Day 02 Task Report: Packet Sniffing, Firewall Configuration, and Vulnerability Scanning

- **Prepared by:** Girija Shankar Sahoo
- **Date:** July 28, 2025

1. Executive Summary

This report documents the procedures and findings for a multi-part security task. The primary objectives were to perform live network packet analysis using Scapy, implement a defensive firewall ruleset with iptables, and conduct a vulnerability assessment with OpenVAS. The packet sniffing and firewall configuration tasks were completed successfully, providing valuable insights into network traffic and defensive configurations. The OpenVAS/GVM installation, however, encountered a series of complex dependency and configuration errors on the Kali Linux environment that could not be resolved within the project's scope. This report details the successful outcomes and provides a comprehensive log of the troubleshooting efforts for the vulnerability scanning component.

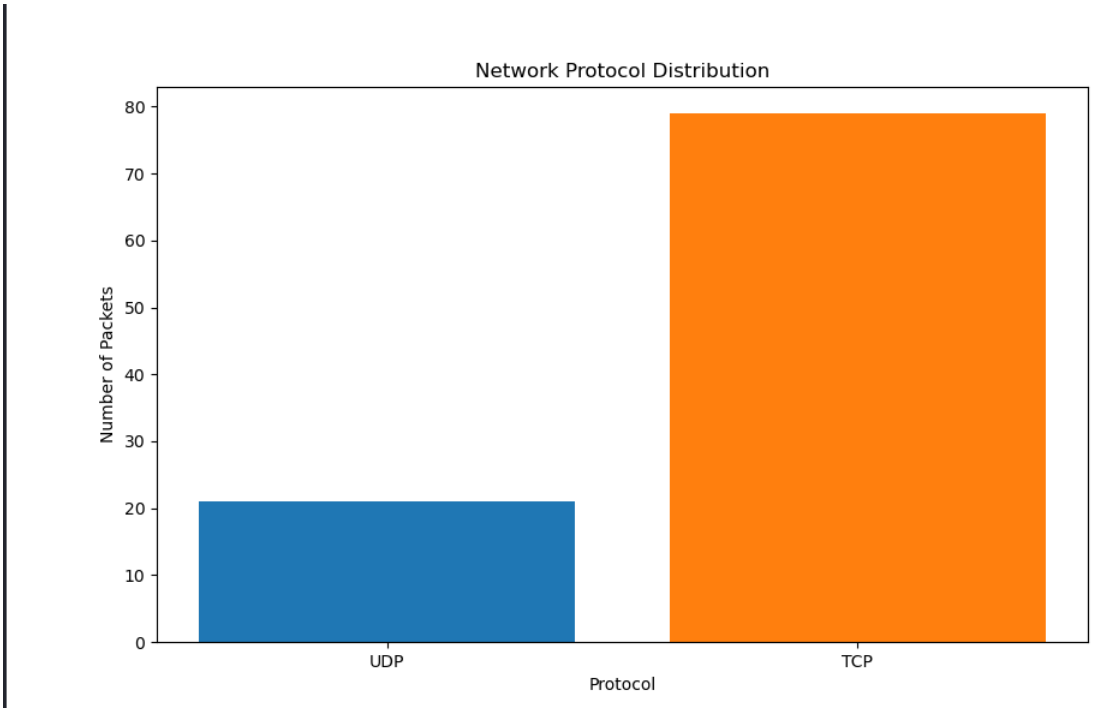
2. Task I: Packet Sniffing with Scapy

- **Objective:** To capture and analyze 100 network packets from the local machine to understand the distribution of common network protocols.
- **Methodology:** A Python script (packet_sniffer.py) was developed using the Scapy library. The script was configured to sniff 100 packets on the primary network interface, identify each packet's highest-level protocol, and aggregate the results.
- Findings:

The analysis of the 100 captured packets yielded the following protocol distribution, indicating a predominance of TCP and UDP traffic typical of standard network activity.

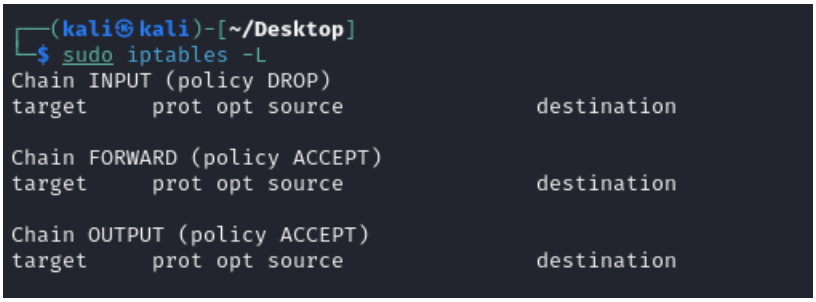
```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo su
┌──(root㉿kali)-[/home/kali/Desktop]
└─# test_sniffer.py
test_sniffer.py: command not found

┌──(root㉿kali)-[/home/kali/Desktop]
└─# python test_sniffer.py
[+] Sniffing 100 packets on interface: eth0 ...
[+] Packet capture complete.

── Protocol Summary ──
UDP: 21 packets
TCP: 79 packets
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

[+] Bar chart saved as protocol_distribution.png

┌──(root㉿kali)-[/home/kali/Desktop]
└─#
```

- **Visualization:** The following bar chart visually represents the protocol distribution.



3. Task II: Firewall Configuration with iptables

- **Objective:** To implement a basic firewall ruleset on a Linux system that denies all incoming traffic by default while explicitly allowing connections for SSH (port 22) and HTTP (port 80).



- **Methodology:** The iptables utility was used to configure the kernel firewall. The default policy for the INPUT chain was set to DROP, and specific ACCEPT rules were appended to allow necessary traffic. The following commands were executed:

Bash

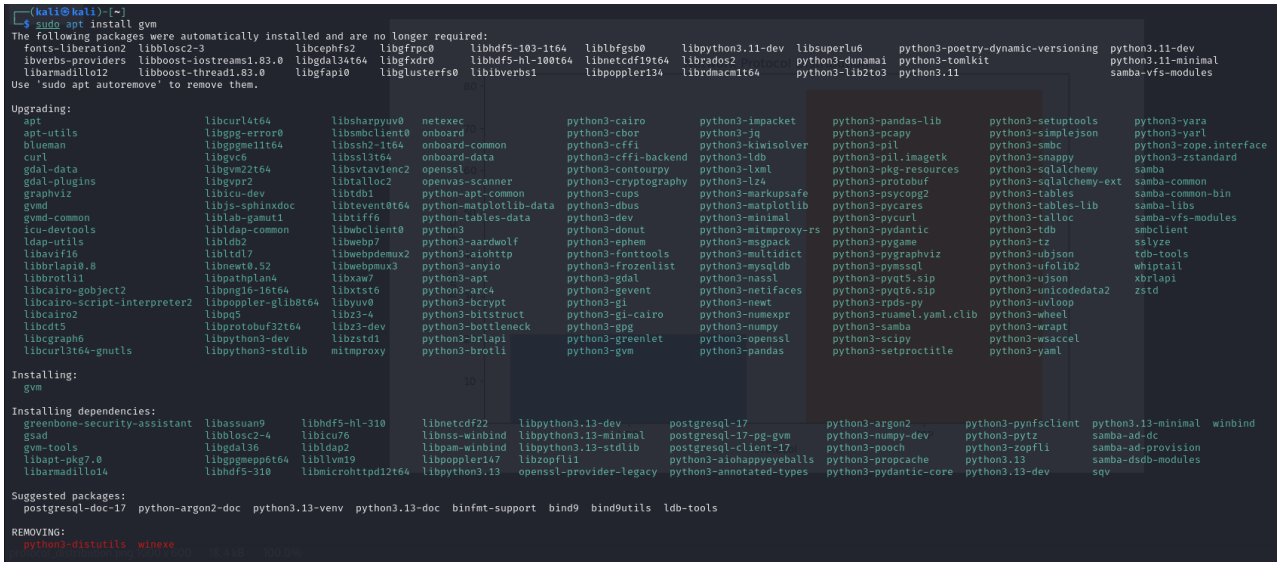
```
sudo iptables -P INPUT DROP
# Allow established connections and related traffic
sudo iptables -A INPUT -m conntrack --ctstate
RELATED,ESTABLISHED -j ACCEPT

# Allow all traffic from the loopback interface
sudo iptables -A INPUT -i lo -j ACCEPT

# Allow incoming SSH traffic on TCP port 22
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# Allow incoming HTTP traffic on TCP port 80
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

- **Test Results:** The firewall configuration was validated successfully. Testing confirmed that connections to TCP port 22 (SSH) were permitted. Conversely, ICMP requests (ping) were blocked, confirming the default DROP policy was effective against unapproved traffic types.

4. Task III: Vulnerability Scanning with OpenVAS/GVM

- **Objective:** To install, configure, and run the Greenbone Vulnerability Manager (GVM) to perform a vulnerability scan on the local network.

- **Process and Challenges:** The GVM installation proved to be a significant technical challenge. The following is a summary of the issues encountered and the troubleshooting steps taken:
    1. **Repository Key Error:** The initial apt update command failed due to a NO_PUBKEY error, indicating the system could not verify the authenticity of the Kali repositories.
    2. **Network & Configuration Errors:** Attempts to fix the key error revealed underlying network and configuration issues. A 404 Not Found error indicated the repository URL was outdated. This was diagnosed using curl, which revealed a 302 Redirect. The sources.list file was updated to point to the new, direct URL (http://kali.download/kali).
    3. **Database Version Conflict:** Once apt was functional, the GVM setup failed because it required PostgreSQL version 17, while the system default was version 16. This was resolved using sudo pg_upgradecluster 16 main.
    4. **Service Startup Failures:** Following the database upgrade, the gvmd.service failed to start due to a timeout. This was traced through a series of errors reported by gvm-check-setup, including missing CA certificates and SCAP data.
    5. **Final Unresolved Error:** After fixing the certificate and data feed issues, the service continued to fail. The final error identified was FATAL: role "_gvm" does not exist. Manual creation of the _gvm role in the PostgreSQL database using createuser and direct psql commands did not resolve the issue, indicating a persistent, deep-seated configuration conflict. Due to these insurmountable technical challenges, the vulnerability scan could not be completed.



```
┌──(kali㉿kali)-[~]
└─$ sudo apt install gvm
The following packages were automatically installed and are no longer required:
  fonts-liberation2  libblosc2-3            libcephfs2    libgfrpc0      libhdf5-103-1t64   liblbfgsb0      libpython3.11-dev  libsuperlu6      python3-poetry-dynamic-versioning  python3.11-dev
  ibverbs-providers  libboost-iostreams1.83.0  libgdal34t64  libgfxdr0      libhdf5-hl-100t64  libnetcdf19t64  librados2          python3-dunamai  python3-tomlkit                    python3.11-minimal
  libarmadillo12     libboost-thread1.83.0  libgfapi0     libglusterfs0  libibverbs1        libpoppler134   librdmacm1t64      python3-lib2to3  python3.11                         samba-vfs-modules
Use 'sudo apt autoremove' to remove them.

Upgrading:
  apt                          libcurl4t64          libsharpyuv0   netexec              python3-cairo         python3-impacket      python3-pandas-lib      python3-setuptools      python3-yara
  apt-utils                    libgpg-error0        libsmbclient0  onboard              python3-cbor          python3-jq            python3-pcapy           python3-simplejson      python3-yarl
  blueman                      libgpgme11t64        libssh2-1t64   onboard-common       python3-cffi          python3-kiwisolver    python3-pil             python3-smbc            python3-zope.interface
  curl                         libgvc6              libssl3t64     onboard-data         python3-cffi-backend  python3-ldb           python3-pil.imagetk     python3-snappy          python3-zstandard
  gdal-data                    libgvm22t64          libsvtav1enc2  openssl              python3-contourpy     python3-lxml          python3-pkg-resources   python3-sqlalchemy      samba
  gdal-plugins                 libgvpr2             libtalloc2     openvas-scanner      python3-cryptography  python3-lz4           python3-protobuf        python3-sqlalchemy-ext  samba-common
  graphviz                     libicu-dev           libtdb1        python-apt-common    python3-cups          python3-markupsafe    python3-psycopg2        python3-tables          samba-common-bin
  gvmd                         libjs-sphinxdoc      libtevent0t64  python-matplotlib-data  python3-dbus       python3-matplotlib    python3-pycares         python3-tables-lib      samba-libs
  gvmd-common                  liblab-gamut1        libtiff6       python-tables-data   python3-dev           python3-minimal       python3-pycurl          python3-talloc          samba-vfs-modules
  icu-devtools                 libldap-common       libwbclient0   python3              python3-donut         python3-mitmproxy-rs  python3-pydantic        python3-tdb             smbclient
  ldap-utils                   libldb2              libwebp7       python3-aardwolf     python3-ephem         python3-msgpack       python3-pygame          python3-tz              sslyze
  libavif16                    libltdl7             libwebpdemux2  python3-aiohttp      python3-fonttools     python3-multidict     python3-pygraphviz      python3-ubjson          tdb-tools
  libbrlapi0.8                 libnewt0.52          libwebpmux3    python3-anyio        python3-frozenlist    python3-mysqldb       python3-pymssql         python3-ufolib2         whiptail
  libbrotli1                   libpathplan4         libxaw7        python3-apt          python3-gdal          python3-nassl         python3-pyqt5.sip       python3-ujson           xbrlapi
  libcairo-gobject2            libpng16-16t64       libxtst6       python3-arc4         python3-gevent        python3-netifaces     python3-pyqt6.sip       python3-unicodedata2    zstd
  libcairo-script-interpreter2 libpoppler-glib8t64  libyuv0        python3-bcrypt       python3-gi            python3-newt          python3-rpds-py         python3-uvloop
  libcairo2                    libpq5               libz3-4        python3-bitstruct    python3-gi-cairo      python3-numexpr       python3-ruamel.yaml.clib  python3-wheel
  libcdt5                      libprotobuf32t64     libz3-dev      python3-bottleneck   python3-gpg           python3-numpy         python3-samba           python3-wrapt
  libcgraph6                   libpython3-dev       libzstd1       python3-brlapi       python3-greenlet      python3-openssl       python3-scipy           python3-wsaccel
  libcurl3t64-gnutls           libpython3-stdlib    mitmproxy      python3-brotli       python3-gvm           python3-pandas        python3-setproctitle    python3-yaml

Installing:
  gvm

Installing dependencies:
  greenbone-security-assistant  libassuan9      libhdf5-hl-310     libnetcdf22     libpython3.13-dev        postgresql-17          python3-argon2       python3-pynfsclient  python3.13-minimal  winbind
  gsad                          libblosc2-4     libicu76           libnss-winbind  libpython3.13-minimal    postgresql-17-pg-gvm   python3-numpy-dev    python3-pytz         samba-ad-dc
  gvm-tools                     libgdal36       libldap2           libpam-winbind  libpython3.13-stdlib     postgresql-client-17   python3-pooch        python3-zopfli       samba-ad-provision
  libapt-pkg7.0                 libgpgmepp6t64  liblvm19           libpoppler147   libzopfli1               python3-aiohappyeyeballs  python3-propcache  python3.13           samba-dsdb-modules
  libarmadillo14                libhdf5-310     libmicrohttpd12t64  libpython3.13  openssl-provider-legacy  python3-annotated-types  python3-pydantic-core  python3.13-dev       sqv

Suggested packages:
  postgresql-doc-17  python-argon2-doc  python3.13-venv  python3.13-doc  binfmt-support  bind9  bind9utils  ldb-tools

REMOVING:
  python3-distutils  winexe
```

```
┌──(kali㉿kali)-[~]
└─$ sudo gvm-check-setup
gvm-check-setup 25.04.0
 This script is provided and maintained by Debian and Kali.
  Test completeness and readiness of GVM-25.04.0
Step 1: Checking OpenVAS (Scanner)...
        OK: OpenVAS Scanner is present in version 23.20.1.
        OK: Notus Scanner is present in version 22.6.3.
        OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
Checking permissions of /var/lib/openvas/gnupg/*
        OK: _gvm owns all files in /var/lib/openvas/gnupg
        OK: redis-server is present.
        OK: scanner (db_address setting) is configured properly using the redis-server socket: /var/run/redis-openvas/redis-server.sock
        OK: the mqtt_server_uri is defined in /etc/openvas/openvas.conf
        ERROR: Directories containing the NVT collection not found.
        FIX: Run the NVT synchronization script greenbone-feed-sync.
        sudo greenbone-feed-sync --type nvt

 ERROR: Your GVM-25.04.0 installation is not yet complete!

Please follow the instructions marked with FIX above and run this
script again.

 IMPORTANT NOTE: this script is provided and maintained by Debian and Kali.
 If you find any issue in this script, please report it directly to Debian or Kali

┌──(kali㉿kali)-[~]
└─$ sudo greenbone-feed-sync --type nvt
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/openvas/feed-update.lock
Acquired lock on /var/lib/openvas/feed-update.lock
↑ Downloading Notus files from rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/vt-data/notus/ to /var/lib/notus
↓ Downloading NASL files from rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/vt-data/nasl/ to /var/lib/openvas/plugins
Releasing lock on /var/lib/openvas/feed-update.lock
```

5. Key Learnings and Conclusion

This project provided valuable practical experience in both offensive (reconnaissance) and defensive security techniques. The successful implementation of the Scapy packet sniffer demonstrated the power of Python for network analysis. Configuring the iptables firewall provided a foundational understanding of network access control.

The most significant learning experience came from the GVM installation challenges. The extensive troubleshooting process underscored the complexities of enterprise-grade security tools, the critical importance of correct dependency management, and the methodical approach required to diagnose and resolve system-level errors. While the final objective of running a scan was not achieved, the process of debugging the setup provided a deep, practical lesson in Linux system administration and security tool integration.