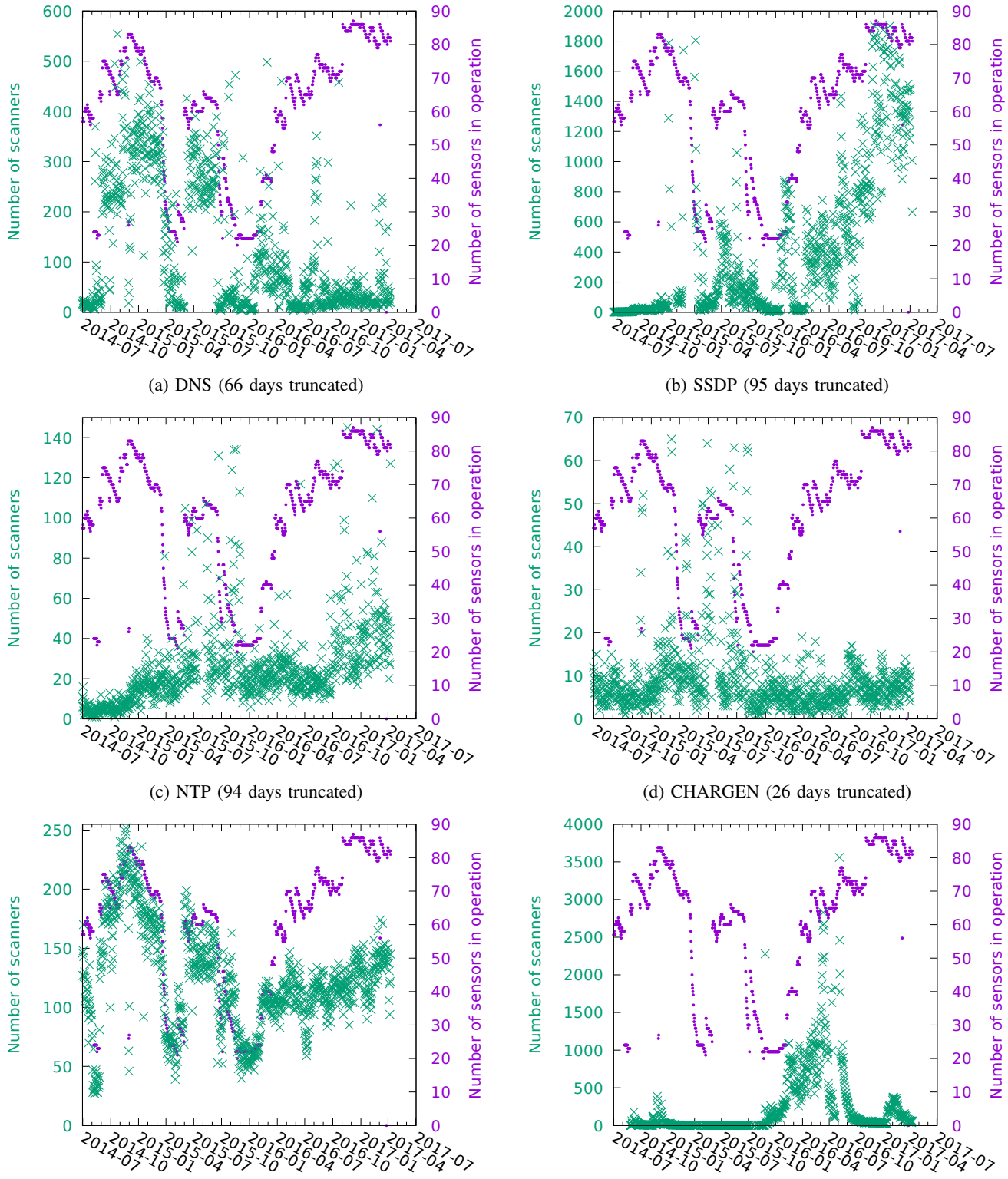
The second complication in our attempt to distinguish between scanners and victims is the attacks we see against DNS servers that involve requests to resolve randomly chosen subdomains. The randomness of the request means that the answer will never be cached and so the recursive resolvers consult the authoritative DNS server for the domain – which means that this is actually a DDoS on the authoritative server. Our experience is that in these attacks the requesting IP addresses are randomly chosen (often they are not even routable) and so we have manually identified the domain names used in these attacks (a relatively simple exercise since most other DNS attacks use a small set of queries). We analyse these attacks in §VI-C below, and ignore them entirely elsewhere.

We do not believe that there are any other kinds of attack that might influence our analysis as while there are scanners searching for exploitable services, these are scanning on ports other than those Hopscotch reflects from.

V. Scanning behaviour

Our sensors record scans for every UDP port, including those for which we have not (yet) implemented any reflection. By examining how many sensors detect any particular scan we can describe some rather different approaches to scanning. Known 'white-hat' scanners are excluded from this analysis as described in §III-C.

Our data shows that across all protocols there are an average of 5 070 (95% CI [258, 14 300]) IPv4 addresses scanning each day with a median of 1 450. However, there is a wide variation in the number of scanners between protocols with a median of just 1 QOTD scanner, but 65 DNS scanners, per day. Statistics for the protocols abused in DDoS attacks are given in Table II.



(a) DNS (66 days truncated)

(b) SSDP (95 days truncated)

(c) NTP (94 days truncated)

(d) CHARGEN (26 days truncated)

(e) SIP

(f) port 53413 (Netis routers)

Fig. 1: Count of distinct IP addresses scanning per day for six different UDP protocols (×) under plotted with the daily number of operational sensors (·). Note that the figures have different y-axes scales for clarity. The correlation between the number of scanners and the number of sensors is discussed in §V-A and the need for the vertical truncation is considered in §V.

Figure 1 shows the daily count of distinct IP addresses scanning for six interesting protocols, as well as the number of sensors in operation. Care is required in interpreting these graphs because the vertical scales differ in order to present the data at the best resolution for each protocol.

In addition, we have truncated the vertical scale to exclude days which have high numbers of scanners, even after removal of prefix attacks (see §VI-B). The number of days excluded from the plot by this truncation is indicated in the figure captions. We do not currently believe that there are sudden spikes in the number of people undertaking scans, so we assume that there must be some linkage between the disparate scanning IP addresses, but we continue to investigate this phenomenon.

The first four figures (Figures 1a-1d) show how fashions change in DDoS – interest in scanning for DNS grew over time but declined in 2016 whereas scanning for NTP continues to grow, albeit at lower levels. Figure 1d shows ongoing interest (again at low levels) in scanning for CHARGEN reflectors.

The last two figures (1e, 1f) show potentially different criminal behaviour, since the attackers are likely seeking to exploit a vulnerability of the host itself rather than detect a UDP service suitable for use in reflection attacks.

Figure 1e seems to show somewhat variable interest in scanning for Session Initiation Protocol (SIP) [30]. SIP is frequently used for Voice Over IP (VOIP) communications and insecure devices can be exploited by criminals who monetise the services they provide. The final graph (Figure 1f) shows how much scanning there is of port 53413. We believe these are scans by criminals who seek to locate and compromise certain models of Netis router to build a botnet [40]. There are two spikes in 2014 (the first of which begins shortly after initial press coverage of the Netis vulnerability) and there is then much increased interest at the end of 2015 and during 2016, which we cannot currently link to any particular publicity.

While we have had some success tying scans and attacks together using techniques similar to those used by Krupp et al. [14] which rely on information on the TTL and which subset of sensors was used for an attack, further work is required to make it sufficiently robust in the absence of ground truth, so we do not present it here.

A. How sensor numbers affect our data

Although we have just said that the SIP scanning graph (Figure 1e) apparently shows variable interest by criminals in scanning for SIP, the shape is in practice an artefact of the number of sensors that we had operational at any given time. We believe that those seeking to compromise SIP devices stop their scanning once they have sufficient devices for their immediate needs. In contrast, we believe, those planning to perform a DDoS attack want to identify as many reflectors as possible.

To show this effect, all the figures in Figure 1 also plot the daily number of operational sensors and some correlation is evident to the eye. Calculating Spearman's ρ correlation coefficient between the daily counts of operating sensors and the number of scanners observed reveals significant correlation (according to Fisher Z and Student T tests at 3 standard deviations) for SSDP, NTP, and SIP but not for CHARGEN, DNS, and 53413. For SIP the ρ value of 0.601, indicates that on the days we see fewer scanners a major reason is that fewer sensors were operating. That is, SIP scanners are being fairly selective in how big a range of IP addresses they scan each day. Similarly SSDP has substantial correlation with ρ value of 0.503, this may not be causal, recent increase in interest in SSDP happened to occur during the period of highest daily sensor counts [1]. NTP is less correlated with a ρ value of 0.128.

Protocol	#	mean	median	5%	95%
CHARGEN	560	7.19	4	1	19
DNS	11 100	7.28	2	1	28
NTP	2 730	9.74	9	2	19
SIP	7 240	14.4	9	2	42
SNMP	430	15.1	5	1	56.8
SSDP	8 080	18.2	12	2	52.2
Netis	5 360	7.81	6	2	19

TABLE III: Scanners with more than 10 scan events; giving the number of scanners (#) and statistics (other columns) for the number of days on which they scanned.

B. Lifetime of scanners

If the IP addresses used for malicious scanning change regularly then it is harder to block or investigate them, so it is useful to understand the extent to which scanners continue to use the same IP address for long periods. Hence, we consider the IP addresses of 'very active' scanners for which we have observed over 10 scan events (as described in §IV). Table III records how many of these very active scanners we observed per protocol and the average number of days on which they operated.

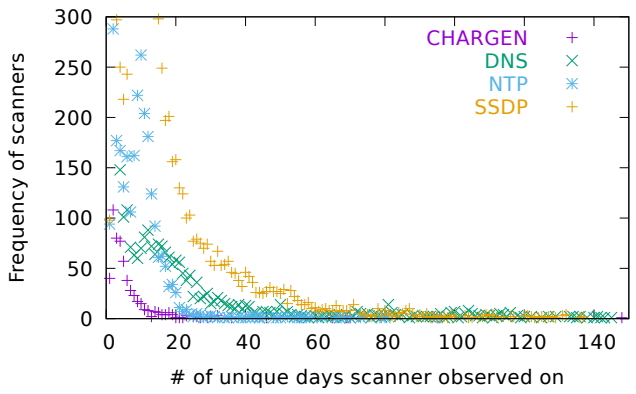
In Figure 2 we present scatter plots for this data, the upper graph showing the protocols which are abused in DDoS attacks (and which we reflect) and the lower graph showing protocols that we do not currently reflect.

For clarity, the small number of scanners that were observed on more than 150 days are not shown. Also, many scanners trigger our threshold of 10 scan events but do so on rather fewer than 10 days – that is they interact with our sensors several times on the same day, but those interactions are spread far enough apart for us to consider them separate scans.

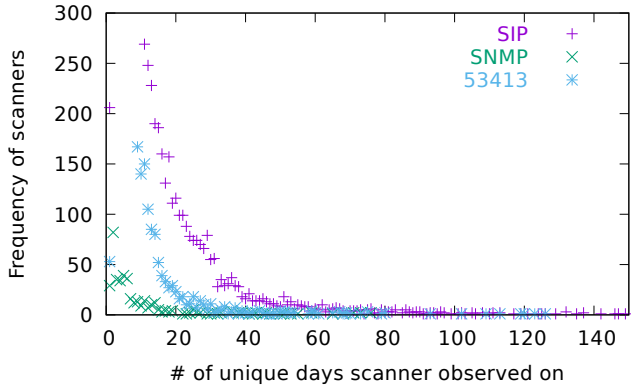
VI. Attack behaviour

Having explained what we have learnt about scanning, we now turn to analysing the data we have collected on DDoS attacks. Our approach only collects data about reflective UDP amplification attacks and only for attackers that do their own scanning to identify our sensors.

We observe 5 120 (95% CI [594, 10 200]) attacks per day on average with a median of 5 150 and maximum of 13 800. We have received 3.31 trillion packets during these attacks with an average of 3.26 billion (95% CI [23.7 million, 11.2



(a) Unique days (reflected protocols)



(b) Unique days (not reflected protocols)

Fig. 2: Frequency of observing scanners for different numbers of days for various protocols. The upper graph is for protocols we reflect, the lower graph is for protocols we do not reflect.

billion]) per day and median of 2.6 billion. On average we find that our sensors are each currently (April 2017) receiving around 200GB of traffic per month.

Table IV summarises the attacks during the entire measurement period, broken down by protocol and again by number of attacks, packet count and total attack hours. The data is normalised by the number of days of operation to account for the progressive deployment of reflection services. It is clear that the four most abused protocols are NTP, DNS, SSDP and CHARGEN, in that order. It is perhaps unsurprising that NTP and DNS services are widely used for UDP reflection attacks since they have some of the highest amplification factors.

Packets received by Hopscotch nodes on ports with no UDP service cannot form part of an attack because that node has never responded to a scanner. However, we incorrectly classify 563×10^6 packets ($< 0.1\%$ of the total) to be attacks, even though this was impossible. The cause is that scanners decided (for unknown reasons) to send more than 5 packets to our sensor, so we misclassified the event. Although a small number of packets, the effect is to cause us to record an additional 229 attacks per day in Table IV than is in fact the case.

Table V shows the top 10 domains by number of attacks used for reflective DNS attacks. It is comparable with a similar table given in AmpPot [13, p630], but the substantial differences are because this dataset was collected over a much longer period, and measures substantially more attacks. These were the domains which contained records used for reflective DNS attacks.

A. *Predicting the duration of attacks*

In Table VI we present percentiles of attack duration and packet count. For example, half of all attacks last for less than 10.97 minutes; similarly our sensors see fewer than 7 790 packets in over half of all attacks. Note that this does *not* imply that half of all attacks take less than 10.97 minutes *and* fewer than 7 790 packets.

From our data we conclude that UDP reflection attacks are typically short in duration and low in overall packet volume, but may still consume significant bandwidth while the attack is taking place. The distribution of attack duration has a long tail: there are some attacks which continue for hours, or in a few cases, days.

Given the distribution of attack durations, a victim may wish to know the probability of an attack finishing within the next five minutes, conditional on the fact that the attack has already run for a certain period of time. Such a calculation will help a victim determine whether action should be taken, such as reconfiguring their network and/or servers, or whether simply waiting for the attack to end is a strategy worth considering.

We can estimate the probability of an attack finishing within the next five minutes as follows. First, we divide attack duration into five-minute periods, starting from zero, so the zero-th period is $(0, 5]$, the first $(5, 10]$, and so on. Then, let n_i represent the number of attacks which finish within the ith five-minute time period $(5i, 5(i+1)]$. For example, across all protocols, $n_3 = 584\,132$, which means 584 132 attacks lasted between 15 and 20 minutes in duration. Finally, we can estimate the probability of an attack finishing within the next five minutes, given that attack duration D has already reached time period t, as follows:

$$P(\text{finishes in 5 mins} | D = t) = \frac{c_t}{\sum_i c_i - \sum_{j=0}^{t-1} c_j}$$

Or, in words, we estimate of the probability of attacks finishing in time period t as the proportion of attacks which finish inside time period t, (c_t), versus those which finish in time period t or any later period.

Figure 3 plots the distribution of attack durations and the probability of an attack finishing within the next five minutes for NTP, DNS, SSDP and CHARGEN. The probability of an attack ending within five minutes is highest in the first 5 to 10 minutes of the attack. For example, for NTP, when an attack has already lasted five minutes, there is a 45% chance it will terminate within the next five minutes. These probabilities generally decline as the attack continues, which is bad news for a victim: an attack generally becomes less and less likely to stop the longer it goes on.

Protocol	Attacks per day		Packets per day		Attack-hrs per day		Period	BAF	PAF
	#	%	# (10^6)	%	hours	%	days		
NTP	3 430	51.7	3 030	93.2	1 090	49.1	1 020	556.9	10.61
DNS	1 290	19.4	22.8	0.7	406	18.3	1 020	28.7	1.32
SSDP	1 040	14.5	26.8	0.8	386	16.1	927	30.8	9.92
CHARGEN	601	9.1	110	3.4	189	8.5	1 020	358.8	1
Portmap	237	3.4	137	2.4	104	4.6	470		
SQLMon	7.3	< 0.1	0.2	< 0.1	1.5	< 0.1	835		
QOTD	0.7	< 0.1	< 0.1	< 0.1	0.2	< 0.1	1 020	140.3	1
mDNS	7.0	< 0.1	0.2	< 0.1	8.5	0.4	349		
Total	6 400	96.5	3 250	100.0	2 090	94.3			
Misclassified	229	3.3	0.6	< 0.1	127	5.4			

TABLE IV: Attacks seen, broken down by protocol. The Attacks/day column records the number of attacks, averaged across the measurement period for each protocol, together with their proportion relative to all protocols. The packets/day, and attack-hours/day are derived in the same manner. Attack hours/day can exceed 24 as many attacks can be in progress at any one time. For reference, the average Bandwidth Amplification Factor (BAF) and Packet Amplification Factor (PAF) from Rossow's *Amplification Hell* paper [31] is recorded where available.

Domain	First seen	Duration	Attacks
cpsc.gov.	2014-08-10	884	285 000
fullenlaces.com.	2014-09-04	400	250 000
mg1.pw.	2015-04-10	145	238 000
067.cz.	2014-08-26	852	200 000
eda.gov.	2014-08-10	703	167 000
dhs.gov.	2015-02-24	68	133 000
doleta.gov.	2014-08-10	561	121 000
ohhr.ru.	2014-11-20	152	89 300
msk.su.	2014-12-28	729	65 600
r3a.es.	2015-03-19	28	64 500

TABLE V: Top 10 domains used in reflective DNS attacks by number of attacks. Duration in days.

Percentile	Duration (min)	Packet count (#)
10%	4.73	43
20%	5.05	255
30%	7.00	985
40%	10.00	2 970
50%	10.97	7 790
60%	13.87	20 600
70%	17.43	56 500
80%	23.10	170 000
90%	35.67	625 000
95%	57.52	1 590 000
99%	130.52	7 450 000

TABLE VI: The maximum duration and packet count for a given percentile of attacks. 90% are less than 35.67 minutes; in 90% we record fewer than 625 000 packets. CDFs are plotted in Appendix A-D.

There are two notable exceptions to this general trend: at around 60 and 120 minutes, NTP attacks become relatively more likely to cease. We note that the booter/stresser services which we have examined commonly let users specify attack durations of 5, 10 and 60 minutes and it seems very likely that we are seeing these durations reflected in our attack data.

Since Karami et al. [11, 12] found that the identity of the victim strongly influenced the likely length of attacks, it would be useful to calculate the probability that attacks will shortly cease based upon the experience of victims of a similar type (games players, corporations, etc.). We leave the calculation of such attack duration percentile tables to future work.

B. Attacks on IP address prefixes

As we explained in §IV, some DDoS attacks do not target individual hosts but attack many hosts within a block of addresses – sometimes just a /24 prefix, but we have also seen attacks on a /12 and a /13. The attack is effective when all the addresses in the prefix range are behind the same router, so that when the capacity of the router is exhausted, connectivity to all the addresses is lost.

For example between 14th and 29th of December 2015 (16 days) we observed 56 attacks on 27 different Turkish prefixes. This coincided with claims by Anonymous to be attacking Turkey as part of `#OpTurkey` [36].

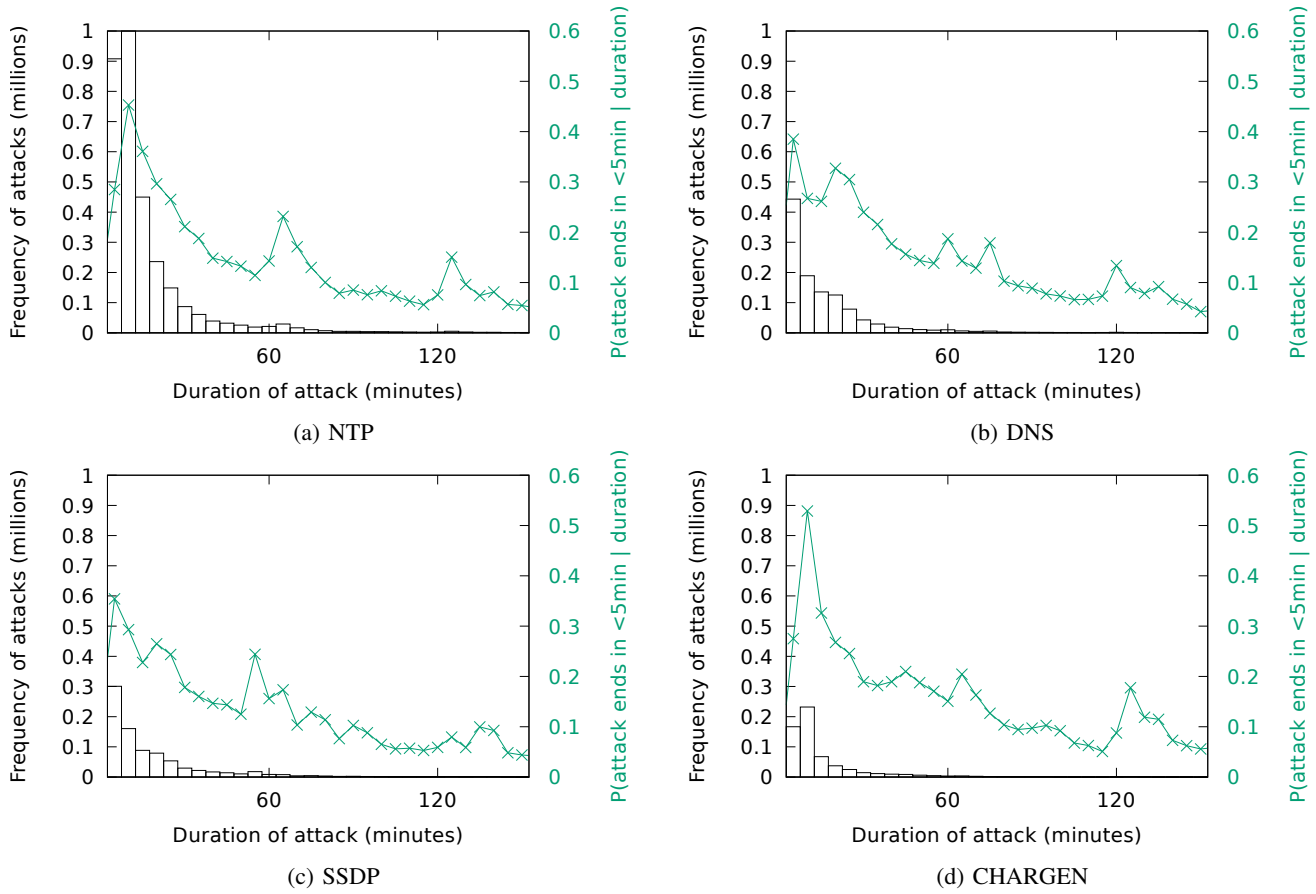
When compared with the total number of observed attacks per day of 5 120, prefix attacks are rare. Overall, we observed just 6 000 attacks on prefixes over the whole data collection period with an average daily rate of 5.93 (95% CI [0, 33]) and median of 3. Figure 4 shows the consistent behaviour of prefix attacks over the entire measurement period.

C. Attacks on authoritative DNS servers

Our Hopscotch sensors act as recursive DNS servers and in §IV we discussed the attacks that involve asking large numbers of recursive DNS servers to resolve random sub-domains in order to mount an attack on the authoritative server for the domain.

We have detected an average of 6 550 (95% CI [0, 19 900], median 0) packets per day targeting the authoritative DNS servers for 1285 different domains. Figure 5 shows that this type of attack has become significantly less common over the last year.

Table VII records the authoritative servers for the top 10 domains by traffic volume together with attack duration. We also observed one attack, presumably from a competing booter, on the authoritative server for a domain that had been created solely to providing records for reflective DNS attacks.



(a) NTP

(b) DNS

(c) SSDP

(d) CHARGEN

Fig. 3: Frequency and attack duration curves for DNS, NTP, SSDP and CHARGEN. Frequency plots show the number of attacks which last for the duration shown on the horizontal axis. The duration curves show the probability that an attack will finish within the next five minutes, conditional on the current attack duration reaching t.

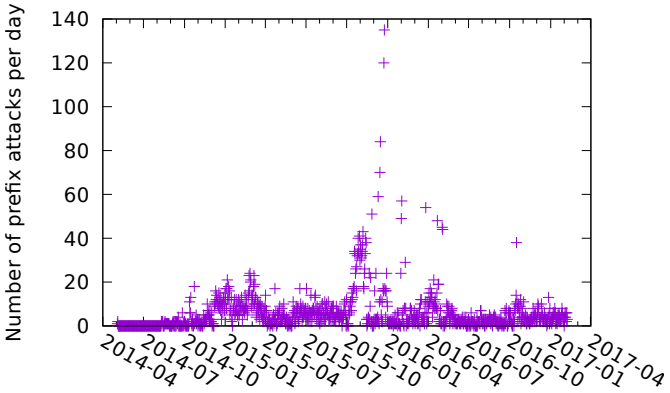


Fig. 4: Number of prefix attacks (on an IP subnet) seen each day.

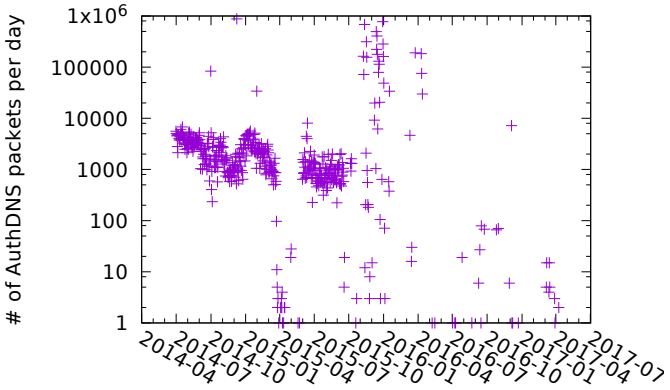


Fig. 5: Number of packets per day received in attacks on authoritative DNS servers. Note log scale for y-axis.

Total packets observed	Total duration of attacks (hours)	Targeted domain
2 690 000	5 420	x99moyu.net.
909 000	47	zyngamail.com.
401 000	3 750	edgesuite.net.
255 000	3 440	ylzc001.com.
183 000	2 840	9zhidao.com.
154 000	2 490	uw99.com.
152 000	366	com.co.
134 000	1 580	logicielfull.com.
120 000	403	appvod.com.
111 000	1 090	gratuitzone.com.

TABLE VII: Top 10 domains by number of observed authoritative DNS attack packets.

D. Attack detection coverage

The data gathered by our sensor network is unlikely to include details of every reflective UDP attack by attackers who use Internet scans to find reflectors. This is because there are a large number of potential reflectors on the Internet and therefore an attacker may not find all reflectors in any one scan and for attacks of short duration or limited bandwidth an attacker may use a subset of available reflectors. Despite this, we can estimate the total number of attacks, including the attacks that we did not observe, with a *capture-recapture* statistical analysis.

Capture-recapture analysis is designed to estimate the total size of a population by using two or more independent sets of samples drawn from the population and measuring the relative size of the overlap between them. The technique was originally developed for ecology, so the classic explanation goes as follows. First, you take some fish out of a lake, count them (n), mark them, and put them back. Later you return for a further day's fishing and catch (K) fish of which (k) are found to be marked as caught earlier. The total number of fish in the lake (N) can then be estimated with the Lincoln-Petersen estimator ($\hat{N}$) [27]:

$$\hat{N} = \frac{nK}{k} \tag{1}$$

While other estimators exist, our sample sizes are large enough that the Lincoln-Peteresen estimator provides a reliable estimate of the total number of attacks.

Originally capture-recapture was developed for just two samples but in our data we have up to 87 sensors and hence samples on any one day. General solutions have been developed involving 2^k contingency tables [8]. However while this is tractable for $k = 5$ the computation simply does not scale to $k = 87$.

A k^2 strategy of computing pairwise capture-recapture estimates for each pair of sensors and averaging the result could be used (Appendix B-A). However, when there are many small samples, two samples may happen to have a large overlap between them and this can bias results: two small sets of attacks could predict a small total population size with high confidence. Indeed this method sometimes gave an estimate that was much smaller than the total number of observed attacks across all our sensors.

Our solution is to split our sensors into two groups, a core set of sensors that have run continuously since the beginning of this work (including those in the /28), plus a few additional sensors; and all other sensors (Appendix B-B). This also avoids any problems of correlation between sensors in the /28.

We compute the uncertainty in Equation 1 by treating the number of attacks observed by each sensor as a counting experiment with square root uncertainty and propagate this through the computation [37] to give Equation 2:

$$\hat{N} = \frac{nK}{k} \pm \frac{nK}{k}\sqrt{\frac{1}{n} + \frac{1}{K} + \frac{1}{k}} \tag{2}$$

Figure 6 shows the results of this analysis and plots the estimated total number of attacks each day for CHARGEN, DNS, NTP and SSDP. There are a few interesting trends visible in the data. In late 2014 and early 2015, DNS was generally the most popular protocol of choice for DDoS, punctuated by short periods when SSDP was used. Since mid-2015, NTP has emerged as the dominant protocol of choice, and usage of DNS, CHARGEN and SSDP fell significantly. However, on some days DNS has been much more popular than NTP and at the end of 2016 usage of SSDP has substantially increased. For DNS, since spring 2016, our data has often been insufficient to be confident of the actual number of attacks, on many days our capture-recapture algorithm gives no estimate as the sets it uses are are too small. This is either due to a low number of attacks or a low coverage of attacks by our sensors. On days when we do have enough data to produce an estimate the estimated coverage is often low, so it is likely that poor coverage is the cause of the problem.

In *Amplification Hell* [31] Rossow said that he feared that attackers would discover powerful amplification vectors like NTP and SSDP. Our data shows that attackers have now discovered this fact and have moved towards NTP, and from time to time, SSDP.

There were spikes of SSDP attacks in early 2015 but these subsided substantially from autumn 2015 to autumn 2016, possibly because many networks have decided to filter port 1900 entirely. It has picked up at the end of 2016. In 2015 there were a median of 1 280 attacks per day with 95% CI [109, 7 670] and in 2016 a median of 691 attacks per day with 95% CI [105, 2 200] for SSDP. In *Exit from Hell* [15] Kührer et al. found some evidence that NTP was blocked by network providers. However, there is complexity in blocking the NTP 'monlist' packets whilst allowing small timekeeping packets to get through – complexity that is entirely absent from decisions about blocking SSDP, which was only intended for use over multicast on local networks, so blocking it entirely on the open Internet appears simple and obvious to many.

Table VIII gives the estimated total attacks for each protocol, showing how different types of attack vary in popularity. The 5th and 95th percentiles illustrate the wide variation in the estimated number of attacks per day over the observation period while Figure 6 shows the relative consistency in the total number of attacks per day over shorter periods.

Protocol	Mean	Median	5%	95%
CHARGEN	761	488	64.8	2 200
DNS	4 000	1 930	142	10 200
NTP	3 860	4 130	321	6 740
SSDP	1 360	972	145	3 650

TABLE VIII: Estimated number of attacks per day.

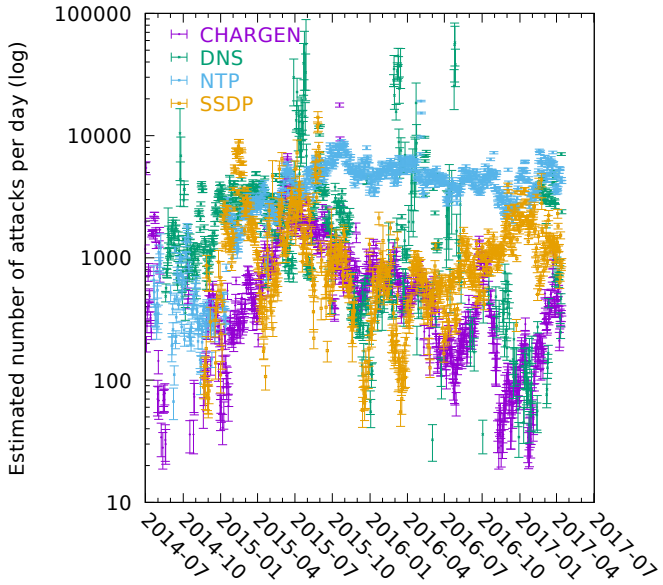


Fig. 6: Estimated total number of attacks each day for different protocols. Note that the y axis uses a log scale and so is visually misleading.

These estimates are not directly comparable with those in Table IV as the figures in Table VIII only consider days for which we had sufficient data to make estimates while Table IV uses the whole collection period.

Our capture-recapture analysis suggests that we observe almost all of the estimated total number of attacks for NTP and CHARGEN and most SSDP and DNS attacks. This is shown in Figure 7 which records the cumulative proportion of attacks that were observed for different protocols. Table IX summarises this with the average estimated proportion of attacks observed. This is also consistent with the correlation between the number of sensors in operation and number of scanners observed, for DNS and SSDP, discussed in §V-A.

The table also shows the number of days for which we had sufficient data to make estimates for each protocol. This shows that we have better coverage for CHARGEN and NTP than for DNS and SSDP, which we believe stems from the population of DNS and SSDP reflectors being much larger than for NTP and CHARGEN, so our sensors are less likely to be utilised. The number of days for these protocols differs not only because the reflection functionality for each protocol was deployed at different points in time but also because some protocols were sufficiently unpopular during some periods that we have insufficient data to make an estimate.

Appendix B-C discusses the variation in estimated coverage

Protocol	Mean	Median	# Days	Deployed
CHARGEN	0.958	0.997	923	2014-03
DNS	0.851	0.981	727	2014-03
NTP	0.966	0.992	948	2014-03
SSDP	0.906	0.98	822	2014-10

TABLE IX: Estimated proportion of attacks observed, and number of days of data, for different protocols.

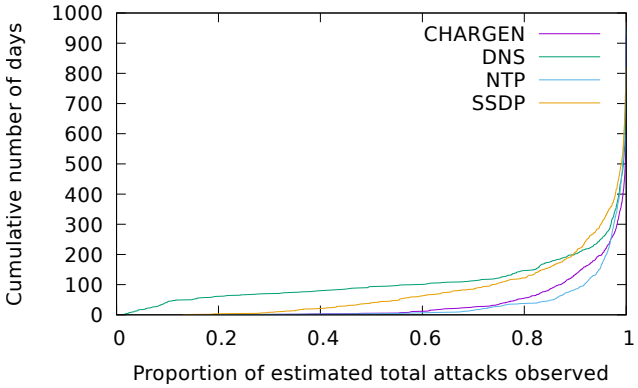


Fig. 7: Cumulative frequency of observing different proportions of attacks different protocols.

over time and with the number of sensors in operation.

E. Comparison with other data sets

We came across one booter service (Vdos) which had a real-time display of all currently active attacks. We believe this was implemented to allow users to know how many people they would be sharing the available attack bandwidth with. Between 2014-10-15 and 2014-11-04 (when the functionality was removed[2]) we recorded information about the attacks that the booter claimed were in progress. Later the Vdos booter was hacked and its database leaked (with records of 170 000 attacks) along with the database of the cnbooter (60 attacks), ustress (2 370 attacks), and vstress (2 attacks) booters that were hosted on the same server. In addition logs from the Vdos backend used for all these frontends for 896 000 attacks between 2015-08-30 and 2016-09-01 were leaked. We also separately obtained the database for CMDbooter with 25 800 attacks recorded from 2015-07-08 to 2015-08-06. Table X shows, for the DDoS strategies we would expect to observe, the number of recorded attacks that we did observe.

For the NTP attacks recorded by scraping Vdos we observed 92% of the attacks. However, for SSDP we only observed 11 and missed 15 500. This is unsurprising as we had only just started reflecting SSDP and were clearly not yet on this booter's list of amplifiers. Later on, the Vdos backend logs show that we observed 91% of NTP attacks, 27% of SSDP attacks, 2% of DNS attacks and 0.5% of Portmap attacks. Here the Portmap result is unsurprising as Hopscotch had only just started reflecting Portmap. However, DNS has been supported

[2]It was later re-added and other researchers collected similar data for the period 12-2014 to 02-2015, but were not able to share it with us [12].

Source	Period	Protocol	Observed	Missed	Proportion
Vdos scrape	2014-10-15 – 2014-11-04	NTP	2 070	177	0.921
		SSDP	11	15 500	0.000 71
Vdos API	2015-08-30 – 2016-09-01	DNS	10 800	469 000	0.022 5
		NTP	73 900	6 970	0.914
		Portmap	67	12 000	0.005 54
		SSDP	15 700	42 300	0.271
CMDBooter	2015-07-08 – 2015-08-06	CHARGEN	512	5 090	0.091 4
		DNS	5	510	0.009 71
		NTP	1 200	7 460	0.139
		SSDP	156	8 030	0.019 1

TABLE X: The number of attacks that we might have observed versus those that we did observe based on records from booters. Since we did not start deploying Portmap support to our sensors until January 2016 it is expected that we would only observe a small number of Portmap attacks. However coverage of DNS is low. These are the raw number of attacks on IP addresses, there is no collation of consecutive attacks against the same target or attempt to resolve domain names.

from the beginning and only 2% of attacks were observed. The capture-recapture analysis reflects this, as it estimates our coverage of DNS attacks as low on many days (see Figure 12) and has insufficient data to provide an estimate for other days, particularly for many days in late 2016. Hopscotch's coverage of CMDBooter is rather worse with only 14% of NTP attacks observed, 9% of CHARGEN, 2% of SSDP and 1% of DNS.

This illustrates an inherent shortcoming of our capture-recapture analysis. When our sensors have been around long enough to be found by scanners we record a high proportion of attacks and can robustly estimate what we miss. However, when our sensors are new (or otherwise ignored by attackers, who perhaps identify them as honeypots, or who have more reflectors to hand than they need) then they are never used and we cannot extrapolate meaningful data from a lack of activity.

VII. Conclusions

In this paper we have analysed the data collected from a network of Hopscotch honeypot UDP reflectors of median size 65 over the period July 2014 until January 2017. We saw a median of 1 450 malicious scanners per day across all UDP protocols, but we find that the number of scanners operating at any one time for many protocols used in reflected UDP amplification DDoS attacks is relatively low and that in turn suggests that intervention strategies that deal with one instigating entity at a time might be tractable.

We recorded details of 5.18 million attacks involving in excess of 3.31 trillion packets. We observed that the proportion of attacks using DNS reduced and the proportion using NTP increased over the observation period, in line with the fears expressed in earlier work [31]. We also observed that SSDP, about which there were also fears, was briefly more popular than DNS then fell away – but is lately of more significance.

This is the first paper we are aware of to leverage disparate data collection locations and use capture-recapture statistics to estimate the total number of reflected UDP amplification attacks. Over the observation period, the median estimated total number of attacks per day for CHARGEN is 488, for DNS is 1 930, for NTP is 4 130, and for SSDP is 972.

Naturally the soundness of applying capture-recapture analysis depends upon assumptions about the independence of our samples. It may be that some attackers never use our reflectors at any time, and comparison with leaked booter datasets shows that for some protocols at some periods this is an issue. However, for some protocols we achieve high level of detection (between 85.1% and 96.6% of attacks) and so the size of our sensor network is not so limited as to miss whole swathes of relevant activity. The sensors have a very small footprint – the main impact on the hosting system is the arrival of circa 200GB of incoming (but not of course reflected) traffic per month. This means that fully duplicating our setup on low-cost hosting providers is possible for well under USD 1500 per annum.

Although we originally built the Hopscotch honeypot for scientific purposes, to determine the prevalence UDP amplification attacks and characterise the victims, it has proved to be of practical assistance to numerous organisations who have been suffering DDoS attacks. It might be thought that a victim would not care about ineffective attacks or would have a surfeit of data about a damaging DDoS, but in practice telemetry and measurement tools are often inadequate and we have been able to assist numerous people in understanding what just hit them.

Dataset

We are unable to release our dataset as open data because it contains sensitive information about victims and scanners as well as identifying the location of our Hopscotch nodes and the identity of our partners. However, all of the data used in this analysis is available through the Cambridge Cybercrime Centre (https://cambridgecybercrime.uk) to academic researchers. We also have other databases for other booters that do not overlap with the Hopscotch data including those used by Santanna et al. [33].

Acknowledgements

We are extremely grateful to the organisations and individuals who have hosted Hopscotch nodes, and in particular the ShadowServer Foundation and Digital Ocean Inc. Daniel R. Thomas is supported by a grant from ThreatSTOP Inc.

Richard Clayton is supported by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHSS&T/CSD) Broad Agency Announcement 11.02, the Government of Australia and SPAWAR Systems Center Pacific [contract number N66001-13-C-0131]; and the EPSRC [grant number EP/M020320/1]. Alastair R. Beresford is partly supported by the EPSRC [grant number EP/M020320/1]. The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect those of any of the funders.

REFERENCES

[1] Akamai Inc. *Akamai's [state of the internet] / security Q4 2016 Report*. 2017.

[2] 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)*. USENIX, 2012.

[3] Cheshire, S., AND Krochmal, M. *Multicast DNS*. RFC 6762 (Proposed Standard). Internet Engineering Task Force, 02/2013.

[4] 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: *Internet Measurement Conference (IMC)*. ACM, Vancouver, BC, Canada, 11/2014, 435–448. ISBN: 9781450332132. DOI: 10.1145/2663716.2663717.

[5] Durumeric, Z., Bailey, M., AND Halderman, J. A. An Internet-wide view of Internet-wide scanning. In: *Proceedings of the 23rd USENIX Security Symposium (USENIX Security)*. USENIX, 08/2014, 65–78.

[6] Durumeric, Z., Wustrow, E., AND Halderman, J. A. ZMap: Fast Internet-wide scanning and its security applications. In: *Proceedings of the 22nd USENIX Security Symposium (USENIX Security)*. USENIX, 08/2013.

[7] Ferguson, P., AND Senie, D. *Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing*. RFC 2827 (Best Current Practice: BCP38). Internet Engineering Task Force, 05/2000.

[8] Fienberg, S. E. The multiple recapture census for closed populations and incomplete 2k contingency tables. *Biometrika* 59, 3 (1972), 591–603.

[9] Hutchings, A., AND Clayton, R. Exploring the provision of online booter services. *Deviant Behavior* 37, 10 (2016), 1163–1178. DOI: 10.1080/01639625.2016.1169829.

[10] Jacobson, V., Leres, C., AND McCanne, S. libpcap, Lawrence Berkeley Laboratory, Berkeley, CA. *Initial public release* (1994).

[11] Karami, M., AND McCoy, D. Understanding the emerging threat of DDoS-as-a-Service. In: *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*. USENIX, Berkeley, CA, 2013.

[12] 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 (WWW)*. ACM, 2016, 1033–1043.

[13] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K., AND Rossow, C. AmpPot: Monitoring and defending against amplification DDoS attacks. In: *Research in Attacks, Intrusions, and Defenses (RAID)*. Vol. 9404 LNCS. Springer, Kyoto, Japan, 11/2015, 615–636. DOI: 10.1007/978-3-319-26362-5_28.

[14] Krupp, J., Backes, M., AND Rossow, C. Identifying the scan and attack infrastructures behind amplification DDoS attacks. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Vienna, 2016, 1426–1437. ISBN: 9781450341394. DOI: 10.1145/2976749.2978293.

[15] Kührer, M., Hupperich, T., Rossow, C., AND Holz, T. Exit from Hell? Reducing the impact of amplification DDoS attacks. In: *Proceedings of the 23rd USENIX Security Symposium (USENIX Security)*. USENIX, 08/2014, 111–125.

[16] Matherly, J. C. *The search engine for the Internet of Things*. 2013. URL: https://www.shodan.io/.

[17] Microsoft Inc. *[MC-SQLR]: SQL server resolution protocol*. 2007. URL: https://msdn.microsoft.com/en-us/library/cc219703.aspx.

[18] Mills, D., Martin, J., Burbank, J., AND Kasch, W. *Network Time Protocol version 4: Protocol and algorithms specification*. RFC 5905 (Proposed Standard). Internet Engineering Task Force, 06/2010.

[19] Mirkovic, J., AND Reiher, P. A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Comput. Commun. Rev.* 34, 2 (04/2004), 39–53.

[20] Mockapetris, J. *Domain Names – Implementation and specification*. RFC 1035 (Internet Standard). Internet Engineering Task Force, 11/1987.

[21] Moore, D., Voelker, G. M., AND Savage, S. Inferring internet denial-of-service activity. In: *Proceedings of the 10th USENIX Security Symposium (USENIX Security)*. USENIX, 2001.

[22] Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., AND Savage, S. Inferring internet denial-of-service activity. *ACM Trans. Comput. Syst.* 24, 2 (05/2006), 115–139.

[23] Network Administrator *smurf's attack*. NANOG mailing list, 5 Sep. 1997. URL: https://www.nanog.org/mailinglist/mailarchives/old_archive/1997-09/msg00060.html.

[24] Newman, L. H. *The Botnet that Broke the Internet Isn't Going Away*. 12/2016. URL: https://www.wired.com/2016/12/botnet-broke-internet-isnt-going-away/.

[25] Noroozian, A., Korczynski, M., Ganan, C. H., Makita, D., Yoshioka, K., AND Van Eeten, M. Who gets the boot? Analyzing victimization by DDoS-as-a-Service. In: *Research in Attacks, Intrusions and Defenses (RAID)*. Vol. LNCS 9854. Springer, 09/2016, 368–389. DOI: 10.1007/978-3-319-45719-2_17.

[26] Paxson, V. An analysis of using reflectors for distributed denial-of-service attacks. *SIGCOMM Comput. Commun. Rev.* 31, 3 (07/2001), 38–47.

[27] Petersen, C. G. J. The yearly immigration of young plaice into the Limfjord from the German Sea. *Report of the Danish Biological Station* 6 (1896), 1–48.

[28] Postel, J. *Character generator protocol*. RFC 864 (Internet Standard). Internet Engineering Task Force, 05/1983.

[29] Postel, J. *Quote of the day protocol*. RFC 867 (Internet Standard). Internet Engineering Task Force, 05/1983.

[30] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., AND Schooler, E. *SIP: Session Initiation Protocol*. RFC 3261 (Proposed Standard). Internet Engineering Task Force, 06/2002.

[31] Rossow, C. Amplification Hell: Revisiting network protocols for DDoS abuse. In: *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*. 02/2014.

[32] 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. *IFIP/IEEE International Symposium on Integrated Network*

Management (2015), 243–251. DOI: 10.1109/INM.2015.7140298.

[33] Santanna, J. J., Durban, R., Sperotto, A., AND Pras, A. Inside booters: An analysis on operational databases. *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management* (2015), 432–440. DOI: 10.1109/INM.2015.7140320.

[34] Scacco *Possible denial of service using DNS*. 07/1999. URL: https://marc.info/?l=bugtraq&m=93348057829957&w=2 – https://archive.is/CcNJZ (visited on 2016-05-26).

[35] Srinivasan, R. *Binding protocols for ONC RPC version 2*. RFC 1833 (Proposed Standard). Internet Engineering Task Force, 08/1995.

[36] Syrmopoulos, J. *Anonymous launches massive attack on Turkey for supporting ISIS — 40,000 sites down in 7 days*. 2015. URL: http://thefreethoughtproject.com/anonymous-targets-turkey-islamic-state-support-takes-40000-turkish-websites-offline/ – https://archive.is/piVa0.

[37] Taylor, J. R. *An introduction to error analysis*. 2nd ed. University Science Books Sausalito, California, 1997. ISBN: 093570275X.

[38] UPnP Forum *UPnP™ device architecture 1.1*. 10/2008. URL: http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

[39] US CERT *UDP-based amplification attacks*. Alert (TA14-017A). 01/2014.

[40] Yeh, T. *Netis routers leave wide open backdoor*. 08/2014. URL: http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/ – https://archive.is/tVkpz.

APPENDIX A
PARAMETER SELECTION

We have used a number of apparently arbitrary numbers for various aspects of our analysis. We now discuss the basis on which these numbers have been chosen.

A. Scan events

When we have observed 10 scan events from the same IP address then we classify that IP address as a very active scanner and use it for the plots in §V. Figure 8 shows the cumulative number of scan events seen for different IP addresses. There are many events for which we only observe one event, this is due to events like unclassified prefix attacks where we only observe one packet per IP address or unclassified authoritative DNS attacks where the IP addresses are randomly generated. A threshold of 10 cuts out noise from these sources.

B. 15 minute inter-attack threshold

We deem an attack to be over when there is no further traffic (for the relevant protocol) to a victim for a period of 15 minutes. This value was chosen to loosely correspond with the availability of short lived attacks (under an hour) from booter systems, while not conflating attacks that might take place many hours apart.

Although in practice for many attacks we see many thousands of packets and it is pretty clear when they start and end, for others where the criminal has attempted to be less conspicuous we may see only a smattering of packets at any particular sensor and the challenge is in distinguishing between attacks and scans.

C. Prefix attacks

To identify attacks targeting a range of IP addresses, rather than an individual IP, we identify any /28 ranges where more than 14 IPs see attacks with a given UDP protocol during a single day. We then expand the ranges and counts through 16 IPs in a /24, 64 in a /22, 128 in a /19 and so on to 256 in a /16. This gives us the smallest possible prefixes that were hit (we prefer to list a /28 rather than a /24, a few /24s rather than a /16).

However, where the whole of a large prefix was hit the traffic may be sufficiently distributed that there are gaps in what we see. So if there are any attacks at all (for the relevant protocol on the relevant day) to any IP address within a /28 adjacent to a prefix range then we add that /28. Finally, for all prefix ranges /$(n+1)$ we check the other half of the larger range /n and expand to the larger range if more than a quarter of that is being attacked – which once again deals with attacks on larger ranges where we miss some of the traffic.

In order not to conflate independent attacks on different entities that happened to be neighbours in the IPv4 space all of this analysis is only done within the boundaries of IP address block allocations as documented by the five Regional Internet Registries (RIRs).

We believe that we have correctly identified the vast majority of prefix attacks (most of which attack a hundred or more IPs per /24). If we have missed any then we will overestimate the number of attacks, or (more likely because packet counts to any individual IP address will be low) we will overestimate the number of scanners for that day.

D. CDFs

Figure 9 shows a CDF of the number of packets in attacks which is a smooth curve. Figure 10 shows a CDF of the duration of attacks, this is not a smooth curve as there are spikes at 5, 10, 15 and 20 minutes, a feature of the available attack lengths provided by booters. This feature also shown in Figure 3. Figure 11 shows a CDF of the number of Hopscotch sensors in operation each day, there are always at least 20 and half the time there are more than 65.

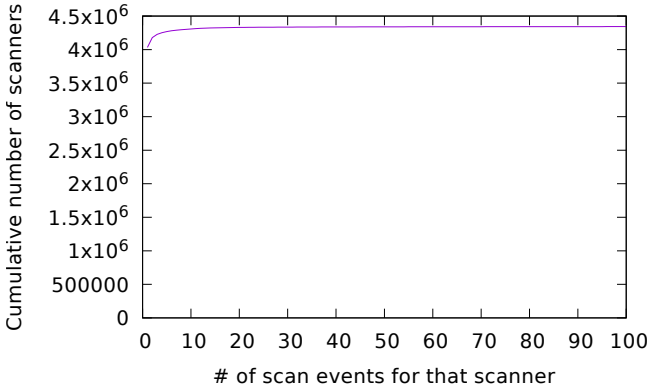


Fig. 8: Cumulative number of scan events seen per scanner.

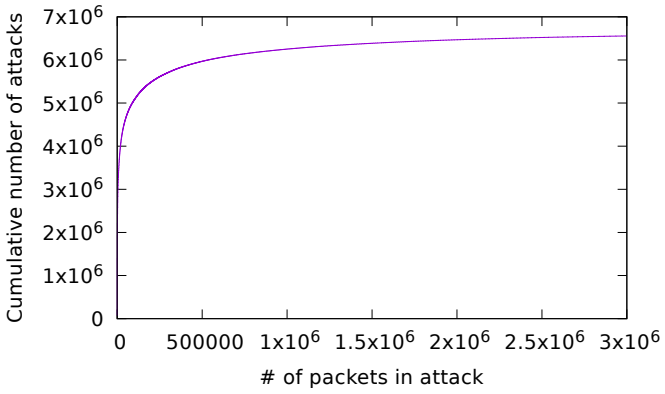


Fig. 9: CDF of number of packets in attacks.

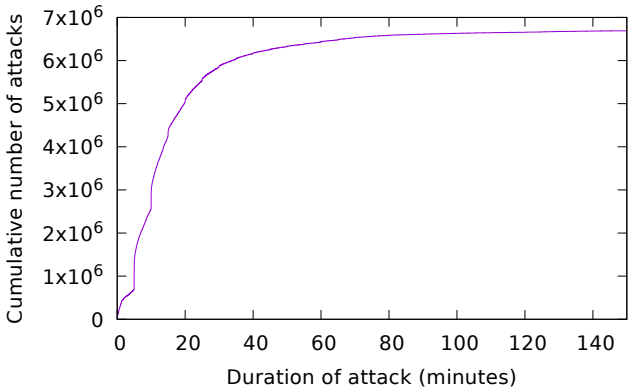


Fig. 10: CDF of duration of attacks.

APPENDIX B
CAPTURE-RECAPTURE CONSIDERATIONS

A. k^2 Capture-recapture

For k^2 rather than 2^k capture-recapture $\hat{N}_{ij}$ is estimated for each pair of samples using a standard capture recapture estimator such as Lincoln-Petersen (Equation 1). Then $\hat{N}$ is calculated by taking the mean or median of the $\hat{N}_{ij}$s.

Unfortunately this is vulnerable to bias if two samples happen to have a large overlap as then they can predict a small value of N with high confidence. This can result in values of $\hat{N}$ that are smaller than the total number of observed events and hence clearly wrong.

B. Merging samples in capture-recapture

Merging together samples to make a smaller set of samples such that 2^k methods become tractable is difficult, particularly when there may be some correlation between samples as if two samples are essentially identical (for example, in our case, two sensors on adjacent IP addresses that were started at the same time) but then end up in two different merged groups of samples, then this may cause the merged groups to also be highly correlated. We leave finding optimal divisions of samples into groups to future work.

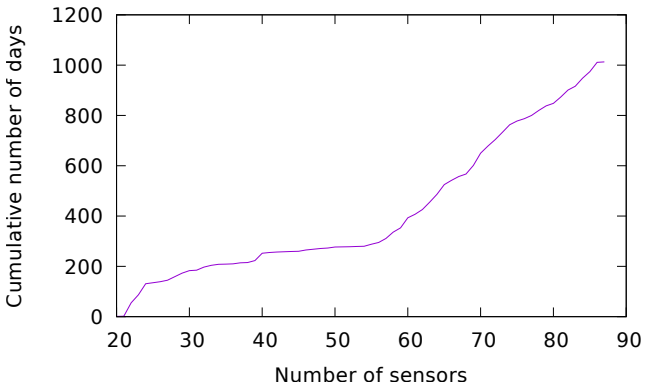


Fig. 11: CDF of number of sensors in operation each day.

Our current approach is to merge (union of the set of all observed attacks) one set of samples which should be highly correlated (adjacent IP addresses and all running the same software version for the same period) and have been participating for the whole period and a few other manually selected sensors added to provide a more even split. We then compare that with the result of merging all other samples.

C. Daily coverage

Figure 12 shows the proportion of the estimated total number of attacks that were actually observed by our sensors plotted along with the number of sensors in operation each day, and the number of those in the first set that is compared with all other sensors for capture-recapture.

The Spearman's rank correlation between the number of sensors and the proportion of the estimated total number of attacks that we observe is 0.633 for CHARGEN, 0.209 for DNS, 0.632 for NTP, and 0.342 for SSDP. This means that when we have more sensors then we observe a larger proportion of the estimated total number of attacks, as we would expect. However, these values are not that high, perhaps because we already have high estimated coverage with our existing set of sensors. The improvement in coverage of SSDP towards the end of 2016 might be caused by an increase in the response packet size for SSDP produced by Hopscotch since August 2016.

The variation in the number of sensors has a number of causes. In April 2015 a temporary problem with our largest sensor hosting partner meant that we did not receive data from them, and at the end of 2015 they ceased sharing data with us. After that we grew our own network of sensors to restore the number of sensors.

D. Daily coverage with booters

For most protocols comparing the proportion of attacks observed on a daily basis with the overall value does not show much difference. However, for NTP and SSDP for the Vdos booter using the backend logs, non-uniform behaviour is observed. Figure 13 shows daily coverage for NTP and while usually it shows very high coverage, for a period in January

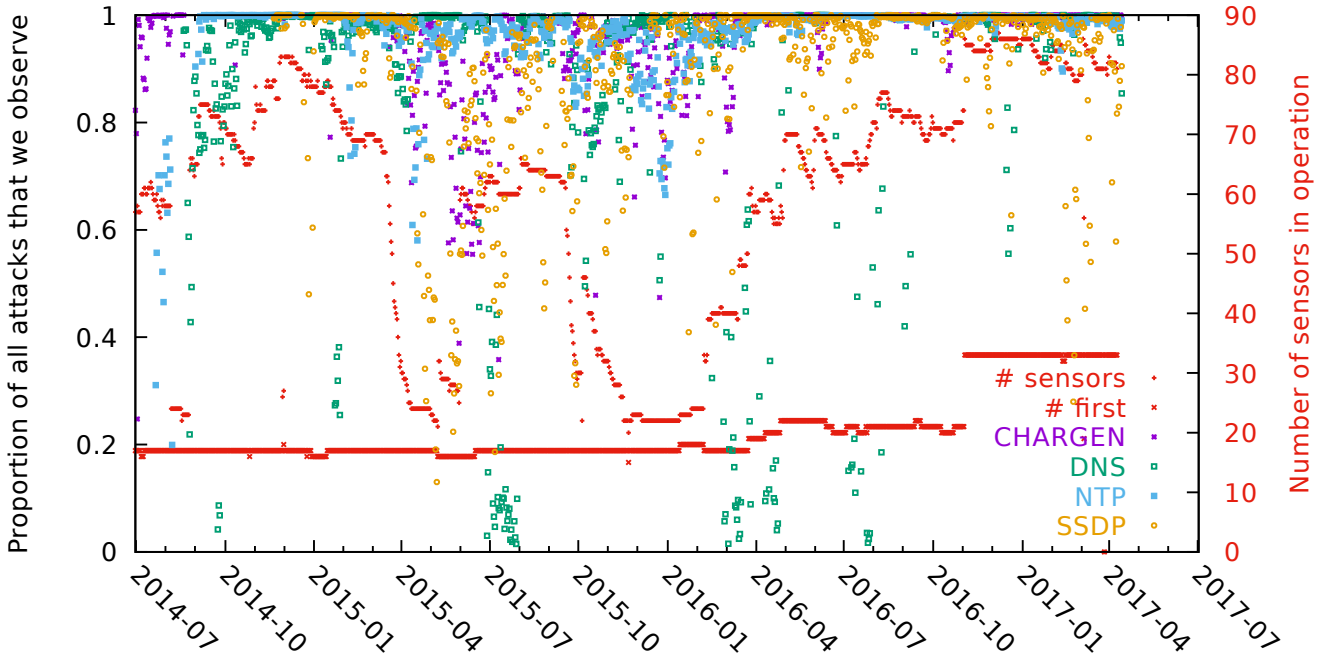


Fig. 12: Proportion of the estimated total number of attacks each day observed by our sensors for different protocols. Total number of sensors in operation each day and the number in the first set used for capture-recapture shown in red.

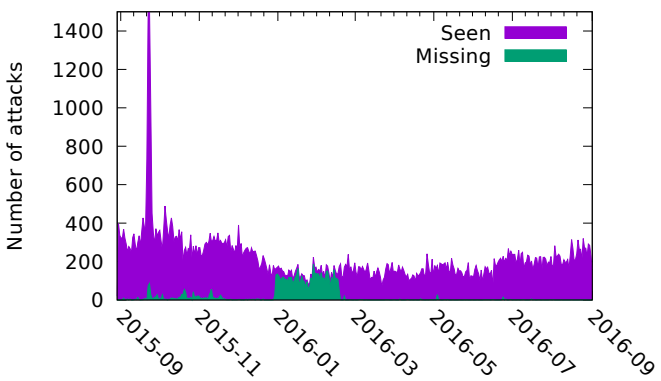


Fig. 13: NTP Vdos seen and missed attacks stackplot.

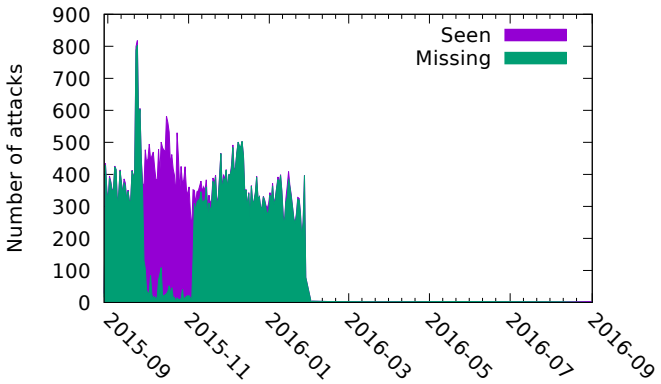


Fig. 14: SSDP Vdos seen and missed attacks stackplot.

and February 2016 it shows much lower coverage. This might have been caused by our sudden loss of sensors at that time as shown in Figure 12. In contrast, Figure 14 shows the same data for SSDP and while SSDP stops being used from February 2016, we only observe good coverage for October 2015, with poor coverage before and after.