Capture network packets in a PCAP file using Scapy in a Python environment. Develop Python code to identify anomalies in the given packets inside the PCAP file based on the following criteria:

- Common destination ports for TCP and UDP
- Excessive Traffic (DDoS)
- Number of packets and packet size
- Unsolicited ARP replies
- Unusually large DNS responses
- Excessive ICMP Echo requests
- Excessive TCP SYN
- IPs scans excessive ports

Example:

```
from scapy.all import rdpcap, DNS, IP, ICMP, TCP, ARP
from collections import Counter
from collections import defaultdict
import time

# Load the PCAP file
packets = rdpcap('example.pcap')

# Inspect packets
for packet in packets:
    print(packet.summary())

non_standard_ports = set()
################################################################
# Detecting Traffic on Non-Standard Ports
################################################################
for packet in packets:
    if packet.haslayer('TCP'):
        tcp_layer = packet['TCP']
        if tcp_layer.dport not in [80, 443, 22]:  # Add standard
destination ports
            non_standard_ports.add(tcp_layer.dport)

print("Non-standard ports detected:", non_standard_ports)

# Rule 2: High Traffic Volume (DDoS Detection)
ip_count = Counter()

for packet in packets:
    if packet.haslayer('IP'):
        ip_layer = packet['IP']
        ip_count[ip_layer.src] += 1
################################################################
```