Unsecured Web Server

By Ross Mintzer

An unsecured web server poses a significant risk to an organization’s data stored on that web server and potentially other assets if credentials are stored there as well. A wide range of information could be exposed to attackers and reduces an organization’s ability to maintain confidentiality, availability, and integrity. Undoubtedly, the problem with HTTP traffic lies in the lack of encryption it provides and the encrypted version HTTPS is almost universally used today. The lack of encryption of HTTP traffic is particularly sensitive to MITM attacks resulting in the interception of credentials, private communication, payment info, etc. HTTPS solves this by encrypting the data in motion or the packets sent between client and server.

In my project, I constructed an azure network with a DVWA web app and an ELK stack running HTTP on port 80 and 5601 respectively. In this use case it was acceptable to do so, since I was only using it for testing and not passing sensitive info. Furthermore, the insecure configuration allows more options to practice penetration testing such as demonstrating HTTP 's lack of encryption by intercepting plain text traffic through Wireshark. I`vn order to prevent the webapps from being exploited, I put in place a firewall rule to only allow my machine access to the network’s virtual machines. Additionally, In a properly secured deployment, both the ELK stack and web pages would need to run HTTP with SSL or TLS encryption (HTTP*S*) .

In order to switch an existing web server over to HTTPS, I would need to purchase an SSL certificate from a site such as godaddy.com who will verify I actually own the domain. Next, I will need to generate both a public and private key on the web server in order for it to be able to encrypt and decrypt communication with clients. From there I have the option to install it myself or pay the certificate provider an additional fee to install it. In a real-world scenario, this decision would depend on the type of certificate acquired with 3 potential options, Domain (lowest tier), Organizational (medium tier), and Enterprise (highest tier). Each successive tier provides additional security but also requires more identity verification resulting in significantly more paperwork. As an information-security analyst in a small organization only requiring a domain level certificate I may handle the paperwork and installation myself. However, in a large enterprise requiring an enterprise level certificate it would be best to pass this to a system administrator, since the process could interfere with my ability to do my job –identifying potential security risks.

After the certificate is installed, all links on the website should be updated to include https:// . Also, many previous users using bookmarks and search engines will need to be redirected to your new address. This can be done by updating your .htaccess file. Once this is set up you can set it and forget it for 825 days until the certificate expires and needs renewal.

In conclusion, a server running HTTP is a critical vulnerability that an organization seeking security/privacy compliance must address. Without intervention, an organization is at risk of data exposure due to plain text packets sent over HTTP. Legal ramifications will ensue since this is a compliance guideline for most industries. Ultimately, obtaining and installing an SSL or TLS certificate will be the best solution to this critical vulnerability.