Aim: Install and configure Bro/Zeek, an open-source network analysis framework. Set up anomaly detection rules in Bro/Zeek. Analyze the generated network logs to identify anomalies.

Bro/Zeek Installation

- **Update & Prepare System**
  - Update packages: `sudo apt update && sudo apt upgrade`
  - Install dependencies: `sudo apt install cmake make gcc g++ flex bison libpcap-dev libssl-dev python3-dev swig zlib1g-dev`
- **Download & Install**
  - Official package (recommended for Ubuntu):

```
curl -fsSL
https://download.opensuse.org/repositories/security:zeek/xUbuntu_24.04/Re
lease.key | gpg --dearmor | tee /etc/apt/trusted.gpg.d/security_zeek.gpg
echo 'deb
http://download.opensuse.org/repositories/security:/zeek/xUbuntu_24.04/
/' | tee /etc/apt/sources.list.d/security:zeek.list
sudo apt update
sudo apt install zeek
```

  - Add Zeek to PATH:

```
echo "export PATH=$PATH:/opt/zeek/bin" >> ~/.bashrc
source ~/.bashrc
```

- **Verify Installation**
  - Check version: `zeek --version`

- Output should show installed Zeek version.

Zeek Configuration

- **Edit Network Definitions**
  - Open Zeek network config: `nano /opt/zeek/etc/networks.cfg`
  - Add your internal network (example):

    ```
    10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
    ```

- **Configure Cluster/Node**
  - Open node config: `nano /opt/zeek/etc/node.cfg`
  - Set interface and node roles. Example for standalone:

    ```
    [zeek]
    type=standalone
    host=localhost
    interface=eth0
    ```

  - For clustered setup, configure logger, manager, proxy, and workers as needed.
- **Deploy & Start**
  - Deploy configuration: `zeekctl deploy`
  - Check status: `zeekctl status`

Setting Up Anomaly Detection Rules

- **Install Community or Custom Scripts**
  - Zeek comes with built-in detection scripts (e.g., `scan.zeek` for port scans), and many custom packages are available.
  - Example: DNS anomaly detection
    - Use Zeek Package Manager (zkg): `zkg install sensorfleet/anomalous-dns`
    - Restart Zeek: `/etc/init.d/zeek restart`

    - Alternatively, copy custom `.zeek` scripts to `/opt/zeek/share/zeek/site/` and add to `local.zeek`.

- **Sample Custom Rule (scan detection pseudocode)**

```
event zeek_scan(num_ports: count, src_ip: string) {
  if (num_ports > 20)
    print("Potential reconnaissance activity from ", src_ip);
}
```

- **Notice/Alert Logging**
    - Custom anomalies can trigger entries in `notice.log` with alert text, for integration with SIEM/workflow automation.

Log Analysis & Anomaly Identification

- **Log Locations**
    - Logs by default are at `/opt/zeek/logs/current/`
    - Key log files:
        - `conn.log` — all network connections
        - `scan.log` — port scans detected
        - `dns.log` — DNS queries and anomalies
        - `notice.log` — alerts/notifications
- **Analyzing Events**
    - Example CLI extraction (port scan):

```
zcat logs/current/scan.log.gz | jq '. | {ts, src_ip: .src, dest_ip:
.dst, num_ports: .num_ports}'
```

    - Elevated DNS NXDOMAIN (failed query) rates and spikes suggest possible DGA or suspicious activity.
    - Large files, extended durations in `conn.log` and `files.log` may indicate exfiltration attempts.

Practical Steps for Your Network

- Install Zeek on a Linux host as per above.
- Configure the local network address.
- Enable anomaly detection scripts as needed.
- Start Zeek, generate potentially anomalous traffic (nmap scans, high-volume DNS, large file transfers).
- Analyze Zeek's log files for alerts and abnormal patterns as outlined.

This document gives all steps needed to install, configure, and practically use Zeek for basic anomaly detection. Customize detection scripts for specific scenarios and regularly monitor logs for real-time threat identification.