



Networking Ports and Protocols

Fundamentals | Ports every SOC analyst and exam candidate must know, secure vs insecure protocols, OSI/TCP-IP layers, TCP flags and nmap scans (addressing, ICMP, DHCP and triage: see the sheet “IP Addressing, ICMP, DHCP and Network Triage”)

#Ports #Protocols #TCPIP #OSIModel

How to read the port tables

INFO

A **port** is a 16-bit number (0-65535) that identifies a service on a host; with the IP address and the transport (TCP = reliable, connection-oriented; UDP = connectionless datagrams) it forms the socket. **Well-known** ports 0-1023 need admin rights to open, **registered** 1024-49151, **dynamic** / **ephemeral** 49152-65535 (client side). The official list is kept by IANA (Internet Assigned Numbers Authority). An RFC (Request for Comments) is an IETF Internet standard document; its title is given where an RFC number appears. Exam codes: SY0-701 = CompTIA Security+, N10-009 = CompTIA Network+, CISSP Domain 4 = Communication and Network Security.

SSH / RDP / VNC — Secure Shell (encrypted remote terminal and file copy) / Remote Desktop Protocol (Windows graphical session) / Virtual Network Computing (cross-platform screen sharing).

NLA — Network Level Authentication: RDP users authenticate (CredSSP) before a desktop session is created, so unauthenticated attackers never reach the logon screen code.

WinRM — Windows Remote Management: SOAP-over-HTTP(S) remote administration used by PowerShell Remoting.

MTA / MSA — Mail Transfer Agent (server relaying mail to other servers, port 25) / Mail Submission Agent (accepts mail from authenticated users, port 587 or 465).

STARTTLS — command that upgrades a cleartext connection to TLS on the same port; see implicit vs explicit TLS below.

HSTS / QUIC — HTTP Strict Transport Security (header forcing browsers to use HTTPS) / UDP-based transport under HTTP/3 with TLS 1.3 built in.

LDAP / Global Catalog — Lightweight Directory Access Protocol (queries Active Directory) / partial replica of every domain in the AD forest, queried on 3268/3269.

KDC — Kerberos Key Distribution Center (on every domain controller): answers AS-REQ (logon, returns a TGT, ticket-granting ticket) and TGS-REQ (service ticket requests).

AAA / RADIUS / TACACS+ — Authentication, Authorization, Accounting; RADIUS (Remote Authentication Dial-In User Service, UDP, used for Wi-Fi 802.1X and VPN) and TACACS+ (Cisco, TCP, for network device admin logins).

NetBIOS / NBT-NS / LLMNR — legacy Windows naming (NetBIOS Name Service) and Link-Local Multicast Name Resolution: fallbacks that any host can answer, so an attacker (Responder tool) impersonates the target and captures hashes.

MS-RPC / SMB — Microsoft Remote Procedure Call (endpoint mapper 135 then dynamic ports) / Server Message Block (file shares, named pipes, remote admin on 445).

SNMP / IPMI / BMC — Simple Network Management Protocol (device monitoring; v3 adds auth and encryption) / Intelligent Platform Management Interface on the Baseboard Management Controller (out-of-band server management: iLO, iDRAC).

IKE / IPsec / ESP / AH — Internet Key Exchange (negotiates keys, UDP 500, 4500 with NAT traversal) for IPsec; ESP (Encapsulating Security Payload, IP protocol 50) encrypts, AH (Authentication Header, 51) only authenticates.

OT / PLC — Operational Technology (industrial control systems) / Programmable Logic Controller; OT protocols (Modbus, DNP3, S7comm, EtherNet/IP, BACnet) were designed without authentication.

well-known 0-1023 registered 1024-49151 ephemeral 49152-65535 TCP = IP protocol 6
UDP = IP protocol 17 ICMP = IP protocol 1

Port reference by family

Port	Remote access / mail	Security notes
22/tcp	SSH, SCP, SFTP	keys only, no root, admin network
23/tcp	Telnet	cleartext; common on OT and switches
3389/tcp	RDP	NLA + MFA + gateway, never exposed
5900/tcp	VNC	weak auth; tunnel over SSH/VPN
5985 / 5986/tcp	WinRM HTTP / HTTPS	PS Remoting; 5986 outside the domain
25/tcp	SMTP relay (MTA-MTA)	STARTTLS; block outbound from clients
587/tcp	SMTP submission (MSA)	STARTTLS + auth (RFC 6409, Message Submission for Mail)
465/tcp	SMTPS implicit TLS	re-standardised by RFC 8314 (Cleartext Considered Obsolete)
110 / 995/tcp	POP3 / POP3S	prefer 995 or disable POP
143 / 993/tcp	IMAP / IMAPS	prefer 993

Port	Web / directory / auth	Security notes
80 / 443/tcp	HTTP / HTTPS	HSTS; 443/udp = HTTP/3 over QUIC
8080 / 8443/tcp	Alt HTTP / HTTPS	proxies, Tomcat, forgotten consoles
389/tcp,udp	LDAP	cleartext bind; require signing
636/tcp	LDAPS	LDAP over TLS
3268 / 3269/tcp	AD Global Catalog / TLS	forest-wide search
88/tcp,udp	Kerberos KDC	AS-REQ, TGS-REQ; Kerberoasting
464/tcp,udp	Kerberos kpasswd	password change
1812 / 1813/udp	RADIUS auth / accounting	MD5-based obfuscation (BlastRADIUS 2024); RadSec = RADIUS over TLS 2083/tcp; legacy 1645/1646
49/tcp	TACACS+	device AAA, whole packet body encrypted
Port	File transfer / Windows	Security notes
20 / 21/tcp	FTP data / control	cleartext; passive = high ports
989 / 990/tcp	FTPS implicit TLS	explicit FTPS = 21 + AUTH TLS
69/udp	TFTP (Trivial FTP)	no auth; PXE network boot, device configs
445/tcp	SMB direct	block at edge; no SMBv1 (MS17-010 = EternalBlue patch, WannaCry)
137-139/udp,tcp	NetBIOS name, datagram, session	NBT-NS poisoning (Responder)
135/tcp	MS-RPC endpoint mapper	then dynamic 49152-65535/tcp
2049/tcp,udp	NFS (Network File System)	restrict exports, sec=krb5p (Kerberos + encryption)
873/tcp	rsync	modules without auth leak data
Port	Databases / caches	Security notes
1433/tcp, 1434/udp	MSSQL, SQL Browser	xp_cmdshell abuse
3306/tcp	MySQL / MariaDB	bind localhost, TLS
5432/tcp	PostgreSQL	pg_hba.conf allow-list
1521/tcp	Oracle TNS (Transparent Network Substrate) listener	valid node checking (IP allow-list)
27017/tcp	MongoDB	no auth by default, 2017 ransom wipes
6379/tcp	Redis	CONFIG SET writes cron/SSH keys
9200 / 9300/tcp	Elasticsearch REST / transport	open clusters leak datasets
11211/tcp,udp	memcached	UDP amplification: 1.3 Tbps DDoS on GitHub, Feb 2018
Port	Management / infrastructure	Security notes
161 / 162/udp	SNMP agent / trap	v1/v2c cleartext; v3 authPriv
514/udp,tcp; 6514/tcp	syslog; syslog TLS	UDP is lossy; SIEM over 6514
123/udp	NTP (Network Time Protocol)	monlist amplification; NTS = Network Time Security (RFC 8915)

53/udp,tcp	DNS	TCP for answers > 512 B (without EDNS) and AXFR zone transfers; DoT (DNS over TLS) 853, DoH (DNS over HTTPS) 443
67 / 68/udp	DHCP server / client	rogue DHCP; enable snooping
546 / 547/udp	DHCPv6 client / server	mitm6 tool hijacks IPv6 DNS; RA Guard on switches
5353/udp, 1900/udp	mDNS (multicast DNS), SSDP/UPnP (device discovery)	discovery + amplification; block at edge
623/udp	IPMI (BMC, iLO, iDRAC)	cipher 0 = authentication bypass; isolate on a management VLAN

Port	VPN / OT	Security notes
500 / 4500/udp	IKE / IPsec NAT-T	ESP = IP proto 50, AH = 51
1194/udp,tcp	OpenVPN	tls-crypt hides and authenticates the handshake
51820/udp	WireGuard	silent to unauthenticated probes
1723/tcp + GRE 47	PPTP (Point-to-Point Tunneling)	MS-CHAPv2 authentication broken: never use
1701/udp	L2TP (Layer 2 Tunneling)	no encryption itself: only inside IPsec
502/tcp	Modbus/TCP	no auth, no crypto
20000/tcp,udp	DNP3 (Distributed Network Protocol)	electric and water utilities; Secure Authentication v5 rarely deployed
102/tcp	Siemens S7comm (ISO-TSAP)	Stuxnet target
44818/tcp, 2222/udp	EtherNet/IP (Common Industrial Protocol)	Rockwell PLCs
4840/tcp	OPC UA (Open Platform Communications Unified Architecture)	built-in auth, signing and encryption (when enabled)
47808/udp	BACnet/IP	building automation
1883 / 8883/tcp	MQTT / MQTT over TLS (IoT publish-subscribe)	open brokers leak telemetry

 EXAM TIP

Security+ and Network+ test ports by number: 587 vs 25, 993 vs 143, 636 vs 389, 3389 RDP, 1812 RADIUS (UDP) vs 49 TACACS+ (TCP), 22 SFTP vs 989/990 FTPS. FTPS is FTP over TLS; SFTP is an SSH subsystem: they share only the letters.

MYTH

✗ Moving SSH from port 22 to a high port secures it

✗ DNS only needs UDP 53

✗ Blocking all ICMP makes the network safer

✗ Traffic to port 443 is HTTPS

FACT

✓ **It cuts log noise, not the risk**
Internet-wide scanners probe all 65 535 ports and recognise SSH by the 'SSH-2.0-...' banner every server sends first. Key-only login, MFA and a source allow-list are what protect it.
Source: RFC 4253 section 4.2 (protocol version exchange)

✓ **DNS needs TCP 53 as well**
Answers too big for UDP come back truncated (TC flag) and are retried over TCP, common with DNSSEC signatures; zone transfers always use TCP. Dropping 53/tcp causes intermittent failures, not a clean block.
Source: RFC 7766 (TCP support is required of DNS implementations)

✓ **Some ICMP must pass or large transfers hang**
Path MTU Discovery relies on ICMP type 3 code 4 (IPv4 'fragmentation needed') and ICMPv6 type 2 (Packet Too Big); without them TLS and VPN sessions stall after the handshake (MTU black hole). Rate-limit echo instead.
Source: RFC 4890 (recommendations for filtering ICMPv6)

✓ **A port number is only a convention**
Malware tunnels raw TCP, SSH or its own C2 protocol over 443 because egress filters allow it. Application identification on an NGFW, or a session without a TLS handshake, reveals the mismatch.
Source: MITRE ATT&CK T1571 Non-Standard Port

Four port and protocol beliefs that lead to wrong firewall rules or false comfort. NGFW = next-generation firewall; C2 = command-and-control channel of malware; MTU = Maximum Transmission Unit, the largest packet a link carries

Secure vs insecure pairs

Insecure protocol → hardened replacement

- **Telnet 23** → **SSH 22** cleartext session; SSHv2, Ed25519 keys, no root
- **FTP 21** → **SFTP 22 or FTPS 990 / 21 + AUTH TLS** TFTP 69 has no authentication at all
- **HTTP 80** → **HTTPS 443 + HSTS** TLS 1.2 minimum, 1.3 preferred
- **LDAP 389** → **LDAPS 636 or StartTLS + signing** AD: enforce LDAP signing and channel binding
- **SNMPv1 / v2c** → **SNMPv3 authPriv** authPriv = authentication (SHA-2) + privacy (AES-128 encryption)
- **POP3 110 / IMAP 143** → **995 / 993** no LOGIN/PLAIN auth without TLS
- **SMBv1** → **SMBv3 signing + encryption** SMBv1 off by default since Windows 10 1709
- **SMTP 25** → **submission 587 STARTTLS or 465** 25 remains server-to-server; enforce TLS with MTA-STS (policy over HTTPS) or DANE (TLSA records in DNSSEC)
- **RDP 3389** → **NLA + MFA behind RD Gateway or VPN** NLA authenticates before the session; RD Gateway tunnels RDP in HTTPS
- **NTP 123** → **NTP with NTS (RFC 8915)** clock skew breaks log correlation, Kerberos (5 min tolerance), certificate validity
- **DNS 53** → **DoT 853 / DoH 443 + DNSSEC** DoT/DoH: confidentiality to the resolver; DNSSEC (signed records): integrity only

Implicit TLS — TLS before any protocol byte, dedicated port (465, 990, 993, 995, 636).

Explicit TLS (STARTTLS) — Starts in clear, upgrades on the same port (25/587, 21, 143, 110, 389). Downgradable unless enforced (MTA-STS).

SMB 445, RPC 135, NetBIOS 137-139 wormable exploits, NTLM relay; never expose
RDP 3389 without gateway / MFA top ransomware entry point via brute force and stolen pass-

Databases and caches (1433, 3306, 6379, 9200, 27017) data theft and ransom wipes of unauthenticated instances

Telnet 23, SNMP v2c 161, IPMI 623 cleartext or default credentials on devices

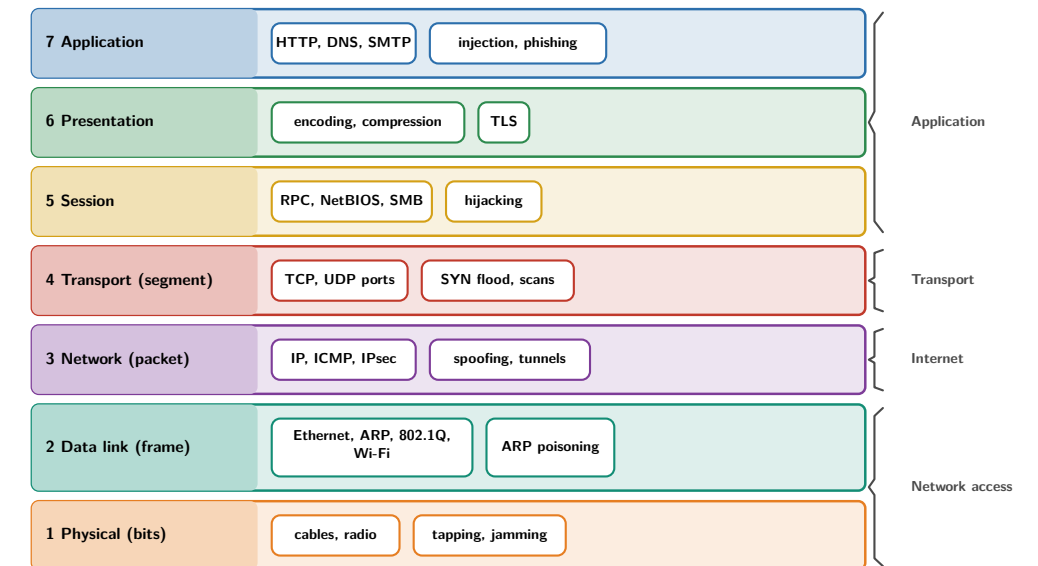
VPN gateways (IKE 500/4500, SSL VPN 443) meant to be exposed, but edge appliances

are heavily targeted: patch within days

HTTPS 443 with patched stack and WAF expected exposure; risk moves to the application

Risk if the service is reachable from the Internet (5 = critical; author estimate based on common incident reports)

Layer models and TCP



OSI mapped to TCP/IP (brackets), with the PDU and the typical attacks per layer: data at 5-7, segment / datagram at 4, packet at 3, frame at 2, bits at 1. TLS is often placed at 6; session hijacking and SMB relay live at 5; SYN floods and port scans at 4; spoofing and ICMP tunnels at 3; ARP poisoning and VLAN hopping at 2; tapping and jamming at 1

Mnemonic: **PDNTSPA**

P Physical 1, bit: 'Please'

D Data link 2, frame: 'Do'

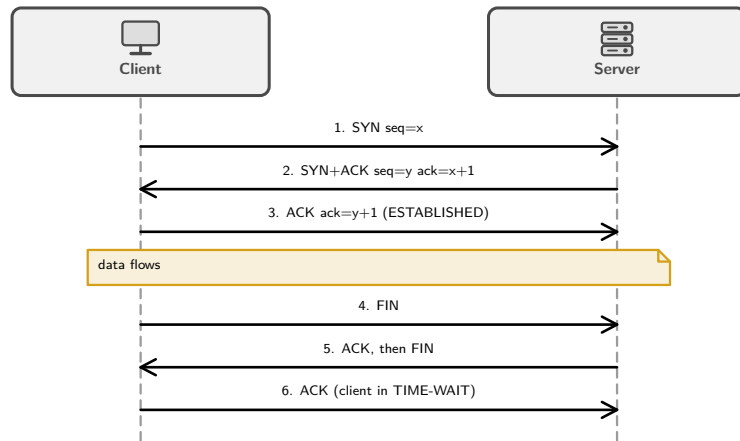
N Network 3, packet: 'Not'

T Transport 4, segment: 'Throw'

S Session 5: 'Sausage'

P Presentation 6: 'Pizza'

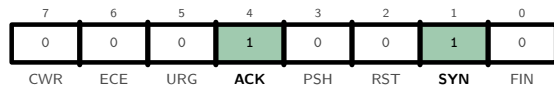
A Application 7: 'Away' (top-down: All People Seem To Need Data Processing)



TCP three-way handshake and teardown (states per RFC 9293, the 2022 TCP specification): the client goes CLOSED → SYN-SENT → ESTABLISHED, the server LISTEN → SYN-RECEIVED → ESTABLISHED. The side that closes first waits in TIME-WAIT for $2 \times \text{MSL}$ (maximum segment lifetime); the other passes through CLOSE-WAIT and LAST-ACK. A SYN flood leaves many half-open connections in SYN-RECEIVED

Flag	Meaning	Seen in
SYN	open, sync sequence numbers	first packet; SYN floods (SYN cookies)
ACK	acknowledgement valid	every packet after the first
FIN	no more data	graceful close, one per side
RST	abort	closed port reply, IDS-injected resets
PSH	deliver to app now	interactive sessions, HTTP
URG	urgent pointer valid	rare; evasion tricks
ECE / CWR	ECN (Explicit Congestion Notification) echo / congestion window reduced	ECN hosts, some scanners

0x12 = 0b00010010



- ☐ CWR (bit 7) clear: no congestion-window reduction signalled
- ☐ ECE (bit 6) clear: no ECN congestion echo
- ☒ ACK (bit 4) set: acknowledgment number (client ISN + 1) is valid
- ☐ RST (bit 2) clear: not a reset: the port is open
- ☒ SYN (bit 1) set: server proposes its own initial sequence number (ISN)

The SYN-ACK flags byte decoded: byte 13 of the TCP header (counting from 0) carries the eight flags, most significant bit first. SYN (0x02) + ACK (0x10) = 0x12. Capture filters test that byte directly: tcpdump/BPF (Berkeley Packet Filter) tcp[13] == 0x12 keeps only SYN-ACKs (open ports answering), tcp[13] & 0x04 != 0 any packet with RST; Wireshark display filter tcp.flags == 0x012. The 4 bits before it are the data offset (header length in 32-bit words) and 3 reserved bits plus AE (accurate ECN, RFC 9293).

nmap scan	Probe	Open	Closed	Filtered
-sS SYN (half-open)	SYN	SYN+ACK then RST	RST	no reply / ICMP 3 (unreachable)
-sT connect	full handshake	SYN+ACK, ACK	RST	timeout
-sA ACK	ACK	RST = unfiltered	RST = unfiltered	no reply = filtered
-sN / -sF / -sX	none / FIN / FIN+PSH+URG	no reply (open filtered)	RST	ICMP 3 code 1,2,3,9,10,13
-sU UDP	empty or protocol payload	reply or none (open filtered)	ICMP 3 code 3	other ICMP 3 codes
-sn host discovery	ICMP echo, 443 SYN, 80 ACK, timestamp	host up	-	-Pn when ICMP is blocked

```

Reading TCP flags in a capture

$ sudo tcpdump -nn -c 3 'host 198.51.100.5 and port 443'
IP 192.0.2.10.51544 > 198.51.100.5.443: Flags [S], seq 1000, length 0
IP 198.51.100.5.443 > 192.0.2.10.51544: Flags [S.], seq 7000, ack 1001, length 0
IP 192.0.2.10.51544 > 198.51.100.5.443: Flags [.], ack 7001, length 0

$ sudo tcpdump -nn -c 2 'host 198.51.100.5 and port 23'
IP 192.0.2.10.51560 > 198.51.100.5.23: Flags [S], seq 2000, length 0
IP 198.51.100.5.23 > 192.0.2.10.51560: Flags [R.], seq 0, ack 2001, length 0

$ 

```

tcpdump flag letters: S = SYN, . = ACK, S. = SYN-ACK, R = RST, F = FIN, P = PUSH. The first capture is a complete three-way handshake to HTTPS (port open); the second is SYN answered by RST-ACK (port closed, host unreachable). No answer at all means filtered. A burst of SYNs to many ports from one source with RSTs back is what a port scan looks like to a defender.

EXAM TIP

NULL/FIN/Xmas scans rely on RFC 793 (original 1981 TCP specification: a closed port answers RST, an open one stays silent): Windows and Cisco answer RST whether open or closed, so every port looks closed. A SYN scan is 'stealth' only because the application never logs a completed connection; the IDS still sees it.

Q1 Ports for Kerberos, LDAP, LDAPS, Global Catalog?

A 88, 389, 636, 3268 and 3269 (GC over TLS)

Q2 Difference between 25, 465 and 587?

A 25 server-to-server relay; 587 user submission with STARTTLS; 465 submission with implicit TLS (RFC 8314)

Q3 Transport for TACACS+ and RADIUS?

A TACACS+ TCP 49, whole body encrypted; RADIUS UDP 1812/1813, only the password obfuscated

Q4 IP protocol numbers 50, 51, 47, 6, 17, 1?

A ESP, AH, GRE (generic routing encapsulation), TCP, UDP, ICMP

Q5 What does RST in reply to an ACK scan mean?

A Port unfiltered (no stateful firewall in the path); says nothing about open or closed

TCP vs UDP from a defender's view

✓ TCP: connection-oriented	✗ UDP: connectionless
+ handshake proves the source IP can receive replies: hard to spoof	– source IP trivially spoofed: reflection and amplification DDoS (DNS, NTP, memcached, SSDP)
+ state (SYN, ESTABLISHED, FIN) lets firewalls track sessions	– no reply does not mean closed: UDP scans are slow and ambiguous
+ retransmission and ordering built in	– used by DNS, DHCP, SNMP, NTP, QUIC, WireGuard, RADIUS

📦 Encapsulation and what each device sees

📄 One HTTPS request, down the stack

- **L7**: the browser builds GET /login for bank.example; TLS encrypts it (only the SNI host-name stays visible in the handshake).
- **L4**: TCP adds source port 51514 (ephemeral) and destination port 443, sequence numbers and flags → a *segment*.
- **L3**: IP adds source 10.0.0.23 and destination 203.0.113.10, TTL 128, protocol 6 (TCP) → a *packet*; NAT later rewrites the source to the public IP.
- **L2**: Ethernet adds source MAC and the MAC of the default gateway (found with ARP) → a *frame*; the MAC changes at every router hop, the IPs do not.
- **L1**: bits on copper, fibre or radio.

Device	Highest layer it reads	Security function / limit
Switch	L2 (MAC)	VLANs, port security, DHCP snooping, DAI (Dynamic ARP Inspection)
Router	L3 (IP)	ACLs on addresses and ports, no session awareness
Stateful firewall	L4 (ports, state)	allows replies to sessions it saw start; blind to content
NGFW (next-generation firewall)	L7	identifies applications and users, TLS inspection optional
Proxy / WAF (web application firewall)	L7 (HTTP)	URL filtering, blocks injection patterns; must terminate TLS to see content
IDS / IPS	L3-L7	signatures and anomalies; IPS inline can drop