Characterizing Optimal DNS Amplification Attacks and Effective Mitigation

Douglas C. MacFarland[1], Craig A. Shue[1], and Andrew J. Kalafut[2]

[1] Worcester Polytechnic Institute
[2] Grand Valley State University

Abstract. Attackers have used DNS amplification in over 34% of high-volume DDoS attacks, with some floods exceeding 300Gbps. The best current practices do not help victims during an attack; they are preventative measures that third-party organizations must employ in advance. Unfortunately, there are no incentives for these third parties to follow the recommendations. While practitioners have focused on reducing the number of open DNS resolvers, these efforts do not address the threat posed by authoritative DNS servers.
In this work, we measure and characterize the attack potential associated with DNS amplification, along with the adoption of countermeasures. We then propose and measure a mitigation strategy that organizations can employ. With the help of an upstream ISP, our strategy will allow even poorly provisioned organizations to mitigate massive DNS amplification attacks with only minor performance overheads.

1 Introduction

In 2013 and early 2014, attackers used DNS amplification in 34.9% of high volume DDoS attacks (those creating at least 20Gbps of attack traffic) and in 18.6% of all network DDoS attacks [8]. In mid-March 2013, attackers used DNS amplification to launch a high-profile attack against Spamhaus, with attack traffic volume exceeding 300Gbps [1]. DNS amplification attacks are particularly valuable to attackers for a few reasons: 1) the amplification effect allows attackers to create a disproportionate amount of traffic at the victim, 2) by IP address spoofing and reflection, the attackers can conceal the identities of the attacking systems, preventing them from being blacklisted or cleaned, and 3) the victim cannot blacklist the IP addresses of the reflecting DNS servers without also hindering legitimate DNS resolutions.

In a typical DNS amplification attack, the attacker sends a DNS query packet from an attack system to a DNS server. In the process of creating this query packet, the attacker forges the packet's source IP address field so that it contains the IP address of the targeted victim, rather than the actual sender of the packet. Upon receiving and processing the query packet, the DNS server then dutifully sends a response back to the indicated source address of the query, which in this case is the address of the victim. When the response packet arrives at the victim, the victim will process the packet, realize it is unsolicited, and discard

it. However, at this point, the attack has already succeeded: the DNS response consumed a portion of the victim's bandwidth and computational resources at the victim's DNS resolvers. Even better from the viewpoint of the attacker, since the DNS response packet from the DNS server is larger than the query packet the attacker sent, the attack traffic at the target is increased by a certain amplification factor.

While DNS amplification attacks are well understood, the best defensive strategy is less obvious. In a July 2013 bulletin, the United States Computer Emergency Response Team (US-CERT) made a few recommendations [16]: 1) reduce the number of open DNS resolvers, 2) disable public recursion on authoritative DNS servers, 3) rate limit responses [18], and 4) limit IP address spoofing. Unfortunately, there is little incentive for organizations to employ these recommendations: these actions help other organizations, not the organization performing the remediation. The spoofing prevention measure, in particular, has been encouraged for over a decade, yet over 25% of Autonomous Systems still allow arbitrary IP spoofing on the Internet [2]. Further, these steps are not actionable for an organization under attack.

While these recommendations may be well intentioned, they likely will not have the desired impact. In particular, efforts to reduce the number of open DNS resolvers will not solve the DNS amplification problem: rather than using an open resolver, attackers can simply query authoritative servers directly and still create effective DDoS conditions.

In this work, we make the following contributions:

1. Measure and Characterize the Attack Potential: We perform DNS queries to the authoritative servers for each of the 129 million DNS domains registered in 9 top-level domains (TLDs) to determine the amplification factor associated with four types of queries. We then focus on the highest amplification factor queries that can be issued and characterize the attack volume that could result. We found that we could create an attack of $1,444$ MBytes/second at the target by sending only 44 MBytes/second of attack traffic at the application layer. We found that such attacks could be scaled up, allowing even relatively small botnets to launch damaging attacks, all without the use of open DNS resolvers.

2. Measure the Adoption of Query Rate-Limiting: We randomly sampled 0.5% of the IP addresses for authoritative DNS servers we previously studied and issued repeated queries to the server to determine whether the domain employed query rate limiting, and if so, what settings were used in the configuration. We found that 2.69% of the studied domains employed rate limiting. Of those, 7.38% rate limited at 5 queries per second or less and the remaining 92.62% used a rate limit between 9 and 14 queries per second.

3. Propose and Evaluate a Novel Mitigation Method: We propose and measure a straightforward mitigation approach that targeted organizations can employ to mitigate attacks. We propose organizations employ remote hosting for their authoritative DNS servers. We then propose organizations request upstream filtering of all DNS traffic, mitigating the DDoS attack. To preserve DNS

functionality for the organization, we propose and test a solution to tunnel DNS queries to a remote DNS resolver, such as a remote VM hosted by a cloud provider or ISP. We found that we could automatically activate a remote DNS resolver, activate the tunnel, and forward all local DNS traffic to the remote node in less than 0.67 seconds, on average. All queries would then have a median additional latency of 16ms. Accordingly, our approach will allow organizations to weather extremely high-volume DNS amplification attacks with minimal effort.

2 Background and Related Work

Traditional reflection attacks, such as the Smurf attack [15], simply forge the source IP address of a packet to be the address of the intended victim. The attacker sends the packet to an innocent third-party system called a *reflector*. The reflector then issues a legitimate reply that arrives at the victim. When a large number of attack packets are sent to reflectors, or when a reflector is a broadcast network address for many hosts, the combined volume at the victim can be crippling.

In a 2001 article, Paxson [11] described how reflectors can be used as part of a distributed reflector denial of service (DRDoS) attack. He argued for five possible defenses against the attacks: 1) filter reflected attack traffic at the victim, 2) prevent source address spoofing, 3) detect and block spoofed packets at the reflector, 4) allow traceback to the origin even through the reflector, and 5) detect the attack traffic from the compromised systems. With the exception of the first defense, in which the victim employs filtering, each of these defenses requires a third-party organization to detect and block attack traffic. The specific third-party organization affected depends on the details of the attack (e.g., the origin of the attack and the particular reflectors in use), but each of them must implement the solution. Solutions which require 100% adoption by third-parties are unlikely to succeed, especially when these third-parties have no incentives for adoption. For example, the second option, source address filtering, is comparatively straightforward for organizations to employ, yet over 25% of Autonomous Systems still allow arbitrary IP spoofing on the Internet [2].

Attackers often try to increase the amount of traffic generated by a reflection attack. These attacks, called *amplification attacks*, typically leverage protocol-specific attributes to increase the attack volume. Recent attacks using NTP amplification [12, 17] were able to create floods of 400Gbps against a victim. In the NTP attack, the attacker found a list of susceptible NTP servers and, spoofing the IP address of the victim, issued a query requesting a list of the last 600 clients that accessed the server. These NTP responses were much larger than the query, creating a massive amplification attack against the victim. Earlier this year, Rossow [13] examined 14 different network protocols to look for reflection attacks that yield significant amplification. While Rossow's analysis did include DNS, it was not the focus of the work and the analysis was not as comprehensive as our own; we compare and contrast our results in the appropriate sections of our paper. Kührer *et al.* discuss the prevalence of DNS amplifiers, compared to

other UDP-based protocols, and discusses fingerprinting techniques [10]; however, they do not expand on the amplification results. The solutions they propose focus on efficient identification, the notification of vulnerable amplifiers for various protocols, and on curtailing ASes that allow spoofing. Their approach is orthogonal to our own solution.

US-CERT recommends that organizations focus on eliminating open DNS resolvers [16], which echoes RFC 5358 [5]. However, this advice ignores the hundreds of thousands of authoritative DNS servers that are, by design, required to answer DNS queries to anyone who asks. These servers are well provisioned and capable of handling large volumes of traffic [7]. Attackers could use these servers to launch crippling attacks, even without using open resolvers. Accordingly, we focus on the risks associated with authoritative servers in this work.

Other reflector and amplification attacks can be damaging. However, we focus on DNS amplification because the protocol is widely used and the amplification attack can be indistinguishable from legitimate usage. Further, measures such as filtering, which may be used to mitigate other amplification attacks, would have unacceptable consequences for DNS (such as leaving a victim without the ability to resolve host names).

3 DNS Amplification Potential

We begin by determining the inherent DNS amplification risks associated with today's DNS authoritative servers. We examined over 129 million domains and over 1.1 million unique DNS authoritative servers to determine the amplification factor associated with common DNS queries.

3.1 Data Collection

As a starting point for our measurements, we used a DNS zone snapshot from July 2, 2013 for a collection of nine generic top-level domains (gTLDs). We obtained the DNS zone files for the `biz`, `com`, `info`, `mobi`, `name`, `net`, `org`, `travel`, and `us` zones from their respective maintainers. These zone files list the domain names and associated name servers for each of the domains registered under these TLDs. We collected records for $129,300,870$ unique domains, each of which had one or more name servers listed, by host name, as authoritative for the domain. In total, $2,771,768$ unique host names were listed as name servers, which upon resolving to IP addresses yielded $1,101,446$ unique name server IP addresses. We collected these records in a distributed fashion and used delays between queries to minimize the impact on other users and the queried servers. We had an opt-out approach for queried providers; however, we did not receive any out-out requests.

Using these zone files, we constructed a set of pairs of the form (`domain_name`, `NS_IP_address`). This resulted in $363,263,970$ unique pairs. For each pair, we issued a set of DNS queries to the associated name server for the domain name without indicating any subdomains or hosts (e.g. a query for `example.com`).

Based on the results reported in our prior work [9], we knew that A records, which provide the IPv4 address for an indicated host name, would be quite common. Queries for A records are commonly issued by hosts on the Internet and are not be likely to be noticed by network operators. Recent DNS amplification attacks have used the ANY record type in their queries, which asks a name server to return any records associated with a host name. Since we used the base domain name, an ANY query would be likely to return the SOA, NS, and MX records associated with the domain, along with an A record for the host name. These four records were the most common in DNS zones in our prior work [9]. While the ANY query may yield the most records, such queries are not as commonly used by normal Internet hosts and their role in attacks may make them more noticeable when queried. Accordingly, we collect data for both the ANY query and the more common A query.

Traditional DNS packets are limited to a maximum length of 512 bytes at the application layer. However, the extension mechanisms for DNS (EDNS) [6] allow larger DNS packets if supported by both the resolver and authoritative server. To communicate support, the resolver sends a pseudo-resource record, OPT, that indicates the supported packet size. The OPT record can indicate DNSSEC support [4], indicating the server should send any associated DNSSEC records.

Attackers have a tactical consideration with using EDNS. Including an OPT record requires the attacker to include an additional 11 byte record in the query. If the server does not support EDNS, or the response would fit within the standard 512 byte limit, the response size remains the same. Accordingly, EDNS use would decrease the amplification factor associated with the query. However, if the EDNS support results in a larger response, it may dwarf the size of the OPT record and increase the amplification factor. Accordingly, we measure amplification, both with and without EDNS enabled (indicating a maximum application layer packet size of 4096 bytes as recommended by RFC 6891).

We also issued queries for AAAA records associated with IPv6. However, they were not widely used and did not provide a meaningful amplification over the other queries types. Accordingly, we omit any further discuss of these records.

In summary, we issued the following DNS queries for each domain: 1) A record without EDNS or DNSSEC support, 2) A record with EDNS and DNSSEC support, 3) ANY record without EDNS or DNSSEC support, and 4) ANY record with EDNS and DNSSEC support.

We issued the queries from July 29, 2013 to Aug. 1, 2013. To perform the massive number of queries quickly, we used a dedicated querying process and a separate packet capture process to collect and store each of the DNS responses sent to our server. Some packets may have been dropped, but for expediency, we accepted these losses and did not attempt a retransmission. Accordingly, each of the results we report will be conservative estimates of possible amplification.

3.2 Analysis of Servers and DNS Responses

We now examine the DNS responses. We exclude analysis of malformed packets, since we are unable to properly parse them, which amounts to 0.07% of the data

set. In Table 1, we show the overall success rates of our queries and statistics on the degree of amplification resulting from each. We calculated all packet sizes at the application layer (i.e., the DNS headers and payload). This excludes extraneous factors, such as the IPv4/IPv6 or UDP/TCP headers, and focuses on DNS. In Figure 1, we show the amplification factor distribution for each of the data sets. For each query type, the attacker receives at least a 129% increase in traffic volume at the application layer using DNS reflection. However, only 0.35% of `A` record queries and 1.54% of `ANY` queries had a packet size greater than the 512 byte limit when EDNS was enabled. Accordingly, the query overhead of using EDNS reduced the average amplification factor for both the `A` and the `ANY` groups. Simply put, an attacker does not benefit from using EDNS in the average case since few responses must be shortened to fit within 512 bytes.

Query Type		Response	Top Million Queries Total (MB)		Amplification Ratio	
Record	Uses EDNS	Rate	Sent	Received	All Queries	Top 1 million
`A`	no	90%	34	485	2.74	14.42
`A`	yes	89%	44	725	2.29	**16.37**
`ANY`	no	84%	35	534	6.22	15.32
`ANY`	yes	85%	44	1,444	5.03	**32.77**

Table 1. DNS Responses to Queries. Results are presented in the aggregate along with statistics on the top 1 million largest responses of each group.

While this degree of amplification may be worthwhile for an attacker, a more potent strategy may be to focus on the queries and responses that yield the greatest amplification factor. To highlight the benefits of doing so, we provide statistics on the top 1 million packets, by response size, of each data set in Table 1. These packets make up roughly 0.3% of each data set. Additionally, while EDNS did not help an attacker sending queries to random domains, it does benefit the attacker who focuses on those providing the most amplification. In both groups, EDNS yielded a notable increase in amplification among the million largest amplifying responses. This selective querying can help an attacker increase the amplification ratio to over 14.42 in the worst case and up to 32.77 in the best case.

The attacker receives the best amplification while using `ANY` queries, but we note that this record type may raise suspicions. An attacker that wishes to use `A` record queries to avoid detection can still achieve an amplification factor of 16.37. As an anecdotal result, in issuing the roughly 1.5 billion DNS queries associated with this study, our organization was contacted only once by a queried organization. That report was from an automated system indicating that the `ANY` query it received from our querying host may be the result of an attacker launching a reflection attack against us. Organizations may begin filtering `ANY` queries to reduce the amplification factor, but the amplification potential of `A` queries is unlikely to change.

To provide context for these results, we consider the theoretical maximum amplification, at the application layer, for DNS with EDNS using the recommended maximum response size of 4096 bytes. The DNS header itself is 12

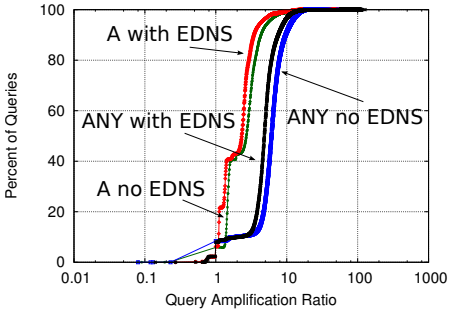

Fig. 1. Cumulative distribution function of the amplification ratio compared to the percent of queries for each data set.

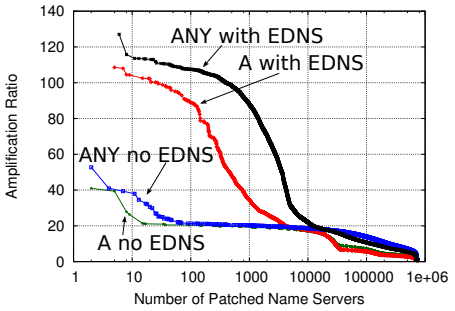

Fig. 2. Amplification ratios ordered from the most amplifying server to the least. Some data points are aggregated for readability.

bytes, with an additional $n+5$ bytes for a query record, with a domain name of length n, and another additional 11 bytes for the OPT record to enable EDNS. The average maximum amplification with EDNS can then be expressed as $\frac{4096}{N+28}$ where N is the average domain name length in the queries. In our dataset, the average domain name length was 17 characters, which yields a maximum average amplification of roughly 91.02. Our overall amplifications are much lower than this, indicating most queried systems are not providing maximum-sized responses. However, looking at our top 10% of amplifying name servers, we see an amplification of 78.13 indicating longer domain names on average with nearly maximum length responses. These highly amplifying servers are closer to the ratios reported in Rossow's DNS_{NS} set [13]. However, our dataset-wide averages are much lower than those in Rossow's data set. These are likely due to our different methodologies: Rossow used the Common Crawl project while we used the zone files themselves. Our data sets are larger and we did not pre-filter based on the deployment of DNSSEC, reducing potential sources of bias.

While attackers want to maximize the amplification factors associated with attacks, they must also ensure they use a large, distributed base of reflectors. If the attackers focus on a small number of highly amplifying reflectors, the reflector bandwidth may become a bottleneck. Even worse, the defenders may be able to filter a small number of reflector IP addresses with little collateral damage. To highlight this point, we note that although we received responses from $669,090$ reflecting name servers, a much smaller pool of servers are responsible for the 1 million highest amplifying queries. For the top 1 million `A` record queries, the number of servers ranges from $24,782$ in the "without EDNS" group to $24,841$ servers in the "with EDNS" group. For the top 1 million `ANY` queries, the number of servers ranges from $22,508$ in the "without EDNS" group to $28,101$ in the "with EDNS" group. In other words, less than 3.8% of authoritative name servers are associated with the highest degrees of amplification. In Figure 2, we demonstrate the amplification ratios associated with each name server.

3.3 Impact of Record Type on Response Size

In Table 2, we show the contributions each resource record makes to the typical DNS packet from the Top 1 million EDNS groups. Attackers may consider which record types have the largest payload for the response and compose queries to elicit these responses. Not all record types are present in each packet. For example, the `SOA` record typically signals that no valid records are being returned. Thus, it is unsurprising it typically represents a large percentage of the responses where it appears. Likewise records associated with DNSSEC tend to be large, constituting a majority of the packet size in the instances where those records occur.

Interestingly, the use of DNSSEC to ensure the authenticity of DNS records has the unintended consequence of improving DNS amplification attacks. As one countermeasure, DNS servers may choose to apply rate-limiting separately to DNSSEC records. If a server continually asks for a response, the servers may discontinue providing DNSSEC records in duplicate responses before cutting the server off entirely. This would effectively decrease the amplification factor of an attack. However, it would limit clients' ability to get authenticated records in cases of high DNS packet loss. Operators may wish to consider these tradeoffs.

	Packet Bytes (Percent)		Packet Occurrence %	
Record Type	`A`	`ANY`	`A`	`ANY`
`A`	171 (22.13%)	115 (7.63%)	87.2%	97.7%
`AAAA`	158 (19.60%)	181 (15.76%)	60.7%	48.8%
`NS`	220 (28.54%)	126 (8.39%)	85.9%	99.5%
`SOA`	70 (11.10%)	63 (3.37%)	12.5%	67.1%
`TXT`	-	141 (9.17%)	-	19.1%
All DNSSEC	623 (71.3%)	1,688 (84.1%)	40.2%	60.0%
`RRSIG`	590 (67.5%)	1,308 (65.2%)	40.2%	60.0%
`DNSKEY`	-	444 (20.8%)	-	47.4%
`NSEC3`	89 (14.4%)	-	11.8%	-

Table 2. Average number of bytes by resource record type for Top 1 million EDNS groups, as well as the occurrence percentages. We omit negligible results for readability.

4 Measuring the Adoption of DNS Rate Limiting

A recent standard specified the rate-limiting of DNS responses at the DNS server to limit the use of DNS amplification in practice [18]. US CERT recommended organizations employ such rate-limiting, where possible, with a limit of five identical responses to the same origin per second [16]. However, CERT acknowledged that some popular DNS servers, notably Microsoft's DNS server, lack response rate limiting functionality, making rate-limiting impractical for many organizations. At the time of writing, this repeated response rate-limiting is the only standardized scheme available at DNS servers. We thus focus our measurement study on this approach.

CERT also acknowledged that rate-limiting may cause legitimate DNS queries to go unanswered if there is significant packet loss or other patterns. In our own

prior work [14], where we monitored the DNS queries being issued to the authoritative servers at the Oak Ridge National Laboratory, we found that over 26,000 DNS resolvers re-issued a repeated query before the expiration of the five-minute TTL associated with the record. We found about 35% of the repeated queries were issued within the first 10 seconds of the original resolution request, likely due to DNS packet loss. Further, we saw that some large Internet service providers load balanced their clients' DNS requests across caching DNS resolvers on contiguous IP addresses. Because the DNS rate limiting standard recommends rate-limiting at the /24 network prefix, it is possible that the combination of packet loss and load balancing will cause legitimate servers to exceed the rate-limit. This will deny clients access to the organization's services. Organizations have an incentive to avoid rate limiting or to set a high rate-limit value to avoid losing business or negatively affecting their customers.

To determine the impact of rate limiting, we used a random 0.5% sample of name servers from our previous study and issued a set of repeated queries to each to find out what limit, if any, the server used for repeated requests. We issued these queries on May 3, 2014. We used an iterative process, ranging from 3 repeated queries to 15 repeated queries, with all queries in a set being issued within a single millisecond. Between iterations, we delay roughly 6 minutes to ensure any rate-limits are reset.

Using this methodology, we declare a particular name server as employing response rate limiting if there is a consecutive sequence from some number, x, to our limit of 15 in which each set of queries is missing at least one response. However, if a set of y queries, where $y > x$, successfully receives all of its responses, it is unlikely that the server uses a rate limit of x, since rate limiting is deterministic by nature. We note that this is a conservative approach, which may cause us to overestimate rate limiting adoption, since some responses could be lost due to chance. However, our methodology will not detect limits set at more than 15 queries a second.

In doing this probing, we found only 149 (2.69%) of the studied name servers employed rate limiting. Of those, 7.38% rate limited at 5 queries/second or less. The remaining 92.62% used a rate limit between 9 and 14 queries/second.

These results show that rate-limiting is rarely used in practice and thus is unlikely to be a significant factor in a DNS amplification attack.

5 Countermeasure: Tunnel to Remote Resolver

Under normal flooding-based DDoS attacks, the victim can employ filtering at the victim's organization. However, victim organizations often also enlist filtering support from the organization's upstream Internet provider. These providers often have greater capacity and can employ filters before the traffic would reach the organization's last-mile link, which is often a bottleneck link. These providers can also employ such filtering at each ingress router to achieve more scalable, distributed filtering for providers with many peering points.

While a similar approach could also be used to filter all DNS response packets destined to the victim organization, this would also prevent legitimate DNS traffic both to and from that organization. Inbound traffic to the organization's authoritative DNS servers can be outsourced to one of the many entities, such as CloudFlare [3], which offer robust, off-site DNS hosting services using anycasting techniques. Since these approaches only focus on protecting externally accessible resources, they do not protect resolvers performing outbound local DNS resolutions.

We propose to address this problem in a simple way: create an off-premises DNS resolver for the organization and create a tunnel, using virtual private network protocols such as IPSec or SSL, between the off-premise resolver and the organization's on-site DNS resolver. We can then configure the on-site resolver to forward all DNS requests through the tunnel to the off-site DNS resolver while configuring the off-site resolver to operate recursively on behalf of the organization. Organizations could then simply request their upstream Internet providers to filter all DNS response traffic to the organization. This will filter the attack traffic, but will not affect the tunneled traffic between the resolvers, allowing organizations to maintain full DNS resolver functionality.

Many cloud providers would allow an organization to cheaply store and run a virtual machine that acts as an off-site DNS resolver. Since the resolver requires minimal computational resources, such hosting would be widely available for less than a dollar per day of use. As long as the organization's upstream provider can filter the attack, organizations could shrug off DNS amplification attacks of arbitrary size with minimal expense. With widespread adoption, the value of amplification attacks would decrease for attackers and their use may decline.

To demonstrate the feasibility of the approach, we used PlanetLab to host a DNS resolver off-site. The remote node was located in Rhode Island, USA, while our local resolver was hosted at our organization in Massachusetts, USA. We used BIND 9.5 as the DNS software on both our local resolver and on the remote PlanetLab resolver. We used OpenSSL to create an encrypted tunnel between the resolvers. We pre-install the DNS and OpenSSL software on each machine.

We then measured the amount of time required to transition from the resolver performing queries locally to performing the queries through the remote resolver. We found that our solution's average start time was 1.36 seconds across 10,000 trials with a 0.55 second standard deviation. This overall time is the sum of the time required to start the remote BIND instance, establish the SSL tunnel, alter the configuration file on the local BIND resolver, and to reload the local resolver to apply these changes. We also determine the client's perspective of perceived downtime during the switch to the solution, after it has been set up, using a host that issued a query every 100ms. Across 10,000 runs, it took an average of an additional 0.66 seconds (standard deviation of 0.81 seconds) from initiating the change until the first response was received by the client.

While using a remote resolver, the latency associated with each query increased to accommodate the propagation delays between the local and remote

resolvers, as shown in Table 3. This had an impact on the latency for lookups. We first measured the delay between issuing a DNS query and receiving its response at the remote machine (which we label the baseline). We then measured the delay between issuing a DNS query and receiving the response at the local resolver, which forwards the query over the encrypted tunnel and to the remote machine for a recursive resolution (which we label the solution). The mean additional latency was 16ms. Naturally, the geographical location and connectivity of the remote resolver will impact the overall latency. However, we can see that the overhead of the solution itself is minimal.

	Query Response Time (ms)			Standard	Number of
Approach	Minimum	Median	Mean	Deviation	Queries
Baseline	7	69	128	166	1,547
Forwarded	22	85	94	62	1,344

Table 3. Latency comparison of DNS resolutions on directly from the remote resolver to those forwarded by a targeted network to the remote resolver.

We note that the adversary could learn about the victim's use of a remote resolver by having a client inside the victim's network, which can cause queries to traverse the remote resolver, and by operating an authoritative server that would be queried by the remote resolver. However, the victim can easily adapt to this by creating N remote resolvers, requiring the attacker to divide their resources. The victim organization may also monitor the attack, discover the colluding entities, and secure the internal client.

While this solution does require the cooperation of a third-party, that third-party is the victim organization's ISP, which has a financial interest in assisting its customer. Furthermore, the involvement of the ISP is minimal, constituting the addition of a simple filter rule.

6 Conclusion

In this work, we analyze the attack potential associated with DNS amplification attacks that focus on using authoritative servers as amplifiers. We find that attackers can launch damaging attacks of $1,444$ MBytes/second of traffic at the target by sending only 44 MBytes/second of attack traffic from the source, and that botnets could scale up such attacks easily. We find that less than 3.8% of authoritative servers are responsible for the highest amplification factors. Further, we note that DNSSEC played a significant role in amplification: by securing the DNS infrastructure, defenders are increasing the amplification potential of DNS reflector attacks. Further, we note that DNS response rate limiting has minimal adoption, with less than 3% of name servers using the approach.

While much discussion has focused on open resolvers, they functionally serve as distributed mirrors of the top amplifying authoritative servers. These resolvers could also let attackers bypass rate-limiting at servers; however, with less than 3% of servers using rate-limiting, open resolvers only seem valuable to have a larger base to distribute attacks.

While attackers have powerful tools at their disposal, we provide a simple mechanism that allows a victim organization to mitigate an on-going attack while incurring only modest latency increases in the organization's own DNS queries. Further, we note that organizations may be able to decrease their role in DNS amplification attacks by rate-limiting DNSSEC responses when repeatedly queried by a single source.

References

1. Bright, P.: Spamhaus DDoS grows to Internet-threatening size. http://arstechnica.com/security/2013/03/spamhaus-ddos-grows-to-internet-threatening-size/ (March 2013)
2. Center for Measurement and Analysis of Network Data, Naval Postgraduate School: Spoofer project: State of IP spoofing. http://spoofer.cmand.org/summary.php (February 2014)
3. CloudFlare: Cloudflare advanced ddos protection. https://www.cloudflare.com/ddos
4. Conrad, D.: Indicating resolver support of DNSSEC. IETF RFC 3225 (December 2001)
5. Damas, J., Neves, F.: Preventing use of recursive nameservers in reflector attacks. IETF RFC 5358 (October 2008)
6. Damas, J., Vixie, P.: Extension mechanisms for DNS (EDNS(0)). IETF RFC 6891 (April 2013)
7. Elz, R., Bush, R., Bradner, S., Patton, M.: Selection and operation of secondary dns servers. IETF RFC 2182 (July 1997)
8. Incapsula, Inc.: 2013-2014 ddos threat landscape report. http://www.imperva.com/docs/RPT_2013-2014_ddos_threat_landscape.pdf (April 2014)
9. Kalafut, A.J., Shue, C.A., Gupta, M.: Touring DNS open houses for trends and configurations. IEEE/ACM Transactions on Networking PP(99), 1 (2011)
10. Kührer, M., Hupperich, T., Rossow, C., Holz, T.: Exit from hell? reducing the impact of amplification ddos attacks. In: USENIX Security Symposium (2014)
11. Paxson, V.: An analysis of using reflectors for distributed denial-of-service attacks. ACM SIGCOMM Computer Communication Review 31(3), 38–47 (2001)
12. Prince, M.: Technical details behind a 400gbps NTP amplification DDoS attack. http://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack (February 2014)
13. Rossow, C.: Amplification hell: Revisiting network protocols for DDoS abuse. In: Network and Distributed System Security (NDSS) Symposium (2014)
14. Shue, C., Kalafut, A.: Resolvers revealed: Characterizing DNS resolvers and their clients. ACM Transactions on Internet Technology (TOIT) 12(4) (July 2013)
15. US-CERT: Smurf ip denial-of-service attacks. Advisory (CA-1998-01): http://www.cert.org/historical/advisories/CA-1998-01.cfm (January 1998)
16. US-CERT: Dns amplification attacks. Alert (TA13-088A): https://www.us-cert.gov/ncas/alerts/TA13-088A (July 2013)
17. US-CERT: NTP amplification attacks using CVE-2013-5211. Alert (TA14-013A) (January 2014)
18. Vixie, P., Schryver, V.: Dns response rate limiting (DNS RRL). http://ss.vix.su/~vixie/isc-tn-2012-1.txt (April 2012)