EXPERIMENT / PRACTICAL -4

Implement Write and analyze the output of various Network commands such as ping, ipconfig, arp, netstat, tracert, nslookup, hostname, systeminfo etc., with sample outputs

AIM:

Implement Write and analyze the output of various Network commands such as ping, ipconfig, arp, netstat, tracert, nslookup, hostname, systeminfo etc., with sample outputs

DESCRIPTION:

Network commands are instructions used to configure, manage, and troubleshoot network devices and connections. These commands can be issued through command-line interfaces (CLI) on network devices such as routers, switches, firewalls, and computers. Here's a detailed description of some commonly used network commands:

Basic Network Commands

1. **ping**:
   - **Purpose**: Tests connectivity between two devices on a network.
   - **Usage**: `ping [hostname/IP address]`
   - **Example**: `ping 192.168.1.1`
2. **traceroute/tracert**:
   - **Purpose**: Traces the path packets take from one device to another.
   - **Usage**:
     - Unix/Linux: `traceroute [hostname/IP address]`
     - Windows: `tracert [hostname/IP address]`
   - **Example**: `traceroute google.com` or `tracert google.com`
3. **ipconfig/ifconfig**:
   - **Purpose**: Displays network configuration details.
   - **Usage**:
     - Windows: `ipconfig`
     - Unix/Linux: `ifconfig`
   - **Example**: `ipconfig /all` or `ifconfig eth0`
4. **netstat**:
   - **Purpose**: Displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
   - **Usage**: `netstat [options]`
   - **Example**: `netstat -a`
5. **nslookup**:
   - **Purpose**: Queries the Domain Name System (DNS) to obtain domain name or IP address mapping.
   - **Usage**: `nslookup [hostname/IP address]`
   - **Example**: `nslookup example.com`
6. **hostname**:
   - **Purpose**: Displays or sets the system's hostname.

- **Usage**: `hostname [new-hostname]`
- **Example**: `hostname`

Network Configuration Commands

1. **route**:
   - **Purpose**: Displays or modifies the IP routing table.
   - **Usage**: `route [add/delete] [destination] [gateway]`
   - **Example**: `route add 192.168.1.0 mask 255.255.255.0 192.168.1.1`
2. **arp**:
   - **Purpose**: Displays and modifies the ARP (Address Resolution Protocol) cache.
   - **Usage**: `arp [options] [hostname/IP address]`
   - **Example**: `arp -a`
3. **iptables**:
   - **Purpose**: Configures the IP packet filter rules of the Linux kernel firewall.
   - **Usage**: `iptables [options]`
   - **Example**: `iptables -L`
4. **ifup/ifdown**:
   - **Purpose**: Activates or deactivates network interfaces on Unix/Linux systems.
   - **Usage**: `ifup [interface]` or `ifdown [interface]`
   - **Example**: `ifup eth0` or `ifdown eth0`

Advanced Network Commands

1. **tcpdump**:
   - **Purpose**: Captures and displays packets on a network.
   - **Usage**: `tcpdump [options]`
   - **Example**: `tcpdump -i eth0`
2. **nmap**:
   - **Purpose**: Scans networks to discover hosts and services.
   - **Usage**: `nmap [options] [hostname/IP address]`
   - **Example**: `nmap -sP 192.168.1.0/24`
3. **wireshark**:
   - **Purpose**: Network protocol analyzer for capturing and analyzing traffic.
   - **Usage**: Typically used with a graphical interface, but can be run with command-line options.
   - **Example**: `wireshark` (GUI) or `tshark -i eth0` (CLI)

Windows-Specific Commands

1. **netsh**:
   - **Purpose**: Configures and displays the status of network components.
   - **Usage**: `netsh [context] [command]`
   - **Example**: `netsh interface ip show config`

2. **Get-NetIPAddress**:
   - **Purpose**: Displays IP address configuration on Windows PowerShell.

- **Usage**: `Get-NetIPAddress`
- **Example**: `Get-NetIPAddress`

Unix/Linux-Specific Commands

1. **dig**:
   - **Purpose**: Queries DNS name servers for information.
   - **Usage**: `dig [hostname/IP address]`
   - **Example**: `dig example.com`

2. **ip**:
   - **Purpose**: Manages IP addresses, network devices, and routing tables.
   - **Usage**: `ip [command]`
   - **Example**: `ipaddr show`

These commands are fundamental tools for network administrators and IT professionals, enabling them to manage and troubleshoot network infrastructure efficiently.

PROCEDURE:

1. `ping`

Description: The `ping` command is used to check the network connectivity between the host computer and a target (usually another computer or a server).

Command:

```
bash
Copy code
ping google.com
```

Sample Output:

```
plaintext
Copy code
Pinging google.com [142.250.64.78] with 32 bytes of data:
Reply from 142.250.64.78: bytes=32 time=14ms TTL=54
Reply from 142.250.64.78: bytes=32 time=13ms TTL=54
Reply from 142.250.64.78: bytes=32 time=15ms TTL=54
Reply from 142.250.64.78: bytes=32 time=12ms TTL=54

Ping statistics for 142.250.64.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 15ms, Average = 13ms
```

Analysis:

- The target (`google.com`) resolves to the IP address `142.250.64.78`.
- All packets sent were received (0% packet loss).
- The round-trip time (latency) varies between 12ms and 15ms, indicating a stable and responsive connection.

2. `ipconfig`

Description: The `ipconfig` command displays the network configuration details of the local machine.

Command:

```
bash
Copy code
ipconfig
```

Sample Output:

```
plaintext
Copy code
Windows IP Configuration

Ethernet adapter Ethernet:
   Connection-specific DNS Suffix  . : example.local
   Link-local IPv6 Address . . . . . : fe80::1c9e:5fbc:b001:6833%12
   IPv4 Address. . . . . . . . . . . : 192.168.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

Analysis:

- The computer has both an IPv4 address (`192.168.1.10`) and a link-local IPv6 address.
- The subnet mask is `255.255.255.0`, indicating a standard class C network.
- The default gateway is `192.168.1.1`, typically the router's address.

3. `arp`

Description: The `arp` command is used to view and manipulate the ARP (Address Resolution Protocol) cache, which maps IP addresses to MAC addresses.

Command:

```
bash
Copy code
arp -a
```

Sample Output:

```
plaintext
Copy code
Interface: 192.168.1.10 --- 0x12
  Internet Address      Physical Address      Type
  192.168.1.1           00-14-22-01-23-45     dynamic
  192.168.1.15          00-25-96-FF-EE-77     dynamic
```

Analysis:

- The ARP cache shows that the IP address `192.168.1.1` maps to the MAC address `00-14-22-01-23-45`.
- Another device on the network with IP `192.168.1.15` maps to the MAC address `00-25-96-FF-EE-77`.

4. `netstat`

Description: The `netstat` command displays network statistics, including current connections, routing tables, and interface statistics.

Command:

```
bash
Copy code
netstat
```

Sample Output:

```
plaintext
Copy code
Active Connections

Proto  Local Address          Foreign Address        State
  TCP    192.168.1.10:55478     ec2-52-87-185-95:443   ESTABLISHED
  TCP    192.168.1.10:55479     216.58.206.14:443      TIME_WAIT
  TCP    192.168.1.10:55480     172.217.10.46:443      ESTABLISHED
```

Analysis:

- The computer has active TCP connections to various remote addresses.
- The state `ESTABLISHED` indicates ongoing communication, while `TIME_WAIT` shows the connection is closing.

5. `tracert`

Description: The `tracert` command traces the path packets take to reach a network host.

Command:

```
bash
Copy code
tracert google.com
```

Sample Output:

```
plaintext
Copy code
Tracing route to google.com [142.250.64.78] over a maximum of 30
hops:

  1     1 ms<1 ms<1 ms  192.168.1.1
  2    12 ms    11 ms    12 ms  10.0.0.1
  3    13 ms    12 ms    11 ms  172.16.0.1
  4    14 ms    13 ms    12 ms  172.217.10.46
  5    15 ms    14 ms    13 ms  google.com [142.250.64.78]
```

Analysis:

- The command shows the route and time taken to reach each hop.
- The final destination (`google.com`) is reached in 5 hops.

6. `nslookup`

Description: The `nslookup` command queries DNS servers to obtain domain name or IP address mapping.

Command:

```
bash
Copy code
nslookup google.com
```

Sample Output:

```
plaintext
Copy code
Server:  resolver1.opendns.com
Address:  208.67.222.222

Non-authoritative answer:
Name:    google.com
Addresses:  142.250.64.78
```

Analysis:

- The DNS server used for the query is `208.67.222.222`.
- The resolved IP address for `google.com` is `142.250.64.78`.

7. `hostname`

Description: The `hostname` command displays the name of the current host (computer).

Command:

```
bash
Copy code
hostname
```

Sample Output:

```
plaintext
Copy code
MyComputer
```

Analysis:

- The name of the host machine is `MyComputer`.

8. `systeminfo`

Description: The `systeminfo` command provides detailed information about the computer's system configuration.

Command:

```
bash
Copy code
systeminfo
```

SAMPLE OUTPUT:

```
plaintext
Copy code
Host Name:                 MyComputer
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.19041 N/A Build 19041
OS Manufacturer:           Microsoft Corporation
System Manufacturer:       Dell Inc.
System Model:              XPS 15 9570
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
```

```
                           [01]: Intel64 Family 6 Model 158
Stepping 10 GenuineIntel ~2200 Mhz
BIOS Version:              Dell Inc. 1.12.0, 1/22/2021
```

ANALYSIS:

- The computer's hostname is `MyComputer`.
- It is running Windows 10 Pro, version 19041.
- The system is a Dell XPS 15 9570 with an Intel processor.

These commands provide a comprehensive view of network and system information, useful for diagnostics and troubleshooting.

RESULT:

The experiment has been conducted successfully.