Reliable The SecOps Group CNSP Guide Files, CNSP Reliable Exam Topics

What's more, part of that Real4test CNSP dumps now are free: https://drive.google.com/open?id=1b68KC7AUmFjfp8YLemgdLpvTaL2pchiG

It is proved that if you study with our CNSP exam questions for 20 to 30 hours, then you will be able to pass the CNSP exam with confidence. Because users only need to spend little hours on the CNSP quiz guide, our learning materials will help users to learn all the difficulties of the test site, to help users pass the qualifying examination and obtain the qualification certificate. If you think that time is important to you, try our CNSP Learning Materials and it will save you a lot of time.

The authoritative, efficient, and thoughtful service of CNSP practice paper will give you the best user experience, and you can also get what you want with our CNSP study materials. I hope our CNSP study materials can accompany you to pursue your dreams. If you can choose CNSP free training materials, we will be very happy. We look forward to meeting you. With the help of our CNSP learning guide, you will get more opportunities than others, and your dreams may really come true in the near future.

>> Reliable The SecOps Group CNSP Guide Files <<

CNSP Reliable Exam Topics & CNSP Well Prep

Many companies think highly of The SecOps Group certifications, and they will spend money on employees' exam fee and preparation materials. They request executive staff to purchase valid CNSP exam questions vce for engineers so that they clear exams and get certifications easily without too much time and energy. Many companies regard us as their good long-term cooperative partner and think highly of our CNSP Exam Questions Vce.

The SecOps Group Certified Network Security Practitioner Sample Questions (Q22-Q27):

NEW QUESTION # 22
What kind of files are "Dotfiles" in a Linux-based architecture?

- A. Driver files

- B. Library files
- C. System files
- D. Hidden files

Answer: D

Explanation:
In Linux, file visibility is determined by naming conventions, impacting how files are listed or accessed in the file system.
Why D is correct: "Dotfiles" are files or directories with names starting with a dot (e.g., .bashrc), making them hidden by default in directory listings (e.g., ls requires -a to show them). They are commonly used for user configuration, as per CNSP's Linux security overview.
Why other options are incorrect:
A: Library files (e.g., in /lib) aren't inherently hidden.
B: Driver files (e.g., kernel modules in /lib/modules) aren't dotfiles by convention.
C: System files may or may not be hidden; "dotfiles" specifically denotes hidden status.

NEW QUESTION # 23
Which of the following services do not encrypt its traffic by default?

- A. FTPS
- B. SSH
- C. DNS
- D. All of these

Answer: C

Explanation:
Encryption ensures confidentiality and integrity of network traffic. Analyzing defaults:
A . DNS (Domain Name System):
Default: Unencrypted (UDP/TCP 53), per RFC 1035. Queries/responses (e.g., "google.com → 142.250.190.14") are plaintext.
Modern Options: DNS over HTTPS (DoH, TCP 443) or DNS over TLS (DoT, TCP 853) encrypt, but aren't default in most systems (e.g., pre-2020 Windows).
B . SSH (Secure Shell):
Default: Encrypted (TCP 22), per RFC 4251. Uses asymmetric (e.g., RSA) and symmetric (e.g., AES) crypto for all sessions.
C . FTPS (FTP Secure):
Default: Encrypted (TCP 21 control, dynamic data ports). Extends FTP with SSL/TLS (e.g., RFC 4217), securing file transfers.
Technical Details:
DNS: Plaintext exposes queries to eavesdropping (e.g., ISP snooping) or spoofing (e.g., cache poisoning).
SSH/FTPS: Encryption is baked into their standards; disabling it requires explicit misconfiguration.
Security Implications: Unencrypted DNS risks privacy and integrity (e.g., Kaminsky attack). CNSP likely pushes DoH/DoT adoption.
Why other options are incorrect:
B, C: Encrypt by default.
D: False, as only DNS lacks default encryption.
Real-World Context: The 2013 Snowden leaks exposed DNS monitoring; DoH uptake (e.g., Cloudflare 1.1.1.1) counters this.

NEW QUESTION # 24
Which of the following is an example of a SUID program?

- A. /usr/bin/curl
- B. /usr/bin/passwd
- C. None of the above
- D. /bin/ls

Answer: B

Explanation:
In Linux/Unix, the SUID (Set User ID) bit allows a program to execute with the owner's permissions, typically root, rather than the caller's. It's denoted by an s in the user execute field (e.g., -rwsr-xr-x). Common SUID programs perform privileged tasks requiring

temporary elevation.
Analysis:
C . /usr/bin/passwd:
Purpose: Updates user passwords in /etc/shadow (root-owned, 0600 perms).
Permissions: Typically -rwsr-xr-x, owned by root. The SUID bit lets non-root users modify shadow securely.
Command: ls -l /usr/bin/passwd confirms SUID (s in user execute).
A . /bin/ls:
Purpose: Lists directory contents, no privileged access needed.
Permissions: -rwxr-xr-x (no SUID). Runs as the calling user.
B . /usr/bin/curl:
Purpose: Transfers data over HTTP/FTP, no root privileges required by default.
Permissions: -rwxr-xr-x (no SUID).
Technical Details:
SUID Bit: Set via chmod u+s <file> or chmod 4755.
Security: SUID binaries are audited (e.g., find / -perm -u=s) due to escalation risks if writable or poorly coded (e.g., buffer overflows).
Security Implications: CNSP likely highlights SUID as an attack vector (e.g., CVE-1996-0095 exploited passwd flaws). Hardening removes unnecessary SUID bits.
Why other options are incorrect:
A, B: Lack SUID; no privileged operations.
D: Incorrect, as /usr/bin/passwd is a SUID example.
Real-World Context: SUID on /bin/su or /usr/bin/sudo similarly enables privilege escalation, often targeted in exploits.

NEW QUESTION # 25
What will be the subnet mask for 192.168.0.1/18?

- A. 255.255.192.0
- B. 255.255.255.0
- C. 255.225.225.0
- D. 255.225.192.0

Answer: A

Explanation:
An IP address with a /18 prefix (CIDR notation) indicates 18 network bits in the subnet mask, leaving 14 host bits (32 total bits - 18). For IPv4 (e.g., 192.168.0.1):
Binary Mask: First 18 bits are 1s, rest 0s.
1st octet: 11111111 (255)
2nd octet: 11111111 (255)
3rd octet: 11000000 (192)
4th octet: 00000000 (0)
Decimal: 255.255.192.0
Calculation:
Bits: /18 = 2