DNS Amplification Attacks: Obsolete or Potent Threats?

Mohit Kolli

Abstract

DNS Amplification attacks are a specific flavor of cybersecurity attack. These exploit the current implementation of the internet architecture. DNS amplification attacks allow for bad actors to take down the servers of a victim by overloading them with network traffic. My research intends to analyze the DNS amplification attack and see whether they are still something to be actively looked at as a dangerous threat to high profile targets. This will be done by looking into the history of DNS amplification attacks, seeing how the attack works from a network perspective, looking at real world utilizations of the attack, approaches to detection, and possible ways the attack may be modified in the future with improvements in AI.

Keywords: cybersecurity, DNS Amplification, AI, detection

Section 1. Introduction

I will be analyzing the potency of DNS amplification attacks in the modern day. While the attacks are fairly old, being first proposed in 1999, they have been consistently used by bad actors for many years (Ryba et al, 2015). With the existence of so many flavors of cybersecurity and even DNS based attacks, it is important to analyze the potency of DNS amplification attacks in the current day, in order to see there are benefits in further research in defense methods against these attacks.

DNS amplification attacks have the large potential to evolve. Many of the current mitigation methods for DNS based attacks still cause problems for end users due to not fully stopping the attack. This means a more sophisticated or larger DNS amplification attack can still cause servers to be taken down. Furthermore, the advent of Artificial Intelligence (AI) based systems allow for more advanced controlling of botnets, which can be used by bad actors to more efficiently take down target servers. While there have been no large scale DNS amplification attacks that utilize AI, just the potential of these attacks in the future warrant further research into the topic.

In my approach to seeing if DNS amplification attacks are obsolete or remain a potent threat to various users, I will first explore the history of amplification attacks. I will see how attack size and amplification attack type have evolved since they were first proposed. Next, I will look at a general overview of DDoS attacks and briefly look at some common flavors of the attacks. This will be followed by a network level analysis of how the DNS amplification attack actually works. After this, I will look at a real application of a DNS amplification attack, see how the attack was carried out, and how it was defended against. Afterwards, I will look at the various ways that DNS amplification attacks are defended against and mitigated. This will be followed

by a brief overview of some of my suggestions for how DNS amplification attacks can be defended against. I will then look at some of the ways DNS amplification attacks may evolve in the future, as well as how defense methods may evolve in the future with the advent of AI.

Section 2. Background:

In this section I will be going through an overview of DDoS attacks, as well as explaining the history behind DNS amplification attacks, and how they functionally work.

Section 2.1 Overview of DDoS attacks.

A distributed denial of service attack is a type of cyber attack where a bad actor takes down a server through a flood of internet traffic, distributed across multiple servers in order to try and bypass detection. This is done to prevent a server from functioning as normal or to “deny service”. One common type of DDoS attack is a DNS amplification attack.

Prior to DDoS attacks, Denial of Service (DOS) attacks were from a single computer. Since these are relatively easy to detect and mitigate, attacks evolved into DDoS attacks that were done from multiple computers, commonly controlled via botnets (Brooks et al, 2022).

One flavor of DDoS attack is the volumetric attack. These use a large volume of traffic generated from the various computers in a botnet (Osterweil et al, 2019). The traffic is used to flood a target server, taking it down through the saturation of network channels. An example of this attack are User Datagram Protocol (UDP) floods, which flood the target with traffic in order to use up the bandwidth of network channels.

Another flavor of DNS attacks are protocol attacks, these target the network layers of target servers with malicious connection requests. Some examples of these are SYN floods, which utilize Transmission Control Protocol (TCP), flooding the target with false connection

requests. More detail on how TCP works is in section 5 (My suggestions for defending against DNS amplification attacks).

Application layer attacks open connections and applications in the aforementioned connections in order to consume resources on the target server. An example of this attack is the page flood attack that overwhelms the target with HTTP requests.

Section 2.2 Brief history of amplification attacks:

In 1999, the cybersecurity nonprofit AusCERT warned about the potential for DNS based amplification attacks utilizing spoofed IP addresses (Ryba et al, 2015). CIAC followed up in the year 2000 with a warning about generalized DDoS attacks utilizing a botnet in order to send packets from more than one location. At the annual hacker convention DEF CON 14 in 2006, info securities experts Randal Vaughn and Gadi Evron presented information about DNS Amplification attacks. In March 2013, there was the first truly notable use of the DNS amplification attack against the anti spam blocklist company SpamHaus. The attack peaked at 300gbps of traffic from a large botnet.

Shortly afterwards in May 2013, there was a 167 gbps DNS attack against Massachusetts Institute of Technology (MIT) servers (Ryba et al, 2015). The attack was large enough to take down the MIT mail servers, as well as a few MIT webpages. In February of 2014, cloudflare reported a 400 gbps amplification attack that relied on NTP amplification. NTP amplification attacks are similar to DNS amplification attacks, in that the attack relies on amplification of packets through server responses.

The cloud computing company Akamai reported a Simple Service Discovery Protocol (SSDP) based amplification attack in October of 2013, peaking at 54 gbps of traffic. In February

of 2018, Github reported a DDoS attack peaking at 1.3 tbps of traffic (Kottler, 2018). This attack was another type of amplification attack, a memcached DDoS attack. Memcached is a system used to cache data in a database for the purpose of speeding up websites and networks. In a memcached DDoS attack, vulnerabilities in this system are abused in order to create an attack. More recently in 2020, AWS mitigated a 2.3 tbps attack against an unknown customer that utilized the Connection-less Lightweight Directory Access Protocol or CLDAP (Bhardwaj et al, 2021). It is important to note the increasing size and formidability of attacks as they become more and more sophisticated over time.

Section 2.3 DNS amplification: Technical description:

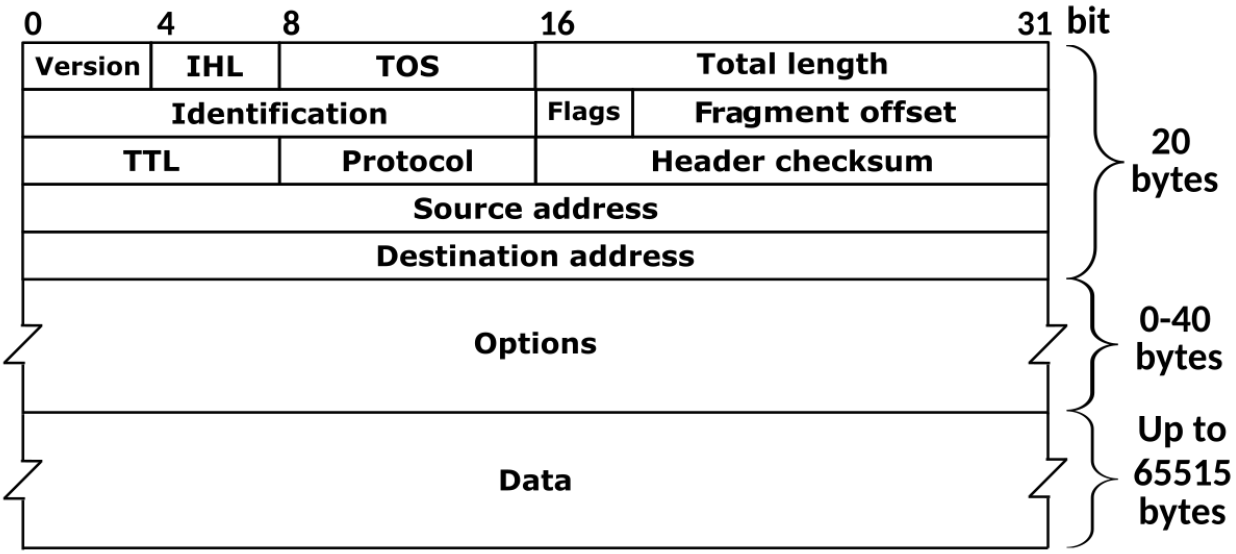
The job of a DNS is to fetch the IP address of a url that is registered. There are multiple steps to this happening as there are specific parts of a DNS request. The recursive resolver first checks its cache, then contacts the root name server, which passes on the information of the top level domain name server, .com for example, to the resolver. The resolver then checks the TLD server which has the IP of the authoritative name server. After the resolver reaches the authoritative name server, it is finally able to find the IP of the web server and sends this to the end user (van der Toorn et al, 2022).

IP spoofing is the modification of the source IP header of a network packet in order to change how the IP is perceived by routers and DNS (Ehrenkranz et al, 2009). In figure 1, you can see the structure of a network packet, with the source IP section preceding the destination IP section. The modification of the source IP can be done with a tool like scapy, changing the source IP address and then sending the packet through a network card where it will be routed

as normal. Since the IP address is spoofed, the DNS response will be sent to whatever IP address is given in the network packet (Ehrenkranz et al, 2009).

Figure 1

How a network packet is structured



Note. From RFC 791, IP Protocol, DARPA Internet Program Protocol Specification [Infographic], by M. Bakni, 1981, IETF, https://tools.ietf.org/html/rfc791, CC by 4.0

Once IP spoofing occurs, bad actors send a DNS registry request for all the IP addresses of a specific domain with a lot of registered subdomains (Anagnostopoulos et al, 2013). This registry request is very small on its own, but the response given by the DNS is large, which is where the amplification part of the attack comes from. The response given by the DNS goes to the end user due to the previous IP spoofing. The bad actor can repeat this process of spoofing IP addresses and giving registry requests until the target's servers are overloaded by DNS registry responses and slow down or crash. In figure 2, you can see how a small stream of DNS

queries from bad actors can lead to a larger amplified attack. In the figure a 3mbps stream of DNS registry requests turns into a 300mbps attack. As can be seen in the diagram, DNS amplification attacks are rarely done from only one computer at a time, rather they are done through many computers and usually controlled via a botnet. Lower computers in the botnet hierarchy take orders from the computers higher up in the botnet, known as botnet controllers. In figure 3, a DNS amplification attack can be seen in the larger context of a networking diagram. In this diagram you can more easily see how the DNS amplification attack has to go through each stage of DNS resolution (the process where urls are converted to an IP address).

Figure 2

The steps behind and basic structure of a DNS amplification attack

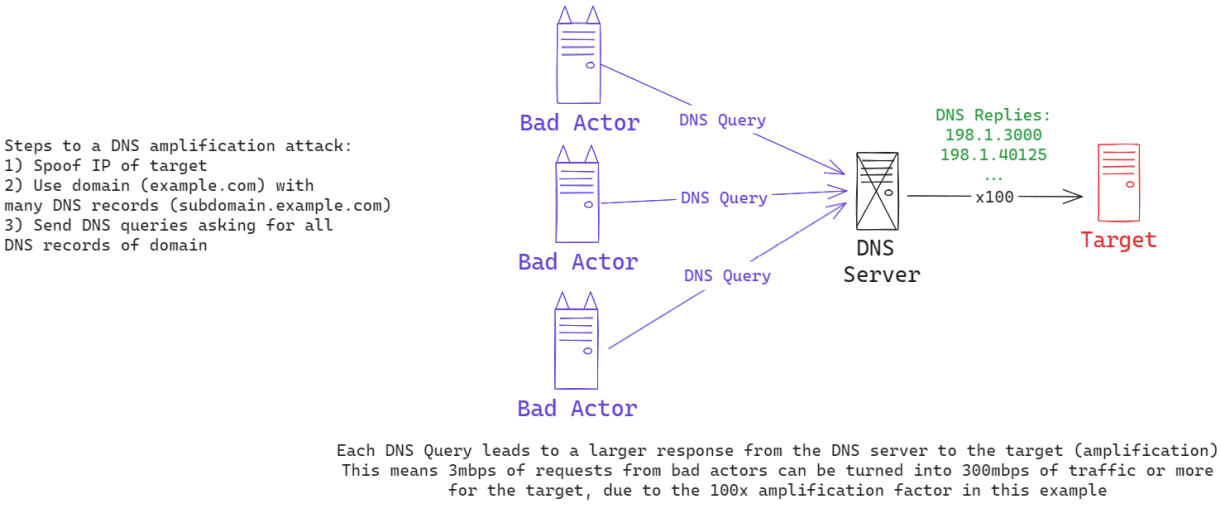
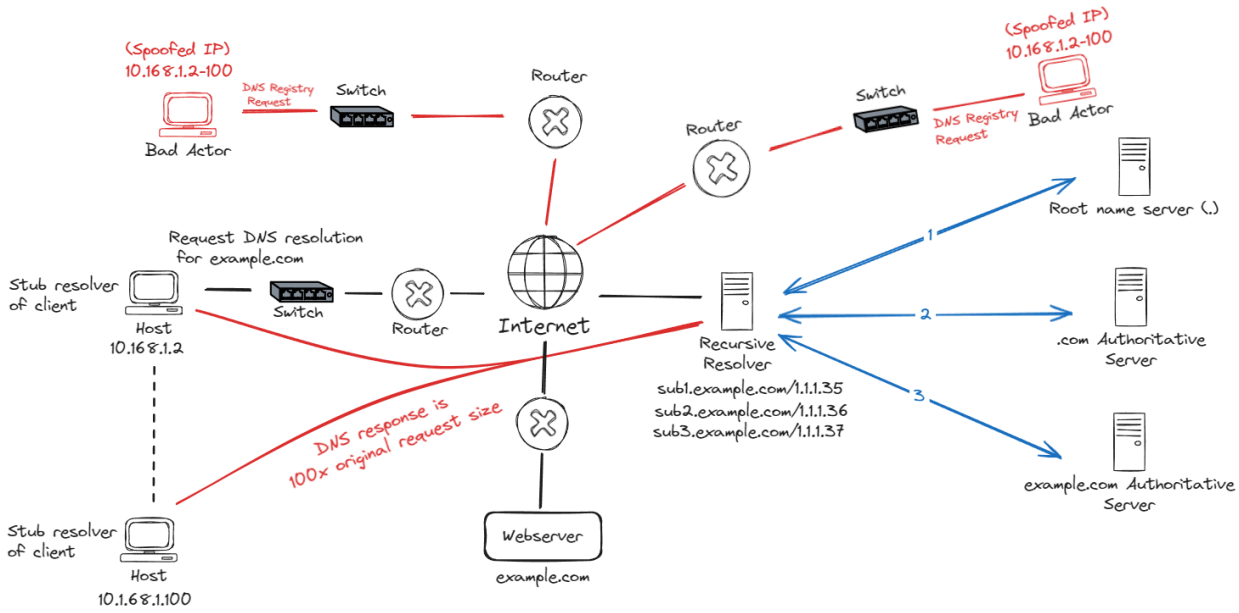


Figure 3

A DNS amplification attack within a larger networking diagram



Section 3. Case study

One example of a DNS Amplification attack being used to take down a prominent target was the 2013 attack on SpamHaus.

Timeline:

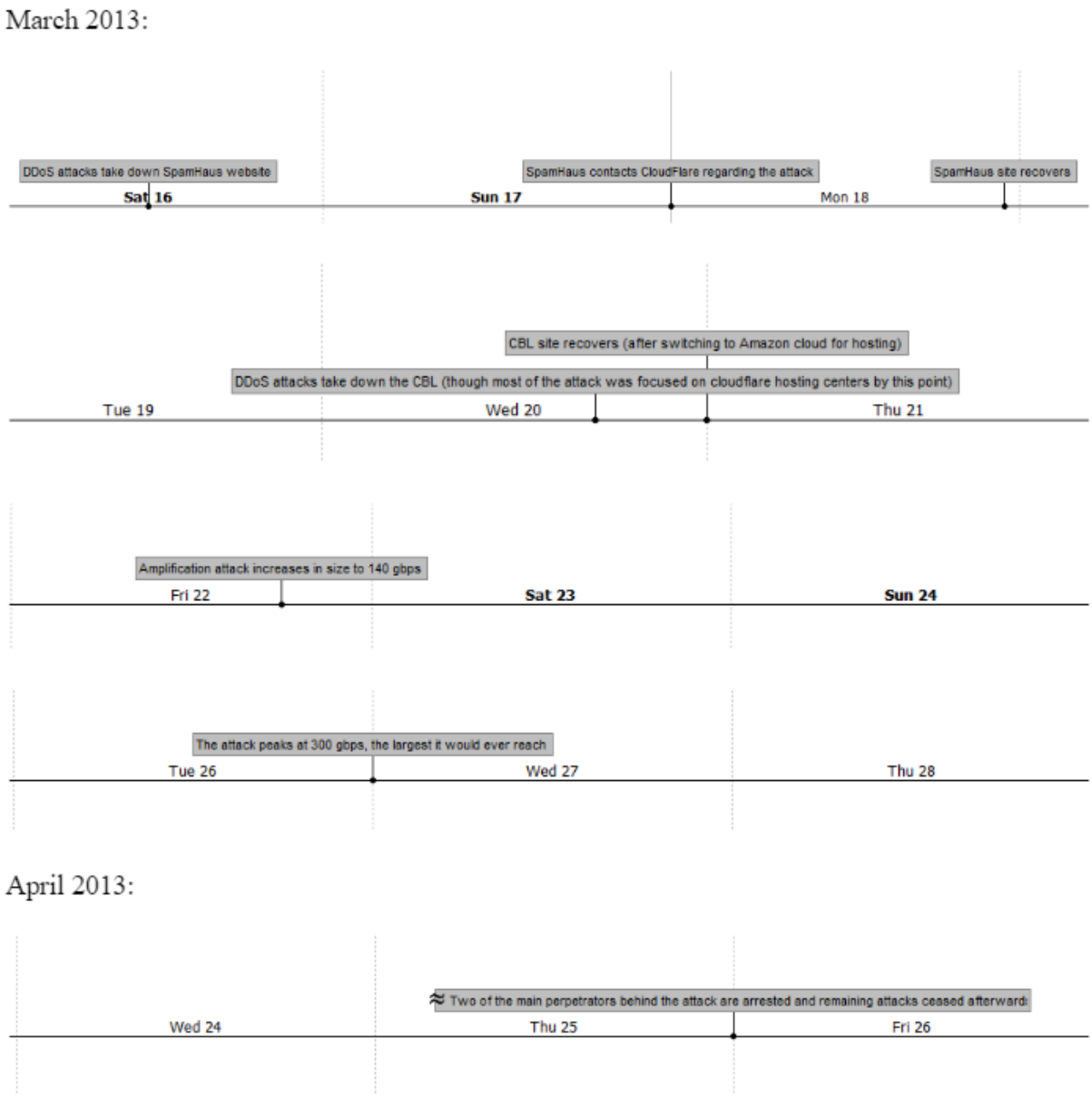
The victim of the attack, SpamHaus is an anti-spam blocklist that provides anti-spam blacklists for emails. The composite blocklist (CBL) is a crucial part of SpamHaus' operations, being a DNS based blocklist that includes information about potential spam sources (Spamhaus, 2020). The perpetrators were StopHaus, an organization that wanted to stop SpamHaus' blocking of emails, likely because a few of the members were perturbed at being included in the block list as spammers themselves.

On March 16 2013, at 12:00am, a DNS amplification attack took down the SpamHaus website (Prince, 2013). Despite attempts by SpamHaus to alleviate the situation, the attack completely overloaded SpamHaus servers. This caused SpamHaus to contact CloudFlare on March 18th to try and come to a solution for the problem. CloudFlare used a series of load balancing techniques to spread the network load across multiple servers. This led to a recovery of the SpamHaus website by 11pm on March 18th.

The attackers, after a brief hiatus, attacked both the CBL as well as CloudFlare servers. The CBL being taken down, affected 1.4 billion users (Handord, 2013). The attacks on CloudFlare servers led to other websites being slowed down to a crawl. On March 21st, the CBL was able to make a recovery after SpamHaus switched to Amazon cloud for hosting. On the same day, the StopHaus domain was taken down. In April 2013, two perpetrators behind the attack were arrested, ceasing further attacks. Figure 4 shows a comprehensive timeline of the events that took place.

Figure 4

A timeline showing various major points during the attack



Technical information:

The attack was carried out through various botnets controlled by StopHaus members. They used a DNS amplification attack to target SpamHaus servers, and later on attacked CloudFlare

hosting servers as well. The botnets consisted of servers from spam friendly server providers as well as unknowing providers. The attack peaked at 300 gbps of traffic, which was the largest attack ever at the time. CloudFlare's mitigation strategy was called AnyCast. This system distributed the traffic of the attack to various data centers, allowing the traffic to not completely take down a single server, but spread the load across many data centers. Despite this, the attacks only completely ceased after the perpetrators were caught, showing the potency of the DNS amplification attack.

Aftermath:

Arrests were made for two of the main StopHaus attackers in April of 2013. Sven Olaf Kamphuis was sentenced to 240 days in jail, of which he served 55 (Perloth, 2013). Seth Nolan McDonagh, a teenager at the time, was also arrested, and sentenced to 240 hours of community service (Corerea, 2015). CloudFlare is the current host for the SpamHaus project. As of now, there have not been any DNS amplification attacks to the same size and scale as the SpamHaus attack

Section 4. Related Work.

In this section I will be looking at various methods of defending against DNS amplification attacks that are widely used.

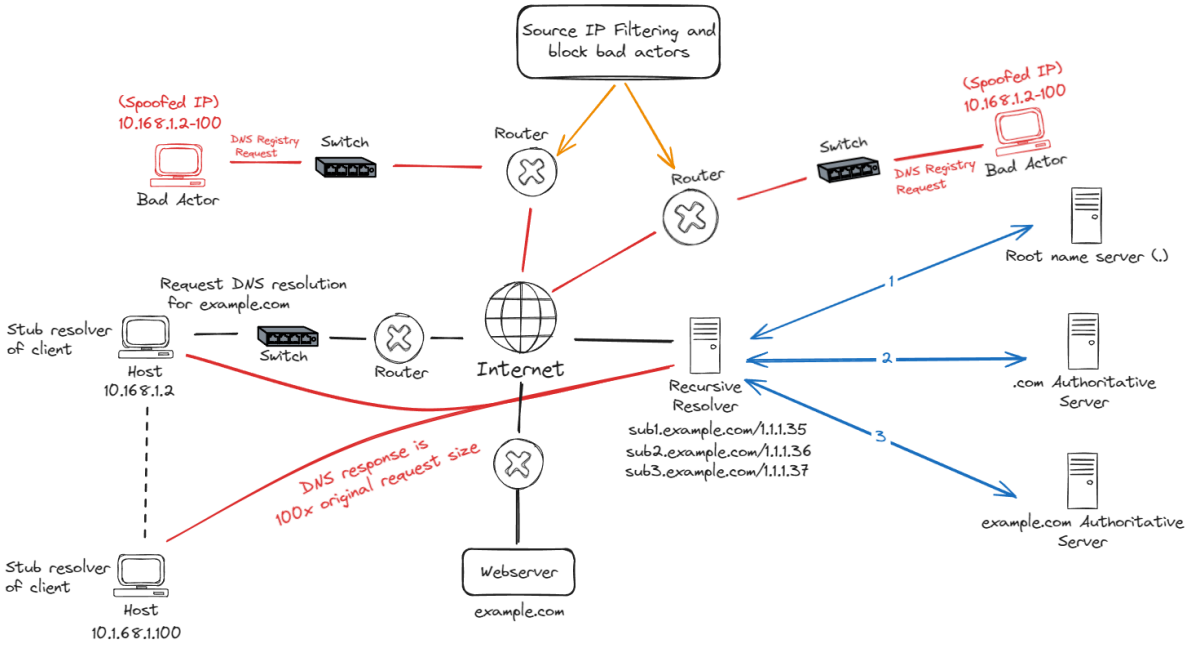
Section 4.1 Established Protocols

Source IP Verification

Mitigation is currently done in a few ways. Source IP verification analyzes the spoofed source IP's routing history by counting the number of hops, or routers that are passed through during the travel to the DNS. The hop count varies by the physical distance between the source router and destination. This means a router with a spoofed IP would have a different hop count than the actual router. By calculating an estimated hop count for the genuine user's router, we are able to detect that the bad actor is spoofing their IP by seeing the discrepancy between the estimated hop count, and the hop count for their route. Once a discrepancy is detected by the DNS, the connection is dropped, allowing for the attack to quickly be stopped. In Figure 5, source IP filtering allows for the blocking of the bad actors featured in the diagram before their queries reach the DNS.

Figure 5

A network diagram showing where bad actors are blocked prior to reaching the DNS



Background on Transport Layer Security

Transport Layer Security (TLS) is used to provide extra security when transmitting data over networks. The three major components of TLS are encryption, authentication, and integrity checks. At the start of establishing a TLS connection with a server, a TLS handshake is done. First the user sends a 'hello' message to the server, providing client hardware and software information, as well as information detailing the highest version of TLS encryption they can support. The server then responds with its 'hello' message, detailing server information, the level of TLS encryption that will be used, and its digital Secure Sockets Layer (SSL) certificate. This certificate contains data about the domain name, certificate authority, and a public key (which will be used later for encryption). The client then checks the validity and trustworthiness of the SSL certificate. If the certificate is valid, the client generates a random session key, which is encrypted with the public key from the server. This encrypted session key is sent over to the server, who decrypts the session key using a private key, and verifies that the decrypted key matches the session key that the browser previously generated. If the decrypted key is valid, the server sends a finish message to the client. The client responds with a finish message to the client. If all these steps occur without issue, all future communications between the client and server are encrypted using the earlier session key.

Recursion

A preventative measure for dealing with DNS amplification attacks is disabling recursion on authoritative name servers (Anagnostopoulos et al, 2013). Since DNS amplification attacks rely on large DNS responses, by not allowing users to give the authoritative name server a DNS registry request, we would prevent the attack from being amplified. Registry requests are known

as recursive because the DNS has to communicate with multiple servers in order to find all the associated IP addresses with the subdomains of a url. A downside to this is that legitimate users might want to know the IP addresses of the subdomains associated with a url, and they would be unable to do so.

Limiting recursion to known users uses similar principles but mitigates the problem rather than preventing it altogether. This method only allows specific IP addresses to give DNS registry requests, meaning that as long as the attack isn't on one of these approved IP addresses, the DNS amplification attack would fail.

Rate Limiting

Rate limiting responses is another method of mitigating the damage of an attack. If a DNS gets too many requests from a particular IP, the connection is dropped (Anagnostopoulos et al, 2013). This would do little to help the target but it would prevent the DNS from being bogged down in requests.

Anycast

Anycast is another method of mitigation. Anycast allows for many machines to share an IP address, allowing for load balancing when the attack reaches the victim (Gupta & Sharma, 2018). The attack would be spread across multiple data centers, spreading out the damage and allowing for legitimate connections to the victim, albeit slower connections.

Section 4.2. Systematic Review of Papers

Spoof detection

As mentioned by Kambourakis et al, the first major step to defending against DNS amplification attacks is through spoof detection, which can be implemented as part of firewall rules for increased security (Kambourakis et al, 2008). Spoof detection can be done in a variety of ways, one of which being analyzing hop counts of network packets. As proposed by Jin et al, hop count filtering inspects the hop counts of incoming packets and determines if they are legitimate or not by comparing incoming packet hop counts to with hop count estimates based on the destination router's location (Jin & Wang 2003). If there is a large discrepancy between the hop count estimate and the hop count of the packet being analyzed, the packet gets dropped. Packets sent from bad actors will usually have greater or lower hop counts than the users they target (greater when physically further away, and lower when physically closer), making it harder for bad actors to try and spoof the IP address of their target without packets being dropped.

DNS based defenses

The next level of defense against DNS amplification attacks would be at the application level, rather than at the network level. One fairly easy to implement method of preventing DNS amplification attacks is by limiting recursion on authoritative name servers, a technique brought up by the Cybersecurity and Infrastructure Security Agency (CISA, 2006). Limiting recursion on authoritative name servers means that only authorized users may perform a DNS registry request. This eliminates the amplification aspect of the DNS amplification attack by not allowing bad actors to request for large DNS registry responses to be sent to the target.

Another method of defense suggested by Kambourakis et al, is implemented on the target's side. Since DNS servers often find themselves as the target for DNS amplification attacks, the method suggested by Kambourakis et al is specifically intended for DNS servers. When a DNS server receives a registry response, it runs through an engine that determines if the registry response was requested by the DNS server or not. For a DNS server that has been the target of a DNS amplification attack, they will receive constant registry responses. Once the engine has determined that a registry response was not requested by the DNS, all further DNS replies are blocked by the addition of a firewall rule (Kambourakis et al, 2008). This allows for the victim of a DNS amplification attack to prevent all further attack attempts from reaching them, so long as the target is a DNS server.

DNS over Transport Layer Security (TLS) is another way to defend against DNS amplification attacks. TLS is a way of establishing a secure connection between a server and a client. Two important parts of TLS are the handshake that is done between the client and the server, as well as the encryption that is part of the protocol. Both of these combined effectively prevents bad actors from impersonating as their targets, preventing IP spoofing. DNS over TLS (DoT) is a way for DNS servers to apply the TLS protocol to any DNS request (Lu et al, 2019). When DoT is applied, the amplification aspect of DNS amplification attacks becomes impossible, due to the handshake portion of TLS.

Section 5. Suggestions for Defending Against DNS Amplification Attacks.

While there are many potential methods of defending against DNS amplification attacks, I still believe there is potential to further improve upon existing methods.

TCP based connections

One potential method of preventing a DNS attack is by making the DNS registry requests TCP based instead of user datagram protocol (UDP) based. TCP connections have a 'handshake' between the source router and the target server. First the client sends a SYN request, allowing the server to know of the client's existence. Then the server sends a SYN-ACK request, effectively letting the client know that they received the SYN request. Finally, the client replies with an ACK request, letting the server know that they revived the SYN-ACK request and that they are ready to request whatever packets they need. TCP connections are usually done to avoid data loss, but there is an added delay and increased resource usage due to the three way 'handshake'.

We can utilize the three way 'handshake' by making all DNS registry requests TCP based. When the bad actor spoofs their IP and sends a SYN request to the server, the target would receive a SYN-ACK, since the target did not send the SYN packet in the first place, they would not respond with an ACK request. Since the server never receives an ACK request, the server would never respond with a registry reponse. This effectively prevents the amplification attack, however it uses a lot of extra time and resources for genuine registry requests.

In a slight modification of the previous method, if a user receives a DNS registry request that they didn't ask for, they can send a message to the DNS saying to switch to a TCP based connection. Everything else would work the same as in the previous solution, except less resources and time are wasted for normal registry requests, as most would be in UDP. The biggest downsides for both of these methods are the large use of resources (TCP connections are more complex than UDP, so they require more resources) as well as extra latency added

when utilizing a TCP based connection versus a UDP based connection (due to the added handshake). This switching between protocols can be seen in Figure 6.

Figure 6

DNS resolver switching between TCP and UDP protocols

A major advantage of this suggested solution over concepts like DoT is that my suggested solution would utilize much less resources and has less latency when compared to TLS based connections. These benefits are due to TCP based connections lacking the same encryption as TLS connections. While the lack of encryption is a definite downside to my proposed solution, I believe the lower latency and less resource use would lead to wider adoption, in contrast to lesser-used DoT. According to Doan et al, “DoT is still only supported by local resolvers for 0.4% of the [Réseaux IP Européens] RIPE Atlas probes” (Doan et al, 2021, p.197). These probes measure internet connectivity and are placed worldwide. A 0.4% rate of being supported is extremely low, showing that DoT is yet to be heavily adopted throughout the world.

Section 6. AI and the Future of DNS Amplification Attacks.

The rise of AI brings countless possibilities for changes in how DNS amplification attacks can be done, as well as defended against. Since DNS amplification attacks are controlled through botnets, advances in controlling a botnet would mean more sophisticated attacks (Anagnostopoulos et al, 2013). Another threat brought up by AI is the identification and detection of valuable targets as well as vulnerabilities through AI based techniques. Alavizadeh et al mention that many current defense techniques are not sufficient enough to prevent AI based botnet attacks (Alavizadeh et al, 2021).

Despite this large challenge in defending against botnets that are becoming more and more sophisticated, there are proposed solutions to this issue. Alavizadeh et al propose increased usage of Moving Target Defense (MTD) to make it harder for botnets to identify targets (Alavizadeh et al, 2021). MTD is a strategy that involves various techniques that mask the identity of a potential target. This is done through IP address randomization, port randomization, and changes in the configuration of server systems (Lei et al, 2018). Another benefit of MTD is that it makes it harder for attackers to estimate the hop counts of their targets (due to IP address randomization), making hop count filtering more effective. Hop count filtering and other packet based filtering methods can also be improved with advancements in AI. Zhang et al propose the use of Reinforcement Learning (RL) in order to train packet filtering systems to be more accurate (Zhang et al, 2019). In a simulation of the RL packet filtering techniques versus traditional packet filtering methods, the RL model had 57.46% of legitimate traffic arriving to a victim of a DDoS attack, compared to 29.95% of legitimate traffic arriving to the victim using traditional port based models. The increase in legitimate traffic arriving to the victim means that the RL model was more accurate in segregating real traffic from attacker traffic.

Section 7. Conclusion.

DNS Amplification attacks are a very potent type of DDoS attack. The attacks leverage the DNS portion of the UDP routing protocol. A famous case of DNS amplification attacks being used is the StopHaus attack of 2013, which succeeded in bringing down SpamHaus, the target. Despite the potency of these attacks there are many ways of detecting and mitigating DNS amplification attacks. The major ways of doing this are through source IP verification, recursion control for the authoritative name server, rate limiting, and anycast. I propose two potential ways of detection and mitigation, using the TCP protocol and its three way handshake in order to avoid bad actors from being able to pretend as the victim. The downsides of both methods are the large resources that are used, as well as added latency with the TCP handshake. While the resource use and latency are a definite downside of TCP based mitigation as suggested, they are not as high as resource use and latency in DoT based solutions. With the rise of AI, there are more options for DNS amplification attacks as well as defense against them. AI allows for bad actors to more easily identify weak targets for an attack, as well as control botnets through the use of AI, allowing for more sophisticated attacks. Moving target defense and hop count filtering also benefit with recent improvements in AI, allowing for more defense options.

Overall, DNS amplification attacks have proven themselves to be very potent threats to cybersecurity. The 2013 SpamHaus attack shows how formidable these attacks can be. While detection methods have improved since the 2013 attack, recent advancements in AI have brought new potential for DNS amplification attacks to evolve. This revolution in AI also has contributed to breakthroughs in defense, allowing for more active defense strategies than

previously used. While DNS amplification attacks may remain a dominant threat in the future, these advancements in defense may allow for attacks to be harder to execute than before.

Section 8. References.

[1] Alavizadeh, H., Jang-Jaccard, J., Alpcan, T., & Camtepe, S. A. (2021). A Game-Theoretic Approach for AI-based Botnet Attack Defence. arXiv [Id='cs.CR' Full_name='Cryptography and Security' is_active=True alt_name=None In_archive='cs' is_general=False Description='Covers All Areas of Cryptography and Security Including Authentication, Public Key Cryptosytems, Proof-Carrying Code, Etc. Roughly Includes Material in ACM Subject Classes D.4.6 and E.3.']. Retrieved from http://arxiv.org/abs/2112.02223

[2] Anagnostopoulos, M., Kambourakis, G., Kopanos, P., Louloudakis, G., & Gritzalis, S. (2013). DNS amplification attack revisited. Computers & Security, 39, 475–485. doi:10.1016/j.cose.2013.10.001

[3] Bhardwaj, A., Mangat, V., Vig, R., Halder, S., & Conti, M. (2021). Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions. Computer Science Review, 39, 100332. doi:10.1016/j.cosrev.2020.100332

[4] Cloudflare. (2011). A diagram showing Unicast and Anycast functionality. In A Brief Primer on Anycast

[5] Corera, G. (2015, July 10). UK teenager sentenced over "biggest" web attack. *BBC News*. https://www.bbc.com/news/technology-33480257

[6] DDoS | second arrest in response to DDoS attack on Spamhaus. (2014, July 7). The Spamhaus Project. https://www.spamhaus.org/resource-hub/ddos/second-arrest-in-response-to-ddos-attack-on-spamhaus/

[7] DNSBL | Update for Composite Blocklist (CBL) Users | Spamhaus. (2020, December 18). The Spamhaus Project. https://www.spamhaus.org/resource-hub/dnsbl/update-for-composite-blocklist-cbl-users/

[8] Doan, T. V., Tsareva, I., & Bajpai, V. (2021). Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times. In O. Hohlfeld, A. Lutu, & D. Levin (Eds.), Passive and Active Measurement (pp. 192–209). Cham: Springer International Publishing.

[9] Gupta, V., & Sharma, E. (09 2018). Mitigating DNS Amplification Attacks Using a Set of Geographically Distributed SDN Routers. 392–400. doi:10.1109/ICACCI.2018.8554459

[10] Handord, S. (2013, March 28). *Chronology of a DDoS: SpamHaus*. Cisco Blogs. https://blogs.cisco.com/security/chronology-of-a-ddos-spamhaus

[11] Jin, C., & Wang, H. (11 2003). Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic. doi:10.1145/948109.948116

[12] Kambourakis, G., Moschos, T., Geneiatakis, D., & Gritzalis, S. (2008). Detecting DNS Amplification Attacks. In J. Lopez & B. M. Hämmerli (Eds.), Critical Information Infrastructures Security (pp. 185–196). Berlin, Heidelberg: Springer Berlin Heidelberg.

[13] Kottler, S. (2018, March 1). February 28th DDoS incident report. The GitHub Blog. https://github.blog/news-insights/company-news/ddos-incident-report/

[14] Krebs, B. (2013). *Inside "The Attack That Almost Broke the Internet" — Krebs on Security*. Krebsonsecurity.com. https://krebsonsecurity.com/2016/08/inside-the-attack-that-almost-broke-the-internet/

[15] Lei, C., Zhang, H.-Q., Tan, J.-L., Zhang, Y.-C., & Liu, X.-H. (2018). Moving Target Defense Techniques: A Survey. Security and Communication Networks, 2018, 1–25. https://doi.org/10.1155/2018/3759626

[16] Lu, C., Liu, B., Li, Z., Hao, S., Duan, H., Zhang, M., … Wu, J. (2019). An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? Proceedings of the Internet Measurement Conference, 22–35. Presented at the Amsterdam, Netherlands. doi:10.1145/3355369.3355580

[17] Osterweil, E., Stavrou, A., & Zhang, L. (2019). 20 Years of DDoS: a Call to Action. arXiv [Cs.NI]. Retrieved from http://arxiv.org/abs/1904.02739

[18] Perlroth, N. (2013, April 26). Dutch man said to be held in powerful internet attack. *The New York Times*. https://www.nytimes.com/2013/04/27/technology/dutch-man-said-to-be-arrested-in-powerful-internet-attack.html

[19] Postel, J. (2019). RFC 791, IP protocol, DARPA internet program protocol specification. Ietf.org. https://tools.ietf.org/html/rfc791

[20] Prince, M. (2013, March 20). The DDoS That Knocked Spamhaus Offline (And How We Mitigated It). The Cloudflare Blog. https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho

[21] Prince, M. (2013, March 27). *The DDoS That Almost Broke the Internet*. The Cloudflare Blog. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

[22] R. R. Brooks, L. Yu, I. Ozcelik, J. Oakley, & N. Tusing. (2022). Distributed Denial of Service (DDoS): A History. IEEE Annals of the History of Computing, 44(2), 44–54. doi:10.1109/MAHC.2021.3072582

[23] Ryba, F.J., Orlinski, M., Wählisch, M., Rossow, C., & Schmidt, T.C. (2015). Amplification and DRDoS Attack Defense - A Survey and New Perspectives. *ArXiv, abs/1505.07892*. van der Toorn, O., Müller, M., Dickinson, S., Hesselman, C., Sperotto, A., & van Rijswijk -

[24] Deij, R. (2022). Addressing the challenges of modern DNS: a comprehensive tutorial. Computer science review, 45, Article 100469. https://doi.org/10.1016/j.cosrev.2022.100469

[25] The Continuing Denial of Service Threat Posed by DNS Recursion (v2.0). (2006). https://cisa.gov/sites/default/files/publications/DNS-recursion033006.pdf

[26] Zhang, Y., & Cheng, Y. (2019). An Amplification DDoS Attack Defence Mechanism using Reinforcement Learning. 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 634–639. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00145

[27] Zhu, Z., Lu, G., Chen, Y., Fu, Z. J., Roberts, P., & Han, K. (2008). Botnet Research Survey. 2008 32nd Annual IEEE International Computer Software and Applications Conference, 967–972. doi:10.1109/COMPSAC.2008.205

Section 9. Appendix/Glossary:

Bad Actor: A bad actor is someone that tries to take down servers or systems for malicious reasons.

Botnet: A series of computers that are either compromised or owned by a single party, who can control all the computers in the botnet through a single channel, allowing for DDoS attacks to be done.

Connectionless Lightweight Directory Access Protocol (CLDAP): This protocol is used by devices in order to share internet directories. This is based on UDP, in order to minimize latency and allow for lightweightness.

Domain Extension: The last letters in a url that proceed the last period in the url, these will be .gov, .edu, .com, etc

Hop Count: A hop count is a measure of how many times a network packet has to travel through devices like routers. Each time a packet has to travel through a different device, the hop count is updated until the packet reaches its final destination.

Hypertext Transfer Protocol (HTTP): A protocol that allows for the communication between clients and web servers through a series of requests and responses.

IP Address: An IP address is a series of numbers that are used for identifying different computers while routing.

Network Time Protocol (NTP): The network time protocol is used for synchronizing system clocks across various networks. It is based on the principles of UDP connections, allowing it to have lower latency.

Simple Service Discovery Protocol (SSDP): This is a protocol for the advertisement of network services, it is a way for devices on the same network to communicate with and discover each other.

SYNchronize (SYN): The SYN message is sent from the client to the server, attempting to establish a connection with the target server. This is the first step of the TCP handshake.

SYNchronize - ACKnowledgement (SYN-ACK): This is a message sent from the target server to the client. This acknowledges the receival of the previous SYN request and intends to see if the client can receive the SYN-ACK message.

Transmission Control Protocol (TCP): A type of internet protocol that is used for the lossless transfer of packets over the internet. This means that packets are not lost during the routing process.

User Datagram Protocol (UDP): An type of internet protocol that is used for the lossy transfer of packets over the internet. This means that packets may be lost during the routing process.