As part of a learning activity, it was required to perform scans on an assigned website. In the case of Team 2, the website given was www.readmyblog.co.uk.

My approach to this activity was to use the programs traceroute, ping, host, dig, and whois. These tools are available for use via online websites, but they can also be used through a Linux command line, which was my chosen approach. It is worth noting that some command line tools required installation via my distribution's package manager.

After using the traceroute command, I found that 20 hops were necessary to connect to the site, from my laptop. Traceroute itself displays the routers which packets travel through, in order to reach the destination server (Gilles, 2010). The addresses and round-trip time are displayed for each of these routers (and in the case of round-trip times, 3 appear as traceroute measures times for 3 packets at each router). By taking the average of the 3 packet round-trip times at each hop, I was able to conclude that the 17th hop was the one that took the longest amount of time.

Thereafter, I made use of the command "host -t ns readmyblock.co.uk" to determine the nameservers for the website, which were found to be ns1, ns2, ns3, and ns4, each as a separate subdomain on a2hosting.com.

I then used the whois command to find the registered contact for the site, which was listed as eNOM, which is actually the domain registrar. Prior to GDPR, more detailed information would be available regarding a registrant's name and address, however, this is no longer the case unless the registrant consents to displaying that information (dnsimple, 2019).

Following this, I used the command "dig staffmatters.co.uk MX" to find the website's MX record. To find the website's host location, I used the ping command to get the website's IP address, and then used it on https://ipinfo.io to find the host location, which was listed as the Netherlands.

Although the scans in this exercise yielded clear and positive results, this might not always be the case in a real world scenario. The command-line tools used in this activity are used early in the process of attacking a website (Mandal & Jadhav, 2016), however, modern web application architectures might affect the results of these command-line tools. As an example, many modern web applications now make use of load balancers, meaning that completely different systems could respond to the same request, leading to inconsistent results (Shaffer, 2010). Penetration tests would become highly inaccurate if this is not catered for, as the systems responding to the request may have completely different software versions or infrastructure, making it harder for recommendations to be made or for vulnerabilities to be found.

**References**

dnsimple. (2019) GDPR and WHOIS Privacy. Available from: https://blog.dnsimple.com/2019/04/gdpr-and-whois-privacy/ [Accessed 6 December 2021].

Gilles. (2010) How to interpret traceroute information? Available from: https://unix.stackexchange.com/questions/1023/how-to-interpret-traceroute-information [Accessed 6 December 2021].

Mandal, N. & Jadhav, S. (2016) 'A survey on network security tools for open source', *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC).* Bangalore, India, 10-11 March. New Jersey: IEEE. 1-6

Shaffer, C. (2010) *Identifying Load Balancers in Penetration Testing*. Web: SANS.