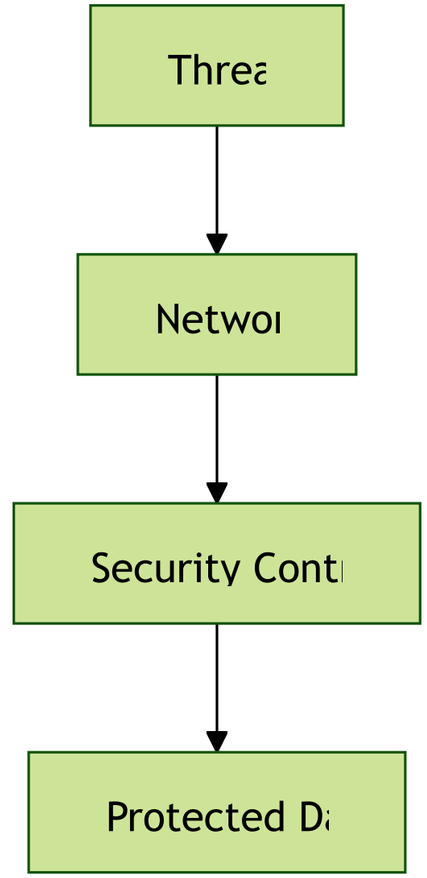
Network Security Architecture: Securing the OSI Layers and Beyond

Exploring security controls, modern frameworks, and breach prevention as of March 28, 2025

1. Introduction to Network Security Architecture

Designing systems to protect networks from threats like the 2013 Target breach.



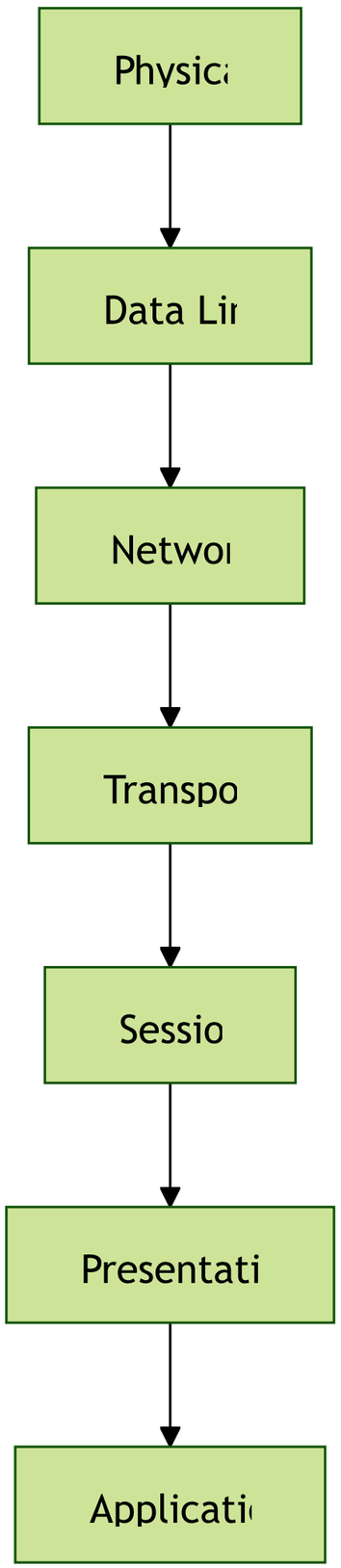
```
# Check network interfaces on Linux
ip link show
```

"Network security architecture is about safeguarding data across all layers. Today, we'll use OSI, modern tools, and frameworks to build a robust defense."

Q: What's the primary goal of network security architecture?

A: To protect the network and its data from unauthorized access, misuse, or destruction.

2. OSI Model Overview



```
# View network traffic on Linux
sudo tcpdump -i eth0
```

"The OSI model helps us categorize threats and controls. Let’s explore each layer next."

3. Physical Layer Security

Protecting hardware and connections: access control, surveillance.

```
# Simulate physical security check on AWS
aws ec2 describe-instances --query "Reservations[*].Instances[*].
```

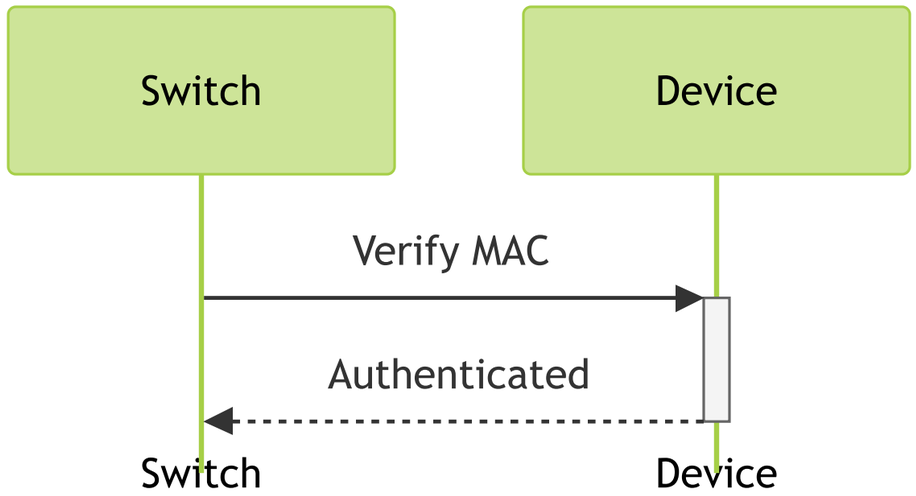
"Physical security prevents tampering. In AWS, ensure instances are in secure regions."

Q: Why is physical security critical?

A: Unsecured hardware can be directly accessed or stolen.

4. Data Link Layer Security

VLANs, port security, MAC filtering to prevent spoofing.



```
# Create VLAN in Docker
docker network create --driver bridge vlan10
```

"Data Link controls stop local network attacks. Try isolating containers with Docker."

5. Network Layer Security

Firewalls, IPsec, VPNs to secure routing and traffic.

```
# Set up iptables firewall on Linux
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -P INPUT DROP
```

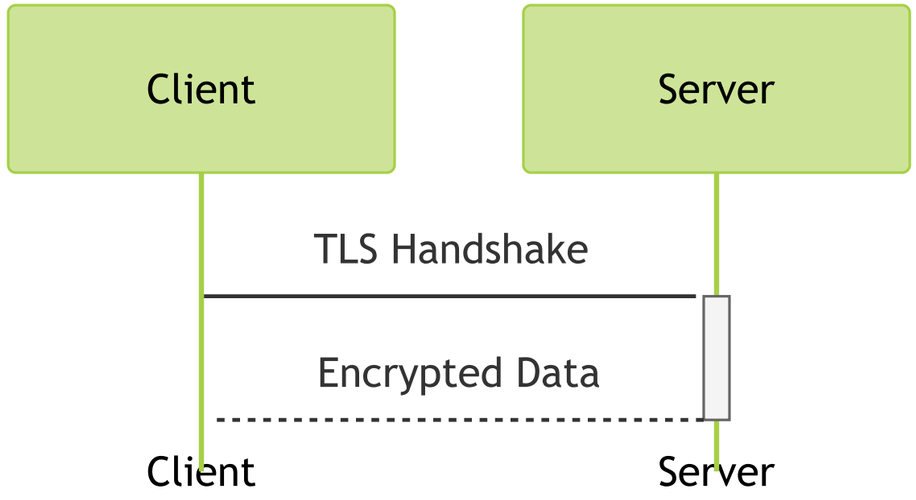
"Network layer filters traffic. This iptables rule allows SSH only."

Q: What's the difference between stateful and stateless firewalls?

A: Stateful tracks connection state; stateless filters based on rules.

6. Transport Layer Security

TLS, port security to ensure end-to-end integrity.



```
# Install SSL on Linux (Ubuntu)
sudo apt install certbot -y
sudo certbot certonly --standalone -d example.com
```

"TLS secures data in transit. Let’s set up HTTPS with Certbot."

7. Session Layer Security

MFA, session timeouts to manage connections.

```
# Enable MFA on AWS
aws iam create-virtual-mfa-device --virtual-mfa-device-name MyMFA
```

"Session security prevents hijacking. MFA adds a layer of protection."

8. Presentation Layer Security

Encryption, antivirus to protect data formats.

```
# Check SSL on MacOS
openssl s_client -connect example.com:443
```

"Presentation ensures data is safe before use. Verify SSL strength here."

9. Application Layer Security

WAF, secure coding to protect user interfaces.

```
# Deploy WAF on AWS
aws wafv2 create-web-acl --name MyWAF --scope REGIONAL --default-
```

"Application layer is the front line. WAFs block exploits."

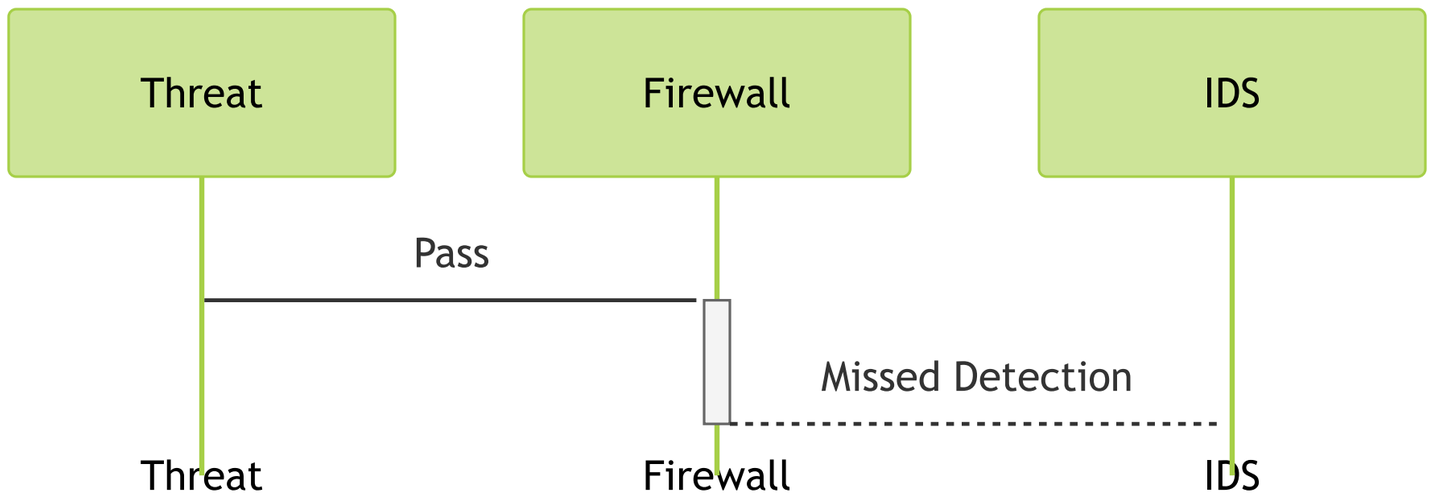
10. Traditional Defense-in-Depth

Layers static controls: firewalls, antivirus.

"Traditional methods stack defenses but can be siloed."

11. Limitations of Traditional Approaches

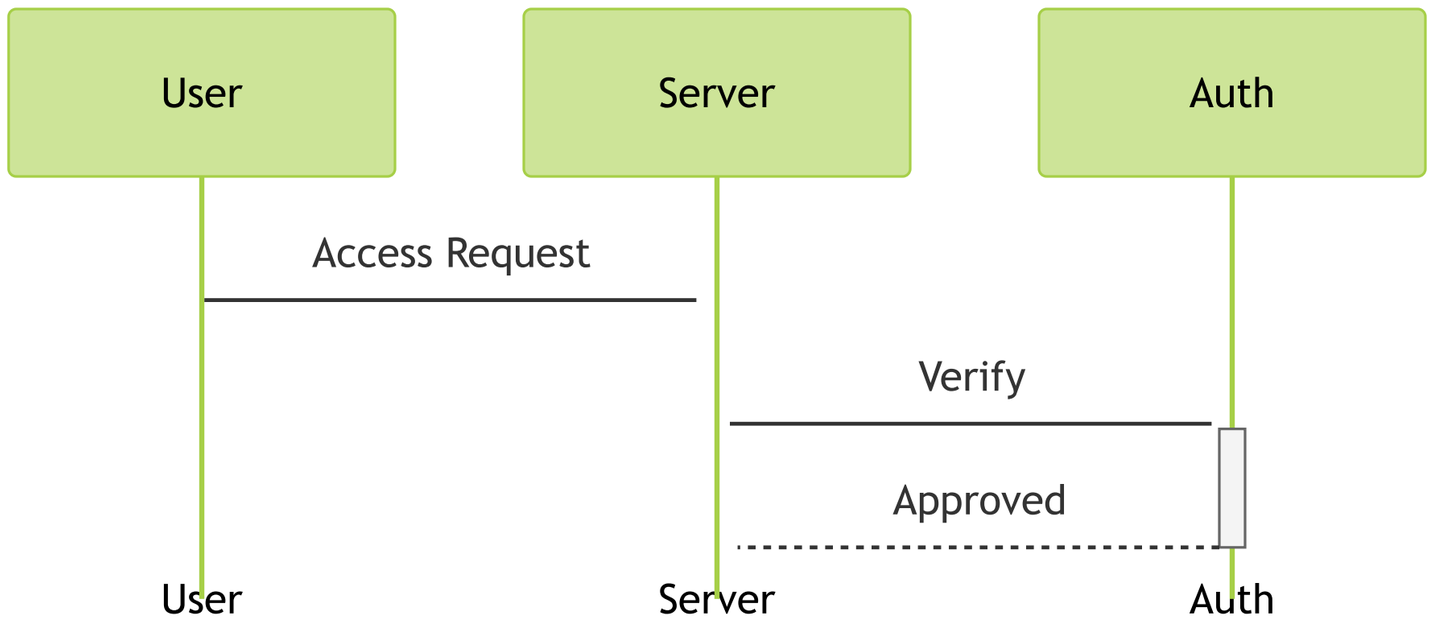
Slow to adapt, lacks integration.



"Siloed tools miss advanced threats like APTs."

12. Zero Trust Architecture

Verify every access request continuously.



```
# Simulate Zero Trust in AWS
aws ec2 authorize-security-group-ingress --group-id sg-123 --prot
```

"Zero Trust assumes breach, verifying all. Restrict SSH here."

Q: How does Zero Trust differ from perimeter security?

A: Verifies every request, not just outside traffic.

13. Extended Detection and Response (XDR)

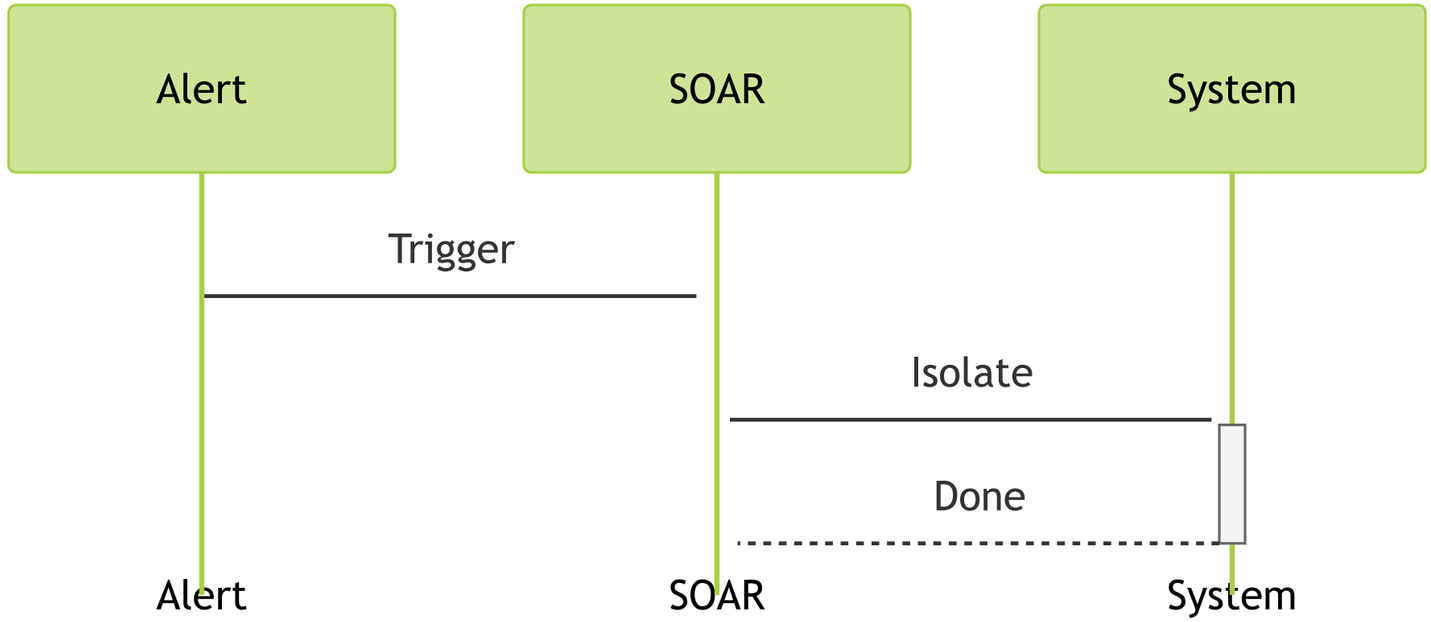
Integrates endpoint, network, cloud data.

```
# Install CrowdStrike Falcon (example)
wget -q https://falcon.crowdstrike.com/download/falcon-sensor.deb
sudo dpkg -i falcon-sensor.deb
```

"XDR gives a unified view, catching what IDS misses."

14. Security Orchestration, Automation, Response

Automates incident response.



```
# Simulate SOAR with script on Linux
echo "if [ \$(netstat -tuln | grep :22) ]; then sudo ufw deny 22;
chmod +x isolate.sh
```

"SOAR speeds up response. This script blocks SSH on alert."

15. AI/ML in Security

Anomaly detection, predictive analytics.

```
# Simple anomaly detection with Python
pip install scikit-learn
python -c "from sklearn.ensemble import IsolationForest; print('M
```

"AI spots unusual patterns, like data exfiltration."

16. MITRE ATT&CK Framework

Threat modeling with adversary tactics.

```
# Map ATT&CK with log analysis
sudo cat /var/log/auth.log | grep "failed"
```

"MITRE helps us understand attack paths, like Target's."

17. TOGAF Framework

Enterprise architecture for security integration.

"TOGAF aligns security with business goals, ensuring scalability."

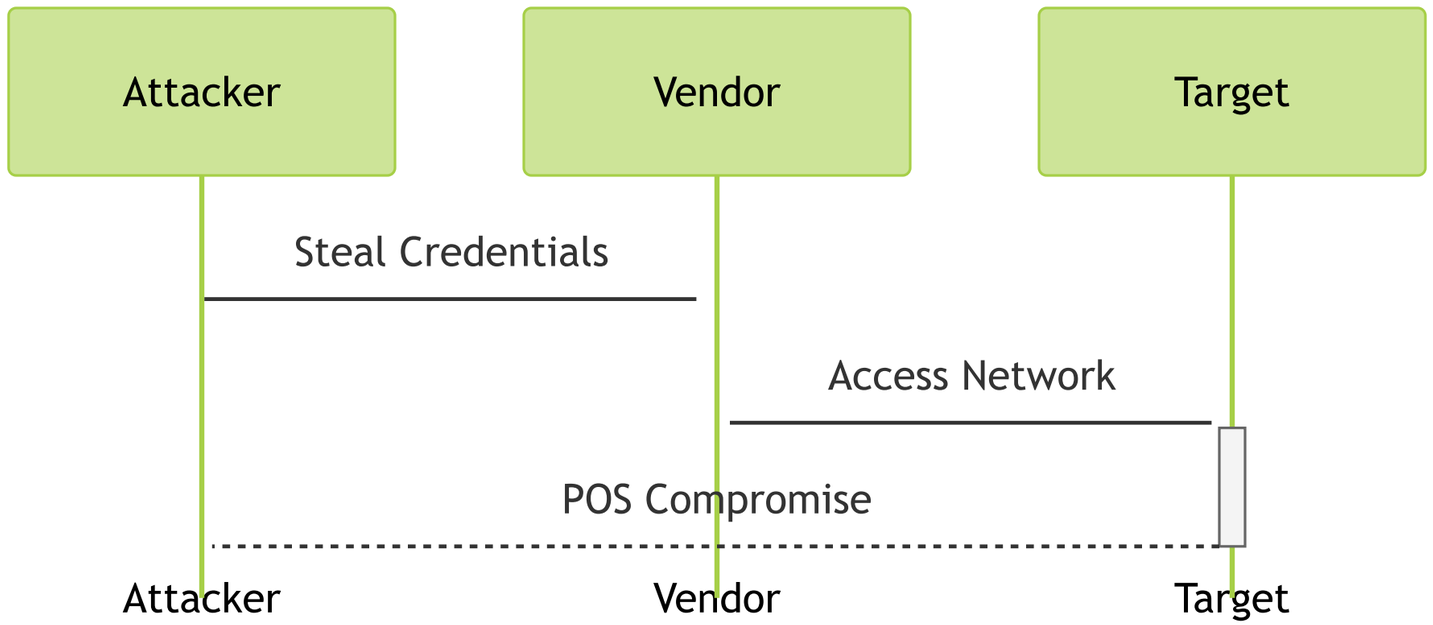
18. SABSA Framework

Risk-driven security design.

"SABSA starts with risks, tailoring controls accordingly."

19. Target Data Breach (2013)

Vendor access led to POS malware, 40M cards stolen.



"Target’s lack of segmentation was key. Modern controls could’ve stopped this."

20. Home Depot Breach (2014)

Vendor credentials, 56M cards stolen.

"Similar to Target, poor vendor security was the entry."

21. Wendy’s Breach (2016)

POS malware, 1,000+ locations affected.

"Distributed systems need centralized monitoring."

22. Sally Beauty Breach (2015)

25,000 cards compromised via POS.

"Detection failed here; XDR could’ve caught it."

23. Neiman Marcus Breach (2014)

350,000 cards stolen via POS malware.

"Slow response worsened this; SOAR helps."

24. Prevention with Modern Architecture

Segmentation, Zero Trust, XDR stop breaches.

"These controls block lateral movement and detect fast."

25. Hands-on: Firewall Setup

Secure a Linux server with ufw.

```
# Enable and configure ufw
sudo apt install ufw -y
sudo ufw allow 22
sudo ufw enable
```

"Let’s secure SSH access now."

26. Hands-on: VLAN Setup

Isolate networks with Docker.

```
# Create and test VLAN
docker network create --driver bridge vlan20
docker run --rm -d --network vlan20 nginx
```

"This isolates traffic, mimicking segmentation."

27. Hands-on: SSL/TLS Setup

Secure a site with HTTPS.

```
# Install and run Certbot
sudo apt install certbot python3-certbot-nginx -y
sudo certbot --nginx -d mydomain.com
```

"HTTPS is critical for transport security."

28. Hands-on: Security Assessment Audit with Lynis.

```
# Install and run Lynis
sudo apt install lynis -y
sudo lynis audit system
```

"Assess your server’s security now."

29. Conclusion and Q&A

Recap: OSI controls, modern tools, breach prevention.

"We've built a secure architecture. Any questions?"

Q: How can we stay ahead of new threats?

A: Use threat intelligence and continuous monitoring.

30. Introduction to TCP/IP

TCP/IP: Core protocol suite for internet communication, simpler than OSI.

```
# Check TCP/IP stack on Linux
netstat –tuln
```

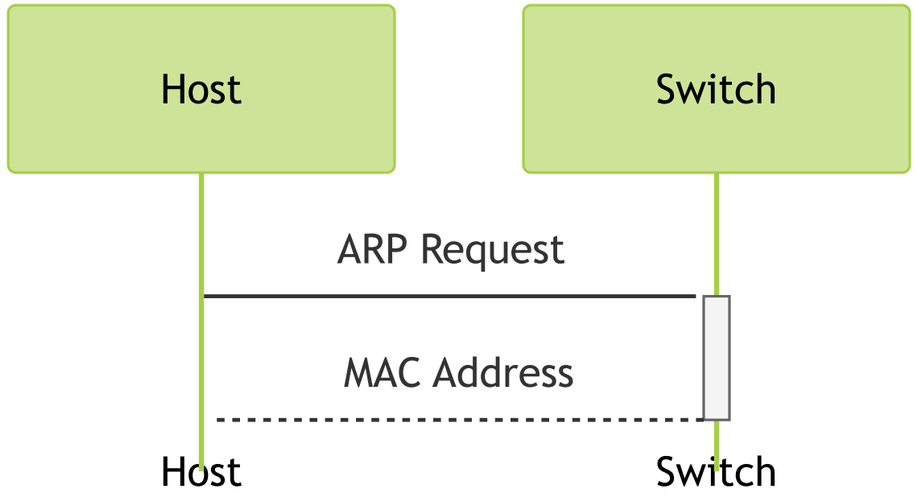
"TCP/IP drives the internet. It’s a 4-layer model we’ll dissect for security."

Q: How does TCP/IP differ from OSI?

A: TCP/IP is practical, with 4 layers vs. OSI's theoretical 7.

31. Link Layer in TCP/IP

Handles hardware addressing, Ethernet, ARP.



```
# View ARP table on Linux
arp -n
```

"Link layer maps IPs to MACs. ARP spoofing is a risk here."

Q: What's an ARP spoofing attack?

A: Faking MAC addresses to intercept traffic.

32. Internet Layer in TCP/IP

IP addressing, routing with IPv4/IPv6, ICMP.

```
# Ping an IP on Linux
ping -c 10 8.8.8.8
```

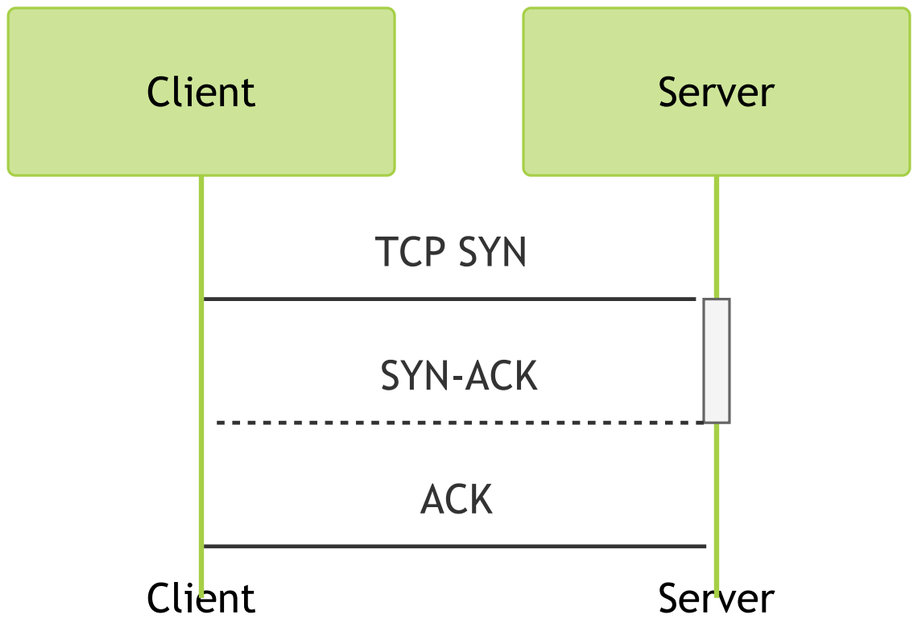
"Internet layer moves packets. ICMP can be abused for reconnaissance."

Q: Why secure ICMP?

A: Prevents ping sweeps and DoS attacks.

33. Transport Layer in TCP/IP

TCP (reliable) vs. UDP (fast), port management.



```
# Check open ports on Linux
ss -tuln
```

"TCP ensures delivery; UDP is lightweight. Both need securing."

Q: When to use TCP vs. UDP?

A: TCP for reliability (e.g., HTTP), UDP for speed (e.g., streaming).

34. Application Layer in TCP/IP

Protocols like HTTP, FTP, DNS for user services.

```
# Query DNS on MacOS/Linux
dig google.com
```

"Application layer is user-facing, prone to exploits like DNS spoofing."

Q: How can DNS be attacked?

A: Spoofing or cache poisoning to redirect traffic.

35. TCP/IP Security Challenges

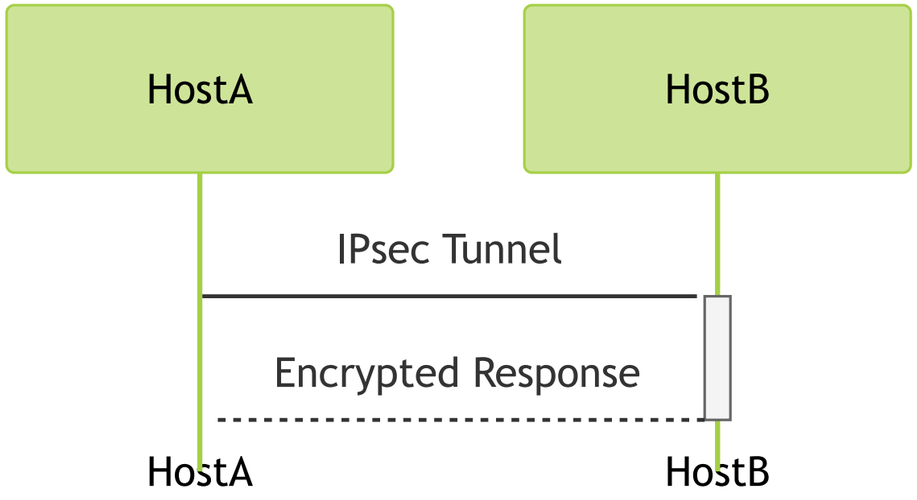
Spoofing, sniffing, session hijacking.

```
# Sniff packets on Linux
sudo tcpdump -i eth0
```

"TCP/IP wasn’t built for security. Sniffing’s easy without encryption."

36. Securing TCP/IP with IPsec

Encrypts and authenticates IP packets.

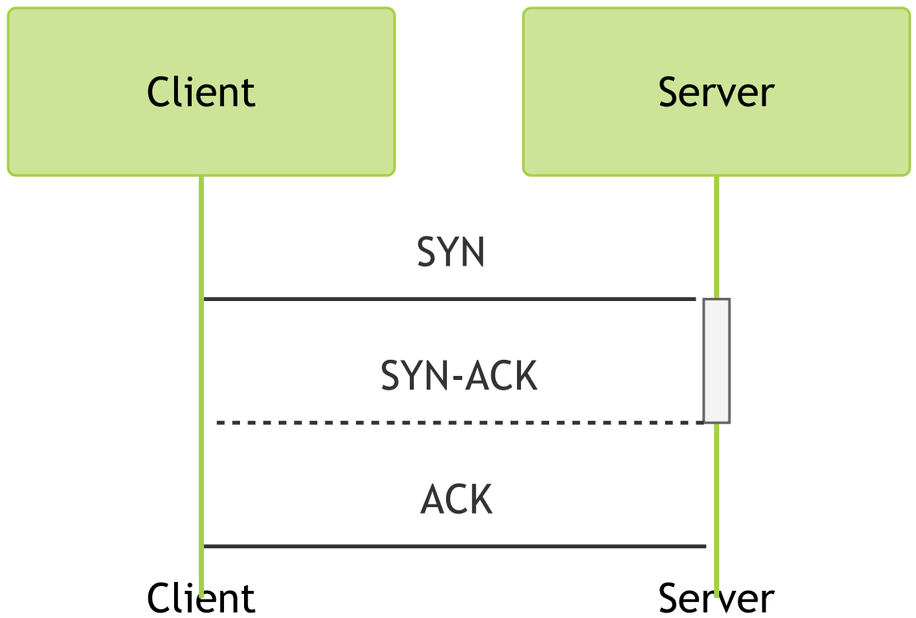


```
# Set up IPsec on Linux (strongSwan)
sudo apt install strongswan -y
sudo vi /etc/ipsec.conf
# Add: conn my-vpn
#      left=192.168.1.1
#      right=192.168.1.2
sudo systemctl restart strongswan-starter
```

"IPsec adds a security layer to IP. Try this basic setup."

37. TCP Handshake Security

Three-way handshake: SYN, SYN-ACK, ACK.



```
# Simulate handshake on Linux
nc -l 12345 & nc localhost 12345
```

"Handshakes can be spoofed for SYN floods. Firewalls mitigate this."

Q: What's a SYN flood?

A: Overwhelming a server with fake SYN requests.

38. Hands-on: TCP Traffic Analysis

Use Wireshark to analyze TCP packets.

```
# Install Wireshark on Linux
sudo apt install wireshark -y
sudo wireshark &
```

"Wireshark shows TCP in action. Capture some traffic now."

39. TCP/IP in Modern Architecture

Integrates with Zero Trust, XDR for security.

```
# Restrict TCP ports on AWS
aws ec2 revoke-security-group-ingress --group-id sg-123 --protoco
```

"Modern tools enhance TCP/IP’s weak spots. Secure ports here."

40. Conclusion and Q&A

TCP/IP secured with layered controls and modern tools.

"We've covered TCP/IP's nuts and bolts. Questions?"

Q: How does TCP/IP fit into OSI security?

A: Maps to OSI layers, needing controls at each for full protection.