Assignment 2: Vulnerability Scanning

For my lab setup, I installed OpenVAS onto my Kali Linux machine to run a vulnerability scan on my target Windows 10 machine. First, I needed to make sure my Kali box has an up-to-date apt repository and has the most recent version of the OS by using the commands: "sudo apt update", "sudo apt upgrade", and "sudo apt dist-upgrade". Once these preliminary commands were run, I was able use "sudo apt install openvas" to download the openvas program. After apt gets the file, it must be set up using the command "gvm-setup". Just for safe measure, I also decided to verify the installation with "gvm-check-setup", which showed that the installation worked successfully. With that, I started OpenVAS using "sudo gvm-start". This app is now accessible via the web browser at the loopback address of 127.0.0.0:9392, 9392 being the default port for the OpenVAS web GUI. After the installation and configuration of OpenVAs was complete, I was ready to begin scanning the Windows 10 machine for vulnerabilities. First, I had to set up a target by navigating to "configuration" and clicking the "new target" button. For the target's name, I chose Win 10 Victim to ensure I was able to pick it apart from any other scans I may run in the future. I then set the IP address to 192.168.56.10 (The IP of the Windows Box) and clicked create at the bottom of the window. Looking at the list of targes, I can see that the target was successfully created. In order to create the scanning task, I then went to the "Scans" tab and clicked the "New Task" button, then gave it the name of "Scan Win 10 Victim" and selected the target from the dropdown. As the task was now created, the scanning process can begin. In the task list, clicking the start button on the task begins scanning the target. This took quite bit of time but when the scan completed, I was presented with a report that showed no major vulnerabilities with the system. The report shows many useful items, such as threat level, severity, and a list of vulnerabilities that the machine is exposed to. The Windows machine was up to date, so I did not expect any meaningful results from this scan. If it did have severe vulnerabilities, this report allows you to view them in detail, as well as export those vulnerabilities to a PDF file for further analysis.

Upon starting this project, I dealt with a plethora of issues. The main issue I ran into didn't have much to do with OpenVAS, but I had stored all my Virtual Machines on a NAS drive, which failed and required me to build my lab up from scratch. This process has taken a while, but from now on it should be taken care of. Initially, I had wanted to find a Windows ISO of a version that was not supported and would likely produce tangible vulnerability results, such as XP. However, I was not able to find one that didn't require a key to set up. Ultimately, I decided to just use the Windows 10 media creation tool to create an ISO to use as my victim machine. One problem someone may run into during this scanning process is a misconfigured IP scheme. For instance, if the target machine were configured with DHCP, the IP address may change and require the administrator to find the IP before they can connect to it. Personally, I set up my victim with a static IP so I wouldn't run into this issue. Also, it may be possible to forget to use the sudo command and run into permission errors while installing. It's also possible that the 9392 port is blocked with a firewall if communicating with the target over a network. This would be a pretty simple fix that's remedied by adding a port forwarding rule to allow that port to be used. The target machine may also block connections to itself, so it's important to disable the local firewall in a closed lab setting. In an open corporate setting, it may be more beneficial to quarantine the machine before running the scan, which would prevent infection to other machines on the network and allow the scan to be completed in a safe environment.