**Peer Response 1**

Context: <https://www.my-course.co.uk/mod/hsuforum/discuss.php?d=290195>

Hi Andrey,

Thanks for your feedback. It is interesting to consider how the different load balancer types and additional middleware (such as Cloudflare and Akamai, as you mention) would influence a vulnerability scan or penetration test in practice. From what I have read, it looks to be the case that DNS load balancers are simple in nature- they lack the ability to dynamically adjust the list of addresses which they respond with, making it possible for a request to be directed to a server which is down or overloaded (Imperva, n.d.). Since it's possible to detect the existence of a DNS load balancer (using the method which you discussed), it becomes easier for an attacker to increase their damage to an organization by taking advantage of the limitations of a DNS load balanced site (assuming this is detected during scanning).

There have also been cases where the additional complexity of a load balancer has also introduced new vulnerabilities. Alp (2020) identified a scenario where sensitive information was leaked due to an improperly configured load balancer. Netscylla Cyber Security (2018) also encountered situations where flaws in load balancers could be exploited to gain network access.

This raises some interesting questions within the context of network scanning and vulnerability testing. I am of the opinion that load balancers shouldn't be part of a penetration test if it is provided and managed by a well-renowned external company- it might just add noise. As an example, if an organization makes use of Amazon Web Services for hosting, it is up to Amazon to manage the lower-and-higher levels of load balancing as this area forms part of their shared responsibility model (Amazon Web Services, 2022). Additionally, cloud providers at this scale often have much more resilient infrastructure by default due to necessity, making it less productive to test their offerings. In a case where an organization provides and manages load balancing on its own, it would then be beneficial to examine it as part of a penetration test as the risk of undiscovered flaws is much greater. Would you agree with this?

**References**

Alp, O. (2020) Weird Vulnerabilities Happening on Load Balancers, Shallow Copies and Caches. Available from: https://medium.com/dataseries/weird-vulnerabilities-happening-on-load-balancers-shallow-copies-and-caches-9194d4f72322 [Accessed 17 January 2022].

Amazon Web Services. (2022) Shared Responsibility Model. Available from: https://aws.amazon.com/compliance/shared-responsibility-model/ [Accessed 17 January 2022].

Imperva. (n.d.) DNS Load Balancing and Failover. Available from: https://www.imperva.com/learn/availability/dns-load-balancing-failover/ [Accessed 17 January 2022].

Netscylla. (2018) Secret Holes Behind the Common Load-Balancer. Available from: https://medium.com/@netscylla/secret-holes-behind-the-common-load-balancer-e7f70fcaa353 [Accessed 17 January 2022].

**Peer Response 2**

Context: <https://www.my-course.co.uk/mod/hsuforum/discuss.php?d=288548>

Hi Taylor,

Thank you for the clear and insightful post. One interesting area you identified is that of timeouts for ICMP packets. Since it relies on networking, traceroute operates at level 3 of the OSI model and from that point of view, various reasons exist for a gateway not responding to an ICMP packet in time (if at all). Apart from an actual timeout, other possible reasons include the gateway not being configured to reply to UDP or ICMP packets either due to its own configuration or a firewall blocked the communication (Carrigan, 2019). It might also be the case that the ports which traceroute queries on the gateway are all in use, affecting replies (IBM, 2020). There are some possible ways around the previously mentioned issues- most traceroute implementations send UDP packets by default (die.net, n.d.), so one could try using the -I flag to send ICMP packets or the -T flag to send TCP packets, which can bypass firewalls. If the cause truly is a timeout, the timeout waiting period could be increased by using the -w parameter.

Within the context of network security, do you think there's much value in attempting to get responses from all traceroute hops for a website?

**References**

Carrigan, T. (2019) Traceroute: Finding meaning among the stars. Available from: https://www.redhat.com/sysadmin/traceroute-finding-meaning [Accessed 16 January 2022].

die.net. (n.d.) traceroute(8) - Linux man page. Available from: https://linux.die.net/man/8/traceroute [Accessed 16 January 2022].

IBM. (2020) traceroute command. Available from: https://www.ibm.com/docs/en/aix/7.1?topic=analysis-traceroute-command [Accessed 16 January 2022].