

LAB- 03 Vulnerability Scanning and Management

By:- Faraz Ahmed

- We enter URL- <https://localhost:9392> on the browser and enter credentials (Username- MGS650 and Password- Change.me!) to login OpenVAS web UI.

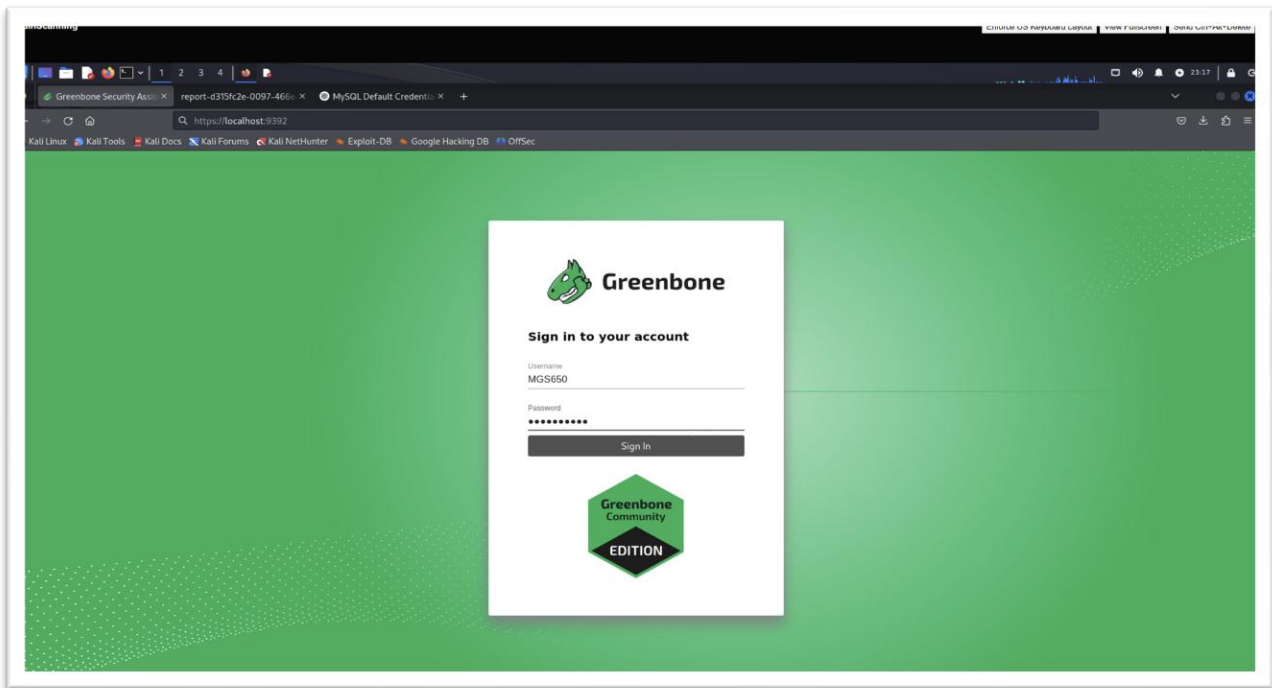


Figure 1: Screenshot of Login for OpenVAS UI.

- Then navigate to Scan>Tasks and click on the little wand icon on top left corner. Then enter “192.168.252.3, 192.168.252.61, 192.168.252.241” and select Start Scan as shown in figure 2.

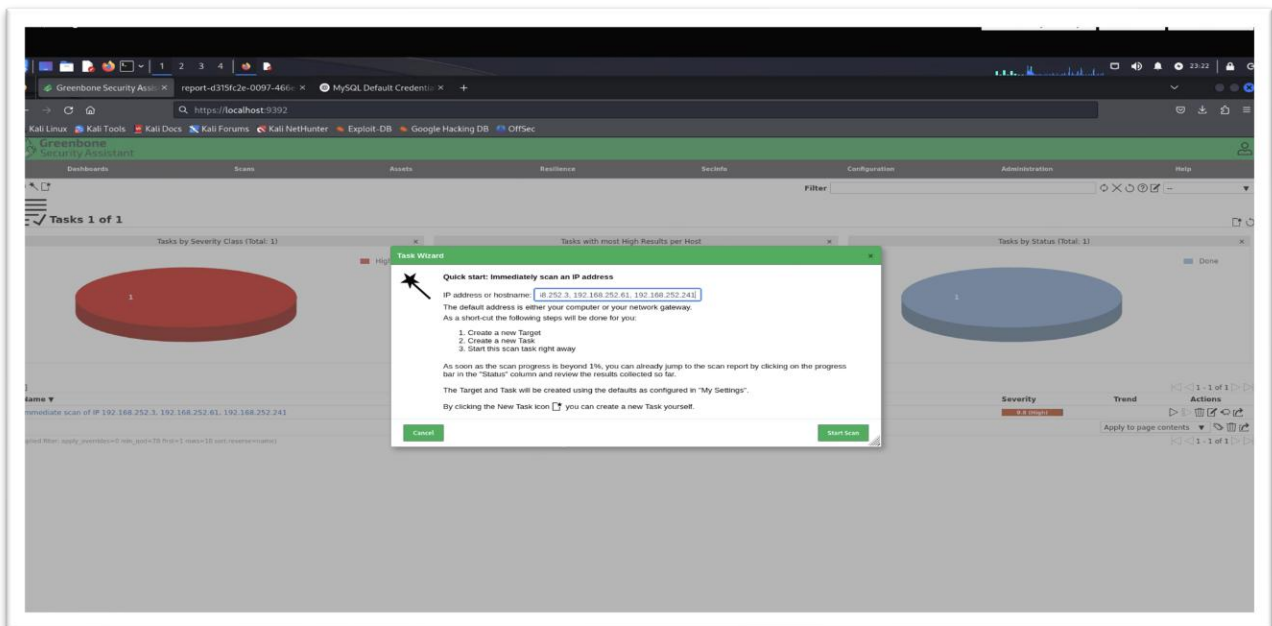


Figure 2: Screenshot of entering all IP address in “Task Wizard”.

- Then we keep that scan running in the background and navigate to Configuration>Port Lists. We can observe 3 different lists as shown in figure 3.

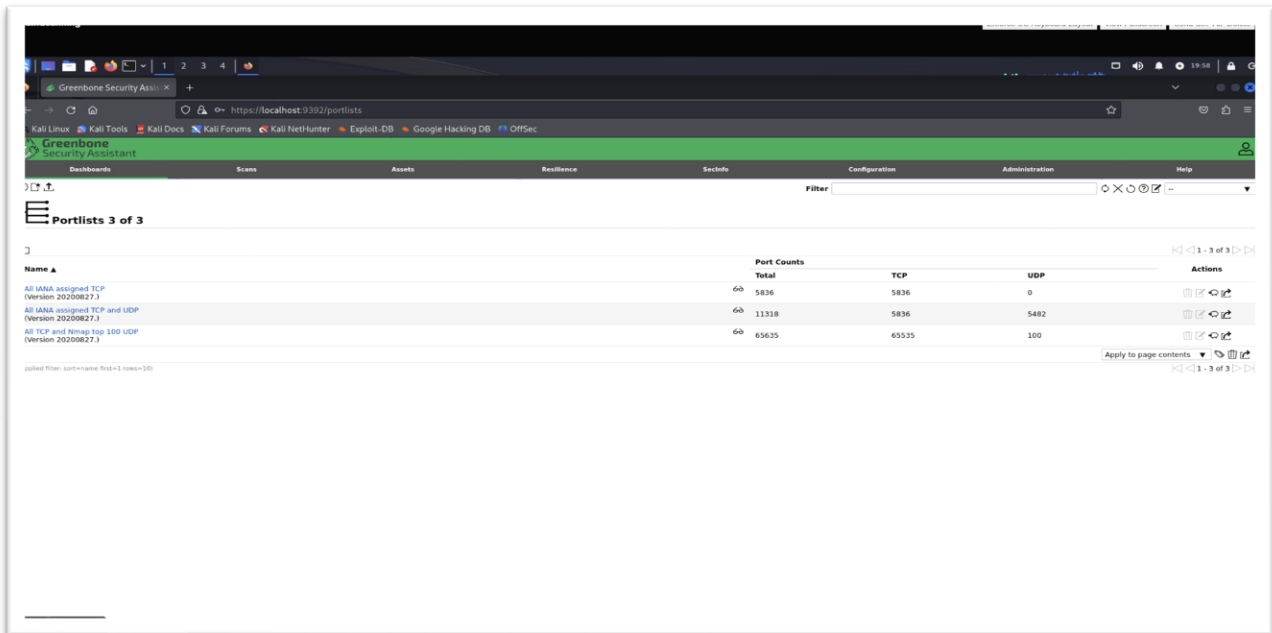


Figure 3: Screenshot of 3 different port lists.

- Then if we select any port, we get into information window where we get more information about that port and we can switch to port ranges to check on all of the ranges linked to it as shown in figure 4 and 5.

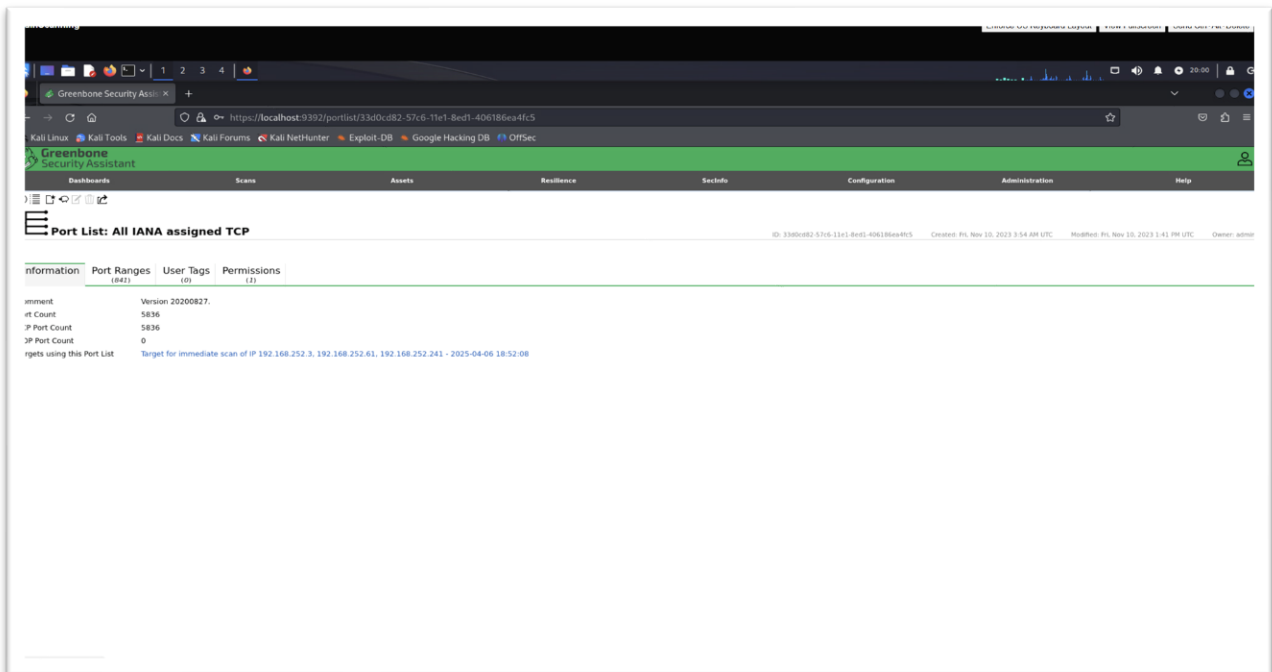


Figure 4: Screenshot of "Information window" inside a port list.

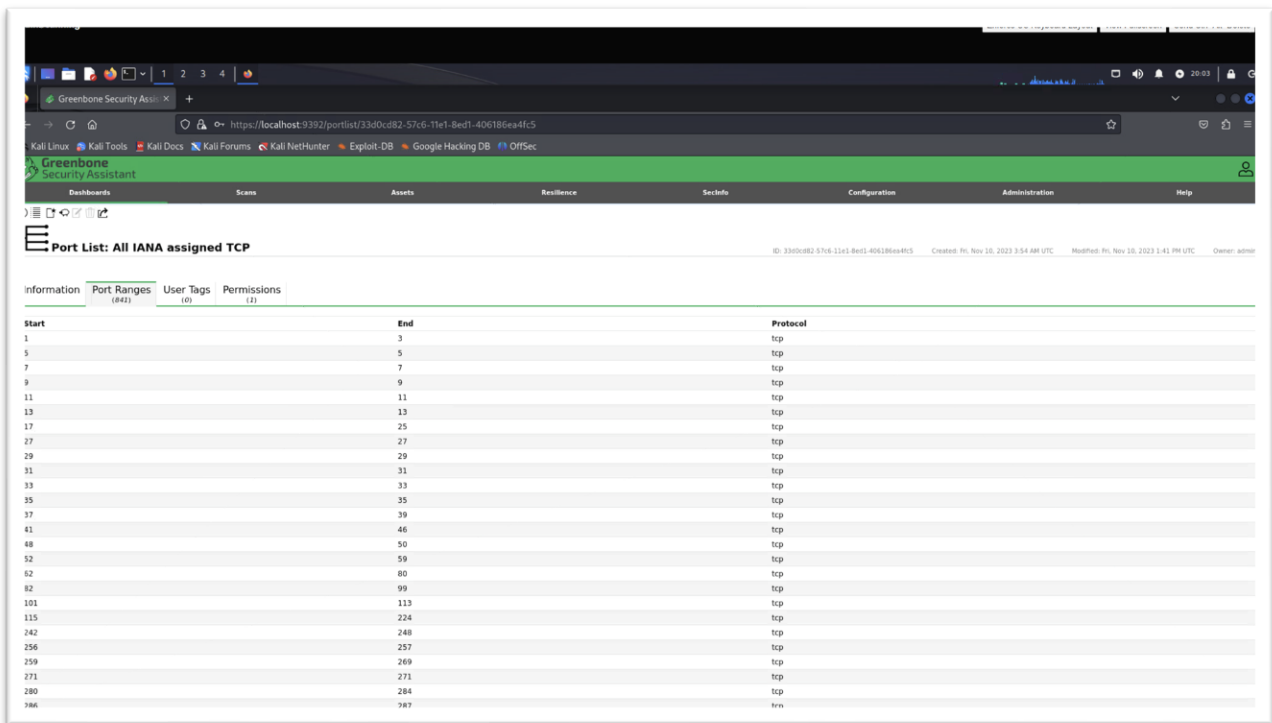


Figure 5: Screenshot of “Port Ranges window” inside a port list.

- We enter command “sudo apt install postfix mailutils” which will help us configure the client to be able to send mail as highlighted below.

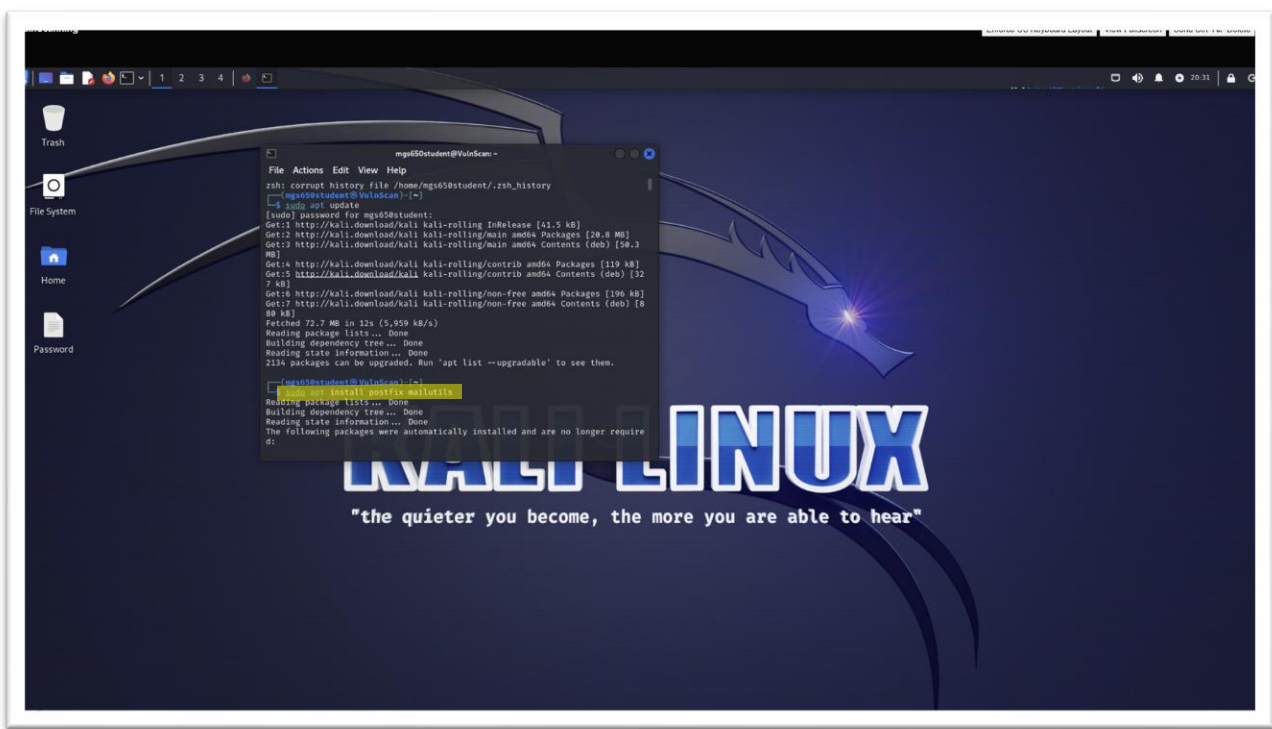


Figure 6: Screenshot of entering “sudo apt install postfix mailutils” command.

- Then run “sudo nano /etc/postfix/main.cf” and scroll down till we find inet_interfaces = all and edit it to “inet_interfaces = loopback-only” as shown in figure 7.

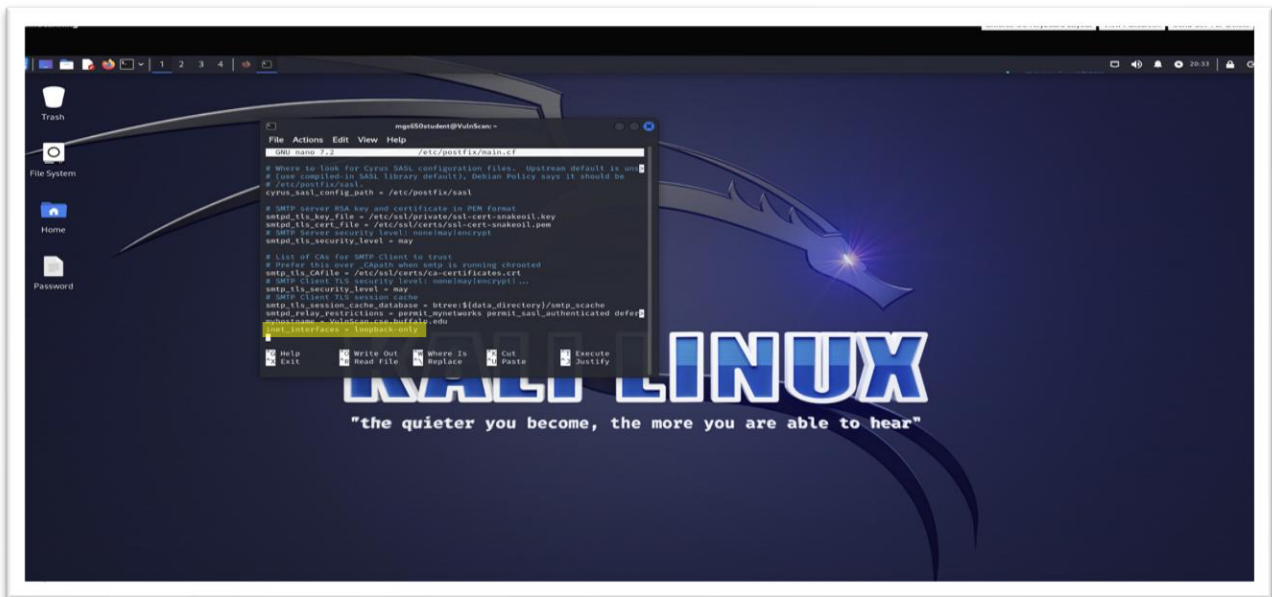


Figure 7: Screenshot of editing in “sudo nano /etc/postfix main.cf”.

- Then we restart the mail services using “sudo systemctl restart postfix” and then finally run these services using “echo “Email Test from Kali” | mail -s “mgs650” fahmed29@buffalo.edu” (figure8). We will receive a mail from this client to our email as shown in figure 9.

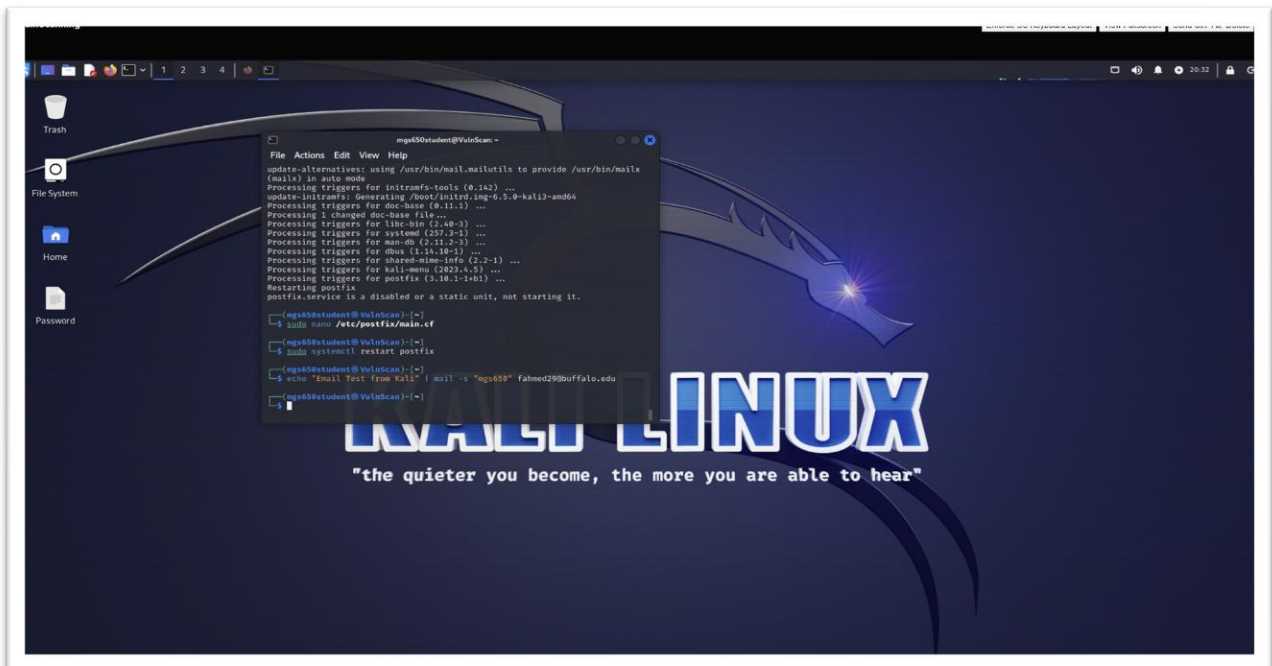


Figure 8: Screenshot of entering “echo “Email Test from Kali” | mail -s “mgs650” fahmed29@buffalo.edu” to send an email.

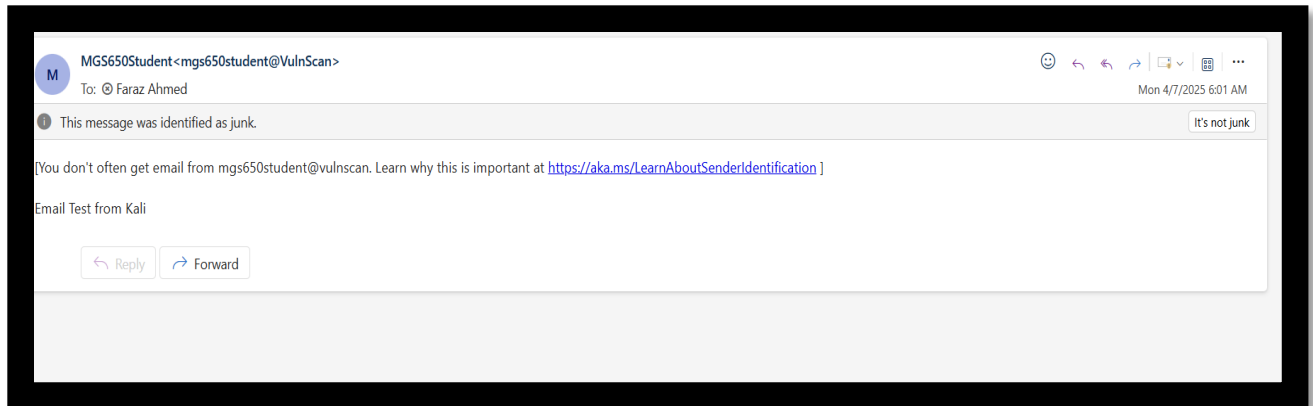


Figure 9: Screenshot of Mail received from Kali Client.

- Now we make a new alert by navigating to Configuration>Alert and that little box icon of top left corner to add new alert. Then the same instructions as shown in figure 10 and 11, then select “Save”.

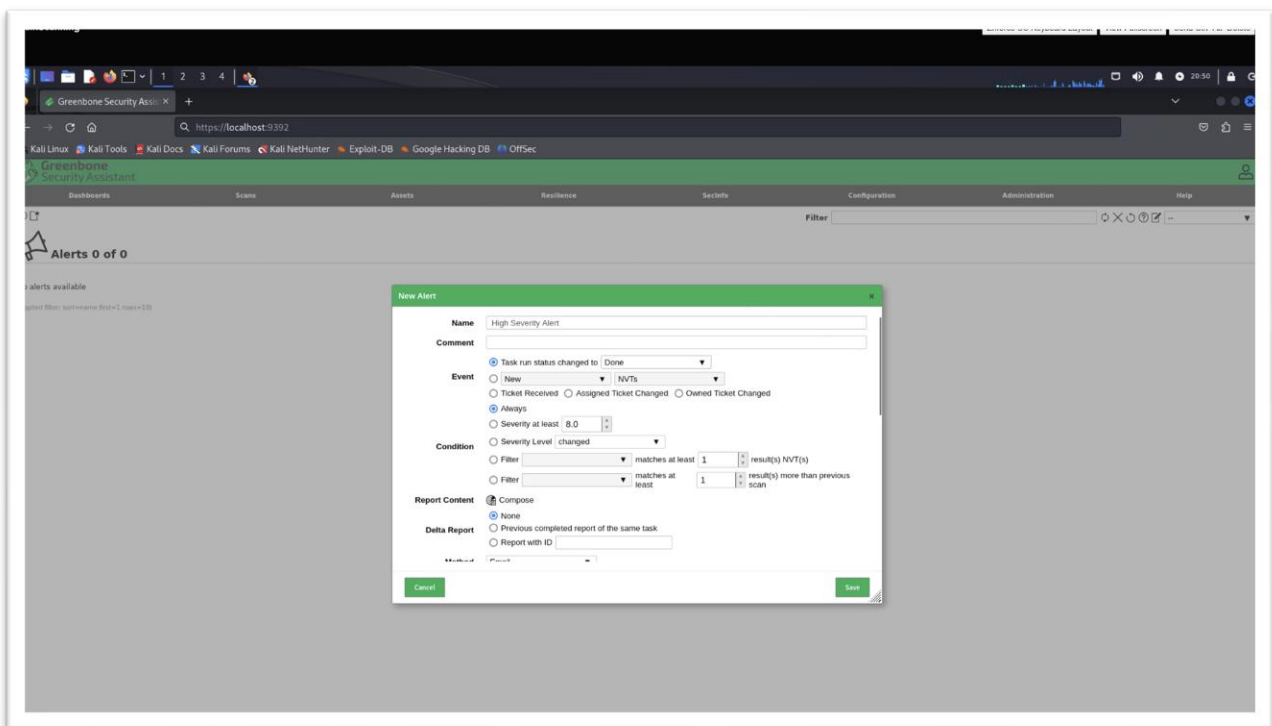


Figure 10: Screenshot of adding details to create a new alert.

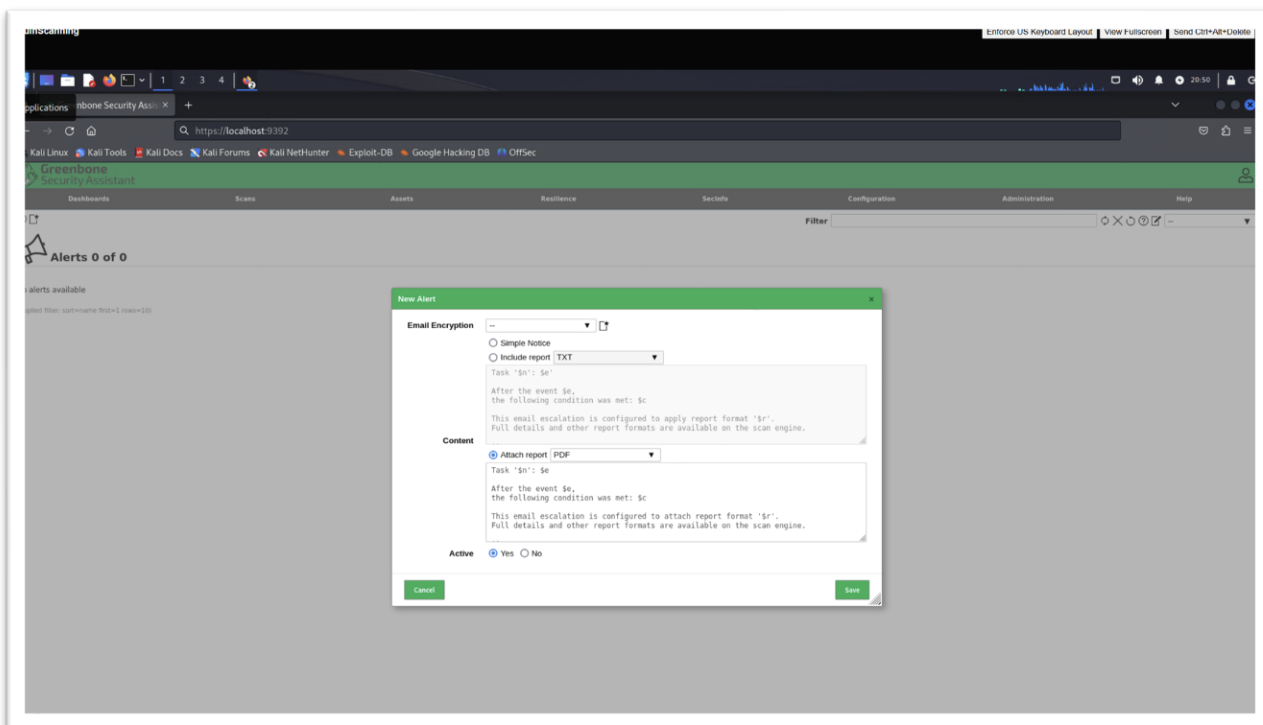


Figure 11: Screenshot of adding details to create a new alert.

- Now we go back to the scan task page and click “Edit task” button in my recent report and then select to add newly created alert which in this case is “High Severity Alert” (Highlighted in figure 12) and then click save.

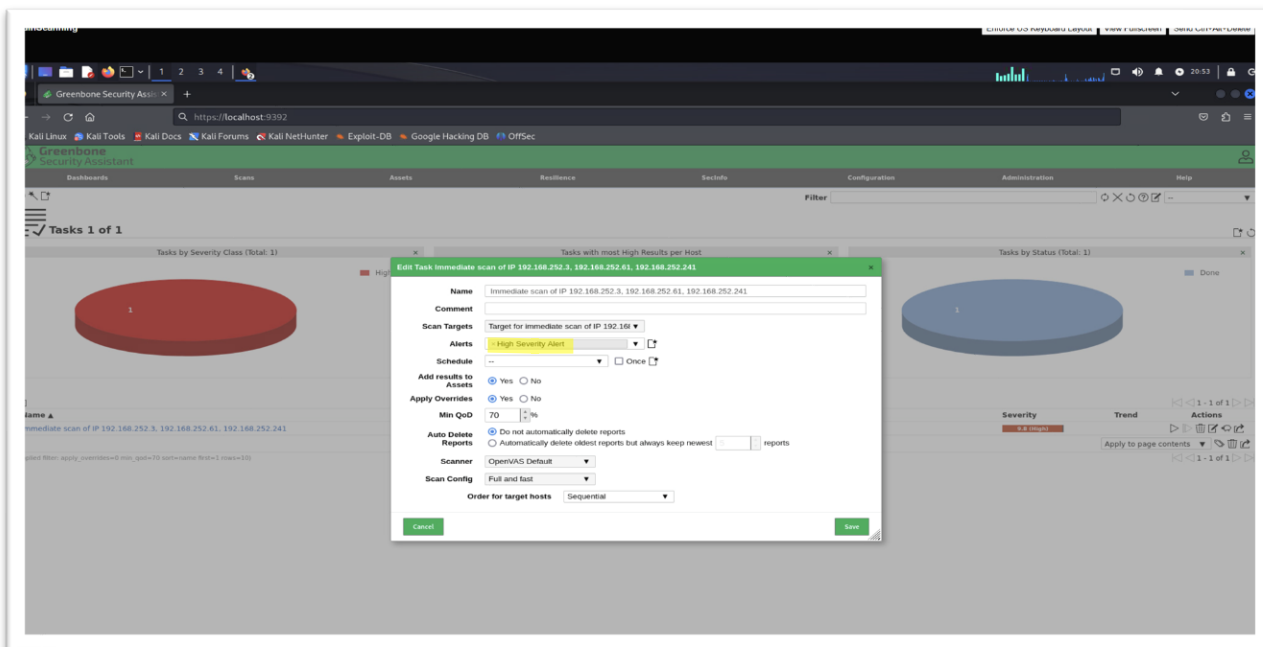


Figure 12: Screenshot of adding “High Severity Alert” to Immediate Scan.

- Then we go to Scans>Reports and check if the scan is fully done. If yes, then we can select the date which will open a new window. Then click on the download icon in top left and select “PDF” in Report Format and press OK. So, this is automatically downloading pdf of report.

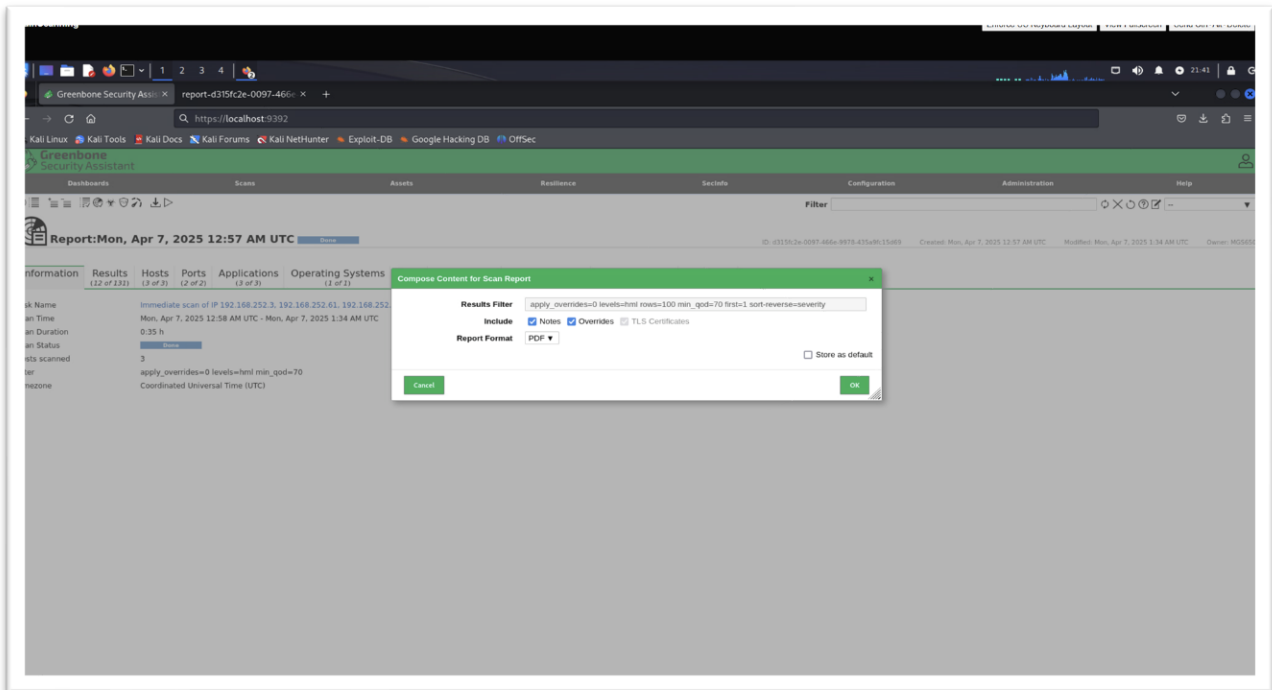


Figure 13: Screenshot of downloading full report in pdf format.

- When the scan is completed, we can navigate to Scans>Results and can observe all vulnerabilities detected during the scans as observed in figure 14.

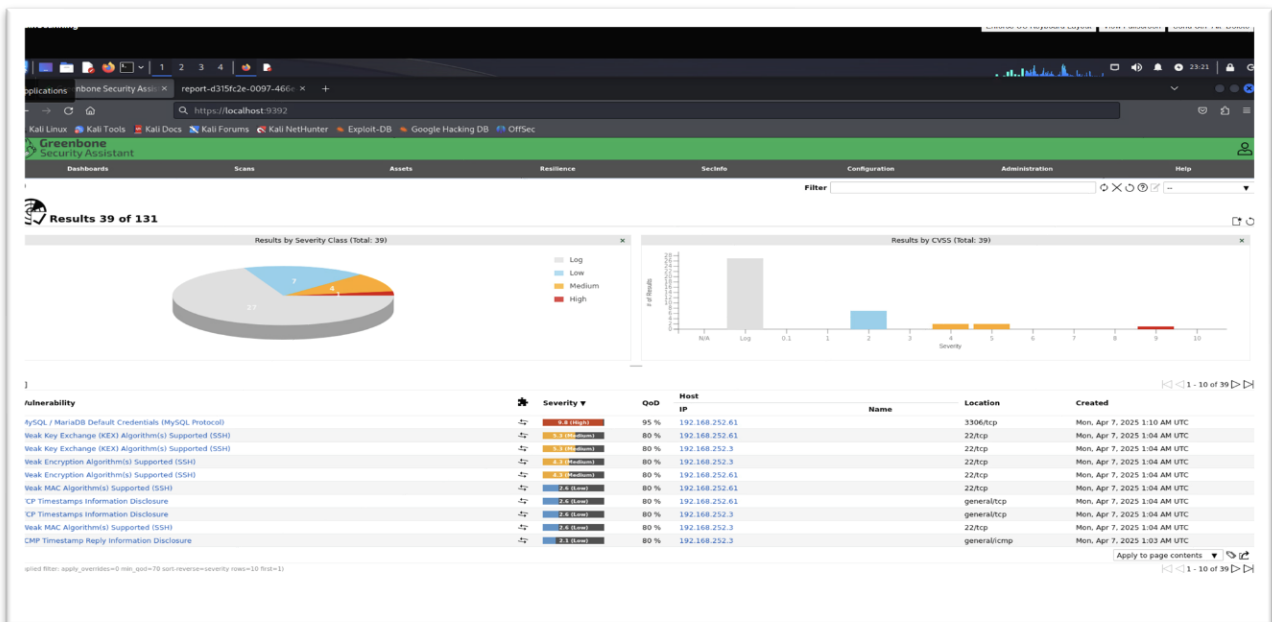


Figure 14: Screenshot of list of vulnerabilities detected during the scan.

- Now go to Scans>Report and select the one report we get from scanning. After that we get every detail of report as observed in figure 15.

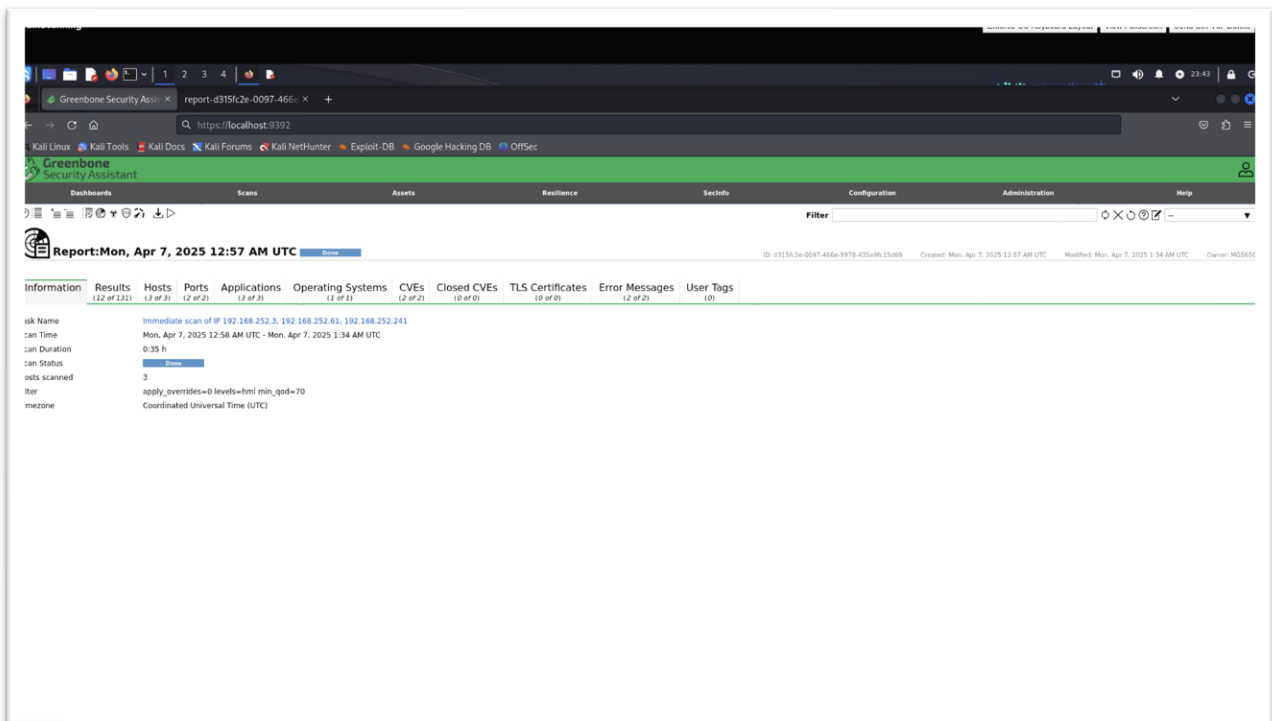


Figure 15: Screenshot on report page of the scan.

Summary of the Vulnerabilities

➤ For 192.168.252.61

- High (CVSS 9.8)- MySQL/ MariaDB Default Credentials- This is a vulnerability reported for weak default password credentials specifically for the root account with an empty password which means that anyone can login to the database using root access (which is highest level of privilege) without any need of entering password. So, the basic solution to this would be just to change the password of the root account to something long and strong and not share that with anyone or contact the vendor of MariaDB or MySQL if there is some other way for these vulnerabilities.
- Medium (CVSS 5.3)- Weak Key Exchange (KEX) Algorithm Supported in SSH- So the SSH server is basically using a weak cryptographic algorithm for a key exchange and this type of algorithm is essential to make a secure connection between two systems. So, if the key exchange is weak then the attacker can easily break the connection and compromise the systems. This uses a 1024- bit MODP group and SHA-1 for hashing which both are outdated and too small for security from modern attacks hence can be exploited easily. So, the basic solutions to this would be to avoid using 1024- bit MODP groups and SHA-1 for key exchanges and instead use elliptic-curve Diffie-Hellmann algorithm which provides better overall security.
- Medium (CVSS 4.3)- Weak Encryption Algorithms Supported in SSH- Again SSH server contains some weak or outdated encryption algorithms such as CBC-based Ciphers (3des-cbc, aes128-cbc, aes192-cbc, aes256-cbc, blowfish-cbc, cast128-cbc, rijndael-cbc@lysator.liu.se) and RC4-based ciphers (arcfour, arcfour128, arcfour256) for both client-to-server and server-to-client side communications. While leaving weak algorithms can lead to some confidentiality breaches and theft of certain sensitive data or information. So, to avoid that, one can update the SSH server configurations to disable all of the listed and apply some other modern strong ciphers like aes256-ctr.
- Low (CVSS 2.6)- Weak MAC Algorithm Supported in SSH- SSH server uses weak Message Authentication Code (MAC) algorithm which is used to ensure the security, integrity and authenticity of SSH. So, the use of these outdated or weak MACs can some connection to attackers. Some of these also are MD5-based (hmac-md5, hmac-md5-96, hmac-md5-etm@openssh.com, hmac-md5-96-etm@openssh.com) and SHA1 (hmac-sha1-96, hmac-sha1-96-etm@openssh.com) and both MD5 and SHA1 are broken and not safe for use. So, we can update SSH server configuration to disable all of those weak MACs and use strong MAC algorithms like hmac-sha2-256 and hmac-sha2-512.
- Low (CVSS 2.6)- TCP Timestamps Information Disclosure- TCP packets consist of timestamp options which can improve performance of system by utilizing features like Round Trip Time (RTT) and Protection Against Wrapped Sequences (PAWS). So, an attacker can use uptime information to identify reboot cycle and crash frequencies and also enable OS fingerprinting. So, in order to mitigation this issue, we can disable TCP timestamps if it's not required for normal use.

- Low (CVSS 2.1)- ICMP Timestamp Reply Information Disclosure- Remote host basically response to an initial ICMP Timestamp Request (Type 13) and get a reply Timestamp (Type 14). So, this reveals some system's time information which an attacker can exploit to hamper with overall system's performance, modify local time and systems uptime. So, to mitigate this, either we can disable the ICMP Timestamp Replies or block ICMP timestamp requests at the firewall and router.

➤ For 192.168.252.3

- Medium (CVSS 5.3)- Weak Key Exchange (KEX) algorithm supported in SSH- This vulnerability is similar to "Medium 22/tcp" vulnerability in port 192.168.252.61.
- Medium (CVSS 4.3)- Weak Encryption Algorithms Supported in SSH- This vulnerability is similar to "Medium 22/tcp" vulnerability in port 192.168.252.61.
- Low (CVSS 2.6)- TCP Timestamps Information Disclosure- This vulnerability is similar to "Low general/tcp" vulnerability in port 192.168.252.61.
- Low (CVSS 2.6)- Weak MAC Algorithms Supported in SSH- This vulnerability is similar to "Low 22/tcp" vulnerability in port 192.168.252.61.
- Low (CVSS 2.1)- ICMP Timestamp Reply Information Disclosure- This vulnerability is similar to "Low general/icmp" vulnerability in port 192.168.252.61.

➤ For 192.168.252.241

- Low (CVSS 2.1)- ICMP Timestamp Reply Information Disclosure- This vulnerability is similar to "Low general/icmp" vulnerability in port 192.168.252.61.