Report for Presentation

Syafiq Al Atiiq

March 17, 2021

1 Summary

From a higher perspective, the DNS system consists of two main entities: i.) authoritative name servers, and ii.) resolvers. The purpose of the former is to store the DNS data in a hierarchical and dynamic database that has a highly distributed fashion of their location. While the latter is to serve the clients directly, translating the domain into IP address. The location of resolvers can be locally at the client's service provider/local organizations, or the cloud public service (i.e. Google 8.8.8.8, and OpenDNS 208.67.222.222). Resolvers walk through the hierarchical structure of authoritative name servers to fetch the domain name resolutions. The process of walking through the structure creates an interaction between resolvers and authoritative name servers, which in some way introduces a new vulnerability. This is the direction taken in the paper.

When the resolver performs the "walking through" process towards the hierarchy, it gets delegated from one authoritative name server to another. The reason why an authoritative server needs to delegate is that one server does not contain everything. The request needs to reach the correct authoritative server that has the required mapping of the domain name in question to the IP address. Top-level authoritative domains (TLDs), second-level domains (SLDs), and other authoritative servers are not allowed to serve the requests of IP addresses for domains that do not reside in the same zone origin. The messages to perform the delegation process is called referral responses. Citing from the author's paper, when the delegation happened, roughly speaking, an authoritative server would tell the recursive resolver something like: "I do not have the answer, go and ask one of these name servers, e.g., ns1, ns2, etc., that should get you closer to the answer". However, the information in the NS referral responses (at different steps), in combination with the actions taken by the recursive resolvers, might introduce humongous communication overheads in terms of additional messages exchange.

This overhead creates a rather bad implication such that an adversary can exploit that to perform a new type of attack, namely *NXNSAttack*. If an adversary managed to own an authoritative server, she is able to craft a response to the resolver in a way that contains n new and non-existent name server. This response message would trigger the resolver to start processing F new resolutions. The authors show that the range of F can be between 74 and $2 \times n$, where n is the number of name server names mentioned in the referral response message. If the adversary is able to generate the referral response multiple times, this would lead to a DDoS attack towards, either:

- The resolver, or
- The corresponding authoritative server

The authors propose several solutions for this vulnerability with a new enhancement to the recursive resolver algorithm, one of them is MaxFetch(k). Basically, MaxFetch(k) would prevent the resolver to resolve all the name server domains in a referral response at once, but more of a sequential process, where the resolver can resolve a maximum k for every original client request.

2 Importance of the Work

The paper discloses a new vulnerability in the DNS system and introduces the corresponding attack, namely NoneXistent Name Server Attack (*NXNSAttack*). This attack is more destructive than the previous DNS-based flooding attack [1], which may disrupt and destroy our DNS system. The authors also proposed some

possible mitigation to the attack, one of them is MaxFetch(k). All the process of disclosing the vulnerability and possible patch has been done prior to the publication of the paper. That said, several DNS vendors have been notified in advance so they already issued a CVE and patched their system.

3 Improvements

I don't think I have a specific criticism of the content of the paper. It's a great contribution and has a profound impact on our life directly, after all. The authors also explicitly said in the conclusion section that they had no intention to look at this specific vulnerability, but rather researching the efficiency of recursive resolvers. As they said, "*You never know what you might find when you go searching for your lost donkey.*"

However, one thing that popped up in my mind is that the solution of MaxFetch(k) might deviate from the RFC [2][3] of DNS. If the vendors decided to follow the advice from the authors, they basically run a non-standard DNS system. Security-wise, this might be a better option now. But, one should work towards re-designing the DNS standard such that this kind of attack does not happen in the future.

References

[1] Xi Luo et al. "A Large Scale Analysis of DNS Water Torture Attack". In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. CSAI '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 168–173. ISBN: 9781450366069. DOI: 10.1145/3297156.3297272. URL: https://doi.org/10.1145/3297156.3297272.

[2] P. Mockapetris. *DOMAIN NAMES - CONCEPTS AND FACILITIES*. RFC 1034. Nov. 1987. URL: https://tools.ietf.org/html/rfc1034.

[3] P. Mockapetris. *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*. RFC 1034. Nov. 1987. URL: https://tools.ietf.org/html/rfc1035.