Cybersecurity Portfolio Project Report

Project Title: Configure Extended Numbered and Named ACLs to Filter Traffic
Tool Used: Cisco Packet Tracer
Student Name: Thabiso K Dzveta

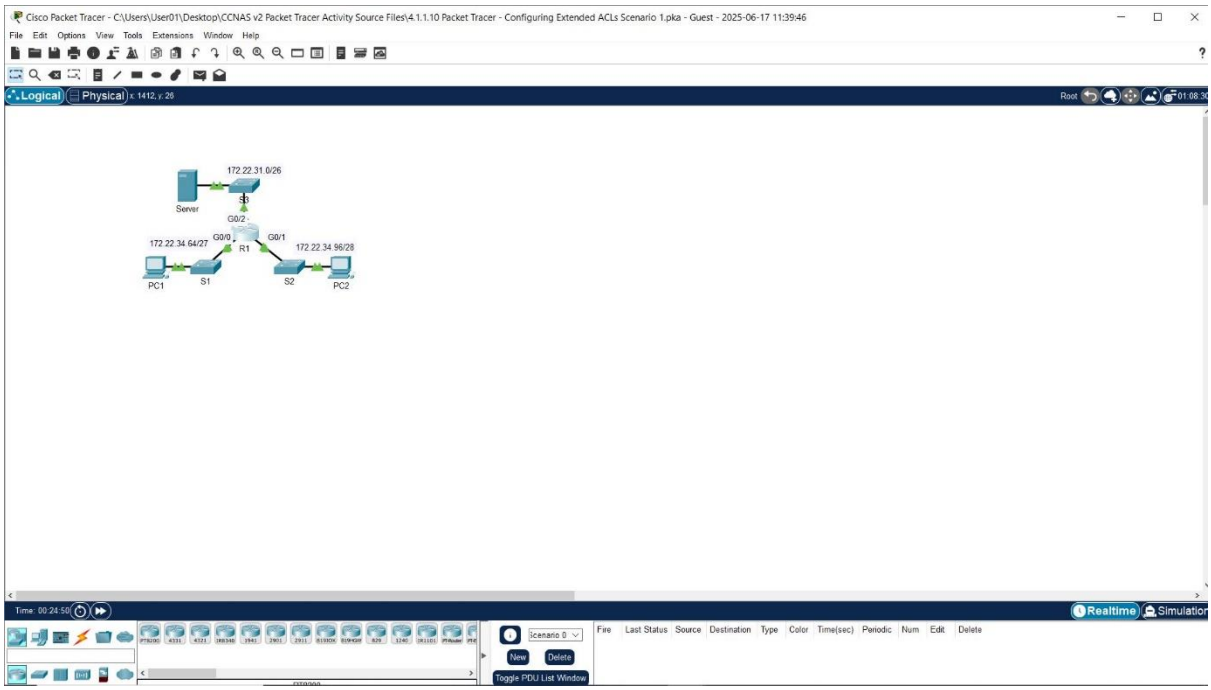
---

1. Project Overview

This project demonstrates the configuration and application of extended numbered and named Access Control Lists (ACLs) on a Cisco router to filter traffic based on IP addresses, protocols, and port numbers. The main objective was to allow specific services (FTP and HTTP) to designated hosts while denying all other traffic, thereby improving internal network segmentation and control.

---

2. Network Topology



---

3. IP Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	172.22.34.65	255.255.255.224	N/A
R1	G0/1	172.22.34.97	255.255.255.240	N/A
R1	G0/2	172.22.34.1	255.255.255.192	N/A
Server	NIC	172.22.34.62	255.255.255.192	172.22.34.1
PC1	NIC	172.22.34.66	255.255.255.224	172.22.34.65
PC2	NIC	172.22.34.98	255.255.255.240	172.22.34.97

---

4. ACL Configuration Summary

Part 1: Extended Numbered ACL (FTP & ICMP for PC1)

Step-by-Step:

1. **Permit FTP Access to Server from PC1's Network**

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp
```

2. **Permit ICMP (ping) Access**

```
R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62
```

3. **Apply ACL 100 Inbound on G0/0**

```
R1(config)# interface g0/0
R1(config-if)# ip access-group 100 in
```

R1

Physical Config CLI Attributes

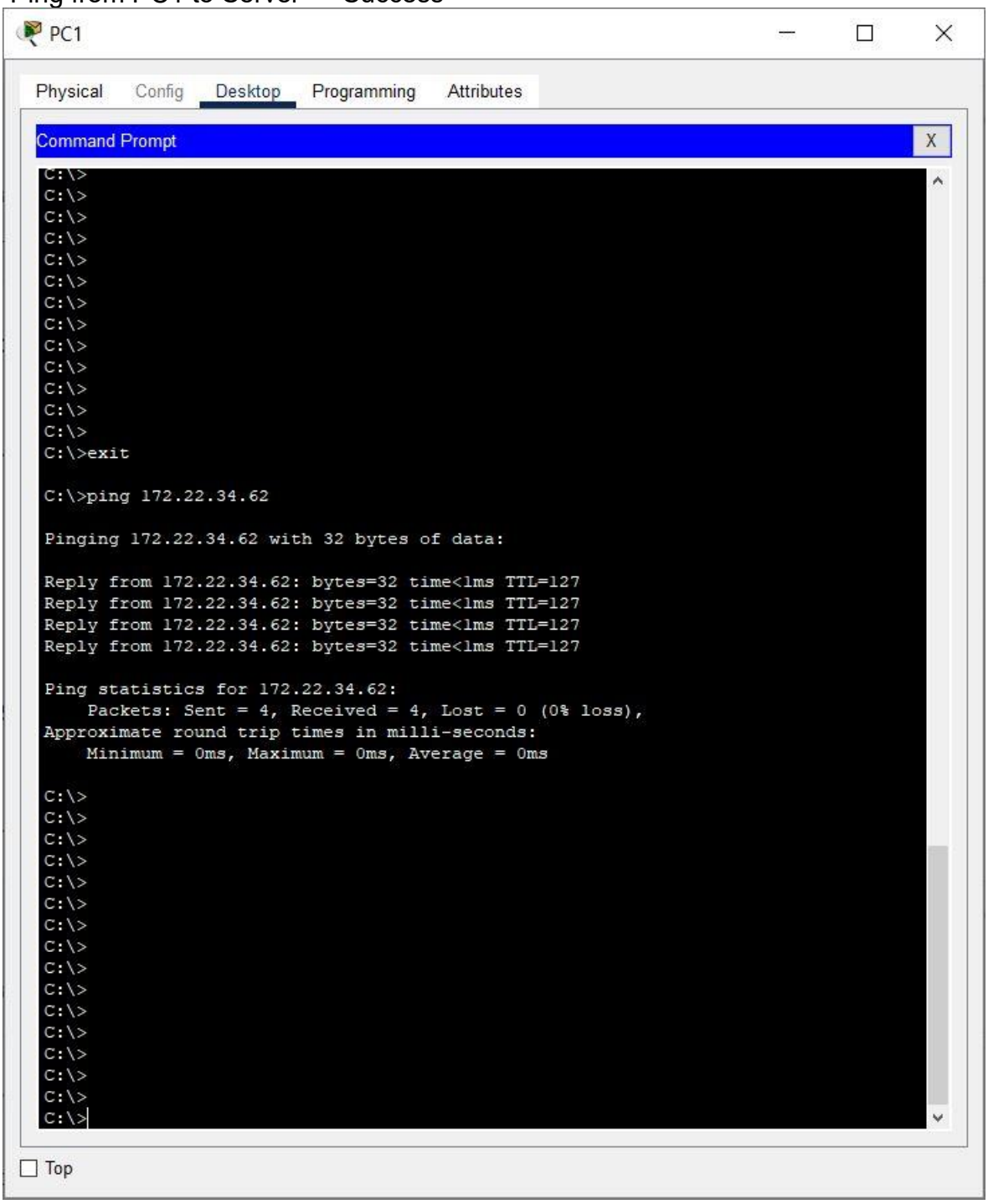
IOS Command Line Interface

```
changed state to up


R1>enable
R1#access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq
ftp
                ^
% Invalid input detected at '^' marker.

R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 100 permit tcp 172.22.34.64 0.0.0.31 host
172.22.34.62 eq ftp
R1(config)#access-list 100 permit icmp 172.22.34.64 0.0.0.31 host
172.22.34.64
R1(config)#interface g0/0
R1(config-if)#ip access-group 100 in
R1(config-if)#exit
R1(config)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console
write memory
Building configuration...
[OK]
R1#
R1#
R1#
R1#
R1#
R1#









R1 con0 is now available
```

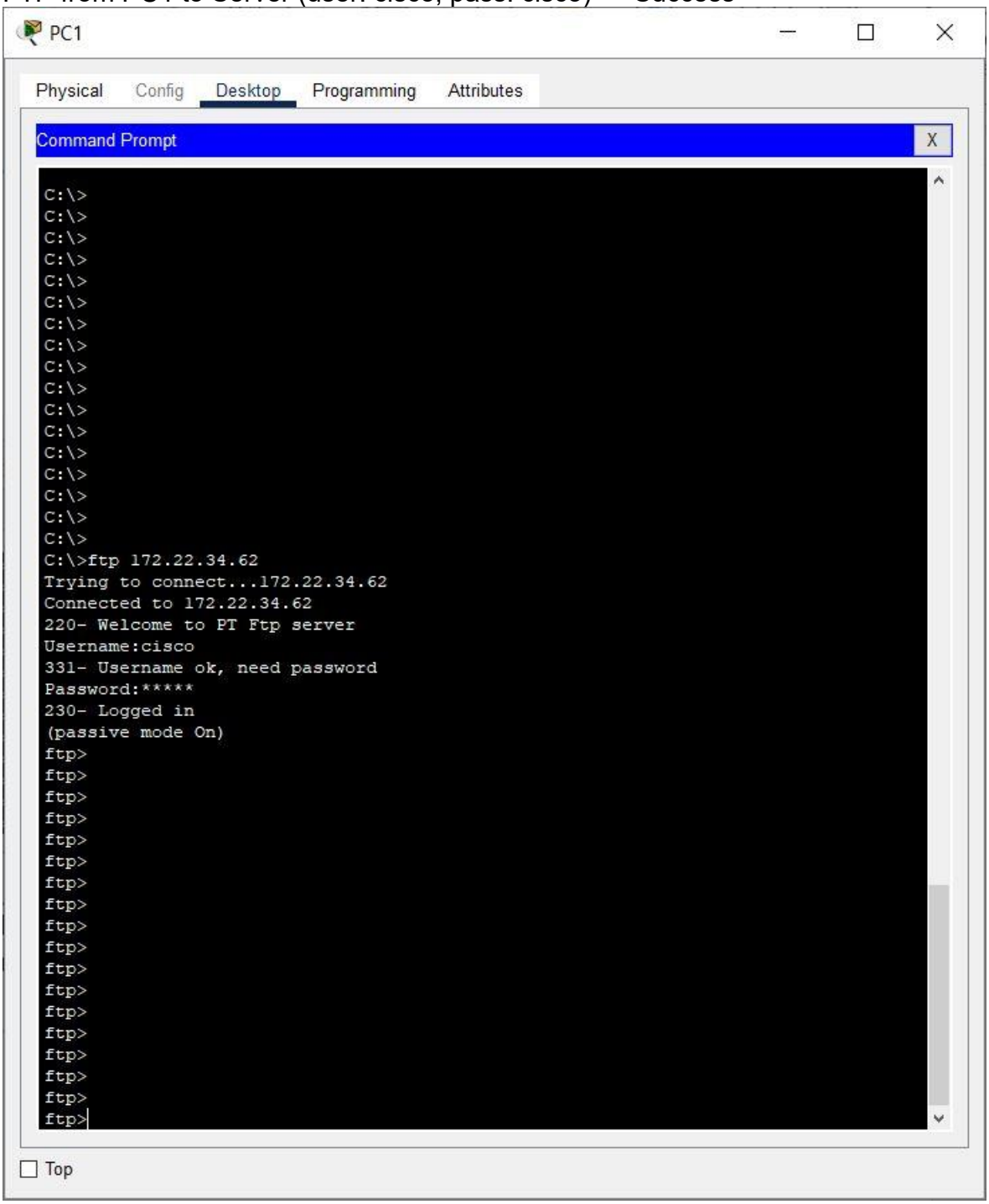
Copy Paste

Top

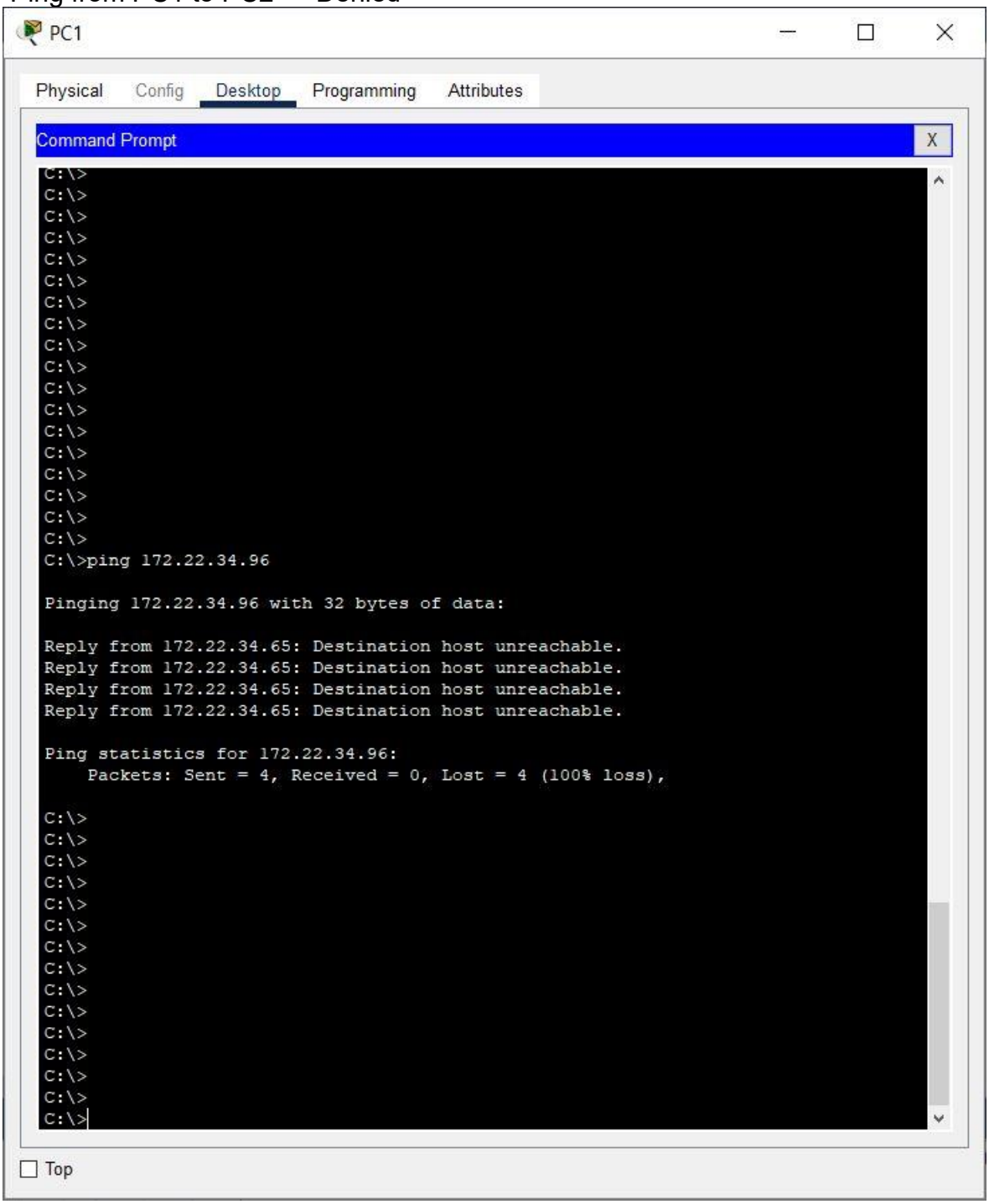
4. **Verification:**
   - Ping from PC1 to Server — Success



- FTP from PC1 to Server (user: cisco, pass: cisco) — Success



- Ping from PC1 to PC2 — Denied



Part 2: Extended Named ACL (HTTP & ICMP for PC2)

Step-by-Step:

1. **Create Named ACL 'HTTP_ONLY' and Permit HTTP**

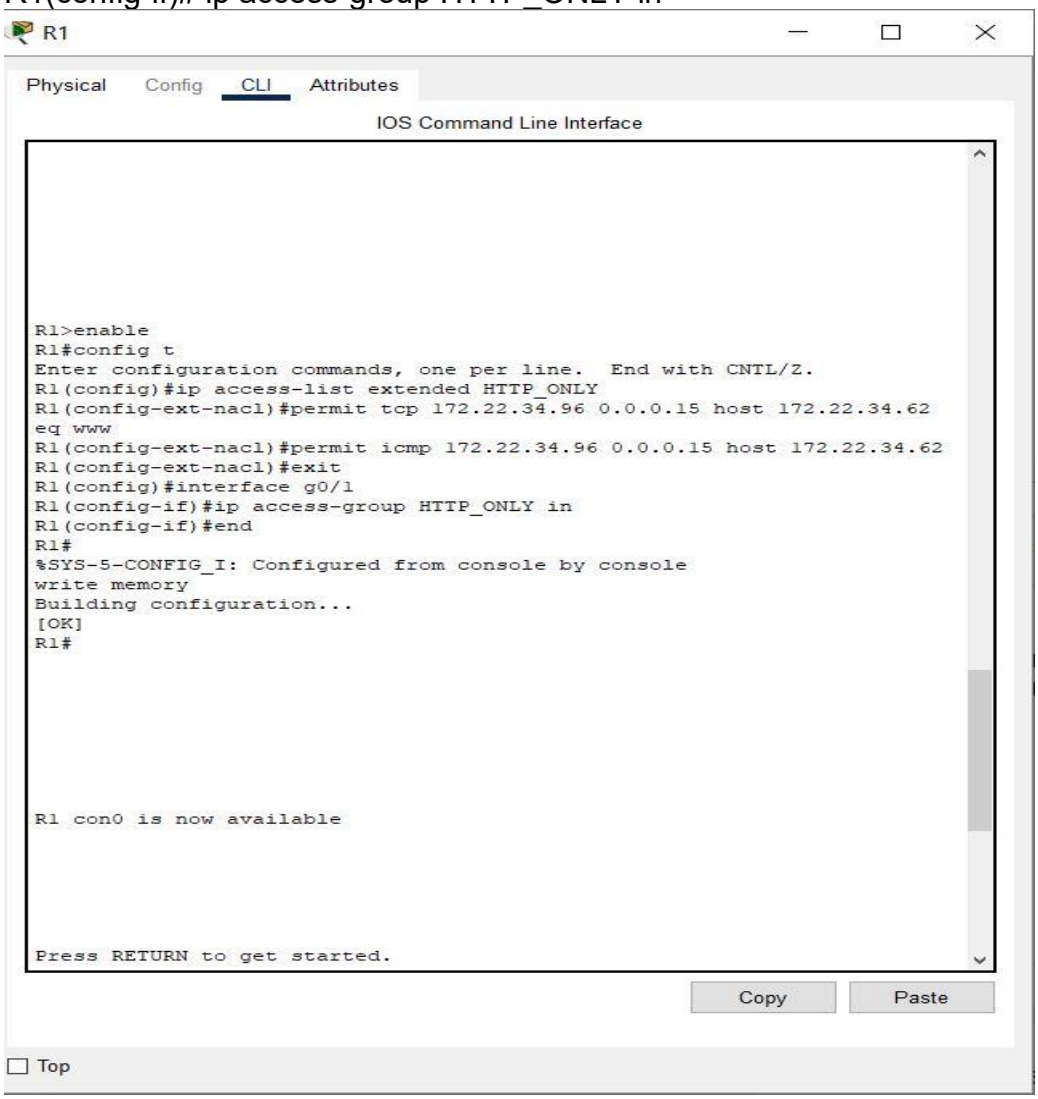
R1(config)# ip access-list extended HTTP_ONLY
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www

2. **Permit ICMP from PC2's Network**

R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62
R1(config-ext-nacl)# exit

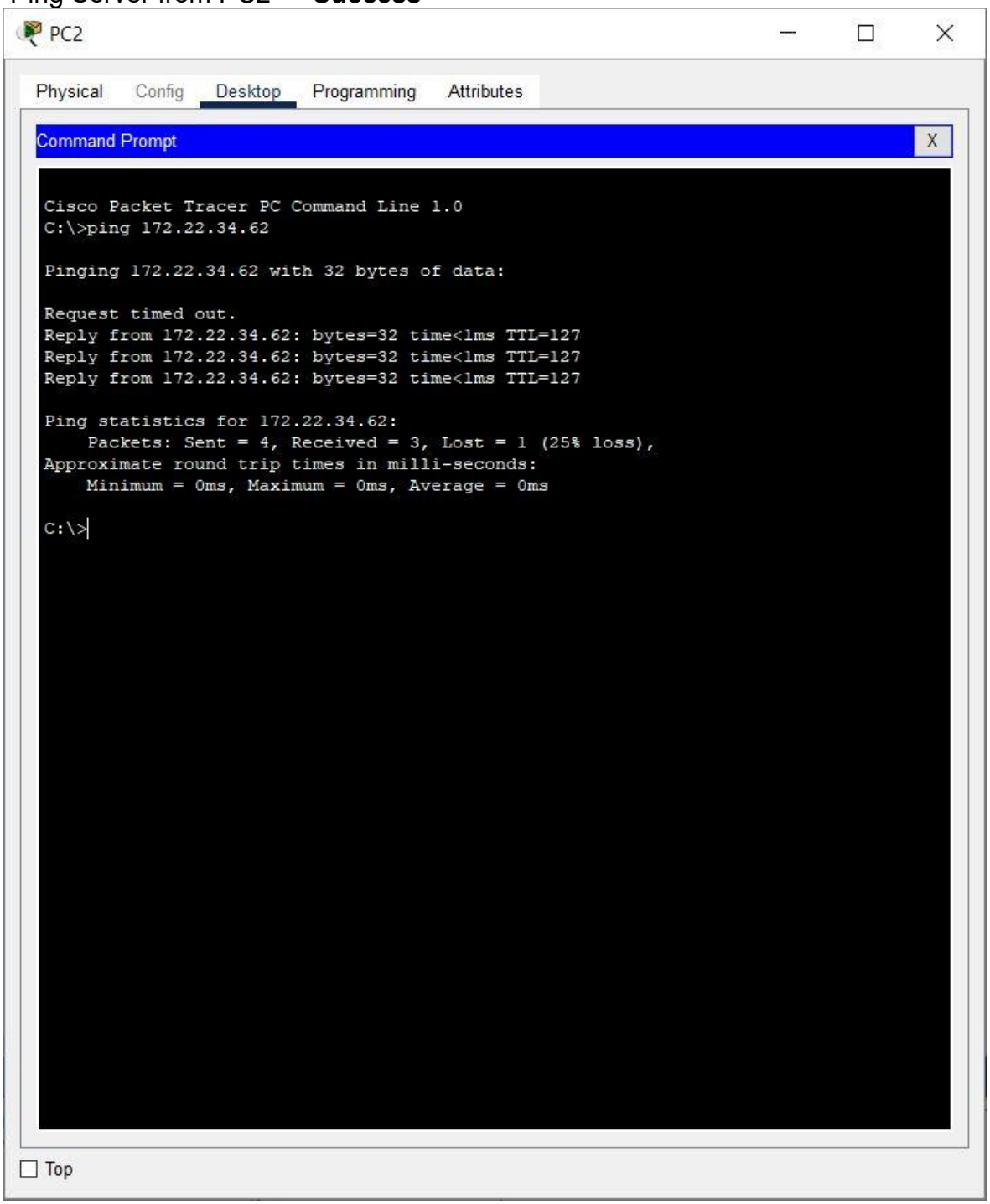
3. **Apply Named ACL on G0/1**

R1(config)# interface g0/1
R1(config-if)# ip access-group HTTP_ONLY in

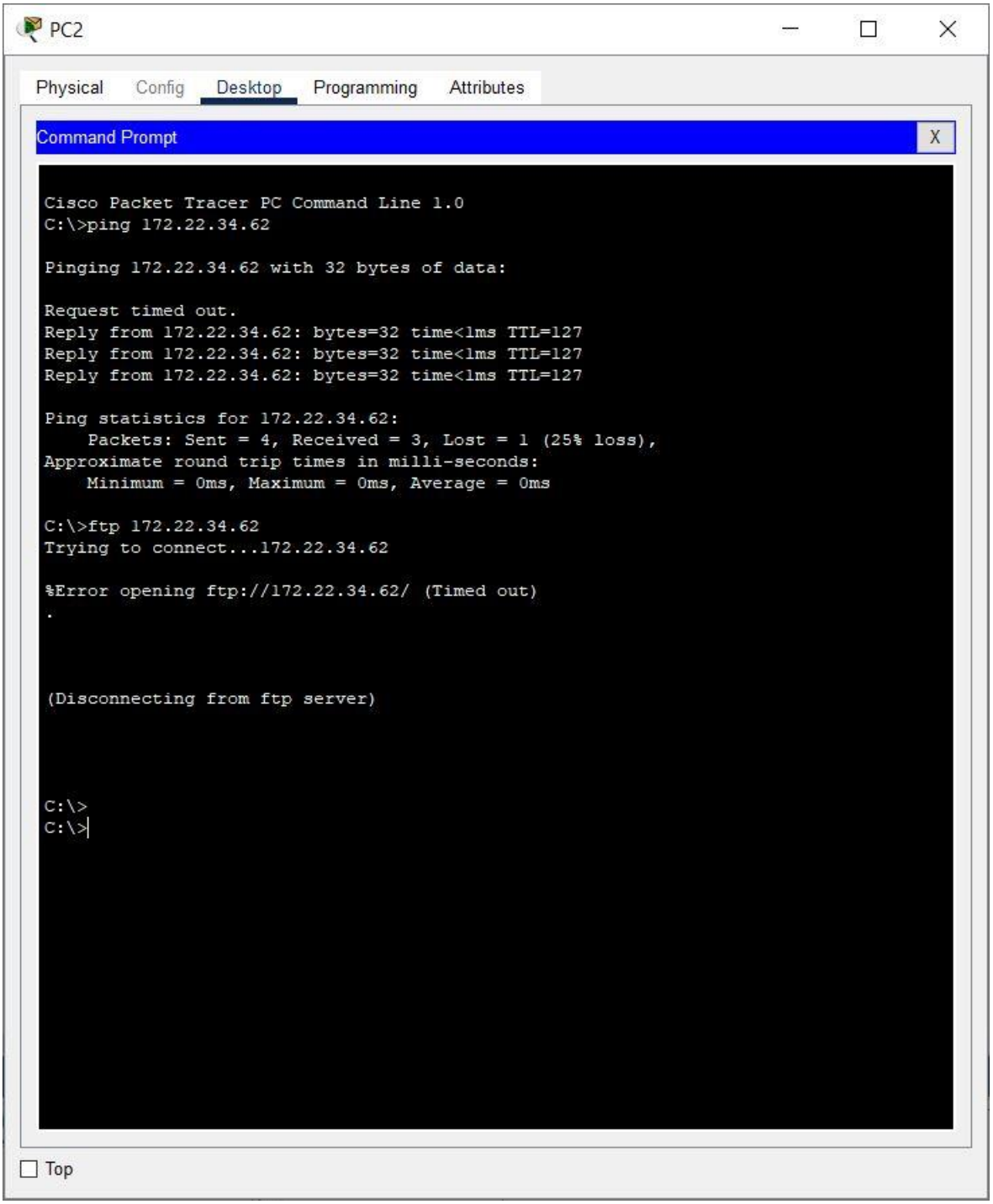


Verification:

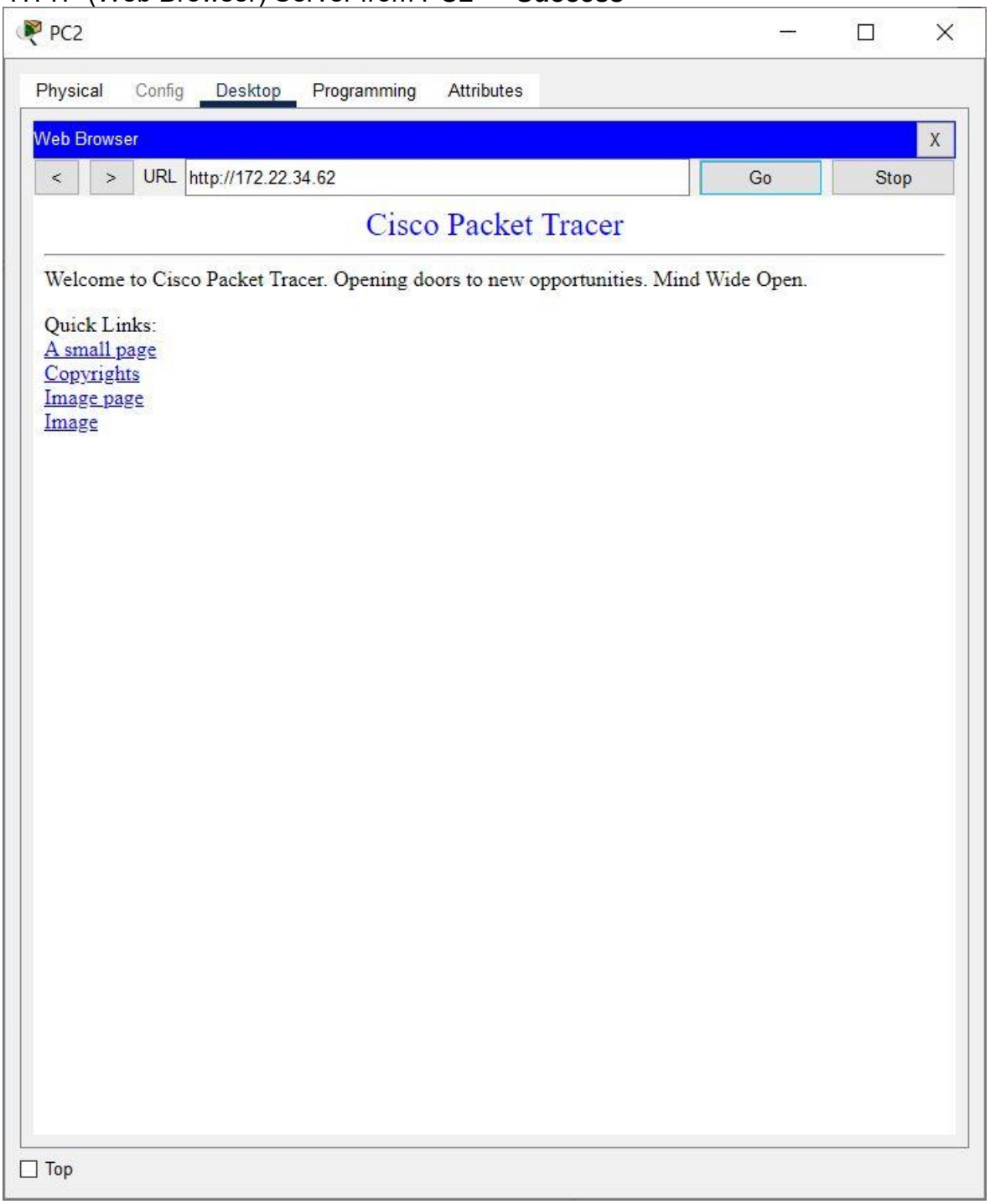
- Ping Server from PC2 — **Success**



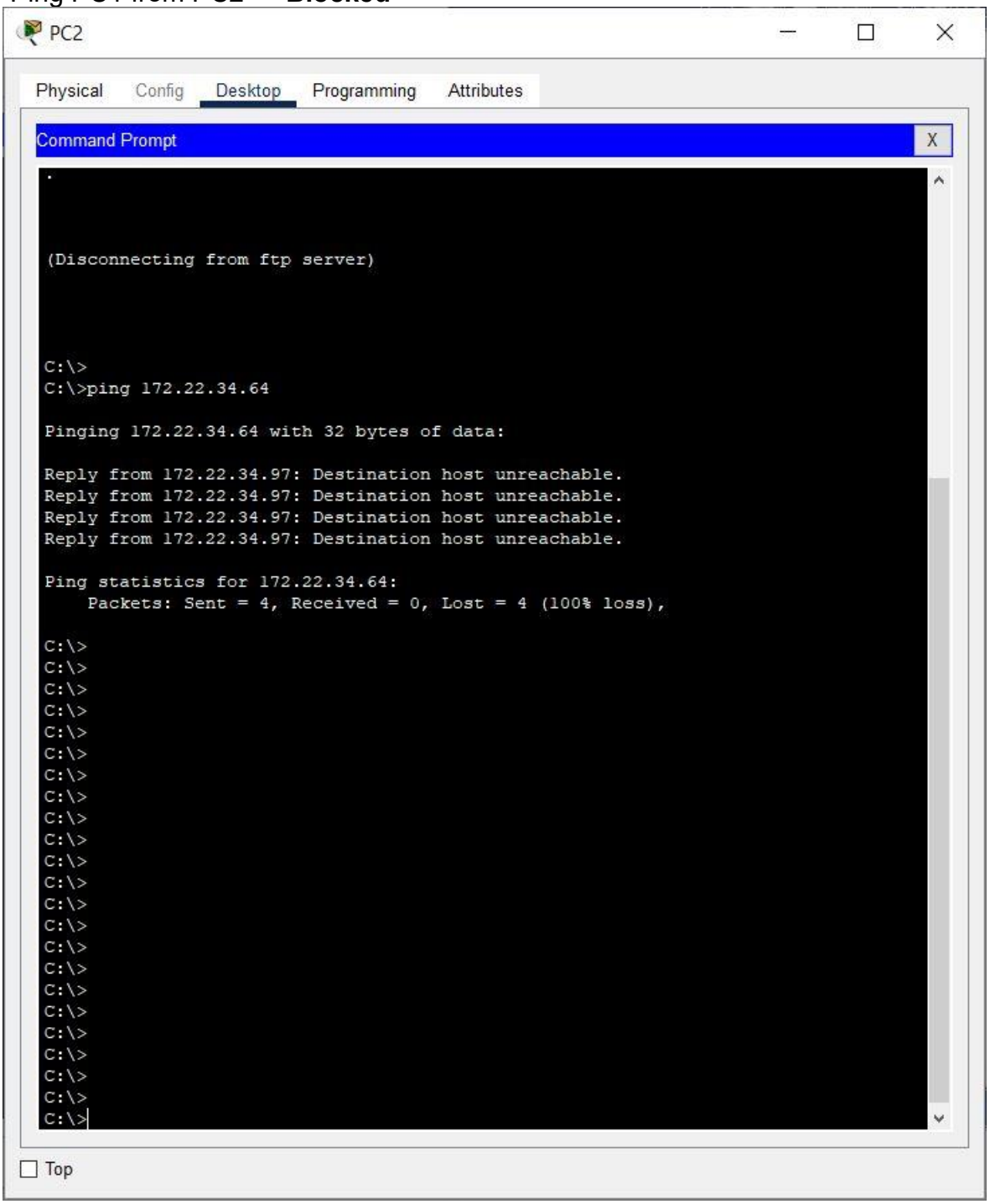
- FTP Server from PC2 — **Blocked**



- HTTP (Web Browser) Server from PC2 — **Success**



- Ping PC1 from PC2 — **Blocked**



---

5. Verification Results

Test	Source	Destination	Protocol	Result
Ping Server	PC1	172.22.34.62	ICMP	Success
FTP Server	PC1	172.22.34.62	FTP	Success
Ping PC2	PC1	172.22.34.98	ICMP	Denied
Ping Server	PC2	172.22.34.62	ICMP	Success
FTP Server	PC2	172.22.34.62	FTP	Denied
HTTP Server	PC2	172.22.34.62	HTTP	Success
Ping PC1	PC2	172.22.34.66	ICMP	Denied

---

6. Conclusion

This project showcased how to enforce **protocol-specific filtering** using extended ACLs. By applying **numbered and named ACLs** at the interface level, we were able to limit host-to-host access, protect server services, and enforce proper traffic segmentation — key practices in network security and access control.

---

7. Reflection

This project helped reinforce my understanding of:

- Extended ACL syntax and structure (numbered vs named)
- Wildcard mask calculations
- Strategic ACL placement (inbound vs outbound)
- Real-world firewall rule logic at the router level
