The purpose of the penetration test is to find any vulnerabilities on the web application as that will help to give more vision about the security risks and guide to implement the web application in best practice standards. In addition, performing the penetration test frequently provides an understanding of security weaknesses and threats that malicious staff or unauthorised users could use to theft or change the data inside the web application and its database.

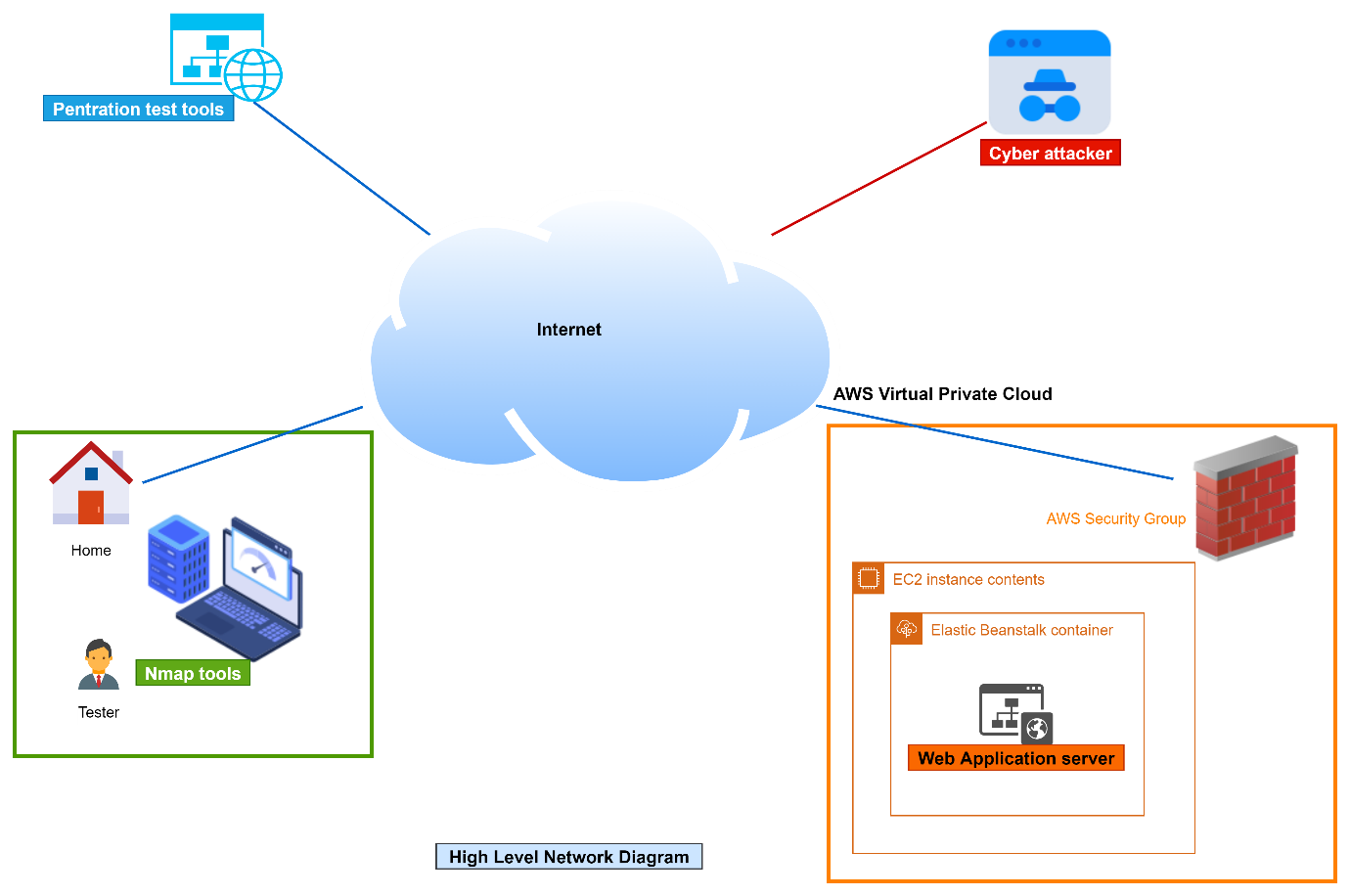
To cover the best practice penetration test, scanning was done from different zones using multiple tools. First of all, the reconnaissance and discovery phase did on the web application like fingerprinting, DNS forward and reverse lookup, DNS Zone transfer, ping, and traceroute. Then penetration tools are used to scan software, application and operation system. For server software, we use the Penetration-tools.com website to give us more information about version-based vulnerabilities detection, the insecure HTTP headers, common insecure cookie settings and deprecated server software.

ImmuniWeb Community tools have been used to scan and get more information and verify the webserver security; these tools also help us fingerprint the CMS components and vulnerabilities. Also, it helps us to check and gather more details about the GDPR and PCI DSS compliance, HTTP Headers, Content Security Policy, Cookies, and external content security tests.

To discover the open doors at the network level, ports scan tools like Penetration-tools.com and Nmap tools are used to figure out the TCP and UDP ports open from the public internet side to the web application server. Nmap tools identified the ports along with version information, traceroute to the web application server and details about the host server like the host’s name, IP address, uptime and the operating system.

Some extra tools like SHHGIT and SkyArt were installed in our AWS environment. SHHGIT gave us some excellent results about secret leaks across Bitbucket, Github and Gitlab platforms. SkyArt used to discover the sensitive and risky users permission but the limited privileges gave us on AWS prevent us from completing the scan.

After all, fingerprinting and scans done to the web application, we have generated a report from each scan about the security weaknesses and started to analyse the data inside the reports, then gathering all the potential vulnerabilities that attackers and malicious staff can use. Finally, we sort out the vulnerabilities between the software, operating system, network access level, GDPR and DSS PCI compliance test. In this report, we have detailed all vulnerabilities and security risks we have found in the scan reports along with the recommendations and advice that can be done to mitigate the security risk and help to avoid any problems with the confidentiality, integrity, or availability agreements.



* **Recommendations**:

**Software security recommendations**:

1. Use HTTPS to encrypt the data communicating between the web browser and the server by using SSL certificate.
2. Add X-Frame-Options HTTP header to avoid any Clickjacking attacks.
3. Set X-XSS-Protection header with block mode as that can help users to prevent any XSS attack.
4. Set X-content-Type-Options to nosniff to prevent Internet Explorer browser from reinterpreting the content of a web page and thus overriding the value of the Content-Type header, as that could lead to attacks such as Cross-Site scripting or phishing.
5. Configure Referrer-Policy header to avoid any user tacking and inadvertent information leakage.
6. Eliminate web server information by obscuring web server information in heads, using a hardened reverse proxy server and ensuring it has recommended software and security patches.
7. Configure Content Security Header to prevents any exploitation of Cross-Site.
8. Keep PHP and Apache software up-to-date by checking the vendor's website and apply all recommended patches. (In our target web application, PHP should be upgraded to version 8.0 as running PHP 7.3 is outdated, and Apache software should be upgraded to 2.4.48 version released 01 June 2021).

**Network-level recommendations:**

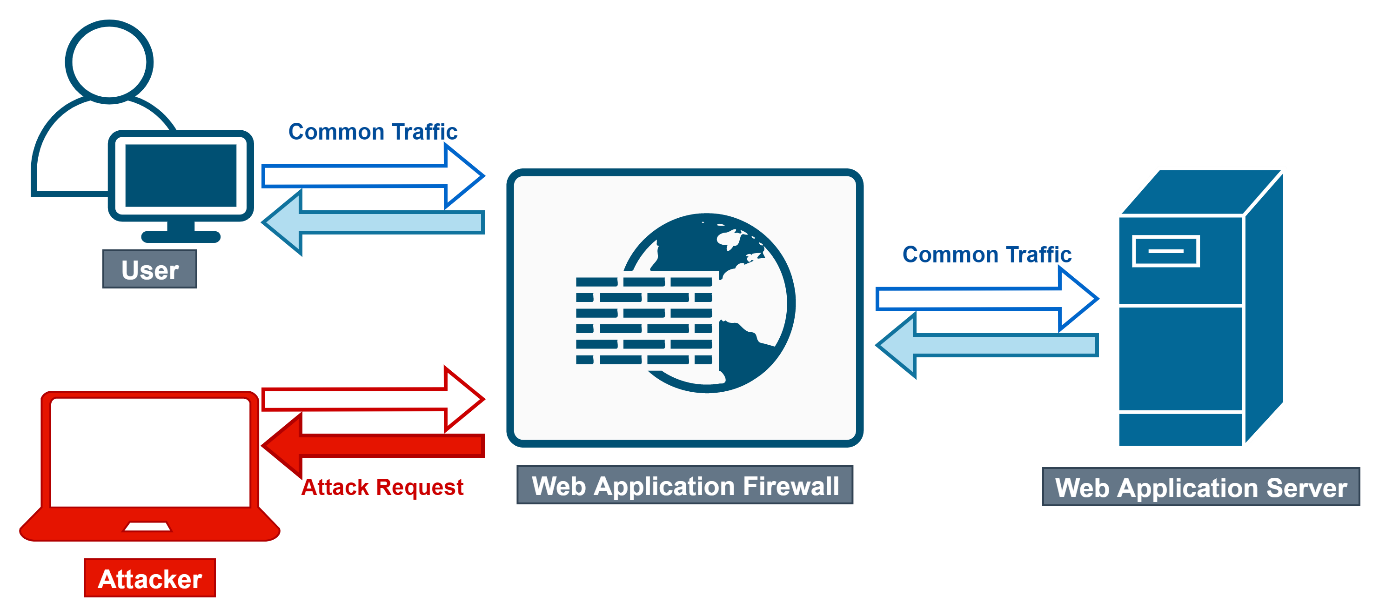
1. Allow Secure DNS to avoid cyber criminals creating any false DNS records.

2- Block SSH port for public internet access and allow it for limited local addresses. If SSH port is required for some customers, firewall policies should be added to allow just required customer IP addresses.

3- Allow HTTPS and block HTTP to make sure all user data is encrypted.

**PCI DSS requirement recommendations:**

1. Install applicable security patches provided by the vendor and critical security patches within one month.
2. Perform training for all developers to learn about secure coding and write down standardisation guidelines that all developers can follow when programming any new web application.
3. Protect all web applications with public internet access to avoid any attack using a web application firewall to detect and block any DDoS or bad traffics.



**Server operating system recommendations:**

Keep the server operating system up-to-date and patched with the latest secure recommended version by working very closely with the vendor and checking all critical security issues in the running version. (For our web application server, it is recommended to upgrade the server to the latest 46bit Amazon Linux 2 version).

**GDPR requirement recommendations:**