Public Review for

On the Potential Abuse of IGMP

Matthew Sargent, John Kristoff, Vern Paxson, Mark Allman

Dear readers, welcome to another episode of our series named *"Oh, if only we had security in mind when we designed connectionless protocols!"*.
In its April 2016 issue, CCR featured a paper about vulnerabilities in NTP, a protocol that has been around for a long time. This time, a different group of authors addresses a vulnerability of another ancient Internet protocol: the Internet Group Management Protocol (IGMP), which allows neighboring hosts and routers to exchange multicast membership status information.
Like the authors of the mentioned NTP paper, Sargent et al. perform a survey of the IPv4 address space to identify vulnerable systems, IGMP responders in this case. However, the vulnerability discussed and analyzed in this paper does not affect functionalities for which the protocol itself was designed, but rather allows an attacker to mount reflective DoS attacks. The severity of this vulnerability is magnified by the amplification opportunities it offers: the authors find that 1% of the responding routers generate responses that are at least 100 times larger than the requests. In other words, the attacker not only can hide behind spoofed addresses but can also "pay less for more", in terms of bandwidth used versus bandwidth of the actual DoS traffic. To conclude the list of analogies with the previous paper of our mini series, the authors informed vendors early, to allow them time to address this issue.
CCR reviewers noted that the number of vulnerable routers is relatively small compared to other vulnerabilities, such as DNS open resolvers, but the opportunity for attackers is undeniable. Reviewers found the paper enjoyable to read, they appreciated the extensive measurements and characterization of vulnerable hosts to gain insight. Finally, for the sake of repeatability, replicability & reproducibility, the editors considered requesting that the authors release the Zmap module they developed and the raw data of their IPv4 survey. However, we concluded that public release of this data may cause harm, and decided to leave to the authors the responsibility of vetting researchers who may want to access data and tools from this paper for their own experiments.
Enjoy your reading, and stay tuned for the next episode (..or hopefully not).

Public review written by
Alberto Dainotti
CAIDA, UC San Diego

On the Potential Abuse of IGMP

Matthew Sargent[†], John Kristoff[§], Vern Paxson[‡,¶], Mark Allman[‡]
[†]Case Western Reserve University, [§]DePaul University,
[‡]International Computer Science Institute, [¶]University of California, Berkeley

ABSTRACT

In this paper we investigate the vulnerability of the Internet Group Management Protocol (IGMP) to be leveraged for denial-of-service (DoS) attacks. IGMP is a connectionless protocol and therefore susceptible to attackers spoofing a third-party victim's source address in an effort to coax responders to send their replies to the victim. We find 305K IGMP responders that will indeed answer queries from arbitrary Internet hosts. Further, the responses are often larger than the requests, hence amplifying the attacker's own expenditure of bandwidth. We conclude that attackers can coordinate IGMP responders to mount sizeable DoS attacks.

CCS Concepts

•Networks → Routing protocols; Denial-of-service attacks; In-network processing;

Keywords

IGMP; Security; Denial-of-Service; Attacks

1. INTRODUCTION

The Internet Group Management Protocol (IGMP) [6, 8, 5] is an IP host extension that allows neighboring hosts and routers to exchange and manage multicast group and routing information. IGMP requests and reports are (*i*) flooded to neighboring multicast hosts and routers via multicast or broadcast destination addresses, or (*ii*) sent via unicast between hosts and routers. In both cases IGMP operates as a connectionless request/response protocol.

Multicasting (or broadcasting) IGMP messages (case *i*) is limited in scope to the local network. However, previous work shows that some routers will respond to unicast IGMP requests (case *ii*) from arbitrary Internet hosts, and that these responses can be leveraged to study network topology [13, 14]. These previous efforts employ the Distance Vector Multicast Routing Protocol (DVMRP) [23], which operates on top of unicast IGMP messages to explicitly request information about a router's multicast neighbors. A DVMRP *AskNeighbors2* request [15] is first sent to a series of routers. DVMRP-enabled routers will respond with a unicast *Neighbors2* response. Each response contains a list of the router's multicast-enabled interfaces as well as some ancillary information about the interface, such as whether it is down/disabled, whether its neighbors are reached via a tunnel, or whether the interface represents a leaf node in the multicast tree [15]. The neighbors in each response can then be probed to investigate the topology.

While we are not interested in topology, the previous work shows that IGMP responders are pervasive. We therefore investigate these routers as a security threat for the following reasons:

- First, the connectionless nature of IGMP's neighbor discovery process provides the possibility of leveraging IGMP-responding routers in reflection attacks, whereby an attacker spoofs an *AskNeighbors2* from some victim V to some router, which in turn sends a *Neighbors2* response to V—hence hiding the attacker's identity.
- Second, since such attacks involve routers, they presumably will often offer significant bandwidth capacity—as opposed to, say, arbitrary end hosts—to bring to bear on a victim.
- Third, preliminary work [11] indicates that there can be significant amplification opportunities in the request/response exchange. As we detail below, for 85% of the routers the responses are larger than the requests—and at least 100 times larger for the top 1% of the routers. Therefore, an attacker can not only hide via reflection, but can also coax routers to increase the amount of data aimed at a victim.
- Finally, our study highlights that while we often only think about UDP when considering reflection attacks, the attack surface for mounting this type of attack actually stretches more broadly.

Given these reasons, we aim to understand the nature of *Neighbors2* responses via a scan of the IPv4 address space. Armed with this data, we are then able to empirically assess the security risks associated with having openly responding DVMRP-enabled routers.

2. RELATED WORK

Previous work uses *AskNeighbors2* probes for studying network characteristics [13, 14]. Using the *mrinfo* [14] and *MERLIN* [13] tools to query routers, researchers are able to learn about network topology by studying the routing information contained in the *Neighbors2* responses. The focus of our work differs from these studies. First, our goal is to understand *Neighbors2* response characteristics globally, by scanning the entire IPv4 address space for routers that will respond to our probes. Both *MERLIN* and *mrinfo* crawl from a seeded list of routers that grows upon receiving responses that contain additional router addresses. As such, their view of the network will be limited to routers that have

a path to their starting seeded list. Second, our goal is to study response characteristics in order to understand the security implications of having routers respond to requests from arbitrary hosts. Previous work has a strong focus on the topology information contained in responses and trying to understand when multiple IP addresses correspond to multiple interfaces on a single router.

Routers that will respond directly to packets from arbitrary hosts create a potential security risk via amplification attacks. Various attack vectors for amplification attacks exist [17] and are well documented (e.g., DNS [4], NTP [21], SSDP [18], CharGen [22]). While amplification attacks themselves have been studied, to the best of our knowlege no research exists on understanding *AskNeighbors2* requests as an attack vector. Understanding an amplification attack that targets routers is particularly interesting as the routers would be capable of receiving and handling large floods of packets, each potentially amplified. This is in contrast to other amplification attacks that leverage open network services, such as DNS, where the target resolvers used for amplification may be located on low-bandwidth residential links [19] that in practice will rate-limit the flooding traffic.

3. INITIAL SNAPSHOT

3.1 Methodology

We begin our study with the goal of scanning the entire IPv4 address space for routers that will respond to *AskNeighbors2* queries. While tools like *mrinfo* [14] and *MERLIN* [13] exist, they are not well suited for scanning the entire network in a timely manner. Both of these tools probe at low rates in an effort to match responses with the exact triggering requests in order to accurately map out network topology. Both tools also process responses in real-time as they expand their list of known routers to probe in the future. We operate under a different set of constraints as we do not process responses in real time. Therefore, we choose a tool specifically designed to scan the entire network in a timely manner, *ZMap* [7, 24]. *ZMap* can either scan at a specified rate or operate with the goal of scanning as quickly as possible based on the available bandwidth on the network. *ZMap* is also extensible through writing custom probe modules that allow arbitrary types of packets to be sent during a scan. We wrote a custom module for *ZMap* that allows us to send *AskNeighbors2* requests over IGMP.

After working carefully with network administrators at our scanning site, we chose a modest scanning rate of 9K packets per second.[1] This moderate rate combined with *ZMap's* random scanning behavior means that we are unlikely to overwhelm any single remote network with traffic. We also realize that observing even a single IGMP packet may come across as potentially alarming on networks that closely monitor traffic and do not expect to observe IGMP. We implemented a blacklist for any complaints we received during our scan.[2] We split the overall scan into 10 slices and updated our blacklist between slices. Finally, we did not probe the multicast address block 224/4, as these probes were answered by our ISP, and therefore do not provide useful insight.

While running our scans with *ZMap*, we simultaneously capture all IGMP packets related to the experiment using *tcpdump* [10] at the border of our network and the wide area network. We then analyze these packet traces to develop our initial understanding of the *Neighbors2* response and responder characteristics. We note that our trace contained all but 458 packets of the 4.2B packets *ZMap* reported transmitting. This shows that there is little loss in both the outgoing edge network and the packet capture process. While we cannot quantify the measurement-based loss on the incoming traffic, the monitor is clearly capable of operating without significant loss of outgoing traffic, and the volume of incoming traffic—263M packets in our initial snapshot, as we discuss below—is an order of magnitude less than the outgoing volume. Therefore, we do not believe measurement-based loss has a significant biasing effect on our dataset.

We define a single response as a series of packets from a single source IP address I that each arrive within 1 second of the previous packet. We chose a 1 second threshold as consistent with previous work [13], and our own data analysis which finds that most response packets arrive within 1 second of sending the probe. Our relaxed definition, which allows responses to keep growing as long as packets arrive within 1 second of the previous packet, ensures we likely capture all responses to a given probe. Note that it is possible that multiple probes may trigger a series of unique responses from a single IP address (i.e., we do find instances where requests to any interface on a router are answered using only a single IP address that is not necessarily the destination of the query). If we issue requests in quick succession that trigger this response behavior, the packets we gather into a single response could have been triggered by multiple requests. However, issuing requests to multiple IPs on the same router this quickly is probabilistically unlikely. Consider the case where a single host responds on behalf of every IP addresses on a given /24, and that the host always responds from the same IP address. Given our scanning rate of 9 Kpps and the random scanning behavior of *ZMap*, we expect that if we have probed a /24 in the past second, the chance of probing that same /24 in the next second is approximately 0.054%. We verified this re-probing rate by analyzing our first slice of data where we find the expectation holds precisely.

Based on our definition of a response, we find that certain routers will send a stream of packets much larger and longer than any reasonable response would be. Sometimes these responses contain thousands of packets and last dozens of minutes. In addition to the above reasoning and analysis, we have manually triggered large streams of packets by sending a single probe to certain routers—i.e., eliminating the chance that the stream is caused by multiple requests. We revisit these large, anomalous responses in § 5.

3.2 Scan Analysis

Table 1 gives an overview of the data collected during our initial scan. Out of the 4.2B IP addresses we probed, we

[1]This rate was chosen based on the tradeoff of scanning speed versus the need to not overwhelm our edge network with scanning traffic, and is based on our particular scanning setup. Scanning at this rate allows our scan to finish in under one week while using only a small portion of the available bandwidth at our edge network.

[2]We received five complaints. The blacklisted prefixes correspond to 132K IP addresses, or about 0.003% of IPv4 address space. Hence, we do not believe the complaints produce a bias in our data collection.

Start Date	Jan 12 2015
End Date	Jan 18 2015
Transmitted Pkts	4.2B
Received Pkts	263M
Responding IPs	305K

Table 1: Overview of data collected.

received responses from 305K hosts.[3] Out of these responding hosts, 8K (2.2%) responded multiple times throughout our scan. This likely indicates that a router responded for multiple interfaces through a single outgoing interface. Additionally, we find 1.6% of responders did not answer direct queries, but did answer on behalf of alternate hosts. At this rate, simple loss of requests or responses likely explains much of the phenomenon.

Given the responses we collected and the addresses we blacklisted, we have an IP-based hit rate of 0.007%. While the hit rate is a small percentage of the IPv4 address space, 305K IP addresses represent a non-trivial number of hosts to leverage during an attack. Further, Table 1 shows that those 305K hosts returned 263M packets—an overall amplification factor of more than 862x. We next turn our attention to understanding the characteristics of responses we observe from the responding hosts.

Figure 1 shows the distribution of byte amplification factor for the responses we observe. About 15% of responding hosts offer no amplification. We observe a median byte amplification factor of 2.4. The largest 6% of responses yield an amplification factor of at least 50 and 1% of responding hosts yield an amplification factor of at least 100. As we are sending packets that are 28 bytes in length, the largest 1% (3K) hosts send responses that are at least 2,800 bytes.

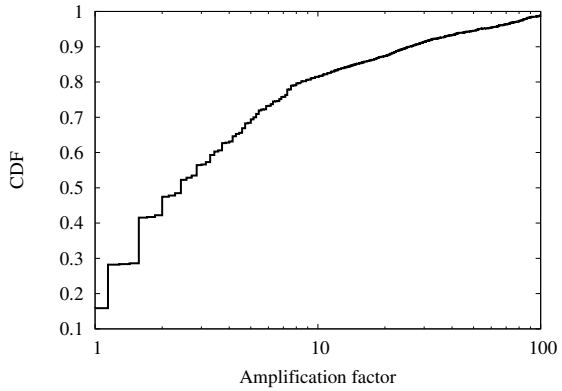

Figure 1: Distribution of byte amplification.

While probes can trigger responses that are large relative to the requests in terms of bytes, responses can also be split across multiple packets. Whereas one potential attack would rely on byte amplification to exhaust a victim's bandwidth, packet amplification could also exhaust the packet processing capacity of the network. Routers have some amount of fixed overhead associated with handling a packet and forwarding it along the correct path, and have a limited number of packet buffering slots available. Inflating the number of extra packets a router must process prevents the router from using its resources to handle legitimate packets. In addition to processing time, extra packets taking up too many slots in a router's buffer could cause legitimate, ongoing connections to have some of their packets dropped due to congestion at the router.

Figure 2 shows the distribution of packets returned in response to a single *AskNeighbors2* probe. For 87% of the responses, we observe no packet amplification. The final 5% of responding hosts send at least 5 packets in response to a single probe. While a specific set of hosts will yield a moderate amount of packet amplification, *AskNeighbors2* packet amplification is not as significant as the byte amplification we observe. Packet amplification is a small enough issue that we do not further consider it in the remainder of this paper.

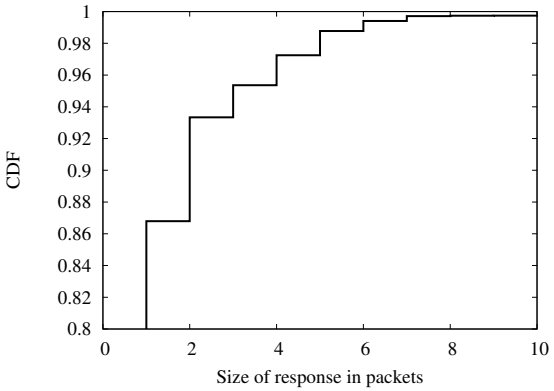

Figure 2: Distribution of packet amplification.

4. RESPONDER LONGEVITY

We next focus our attention on studying the stability of responses from each host over time. A natural question is whether the 305K hosts that respond initially will continue to do so in a consistent manner over time. To understand how routers behave over time, we conducted three additional rounds of probing. For each response from a host H we recorded in our initial snapshot, we send an additional probe to H at times 10, 20, and 30 days after we recorded the initial response from H. We collect packet traces in the same network location as the initial scan.

4.1 Scan Analysis

Out of the 305K hosts that respond to our initial scan, we observe 262K (86%) of them respond to at least one round of our re-probing. We find stability among some routers, as 161K (52.8%) respond to all three rounds of re-probing. For the routers responding to some but not all of our re-probes, 73K (24%) respond to two queries and 28K (9.2%) respond to only one out of three rounds of re-probing. Note that there are 43K (14%) hosts that do not respond to any of our re-probes. This could happen for several reasons, such as (i) the IP address being reassigned to a new router that is not

[3]That is, we received responses that used 305K unique IP source addresses. A given IP address does not necessarily represent a unique router, as routers can have multiple interfaces each with its own IP address. Therefore, we are likely overestimating unique hosts on the network. For the current work and for ease of exposition, we equate an IP address with a host.

Scan	Tot. Routers	Exclusive Routers
Initial	305K	43K
+10 Days	227K	14K
+20 Days	202K	3K
+30 Days	229K	11K

Table 2: Summary of hosts observed during scans.

DVMRP-enabled or openly responding to *AskNeighbors2* requests, (*ii*) filtering of traffic related to our experiment being implemented after our initial scan along the path to the router, (*iii*) the IP address could be an outgoing interface on a router that responds on behalf of its other interfaces but does not respond to probes directly, or (*iv*) response packets from a router could be dropped.[4]

We next consider whether the number of responding hosts we observe decreases over time during our re-probes. Table 2 shows the number of hosts responding to the initial scan and each re-probe. Additionally, the last column indicates the number of routers that respond in only the given round of probing and the initial scan. We find that of 305K hosts that respond to our probes in the initial scan, 202K–228K respond to one of our subsequent requests. We also note that all three re-probes contain hosts that appear exclusively during that re-probe. This shows that there is churn in which hosts will respond and when they respond. Just because a host is unresponsive on one day does not mean it will remain unresponsive in the future. Likewise, a host responding on a given day does not mean it will continue to respond in the future.

4.2 Stable Responders

We now turn our attention to those hosts that respond in each of our re-probes. We refer to these hosts as *stable responders*.[5] Studying stable responders allows us to better assess the security risks involved with having DVMRP-enabled routers that will always respond to requests from arbitrary hosts. Such behavior is advantageous to a potential attacker, who could add stable responders to a "hit list", or a list of known targets to leverage during an attack. Once in possession of a hit list, attackers no longer have to scan for targets to use during an attack, as they can simply leverage hosts on the premade hit list.

Assuming that stable responders represent entries on an IGMP hit list, potential attackers would likely be interested in "guaranteed" amplification from each host. As such, we focus our attention on the minimum amplification factor we observe for each stable responder across our initial scan and three re-probes. Figure 3 shows the distribution of the minimum amplification factor we observe for each stable responder. About 16% of responders yield no amplification in at least one re-probe. We find a median amplification factor of 2.4 (as we did in the initial scan, § 3.2) and that 1.5% of stable responders offer an amplification factor of at least 50. The mode that appears around an amplification factor of 20 is caused by the maximum packet size Cisco routers will send before breaking the response into multiple packets.

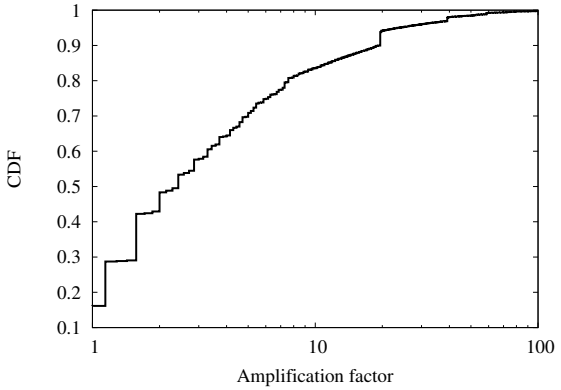

Figure 3: Distribution of byte-amplification factors for stable responders.

One final characteristic of stable responders we consider is how much their response size changes over time. We examine this by determining the minimum and maximum byte amplification that a stable responder with IP address I exhibits, min_I and max_I, respectively. We then plot the distribution of the ratio $max_I : min_I$ across all stable responders as the dotted line in Figure 4. We find that over 84% of stable responders have a consistent response size over a 30-day period. Additionally, we observe that roughly 7% of the routers have size differences that vary by at least a factor of two. Meanwhile, less than 1% of the routers show a discrepancy of at least 10x. These results suggest that while there is some variation, the expected amplification from an IGMP responder is reasonably consistent within a hit list of stable responders.

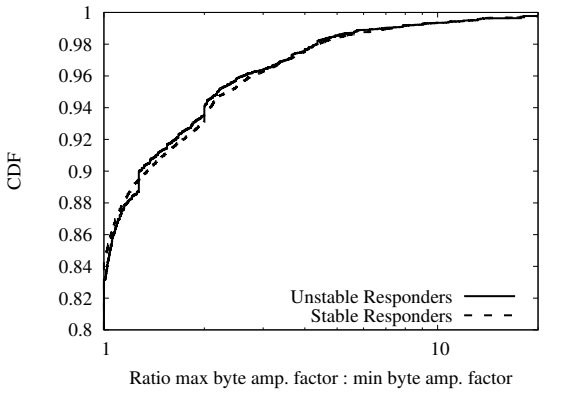

Figure 4: Distribution of ratios of $max_I : min_I$ for each IP address I.

4.3 Unstable Responders

We next turn our attention to hosts that respond during the initial scan, but later are missing a response from at least one of the re-probes. We call these hosts *unstable responders*. Given that we receive no response from the routers in these subsequent probing rounds, we cannot definitively determine why they no longer respond. The reasons are no doubt multiple, including everything from simple loss of query packets

[4]While loss is possible, general loss rates on the Internet are low and would not explain the broad trend of hosts not responding.

[5]Our experiments are conservative in that there may be more responders that would be useful in an attack if an adversary wished to use a wider range of responders, but with less guaranteed success.

to operators upgrading their routers' operating systems to a version with new default settings or applying recommended policy constraints that prevent routers from answering arbitrary *Neighbors2* queries. We seek to understand how unstable responders differ from their stable counterparts. Obviously, unstable responders do not respond as consistently as stable responders, but a natural question to consider is whether unstable responders also differ from stable responders in terms of amplification. We find unstable responders exhibit no amplification 18% of the time. The median byte amplification factor for unstable responders is 2 and 1.2% of unstable responders have a byte amplification of at least 50. The distributions of amplification factors for stable and unstable responders differ by less than an order of magnitude.

Finally, we examine how much byte amplification for unstable responders changes over time for the set of unstable responders that respond during the initial probe and at least one re-probing round. The solid line in Figure 4 shows the distribution of the ratio $max_I : min_I$ for unstable responders is similar to the stable distribution plotted, with only slight variation.

5. ANOMALOUS RESPONSES

During both our initial scan and subsequent re-probes we observe single hosts respond with a stream of *Neighbors2* packets over a period of time lasting up to an hour. Each packet in these streams arrives at our monitor within 1 second of the previous packet, and the streams last for an unpredictable amount of time. The packets in the stream contain no routing information, but they are valid *Neighbors2* responses. Individually each packet would yield no amplification, but together they represent amplification factors that can grow to be millions.

More curious, we cannot always replicate this behavior at will and therefore this behavior remains puzzling. However, we make several comments about these responses:

- Anecdotal evidence [11] documents sustained streams of packets in response to an *AskNeighbors2* request. While evidence suggests that some routers will send hundreds of thousands of packets or more, probing these routers manually does not yield streams of packets at the time of this writing.
- We observe a host that responded with byte-amplification factors of 817K, 1.3M, and 120K for our initial probe, re-probe 1, and re-probe 2, respectively. The large response disappeared in the third re-probe, and manual probes sent to this host's IP address yield single-packet responses at the time of this writing. While we cannot currently trigger this response, the host did display this anomalous behavior over a period of time.
- Through manual testing of anomalous responders, we have been able to identify a host that responds with a stream of packets when we send it a single probe. To better understand this behavior, we began sending this host 1 probe per hour over the course of 1 week (168 probes total). We observe streams of responses for each of the first 135 probes we send to the host. The responses vary in time from 20 seconds to 101 seconds long with a median value of 68 seconds. We receive a minimum of 274 KB in response to a single 28 byte probe, and a median of 696 KB. This corresponds to byte-amplification factors of 9.8K in the minimum case and 24.8K in the median. The maximum amplification factor this host yields is 40.4K, which corresponds to 1.13 MB of data. For the final 33 probes we sent to the host we observe no response packets, and the host remains unresponsive at the time of this writing. This same host had byte-amplification factors of 6K, 20K, and 18K during the initial scan, re-probe 1, and re-probe 3, respectively. It was unresponsive during re-probe 2.

	All Resp.	**Stable Resp.**
Total	305K	161K
/8s	188	185
/16s	9.5K	6.5K
/24s	99K	62K
AS	3.5K	1.9K

Table 3: Responder breadth

The observations above leave us perplexed when it comes to understanding these large responses. We have evidence of large amplification happening in response to single packets across time coming from various hosts. Observing hosts exhibit this behavior across time leads us to believe that the responses are not caused by some sort of measurement artifact. Likewise, being able to manually trigger a stream of responses with a single packet adds confidence that previously observed responses were also triggered by single packets and not measurement errors. On the other hand, we do not currently understand why a host "fixes" itself and stops sending streams of packets in response to a probe, although some candidates may be patching the router or changing its configuration to prevent sending to arbitrary hosts. Another possibility is that a router only exhibits anomalous behavior when it is in a specific, but rare, state and that a bug in a router's software will occasionally be triggered when receiving an *AskNeighbors2* request in this state. We observe 16 hosts that respond with a minimum of 500 packets and a maximum of 8.1M packets across our scans.

In addition to anomalous behavior that appears to be largely unpredictable, we also observe a set of 12 hosts belonging to a single /16 that send large, predictable streams of packets. In response to a single packet, the hosts will respond with 288 identical 464-byte packets. These hosts behave identically in each of our three re-probes and when sending probes manually after the initial set of scans. These responders each have a byte amplification of 4.7K, representing the largest predictable byte amplification factor we observe.[6]

6. RESPONDER CHARACTERISTICS

6.1 Responder Location

Next, we aim to understand how widespread the hosts are that respond to our probes. Table 3 shows our results. In the context of the entire Internet we do not find responders in a large fraction of the networks (there are more than 50K ASes in the routing table [9]). However, we do find

[6]Note, we have sent our findings to the operators of this network.

	All Resp.	Stable Resp.
Mean	87	87
Median	3	3
75^{th} perc.	15	17
95^{th} perc.	199	207
Maximum	19.5K	14.4K

Table 4: Responders per AS

responders in thousands of networks, suggesting that this is not a concentrated phenomenon.

To better understand the concentration of responders, we summarize the number of responders per AS in Table 4. We find the distributions of all and stable responders per AS to be similar. We find that while there are ASes with a significant number of responders—e.g., AS 3292 has 14.4K stable responders—the majority of the ASes have modest numbers of responders. In particular, half the ASes have no more than three responders and three-quarters of ASes have no more than 17 responders. This shows the breadth of the problem and that fixing the issue is not likely a quick fix for only a few network administrators.

Finally, we compared the set of responders to the SpamHaus PBL [20] to understand whether the responders represent infrastructure nodes—as we expect—or end-hosts. The PBL from the date our initial scan started identifies 20% of all responders and 19% of the stable responders as end-hosts. This confirms our intuition that IGMP responders are largely infrastructure nodes.

6.2 Vendors

Neighbors2 responses include a "version" field. According to the specification this should be the protocol version number [6, 8, 5]. We find that only 8% of all responders (and 8% of the stable responder subset) adhere to this part of the specification. Cisco routers instead place their OS major and minor version numbers in this field. This allows us to determine that 80% of all responders and 83% of the stable responders are Cisco routers. This leaves 12% of all responders and 9% of stable responders with some other information in the version number field.[7] The prevalence of Cisco routers in our set of responders suggests that a default setting that provides open responses to *Neighbors2* requests could be at play.

Table 5 shows the breakdown of the IOS versions we find in the responses from Cisco routers. We include only version numbers we observe in at least 1% of the responses. The oldest IOS version we note is 10.0 which was released two decades ago! The most prevalent IOS version in our dataset is 12.x which covers 73.7% of the Cisco routers that responded to our probes. This version of IOS has not been supported by Cisco in four years [1]. The modern 15.x operating systems are responsible for 22.5% of the responders, showing that the problem is not simply a matter of old routers not being upgraded.

7. RATE LIMITING OF RESPONSES

We next turn to the question of whether responders somehow limit the rate at which queries will be answered. Such a limit—and its magnitude—would clearly make the responders less attractive to leverage in an attack. We did not want to launch a high-rate stream of requests at all responders we detected, as then our experiment would itself be an attack. Therefore, we randomly chose 126 responders to probe at various modest rates as a way to get an indication about possible rate limits.[8] The results of these tests are not conclusive, but rather suggestive. We were not comfortable extending the experiment to faster rates.

[7]It is possible that these unclassified hosts are Cisco routers that use a version number not included in the conservative set we match against.

IOS Version	% Cisco Routers
12.2	59.5
12.4	11.7
15.0	9.0
15.1	6.3
15.2	4.6
15.3	2.6
12.0	2.5
10.0	1.3
Other	2.5

Table 5: Breakdown of vulnerable Cisco routers by IOS version.

For each responder we first send a single request. This allows us to characterize a normal response from the given responder. We pause for 5 seconds and then transmit 10 requests as fast as possible. We repeat this cycle of waiting 5 seconds and then sending a volley of requests for volleys of 40, 70 and 100 requests. In total we send 221 requests to each of the 126 responders over roughly 20 seconds—or, roughly 11 pps. We believe this rate, pattern and duration is unlikely to cause difficulties for the responders (and in fact we received no complaints about this experiment).

We find that 107 responders (85%) sent at least 95% of the expected responses across all volleys. Another 6 responders (5%) sent at least 95% of the expected responses to the volley of 100 requests. However, these 6 responders did not meet the 95% threshold for at least one of the smaller volleys and so are limited in some fashion, but not by a simple count of the requests. The final 12 responders (10%) can handle a volley of 70 requests, but not a volley of 100 requests. One of these responders did not answer any of the 100 requests—after answering each of the 70 requests sent in the previous volley. The remaining responders answered at least 56% of the requests, with seven of the responders answering at least 81% of the requests. This small-scale experiment is suggestive that in general responders do not apply rate limits when answering IGMP requests. Finally, we note that Juniper has disclosed that there is no rate limiting of *Neighbors2* responses from their routers [2].

8. ATTACKS

Our discussion so far has aimed to understand *Neighbors2* responders and responses. We now turn to a brief discussion of the attack threat these responders represent.

8.1 Basic Denial of Service Attacks

Consider an attacker that (*i*) controls a moderately-sized botnet of 2K bots, (*ii*) compiles a list of the 48K stable *Neighbors2* responders that offer a minimum byte-amplification of 5, and (*iii*) wishes to exhaust a victim's

[8]Each host in our stable responder list was included in this sample with a probability of 0.08%.

10 Gbps link. The attack requires each bot to send at 0.285 Mbps across 24 responders. Sending in round-robin fashion means that each responder will receive 53 requests per second (well within the capability of the responders as we show in § 7).[9] This will then unleash an aggregate of 10.2 Gbps towards the victim with an expenditure of 570 Mbps across the 2K bots—an amplification of 17.8x.

In addition to a sustained attack designed to consume all the victim's capacity, a pulse attack is also possible [12]. In this attack the aim is to send a pulse of data to momentarily clog the victim's link and therefore force all ongoing connections to reduce their sending rate in response to the congestion. By pulsing at regular intervals the attacker can limit the use of the network capacity for legitimate purposes without expending the effort to constantly fill the victim's network. In addition, attackers could leverage "lensing" to launch such attacks with limited bandwidth [16].

8.2 Infinite Loop Attack

In scrutinizing our data we identified another vulnerability within deployed IGMP implementations. The expected response, if any, to an *AskNeighbors2* request is a *Neighbors2* packet. However, we observe 79 hosts that instead respond to an *AskNeighbors2* request by sending back the same request they received! Such behavior enables an attack on each router exhibiting this behavior. The attacker identifies two misbehaving routers R_1 and R_2 that will respond to an *AskNeighbors2* packet with the same *AskNeighbors2* packet. The attacker sends an *AskNeighbors2* packet to R_1 and spoofs the source IP address of the packet as R_2. R_1 responds to the *AskNeighbors2* request by sending the same request to R_2 (the purported sender of the initial request). Likewise, when R_2 receives the packet from R_1, it will respond to R_1 with an *AskNeighbors2* packet. The routers will continue to circulate the *AskNeighbors2* request back and forth to each other in an infinite loop. Given the connectionless nature of this interaction, a packet that is dropped while being transmitted between the routers will stop the infinitely circulating behavior. However, an attacker can simply introduce additional packets into the infinite loop to combat any packet loss that would occur. In addition to combating packet loss, each additional packet introduced between the two routers increases the severity of the attack. A crafty attacker could start thousands of multi-packet infinite loops by sending packets to each misbehaving router addressed from each of the other misbehaving routers. This will consume capacity and router processing without any ongoing effort from the attacker.

9. CONCLUSION

In this paper we describe a scan of the Internet that identified 305K hosts that respond to IGMP *AskNeighbors2* requests from arbitrary hosts on the Internet. While this type of request has been leveraged to study network topology, our focus is on examining response characteristics as they relate to network security. Our conclusion is that while we do not find vast numbers of IGMP responders—e.g., as compared to open DNS resolvers [19, 3]—the responders we do find represent significant firepower that can be brought to bear in a DDoS attack. We illustrate that a subset of the responders are stable and amplify requests enough to easily saturate a victim's 10 Gbps link. Further, we note that an attacker could use the IGMP responders in concert with a multitude of other attack types (e.g., DNS, NTP, etc. [22]). I.e., attacks such as this easily aggregate together. Fortunately, this attack has a relatively simple fix, as there is little reason for hosts to respond to *AskNeighbors2* requests from arbitrary Internet hosts. As such, we recommend network operators block or ignore these queries except from specific, expected peers.

[9]This particular attack formation is for ease of exposition. Additional—and likely more optimal—configurations are possible.

Acknowledgments

This work was funded in part by NSF grant CNS-1237265. We thank David Johnson for his significant efforts in helping us ensure our scanning and data collection facilities were accurate.

10. REFERENCES

[1] Cisco IOS Software Product Lifecycle Dates & Milestones. http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-software-releases-12-2-mainline/prod_bulletin0900aecd801eda8a.html.

[2] DVMRP Can Be Used to Trigger an Amplification Attack Against a Third Party. https://kb.juniper.net/InfoCenter/index?page=content&id=KB29553.

[3] Open Resolver Project. http://openresolverproject.org/.

[4] A. Aina, J. Akkerhuis, K. Claffy, S. Crocker, D. Karrenberg, J. Ihrn, R. Joffe, M. Kosters, A. Mankin, R. Mohan, et al. SSAC Advisory SAC008 DNS Distributed Denial of Service (DDoS) Attacks, 2006.

[5] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet Group Management Protocol, Version 3, Oct. 2002. RFC 3376.

[6] S. Deering. Host Extensions for IP Multicasting, Aug. 1989. RFC 1112.

[7] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security*, pages 605–620. Citeseer, 2013.

[8] W. Fenner. Internet Group Management Protocol, Version 2, Nov. 1997. RFC 2236.

[9] G. Huston. The 32-bit AS Number Report, Apr. 2016. http://www.potaroo.net/tools/asn32/.

[10] V. Jacobson, C. Leres, and S. McCanne. The tcpdump Manual Page. *Lawrence Berkeley Laboratory*, 1989.

[11] J. Kristoff. DVMRP Ask Neighbors2: an IGMP-based DDoS/Leak Threat, Oct. 2014. https://www.cymru.com/jtk/talks/nanog62-an2.pdf.

[12] A. Kuzmanovic and E. Knightly. Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants). In *ACM SIGCOMM*, Aug. 2003.

[13] P. Mérindol, B. Donnet, J.-J. Pansiot, M. Luckie, and Y. Hyun. MERLIN: MEasure the Router Level of the INternet. In *Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on*, pages 1–8. IEEE, 2011.

[14] P. Mérindol, V. Van den Schrieck, B. Donnet, O. Bonaventure, and J.-J. Pansiot. Quantifying ASes Multiconnectivity Using Multicast Information. In *ACM SIGCOMM Internet Measurement Conference*, 2009.

[15] T. Pusateri. Distance Vector Multicast Routing Protocol, Oct. 2003. Internet-Draft draft-ietf-idmr-dvmrp-v3-11.txt (work in progress).

[16] R. Rasti, M. Murthy, N. Weaver, and V. Paxson. Temporal lensing and its application in pulsing denial-of-service attacks. In *IEEE Symposium on Security and Privacy*, 2015.

[17] C. Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Symposium on Network and Distributed System Security (NDSS)*, 2014.

[18] P. Schmehl. The Microsoft UPnP (Universal Plug and Play) Vulnerability. http://bandwidthco.com/sf_whitepapers/windows/The\%20Microsoft\%20UPnP\%20(Universal\%20Plug\%20and\%20Play)\%20Vulnerability.pdf, 2002.

[19] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. On Measuring the Client-Side DNS Infrastructure. In *ACM Internet Measurement Conference*, Oct. 2013.

[20] SpamHaus. The Policy Block List. https://www.spamhaus.org/pbl/.

[21] C. Systems. Cisco Event Response: Network Time Protocol Amplification Distributed Denial of Service Attacks. http://www.cisco.com/web/about/security/intelligence/ERP-NTP-DDoS.html, Feb. 2014.

[22] P. Technologies. An Analysis of DrDos SNMP/NTP/CHARGEN Reflection Attacks: Part II of the DrDos White Paper Series. http://www.prolexic.com/kcresources/white-paper/white-paper-snmp-ntp-chargen-reflection-attacks-drdos/An_Analysis_of_DrDoS_SNMP-NTP-CHARGEN_Reflection_Attacks_White_Paper_A4_042913.pdf, 2013.

[23] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol, Nov. 1988. RFC 1075.

[24] Zmap. https://zmap.io/.