PASS QUESTION

QUESTION & ANSWER

HIGHER QUALITY, BETTER SERVICE

Provide One Year Free Update!

https://www.passquestion.com

Exam : **CNSP**

Title : Certified Network Security Practitioner (CNSP)

Version : DEMO

1.How many usable TCP/UDP ports are there?
A. 65536
B. 65535
C. 63535
D. 65335
Answer: B
Explanation:
TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) port numbers are defined by a 16-bit field in their packet headers, as specified in RFC 793 (TCP) and RFC 768 (UDP). A 16-bit integer ranges from 0 to 65,535, yielding a total of 65,536 possible ports (2^16). However, port 0 is universally reserved across both protocols and is not considered "usable" for standard network communication.
According to the Internet Assigned Numbers Authority (IANA), port 0 is designated for special purposes, such as indicating an invalid or dynamic port assignment in some systems (e.g., when a client requests an ephemeral port). In practice, operating systems and applications avoid binding to port 0 for listening services, and it's often used in error conditions or as a placeholder in protocol implementations (e.g., socket programming).
Thus, the usable port range spans from 1 to 65,535, totaling 65,535 ports. These ports are categorized by IANA into:
Well-Known Ports (0–1023): Reserved for system services (e.g., HTTP on 80/TCP). Note that 0 is still reserved within this range.
Registered Ports (1024–49151): Assigned to user applications.
Dynamic/Ephemeral Ports (49152–65535): Used temporarily by clients.
From a security perspective, understanding the usable port count is critical for firewall configuration, port scanning (e.g., with Nmap), and detecting anomalies (e.g., services binding to unexpected ports). Misconfiguring a system to use port 0 could lead to protocol errors or expose vulnerabilities, though it's rare. The CNSP curriculum likely emphasizes this distinction to ensure practitioners can accurately scope network security assessments.
Why other options are incorrect:
A. 65536: This reflects the total number of possible ports (0–65535), but it includes the reserved port 0, which isn't usable for typical TCP/UDP communication. In security contexts, including port 0 in a count could lead to misconfigured rules or scanning errors.
C. 63535: This is an arbitrary number with no basis in the 16-bit port structure. It might stem from a typo or misunderstanding (e.g., subtracting 2000 from 65535 incorrectly), but it's invalid.
D. 65335: Similarly, this lacks grounding in protocol standards. It could be a miscalculation (e.g., subtracting 200 from 65535), but it doesn't align with TCP/UDP specifications.
Real-World Context: In penetration testing, tools like Nmap scan ports 1–65535 by default, excluding 0 unless explicitly specified (e.g., -p0-65535), reinforcing that 65,535 is the practical usable count.
Reference: CNSP Official Study Guide (Network Protocols and Ports); RFC 793 (TCP), RFC 768 (UDP), IANA Service Name and Transport Protocol Port Number Registry.

2.Which command will perform a DNS zone transfer of the domain "victim.com" from the nameserver at 10.0.0.1?
A. dig @10.0.0.1 victim.com axrfr
B. dig @10.0.0.1 victim.com afxr

C. dig @10.0.0.1 victim.com arfxr
D. dig @10.0.0.1 victim.com axfr
Answer: D
Explanation:
A DNS zone transfer replicates an entire DNS zone (a collection of DNS records for a domain) from a primary nameserver to a secondary one, typically for redundancy or load balancing. The AXFR (Authoritative Full Zone Transfer) query type, defined in RFC 1035, facilitates this process. The dig (Domain Information Groper) tool, a staple in Linux/Unix environments, is used to query DNS servers.
The correct syntax is:
dig @<nameserver> <domain> axfr
Here, dig @10.0.0.1 victim.com axfr instructs dig to request a zone transfer for "victim.com" from the nameserver at 10.0.0.1. The @ symbol specifies the target server, overriding the system's default resolver.
Technical Details:
The AXFR query is sent over TCP (port 53), not UDP, due to the potentially large size of zone data, which exceeds UDP's typical 512-byte limit (pre-EDNS0).
Successful execution requires the nameserver to permit zone transfers from the querying IP, often restricted to trusted secondaries via Access Control Lists (ACLs) for security. If restricted, the server responds with a "REFUSED" error.
Security Implications: Zone transfers expose all DNS records (e.g., A, MX, NS), making them a reconnaissance goldmine for attackers if misconfigured. CNSP likely emphasizes securing DNS servers against unauthorized AXFR requests, using tools like dig to test vulnerabilities.
Why other options are incorrect:
A. dig @10.0.0.1 victim.com axrfr: "axrfr" is a typographical error. The correct query type is "axfr." Executing this would result in a syntax error or an unrecognized query type response from dig.
B. dig @10.0.0.1 victim.com afxr: "afxr" is another typo, not a valid DNS query type per RFC 1035. dig would fail to interpret this, likely outputting an error like "unknown query type."
C. dig @10.0.0.1 victim.com arfxr: "arfxr" is also invalid, a jumbled version of "axfr." It holds no meaning in DNS protocol standards and would fail similarly.
Real-World Context: Penetration testers use dig ... axfr to identify misconfigured DNS servers. For example, dig @ns1.example.com example.com axfr might reveal subdomains or internal IPs if not locked down.
Reference: CNSP Official Documentation (DNS Security and Tools); RFC 1035 (Domain Names - Implementation and Specification).

3.What is the response from a closed TCP port which is behind a firewall?
A. A FIN and an ACK packet
B. RST and an ACK packet
C. A SYN and an ACK packet
D. No response
Answer: D
Explanation:
TCP (Transmission Control Protocol) uses a three-way handshake (SYN, SYN-ACK, ACK) to establish connections, as per RFC 793. When a client sends a SYN packet to a port: Open Port: The server

responds with SYN-ACK.
Closed Port (no firewall): The server sends an RST (Reset) packet, often with ACK, to terminate the attempt immediately.
However, when a firewall is present, its configuration dictates the response. Modern firewalls typically operate in stealth mode, using a "drop" rule for closed ports rather than a "reject" rule: Drop: Silently discards the packet without replying, resulting in no response. The client experiences a timeout (e.g., 30 seconds), as no feedback is provided.
Reject: Sends an RST or ICMP "Port Unreachable," but this is less common for security reasons, as it confirms the firewall's presence.
For a closed TCP port behind a firewall, "no response" (drop) is the standard behavior in secure configurations, minimizing information leakage to attackers. This aligns with CNSP's focus on firewall best practices to obscure network topology during port scanning (e.g., with Nmap).
Why other options are incorrect:
A. A FIN and an ACK packet: FIN-ACK is used to close an established TCP connection gracefully (e.g., after data transfer), not to respond to an initial SYN on a closed port.
B. RST and an ACK packet: RST-ACK is the host's response to a closed port without a firewall. A firewall's drop rule overrides this by silently discarding the packet.
C. A SYN and an ACK packet: SYN-ACK indicates an open port accepting a connection, the opposite of a closed port scenario.
Real-World Context: Tools like Nmap interpret "no response" as "filtered" (firewall likely present) vs. "closed" (RST received), aiding in firewall detection.
Reference: CNSP Official Study Guide (Firewall Operations and TCP/IP); RFC 793 (TCP).

4.Which of the following statements regarding Authorization and Authentication is true?
A. Authorization is the process where requests to access a particular resource are granted or denied. Authentication is providing and validating the identity.
B. Authentication is the process where requests to access a particular resource are granted or denied. Authorization is providing and validating identity.
C. Authentication includes the execution rules that determine what functionality and data the user can access. Authentication and Authorization are both the same thing.
D. Authentication controls which processes a person can use and which files they can access, read, or modify. Authentication and authorization typically do not operate together, thus making it impossible to determine who is accessing the information.
Answer: A
Explanation:
Authentication and Authorization (often abbreviated as AuthN and AuthZ) are foundational pillars of access control in network security:
Authentication (AuthN): Verifies "who you are" by validating credentials against a trusted source. Examples include passwords, MFA (multi-factor authentication), certificates, or biometrics. It ensures the entity (user, device) is legitimate, typically via protocols like Kerberos or LDAP.
Authorization (AuthZ): Determines "what you can do" after authentication, enforcing policies on resource access (e.g., read/write permissions, API calls). It relies on mechanisms like Access Control Lists (ACLs), Role-Based Access Control (RBAC), or Attribute-Based Access Control (ABAC).
Option A correctly separates these roles:

Authorization governs access decisions (e.g., "Can user X read file Y?").
Authentication establishes identity (e.g., "Is this user X?").
In practice, these processes are sequential: AuthN precedes AuthZ.
For example, logging into a VPN authenticates your identity (e.g., via username/password), then authorizes your access to specific subnets based on your role. CNSP likely stresses this distinction for designing secure systems, as conflating them risks privilege escalation or identity spoofing vulnerabilities.
Why other options are incorrect:
B: Reverses the definitions—Authentication doesn't grant/deny access (that's AuthZ), and Authorization doesn't validate identity (that's AuthN). This mix-up could lead to flawed security models.
C: Falsely equates AuthN and AuthZ and attributes access rules to AuthN. They're distinct processes; treating them as identical undermines granular control (e.g., NIST SP 800-53 separates IA-2 for AuthN and AC-3 for AuthZ).
D: Misassigns access control to AuthN and claims they don't interoperate, which is false—they work together in every modern system (e.g., SSO with RBAC). This would render auditing impossible, contradicting security best practices.
Real-World Context: A web server (e.g., Apache) authenticates via HTTP Basic Auth, then authorizes via .htaccess rules—two separate steps.
Reference: CNSP Official Study Guide (Access Control Fundamentals); NIST SP 800-53 (Security and Privacy Controls).

5.What ports does an MSSQL server typically use?
A. 1433/TCP, 2433/UDP, and 3433/TCP
B. 1433/TCP, 1434/UDP, and 1434/TCP
C. 1433/TCP, 2433/UDP, and 1434/TCP
D. 1533/TCP, 1434/UDP, and 2434/TCP
Answer: B
Explanation:
Microsoft SQL Server (MSSQL) relies on specific ports for its core services, as defined by Microsoft and registered with IANA:
1433/TCP: The default port for the SQL Server Database Engine. Clients connect here for querying databases (e.g., via ODBC or JDBC). It's a well-known port, making it a frequent target for attacks if exposed.
1434/UDP: Used by the SQL Server Browser Service, which listens for incoming requests and redirects clients to the correct port/instance (especially for named instances). It's critical for discovering dynamic ports when 1433 isn't used.
1434/TCP: Less commonly highlighted but used in some configurations, such as dedicated admin connections (DAC) or when the Browser Service responds over TCP for specific instances. While 1433/TCP is the primary engine port, 1434/TCP can be involved in multi-instance setups. Technical Details:
Ports can be customized (e.g., via SQL Server Configuration Manager), but these are defaults. Named instances often use dynamic ports (allocated from the ephemeral range), with the Browser Service (1434/UDP) guiding clients to them.
Firewalls must allow these ports for MSSQL to function externally, posing risks if not secured (e.g., brute-

force attacks on 1433/TCP).
Security Implications: CNSP likely covers MSSQL port security, as vulnerabilities like SQL Slammer (2003) exploited 1434/UDP misconfigurations. Hardening includes restricting access, changing defaults, and monitoring traffic.
Why other options are incorrect:
A. 1433/TCP, 2433/UDP, 3433/TCP: 2433/UDP and 3433/TCP are not MSSQL standards; they’re likely typos or unrelated ports.
C. 1433/TCP, 2433/UDP, 1434/TCP: 2433/UDP is incorrect; 1434/UDP is the Browser Service port. D. 1533/TCP, 1434/UDP, 2434/TCP: 1533/TCP and 2434/TCP aren’t associated with MSSQL; they deviate from documented defaults.
Real-World Context: Tools like netstat -an | find "1433" on Windows confirm MSSQL’s port usage during audits.
Reference: CNSP Official Documentation (Database Security and Ports); Microsoft SQL Server Documentation, IANA Port Registry.