

OCT
2005

BSD Associate Exam Objectives



ABSTRACT

The official Exam Objectives for the BSD Associate (BSDA) certification exam, including detailed descriptions and study aides.

Copyright © 2005 BSD Certification Group

All Rights Reserved.

All trademarks are owned by their respective companies.

This work is protected by a Creative Commons License which requires Attribution. The human friendly version of the license can be viewed at

<http://creativecommons.org/licenses/by/2.5/>

which also provides a hyperlink to the legal code.

The BSD Certification Group specifies its attribution rights as follows: reproduction of this work in whole or in part is granted if a reference is made to the original work found at the [BSD certification website](#).

These conditions can only be waived by written permission from the BSD Certification Group. See the website for contact details.

Cover design by Michael Welsh

PREFACE

Welcome! This document introduces the BSD Associate (BSDA) examination and describes in considerable detail the objectives covered by the exam. The exam covers material across all four major projects of BSD Unix - NetBSD, FreeBSD, OpenBSD and DragonFly BSD.

While the testing candidate is expected to know concepts and practical details from all four main projects, it is not necessary to know all the details of each one. A thorough reading of this document is recommended to understand which concepts and practical details are expected to be mastered.

Throughout this document, a clear distinction is placed on 'recognizing' and 'understanding', versus 'demonstrating' and 'performing'. Certain objectives call for the mere understanding of certain topics, while others call for the ability to demonstrate performance level knowledge of the topic. The difference is an important one, and should be remembered.

Successful mastery of the BSDA examination will, in most cases, require study and practice. The requirements for the exam encompass more background in BSD than is common among casual users or those new to BSD. This is a deliberate decision by the BSD Certification Group- to encourage more cross learning among BSD systems so that breadth of understanding of BSD is as heavily tasked as depth of understanding. The result will be a more well-rounded BSD advocate and a more knowledgeable system administrator.

The BSD Certification Group

www.bsdcertification.org

October 4 , 2005

Table of Contents

- Preface.....3
- Executive Summary.....8
- 1 Introduction.....9
 - 1.1 Definition of Audience for BSDA.....10
 - 1.2 Operating System Versions Covered by BSDA.....10
 - 1.3 Recertification Requirements.....11
- 2 Exam Description.....12
 - 2.1 Official BSDA Examination Description.....12
- 3 Using the BSDA Study Domains.....13
 - 3.1 Domain 1: Installing and Upgrading the OS and Software.....14
 - 3.1.1 Recognize the installation program used by each operating system..14
 - 3.1.2 Recognize which commands are available for upgrading the operating system.....15
 - 3.1.3 Understand the difference between a pre-compiled binary and compiling from source.....15
 - 3.1.4 Understand when it is preferable to install a pre-compiled binary and how to do so.....15
 - 3.1.5 Recognize the available methods for compiling a customized binary..16
 - 3.1.6 Determine what software is installed on a system.....16
 - 3.1.7 Determine which software requires upgrading.....16
 - 3.1.8 Upgrade installed software.....17
 - 3.1.9 Determine which software have outstanding security advisories.....17
 - 3.1.10 Follow the instructions in a security advisory to apply a security patch.....17
 - 3.2 Domain 2: Securing the Operating System.....18
 - 3.2.1 Determine the system's security level.....18
 - 3.2.2 Configure an SSH server according to a set of requirements.....18
 - 3.2.3 Configure an SSH server to use a key pair for authentication.....19
 - 3.2.4 Preserve existing host keys during a system upgrade.....19
 - 3.2.5 Recognize alternate authentication mechanisms.....19
 - 3.2.6 Recognize alternate authorization schemes.....20
 - 3.2.7 Recognize basic recommended access methods.....20
 - 3.2.8 Recognize BSD firewalls and rulesets.....20
 - 3.2.9 Recognize BSD mechanisms for encrypting devices.....20
 - 3.2.10 Recognize methods for verifying the validity of binaries.....21
 - 3.2.11 Recognize the BSD methods for restraining a service.....21
 - 3.2.12 Change the encryption algorithm used to encrypt the password database.....21
 - 3.2.13 Modify the system banner.....21

- [3.2.14 Protect authentication data.....](#)22
- [3.3 Domain 3: Files, Filesystems and Disks.....](#)22
 - [3.3.1 Mount or unmount local filesystems.....](#)22
 - [3.3.2 Configure data to be available through NFS.....](#)22
 - [3.3.3 Determine which filesystems are currently mounted and which will be mounted at system boot.....](#)23
 - [3.3.4 Determine disk capacity and which files are consuming the most disk space.....](#)23
 - [3.3.5 Create and view symbolic or hard links.....](#)23
 - [3.3.6 View and modify ACLs.....](#)24
 - [3.3.7 View file permissions and modify them using either symbolic or octal mode.....](#)24
 - [3.3.8 Modify a file's owner or group.....](#)24
 - [3.3.9 Backup and restore a specified set of files and directories to local disk or tape.....](#)25
 - [3.3.10 Backup and restore a file system.....](#)25
 - [3.3.11 Determine the directory structure of a system.....](#)25
 - [3.3.12 Manually run the file system checker and repair tool.....](#)25
 - [3.3.13 View and modify file flags.....](#)26
 - [3.3.14 Monitor the virtual memory system.....](#)26
- [3.4 Domain 4: Users and Accounts Management.....](#)26
 - [3.4.1 Create, modify and remove user accounts.....](#)26
 - [3.4.2 Create a system account.....](#)27
 - [3.4.3 Lock a user account or reset a locked user account.....](#)27
 - [3.4.4 Determine identity and group membership.....](#)27
 - [3.4.5 Determine who is currently on the system or the last time a user was on the system.....](#)28
 - [3.4.6 Enable accounting and view system usage statistics.....](#)28
 - [3.4.7 Change a user's default shell.....](#)28
 - [3.4.8 Control which files are copied to a new user's home directory during account creation.....](#)28
 - [3.4.9 Change a password.....](#)29
- [3.5 Domain 5: Basic System Administration.....](#)29
 - [3.5.1 Determine which process are consuming the most CPU.....](#)29
 - [3.5.2 View and send signals to active processes.....](#)29
 - [3.5.3 Use an rc\(8\) script to determine if a service is running and start, restart or stop it as required.](#)30
 - [3.5.4 View and configure system hardware.....](#)30
 - [3.5.5 View, load, or unload a kernel module.....](#)30
 - [3.5.6 Modify a kernel parameter on the fly.....](#)31
 - [3.5.7 View the status of a software RAID mirror or stripe.....](#)31
 - [3.5.8 Determine which MTA is being used on the system.....](#)31
 - [3.5.9 Configure system logging.....](#)32
 - [3.5.10 Review log files to troubleshoot and monitor system behavior.....](#)32

- [3.5.11 Understand basic printer troubleshooting.....](#)32
- [3.5.12 Create or modify email aliases for Sendmail or Postfix.....](#)32
- [3.5.13 Halt, reboot, or bring the system to single-user mode.....](#)33
- [3.5.14 Recognize the difference between hard and soft limits and modify existing resource limits.....](#)33
- [3.5.15 Recognize the BSD utilities that shape traffic or control bandwidth.](#) 33
- [3.5.16 Recognize common, possibly third-party, server configuration files.](#) 33
- [3.5.17 Configure a service to start at boot time.....](#)34
- [3.5.18 Configure the scripts that run periodically to perform various system maintenance tasks.....](#)34
- [3.5.19 View the Sendmail or Postfix mail queue.....](#)34
- [3.5.20 Determine the last system boot time and the workload on the system.](#)34
- [3.5.21 Monitor disk input/output.....](#)35
- [3.5.22 Deal with busy devices.....](#)35
- [3.5.23 Determine information regarding the operating system.....](#)35
- [3.5.24 Understand the advantages of using a BSD license.....](#)35
- [3.6 Domain 6: Network Administration.....](#)35
 - [3.6.1 Determine the current TCP/IP settings on a system.....](#)36
 - [3.6.2 Set a system's TCP/IP settings.....](#)36
 - [3.6.3 Determine which TCP or UDP ports are open on a system.....](#)36
 - [3.6.4 Verify the availability of a TCP/IP service.....](#)36
 - [3.6.5 Query a DNS server.....](#)37
 - [3.6.6 Determine who is responsible for a DNS zone.....](#)37
 - [3.6.7 Change the order of name resolution.....](#)37
 - [3.6.8 Convert a subnet mask between dotted decimal, hexadecimal or CIDR notation.....](#)37
 - [3.6.9 Gather information using an IP address and subnet mask.....](#)38
 - [3.6.10 Understand IPv6 address theory.....](#)38
 - [3.6.11 Demonstrate basic tcpdump\(1\) skills.....](#)38
 - [3.6.12 Manipulate ARP and neighbor discovery caches.....](#)38
 - [3.6.13 Configure a system to use NTP.....](#)39
 - [3.6.14 View and renew a DHCP lease.....](#)39
 - [3.6.15 Recognize when and how to set or remove an interface alias.....](#)39
- [3.7 Domain 7: Basic Unix Skills.....](#)40
 - [3.7.1 Demonstrate proficiency in using redirection, pipes and tees.....](#)40
 - [3.7.2 Recognize, view and modify environmental variables.....](#)40
 - [3.7.3 Be familiar with the vi\(1\) editor.....](#)40
 - [3.7.4 Determine if a file is a binary, text, or data file.....](#)41
 - [3.7.5 Locate files and binaries on a system.....](#)41
 - [3.7.6 Find a file with a given set of attributes.....](#)41
 - [3.7.7 Create a simple Bourne shell script.....](#)42
 - [3.7.8 Find appropriate documentation.....](#)42
 - [3.7.9 Recognize the different sections of the manual.....](#)42

- 3.7.10 Verify a file's message digest fingerprint (checksum).....43
- 3.7.11 Demonstrate familiarity with the default shell.....43
- 3.7.12 Read mail on the local system.....43
- 3.7.13 Use job control.....43
- 3.7.14 Demonstrate proficiency with regular expressions.....44
- 3.7.15 Overcome command line length limitations.....44
- 3.7.16 Understand various domain contexts.....44
- 3.7.17 Configure an action to be scheduled by cron(8).....45
- 4 Conclusion.....46
- Appendix A.....47

EXECUTIVE SUMMARY

This document describes the requirements of the BSD Associate (BSDA) certification developed by the BSD Certification Group. It outlines seven *study domains*, each of which expands to about a dozen or more exam objectives. These objectives form the essentials that must be mastered by prospective candidates for this exam.

The exam itself covers topics in system administration for several popular versions of BSD Unix:

- NetBSD: 2.x and 3.x
- FreeBSD: 4.11, 5.x, 6.x
- OpenBSD: 3.6, 3.7, 3.8
- Dragonfly BSD: 1.2, 1.4

This list may change over time. Any changes will be announced on our website, www.bsdcertification.org.

The seven study domains cover installation and upgrades, security, files and disks, user accounts, networking, and several other basic skills in BSD Unix system administration. The document also includes a very useful cross reference tool (in Appendix A) that contains a list of common administrative commands found on the four BSD Unix variants.

The successful candidate can be expected to have a solid grasp on BSD Unix administration and be able to complete many common tasks working under or alongside more senior administrators.

1 Introduction

This document describes the requirements of the BSD Associate (BSDA) certification developed by the BSD Certification Group. The BSDA certification is the first in a series of certifications being developed by the Group to promote BSD Unix system usage around the world.

This document serves two purposes. It describes the certification itself, including the name, description, operating system and version coverage, and recertification requirements. It also introduces seven *study domains*, which are the focus of the exam. Various exam objectives under each study domain are then elaborated in more detail. Taken as a whole, the study domains define the minimum essential knowledge needed to confidently administer BSD systems at the associate level.

This document will be of interest to:

- Potential testing candidates for gauging their current experience against the exam requirements
- Trainers and educational institutions who wish to include BSD training in their offerings
- Courseware and study guide developers who plan to create materials to assist candidates in learning the skills needed to meet the exam requirements

As of this writing, there are no official study materials or recommended training institutions. For this reason, each exam objective details both the theory and commands the BSDA candidate can expect to be tested on. It is likely that the majority of the initial candidates who take the examination will already have most of the required skills and will be able to use self-study to brush up on the rest of the needed skills. Self-study will include each BSD project's handbook and online documentation, man pages, existing tutorials and how-tos, and Google.

There will also be candidates who wish to enter the field of BSD administration and who would benefit from specific study guides or a training program. For this reason, the BSD Certification Group makes the following recommendations:

- That trainers and developers of instructional material include all exam objectives. In addition, the material should provide easy-to-understand background concepts to allow the student to gain understanding of "why" as well as hands-on labs so the student can master the "how".
- That candidates considering purchasing either training or study materials carefully compare their purchase to the exam objectives found in this document. Verify before purchase that each objective is covered to the level of detail required for the candidate to gain mastery of the BSDA examination requirements.

1.1 Definition of Audience for BSDA

The BSDA certification is designed to be an entry-level certification on BSD Unix systems administration. Testing candidates with a general Unix background, but less than six months of work experience as a BSD systems administrator (or who wish to obtain employment as a BSD systems administrator) will benefit most from this certification. Human resource departments should consider the successful BSDA certified applicant to be knowledgeable in the daily maintenance of existing BSD systems under the direction and supervision of a more senior administrator.

The successful BSDA candidate is able to complete common administrative and troubleshooting tasks and has a good understanding of general BSD Unix and networking principles. In addition, the successful candidate demonstrates basic skills with these BSD operating systems: Dragonfly BSD, FreeBSD, NetBSD and/or OpenBSD.

This does not mean that the candidate needs to learn the complete details of four operating systems. It does mean that the candidate is aware of the basic utilities common to these operating systems, and where specified in the exam objectives, of features unique to some of the BSD operating systems. It is recommended that the testing candidate have access to each of these freely available operating systems while preparing for the exam. Online resources such as manual pages, wikis, and email lists also exist for all these BSD versions.

Note: The completion of the BSDA certification is not a requirement for the BSDP (BSD Professional) examination. If you feel that you already have the skills needed as a senior administrator, you should instead consider taking the BSDP examination.

1.2 Operating System Versions Covered by BSDA

In order for a certification to be practical, it must cover working knowledge in versions of BSD operating systems most likely to be seen in the workplace. The initial BSDA offering will cover material applicable to the following versions of these operating systems:

- NetBSD: 2.x and 3.x
- FreeBSD: 4.11, 5.x, 6.x
- OpenBSD: 3.6, 3.7, 3.8
- Dragonfly BSD: 1.2, 1.4

While this appears at first glance to be covering a lot of operating system versions, the testing candidate should keep in mind that the base utilities most likely to be encountered in the BSDA examination rarely change from one operating system revision to another.

The testing candidate is cautioned, however, to carefully review the exam objectives and to research any features the candidate may be unaware of. As a practical example: a candidate using FreeBSD 4.11 would not be experienced in `gmirror(8)` which was introduced in FreeBSD 5.3 and is available in later versions. It is recommended that candidates research the online "Release Notes" for the operating system if they come across an Exam Objective that does not appear on their version of the operating system.

1.3 Recertification Requirements

The exam objectives, the versions of operating systems to be covered, and the question pool for the BSDA will be regularly reviewed and modified as required to ensure that the material being tested is current and relevant. Should the exam objectives or tested operating systems change, an announcement will be made and the new objectives will be made publicly available.

Once a testing candidate has successfully met the requirements for BSDA certification, the candidate does NOT need to recertify when:

- the exam objectives are modified
- the question pool is modified
- a new version of any of the tested operating systems is released

However, the BSDA certification is only valid for 5 years. Existing BSDAs who wish to maintain their certification will need to recertify every 5 years. Details on how to recertify will be publicly available in a document to be published in 2006.

2 Exam Description

2.1 Official BSDA Examination Description

The following is the official BSDA examination description, suitable for display by training and testing centers and for use within training materials:

"The BSDA (BSD Associate) Certification is for those with light to moderate skills in BSD Unix system administration. The successful candidate understands and uses basic system administration commands and demonstrates proficiency in the following areas:

1. Installing and Upgrading the OS and Software
2. Securing the Operating System
3. Files, Filesystems and Disks
4. Users and Accounts Management
5. Basic System Administration
6. Network Administration
7. Basic Unix Skills

Each area contains detailed exam objectives that are fully described in the 'BSDA Certification Requirements Document', available at the BSD Certification Group website www.bsdcertification.org."

The official exam number will be noted in the release announcement as the exam is released. Release announcements for all other exams will also be on the BSDCG website www.bsdcertification.org.

3 Using the BSDA Study Domains

The BSDA certification exam covers seven study domains, each containing several detailed exam objectives:

1. Installing and Upgrading the OS and Software (10 objectives)
2. Securing the Operating System (13 objectives)
3. Files, Filesystems and Disks (14 objectives)
4. Users and Accounts Management (9 objectives)
5. Basic System Administration (24 objectives)
6. Network Administration (15 objectives)
7. Basic Unix Skills (17 objectives)

Testing candidates should understand all exam objectives prior to taking the exam, even though not every exam objective may be tested on every version of the BSDA exam.

Note that **exam objectives** are not the same as **exam questions**. Exam objectives are publicly available for distribution whereas exam questions are the private intellectual property of the BSD Certification Group. Those who have access to exam questions- psychometricians, translators, testing centers, and the testing candidate during the certification exam- are subject to an NDA (Non Disclosure Agreement) and are subject to penalties if the NDA is broken.

To assist the testing candidate in their studies and for those who wish to provide training and study materials, each exam objective has three components:

1. Number and name of exam objective.
2. **Concept** which describes the background material the candidate should master in order to effectively put the exam objective into administrative practice. The level of depth should reflect the BSDA audience described in Section 1.1.
3. **Practical** which provides a list of commands and other content which may appear on the certification exam. Where applicable, the section of the manual is included and candidates are encouraged to familiarize themselves with the contents of specified man pages.

Note: When using the exam objectives, keep the following points in mind:

- Remember that the BSDA certification exam tests practical skills: don't just memorize man pages. Instead, practice the commands until you understand the man pages.

- This is an entry-level examination. You are not expected to know everything, but you are expected to demonstrate that you have experience completing common administrative tasks.
- Watch for the word "recognize" in an exam objective. This indicates that you need to know something exists but not necessarily how to set it up. An example of this is Objective 3.2.11 where the candidate should recognize that BSD systems have the ability to constrain a service and which utilities are available to do so on each BSD. As a BSDA candidate, however, you are not expected to have experience in configuring a jail.
- Where major differences in the utilities used to accomplish a task on the various BSDs exist, it may be noted in the Practical component but the details regarding all differences are not. Instead, you are highly encouraged to refer to the cross reference table in Appendix A as you prepare for the exam. This table contains an alphabetical list of the required commands and their availability on the four listed BSD operating systems.

3.1 Domain 1: Installing and Upgrading the OS and Software

An important aspect of system administration is tracking installed versions of both the operating system and third-party applications. An advantage of using BSD systems is the availability of multiple tools to assist the system administrator in determining software versions and their dependencies. These tools indicate which software is out-of-date or has existing security vulnerabilities. They also assist in upgrading or patching software and its dependencies.

When and how installations and upgrades are done is specific to each organization. The successful BSDA candidate knows how to use the tools which are available for these purposes, and the cautions that are necessary when working on production systems under the supervision of a more senior administrator.

3.1.1 Recognize the installation program used by each operating system.

Concept:

While BSDA candidates are not expected to plan an installation, they should be able to start and complete an installation according to a provided list of requirements. Since the install procedure is operating system dependent, it is recommended that the candidate have prior experience in the default install routine for each tested BSD operating system. The candidate is also expected to have some familiarity with release numbering practices in general (e.g. "dot-zero releases") and where to find the release engineering practices at each BSD project's website.

Practical:

<http://www.bsinstaller.org> for Dragonfly BSD, sysinstall(8) for FreeBSD, sysinst on NetBSD install media, and INSTALL.[arch] on OpenBSD install media

3.1.2 Recognize which commands are available for upgrading the operating system.**Concept:**

The BSDA candidate should recognize the utilities which are used to keep the operating system up-to-date. Some utilities are common to the BSDs, some are specific to certain BSD operating systems and some are third-party applications.

Practical:

make(1) including the 'buildworld', 'installworld', and 'quickworld' and similar targets; mergemaster(8); cvs(1) and the third-party utilities cvsup and cvsnc; build.sh, etcupdate(8), postinstall(8) and afterboot(8); src/UPDATING and src/BUILDING.

3.1.3 Understand the difference between a pre-compiled binary and compiling from source.**Concept:**

The candidate should be familiar with the default location of both the ports collection and the pkgsrc collection and which BSDs use which type of collection. The candidate should also be able to recognize the extension used by packages. In addition, the candidate should be aware of the advantages and disadvantages of installing a pre-compiled binary and the advantages and disadvantages of compiling a binary from source.

3.1.4 Understand when it is preferable to install a pre-compiled binary and how to do so.**Concept:**

The BSDA candidate should be aware that while pre-compiled binaries are quick and easy to install, they don't allow the customization of the binary to a system's particular needs. The candidate needs to know how to install a pre-compiled binary from either a local or a remote source, as well as how to uninstall a pre-compiled binary

Practical: pkg_add(1), pkg_delete(1)

3.1.5 Recognize the available methods for compiling a customized binary.

Concept:

Many applications used by servers support make(1) options to compile a binary with the feature set required by a particular installation. While the BSDs all use make(1), the BSDA candidate should recognize that each BSD uses different mechanisms to use and preserve make(1) options.

Practical:

Dragonfly BSD: mk.conf(5) or make.conf(5), PKG_OPTIONS, CFLAGS

FreeBSD: -DWITH_* or WITH_*=, pkgtools.conf(5), make.conf(5)

NetBSD: PKG_OPTIONS.<pkg>, CFLAGS, mk.conf(5), PKG_DEFAULT_OPTIONS

OpenBSD: bsdpkg.conf(5)

3.1.6 Determine what software is installed on a system.

Concept:

The BSDA candidate should recognize that on BSD systems, software and dependencies are tracked by a package manager if the software was installed using packages, ports or pkgsrc. The candidate should be familiar with querying the package manager to determine what software and their versions are installed on the system.

Practical:

pkg_info(1)

3.1.7 Determine which software requires upgrading.

Concept:

The candidate should recognize the importance of balancing the need to keep software up-to-date while minimizing the impact on a production system. Dragonfly BSD and NetBSD use pkgsrc which provides utilities for determining which installed software is out-of-date. FreeBSD provides pkg_version and third-party utilities are also available which integrate with the BSD package managers.

Practical:

pkgsrc/pkgtool/pkg_chk and make show-downlevel for Dragonfly BSD and NetBSD; pkg_version(1), and the third-party portupgrade

3.1.8 Upgrade installed software.

Concept:

The BSDA candidate should recognize the built-in and third-party commands which are available for upgrading installed software on BSD systems. In addition, the candidate should be able to recognize which BSD systems use pkgsrc.

Practical:

Dragonfly BSD and NetBSD provide pkgsrc/pkgtools/pkg_chk, pkgsrc/pkgtools/pkg_comp, make update and make replace; portupgrade and cvsup are available as third-party utilities

3.1.9 Determine which software have outstanding security advisories.

Concept:

The BSDA candidate should recognize the importance of being aware of software security vulnerabilities. The candidate should also recognize the third-party utilities which integrate with the BSD package managers to determine which software has outstanding vulnerabilities.

Practical:

audit-packages for Dragonfly BSD and NetBSD; portaudit and vuxml for FreeBSD and OpenBSD

3.1.10 Follow the instructions in a security advisory to apply a security patch.

Concept:

The BSDA candidate should be aware that each BSD project maintains security advisories which are available both on the Internet and via mailing lists. The candidate should be able to follow the instructions in an advisory when asked to do so by a supervisor.

Practical:

patch(1), make(1), and fetch(1; ftp(1) and build.sh

3.2 Domain 2: Securing the Operating System

The mark of a good system administrator is the awareness of and adherence to best security practices. The BSDA candidate is expected to be familiar with common security practices. BSD systems are designed with security in mind and provide many mechanisms which allow the system administrator to tune systems to the security requirements of an organization. While the BSDA candidate won't always be responsible for implementing these mechanisms, the candidate should be able to recognize the features and commands available for securing BSD systems.

3.2.1 Determine the system's security level.

Concept:

BSD systems provide security profiles known as securelevels. The BSDA candidate should be able to recognize the restrictions set by each securelevel for each BSD operating system. The candidate should also understand under what circumstances a securelevel can be raised or lowered.

Practical:

init(8), sysctl(8), rc.conf(5)

3.2.2 Configure an SSH server according to a set of requirements.

Concept:

The BSDA candidate should be aware that the sshd(8) built into BSD systems can be configured to limit who can access a system via SSH.

Practical:

sshd_config(5)

3.2.3 Configure an SSH server to use a key pair for authentication.

Concept:

The candidate should understand private/public key theory including: which protocols are available for generating key pairs, choosing an appropriate bit size, providing a seed, providing a passphrase, and verifying a fingerprint. In addition, candidates should be able to generate their own keys and use them for authentication.

Practical:

ssh-keygen(1) including these keywords: authorized_keys, id_rsa, and id_rsa.pub

3.2.4 Preserve existing host keys during a system upgrade.

Concept:

In addition to knowing how to generate a system's SSH keys, the BSDA candidate should know where host keys are located and how to preserve them if the system is upgraded or replaced.

Practical:

/etc/ssh/ssh_host*_key*

3.2.5 Recognize alternate authentication mechanisms.

Concept:

The BSDA candidate is not expected to know how to configure an alternate authentication mechanism. However, candidates should understand basic authentication theory and be aware that providing a username and password is only one way to authenticate on BSD systems. Candidates should have a basic understand of PAM and know it is available on Dragonfly BSD, FreeBSD and NetBSD 3.x. They should also understand basic theory regarding Kerberos, OTP and RADIUS.

3.2.6 Recognize alternate authorization schemes.

Concept:

Candidates should understand basic authorization theory and how MAC and ACLs extend the features provided by the standard Unix permissions.

Practical:

mac(4) and acl(3) on FreeBSD; systrace(1) on NetBSD and OpenBSD

3.2.7 Recognize basic recommended access methods.

Concept:

BSDA candidates should be familiar with standard system administration practices used to minimize the risks associated with accessing a system. These include using ssh(1) instead of telnet(1), denying root logins, using the possibly third-party sudo utility instead of su(1) and minimizing the use of the wheel group.

Practical:

ttys(5), sshd_config(5), ftpusers(5); the possibly third-party utility sudo which includes visudo, suedit and sudoers

3.2.8 Recognize BSD firewalls and rulesets.

Concept:

Each BSD comes with at least one built-in firewall. The BSDA candidate should recognize which firewalls are available on each BSD and which commands are used to view each firewall's ruleset.

Practical:

ipfw(8), ipf(8), ipfstat(8), pf(4), pfctl(8) and firewall(7)

3.2.9 Recognize BSD mechanisms for encrypting devices.

Concept:

The BSDA candidate should be aware that it is possible to encrypt devices on BSD systems and which utilities are available on each BSD system.

Practical:

gbde(4) and gbde(8) on FreeBSD; cgd(4) on NetBSD; vnd(4) on OpenBSD

3.2.10 Recognize methods for verifying the validity of binaries.

Concept:

BSDA candidates should recognize the utility of file integrity utilities such as tripwire. They should also recognize the built-in checks provided on some of the BSDs.

Practical:

security(7) or (8); security.conf(5); veriexecctl(8)

3.2.11 Recognize the BSD methods for restraining a service.

Concept:

The BSDA candidate should recognize the advantages of restraining a service on an Internet facing system and which utilities are available to do so on each of the BSDs.

Practical:

chroot(8); jail(8); systrace(1); the third-party Xen application

3.2.12 Change the encryption algorithm used to encrypt the password database.

Concept:

Given a screenshot of a password database, the BSDA candidate should be able to recognize the encryption algorithm in use and how to select another algorithm. The candidate should also have a basic understanding of when to use DES, MD5 and Blowfish.

Practical:

login.conf(5); auth.conf(5); passwd.conf(5); adduser.conf(5) and adduser(8)

3.2.13 Modify the system banner.

Concept:

The candidate should be aware of the banner(s) that may be seen depending on how a user accesses a system and which files are used to configure each banner.

Practical:

motd(5), login.conf(5), gettytab(5), sshd_config(5)

3.2.14 Protect authentication data.

Concept:

To prevent attacks against system security with password cracking attacks, BSD systems keep encrypted passwords visible to system processes only. The BSDA candidate should have an understanding of the location of the password database files and their proper permission sets.

Practical:

passwd(5), pwd_mkdb(8)

3.3 Domain 3: Files, Filesystems and Disks

The usefulness of any computing system is related to the accessibility of the data stored on it. The BSDA candidate is expected to thoroughly understand how to make data available both locally and remotely and how to use permissions to ensure authorized users can access that data. The candidate should be experienced in backing up data and in resolving common disk issues.

3.3.1 Mount or unmount local filesystems.

Concept:

The BSDA candidate should be familiar with all aspects of mounting and unmounting local filesystems including: how to mount/umount a specified filesystem, how to mount all filesystems, configuring filesystems to be mounted at boot, passing options to mount(1), and resolving mount(1) errors.

Practical:

mount(8), umount(8), fstab(5)

3.3.2 Configure data to be available through NFS.

Concept:

BSDA candidates should be aware of the utilities associated with NFS and the security risks associated with allowing RPC through a firewall. In addition, the candidate should be able to configure a NFS server or client according to a set of requirements on the data to be made available.

Practical:

exports(5), nfsd(8), mountd(8), rpcbind(8) or portmap(8), rpc.lockd(8), rpc.statd(8), rc.conf(5) and mount_nfs(8)

3.3.3 Determine which filesystems are currently mounted and which will be mounted at system boot.**Concept:**

The candidate should be able to determine which filesystems are currently mounted and which will be mounted at boot time.

Practical:

mount(1), du(1), fstab(5)

3.3.4 Determine disk capacity and which files are consuming the most disk space.**Concept:**

The BSDA candidate should be able to combine common Unix command line utilities to quickly determine which files are consuming the most disk space.

Practical:

du(1), df(1), find(1), sort(1), systat(1)

3.3.5 Create and view symbolic or hard links.**Concept:**

Candidates should know the difference between symbolic and hard links as well as how to create, view and remove both types of links. In addition, the candidate should be able to temporarily resolve a low disk space issue using a symbolic link.

Practical:

ln(1), ls(1), rm(1), stat(1)

3.3.6 View and modify ACLs.

Concept:

The BSDA candidate should be able to determine if a FreeBSD system is using ACLs, and if so, on which filesystems. In addition, the candidate should be able to view and modify a file's ACL on a FreeBSD system.

Practical:

mount(8), ls(1), getfacl(1)

3.3.7 View file permissions and modify them using either symbolic or octal mode.

Concept:

The BSDA candidate is expected to have a thorough understanding of traditional Unix permissions including: how to view and modify permissions, why the sticky bit is important on /tmp and other shared directories, recognizing and using the SUID and SGID bits, and the difference between symbolic and octal mode. In addition, the candidate should understand that a shell setting determines the default file and directory permissions and, given a umask value, the candidate should be able to determine the default permission set.

Practical:

ls(1), chmod(1), umask(1) or (2)

3.3.8 Modify a file's owner or group.

Concept:

The BSDA candidate should be able to modify a file's ownership as required. In addition, the candidate should be aware of the importance of verifying one's own identity before creating files.

Practical:

chown(8), chgrp(1); su(1), mtree(8)

3.3.9 Backup and restore a specified set of files and directories to local disk or tape.

Concept:

Candidates should have experience using common Unix command line backup utilities. In addition, candidates should be able to recognize the device names for tape devices on BSD systems.

Practical:

tar(1), cpio(1), pax(1), cp(1), cpdup(1)

3.3.10 Backup and restore a file system.

Concept:

Candidates should recognize the utilities used to backup an entire filesystem and the various dump(1) levels.

Practical:

dump(8), restore(8), dd(1)

3.3.11 Determine the directory structure of a system.

Concept:

BSDA candidates should be able to quickly determine the directory layout used by BSD systems.

Practical:

hier(7)

3.3.12 Manually run the file system checker and repair tool.

Concept:

The BSDA candidate should be aware of the utilities available to check the consistency of a file system and to use them under supervision.

Practical:

fsck(8)

3.3.13 View and modify file flags.

Concept:

The BSDA candidate should understand how file flags augment traditional Unix permissions and should recognize how to view and modify the immutable, append-only and undelete flags.

Practical:

ls(1), chflags(1)

3.3.14 Monitor the virtual memory system.

Concept:

The virtual memory subsystem may have an important impact on a system's overall performance. The BSDA candidate should be able to configure a swap device and review swap usage.

Practical:

pstat(8); systat(1); top(1); vmstat(8); swapctl(8); swapinfo(8)

3.4 Domain 4: Users and Accounts Management

All systems require at least one user account, and depending upon the role of the system, the BSDA candidate's job duties may include supporting end-users in the maintenance of their accounts. Accordingly, the BSDA candidate should be able to create user accounts, modify account settings, disable accounts, and reset passwords. The candidate should also know how to track account activity and determine which accounts are currently accessing a system.

3.4.1 Create, modify and remove user accounts.

Concept:

Managing user accounts is an important aspect of system administration. The BSDA should be aware that the account management utilities differ across BSD systems and should be comfortable using each utility according to a set of requirements.

Practical:

vipw(8); pw(8), adduser(8), adduser.conf(5), useradd(8), userdel(8), rmuser(8), userinfo(8), usermod(8), and user(8)

3.4.2 Create a system account.**Concept:**

Candidates need to understand that many services require an account and that such accounts should not be available for logins.

Practical:

nologin(8); using a * in the password field of passwd(5)

3.4.3 Lock a user account or reset a locked user account.**Concept:**

The BSDA candidate should know how to recognize a locked account and how to remove the lock.

Practical:

vipw(8); chpass(1), chfn(1), chsh(1), pw(8), user(8)

3.4.4 Determine identity and group membership.**Concept:**

In the context of the Unix permission system, determining one's identity and group membership is essential to determine what authorizations are available. The BSDA should be able to determine, and as required, change identity or group membership.

Practical:

id(1), groups(1), who(1), whoami(1), su(1)

3.4.5 Determine who is currently on the system or the last time a user was on the system.

Concept:

BSD systems maintain databases which can be queried for details regarding logins. The BSDA candidate should be familiar with the database names and the utilities available for determining login information.

Practical:

wtmp(5), utmp(5), w(1), who(1), users(1), last(1), lastlogin(8), lastlog(5), finger(1)

3.4.6 Enable accounting and view system usage statistics.

Concept:

The BSDA candidate should be aware of when it is appropriate to enable system accounting, recognize which utilities are available to do so, and know how to view the resulting statistics.

Practical:

ac(8), sa(8), accton(8), lastcomm(1), last(1)

3.4.7 Change a user's default shell.

Concept:

Candidates should know the default shells for both user accounts and the superuser account for each BSD. In addition, they should know how to change the default shell for each BSD operating system.

Practical:

vipw(8); chpass(1), chfn(1), chsh(1), pw(8), user(8)

3.4.8 Control which files are copied to a new user's home directory during account creation.

Concept:

BSD systems use a "skel" directory containing files which are copied over to a user's home directory when a user account is made. The BSDA candidate should be aware of the location of the skel directory on each BSD, as well as how to override the copying of its contents during account creation.

Practical:

pw(8), adduser.conf(5), useradd(8) and usermgmt.conf(5)

3.4.9 Change a password.**Concept:**

The BSDA candidate should be able to change his own password as well as the passwords of other users as required.

Practical:

passwd(1), vipw(8)

3.5 Domain 5: Basic System Administration

An important component of system administration is an awareness of its subsystems and their interactions, as well as how to monitor the health of a running system. The BSDA candidate needs to demonstrate experience in interacting with BSD processes, a running kernel, and the BSD boot process. Candidates should also demonstrate familiarity with BSD devices, the disk subsystem and the mail and print daemons.

3.5.1 Determine which process are consuming the most CPU.**Concept:**

The BSDA candidate should be able to view active processes and recognize inordinate CPU usage. In addition, the candidate should know how to end a process or change its priority.

Practical:

top(1), systat(1), ps(1), nice(1), renice(1), kill(1)

3.5.2 View and send signals to active processes.**Concept:**

The BSDA candidate should be familiar with both the names and numbers of the most commonly used Unix signals and how to send a signal to an active process. The candidate should also recognize the difference between a SIGTERM and a SIGKILL.

Practical:

ps(1); kill(1); killall(1); pkill(1); pgrep(1)

3.5.3 Use an rc(8) script to determine if a service is running and start, restart or stop it as required.**Concept:**

In addition to directly sending signals to processes, the BSDA candidate should realize that BSD systems provide scripts which can be used to check the status of services and to stop, start and restart them as required. The candidate should also be aware of the locations of these scripts on each of the BSD systems. Candidate's should note that this objective does not apply to OpenBSD.

Practical:

rc(8), rc.conf(5)

3.5.4 View and configure system hardware.**Concept:**

BSD systems come with many utilities to determine what hardware is installed on a system. The BSDA candidate should know how to determine which hardware was probed at boot time as well as some BSD specific utilities which can be used to troubleshoot and manipulate PCI, ATA, and SCSI devices.

Practical:

dmesg(8), /var/run/dmesg.boot, pciconf(8), atacontrol(8) and camcontrol(8); atactl(8) and /kern/msgbuf; scsictl(8) or scsi(8)

3.5.5 View, load, or unload a kernel module.**Concept:**

The BSDA candidate should understand the difference between a statically compiled kernel and one that uses loadable kernel modules. The candidate should be able to view, load and unload kernel modules on each BSD system but should be aware that kernel modules are discouraged on NetBSD and OpenBSD systems.

Practical:

kldstat(8), kldload(8), kldunload(8), and loader.conf(5); modstat(8), modload(8), modunload(8), and lkm.conf(5)

3.5.6 Modify a kernel parameter on the fly.

Concept:

BSD systems maintain kernel MIB variables which allow a system administrator to both view and modify the kernel state of a running system. The BSDA should be able to view and modify these MIBs both at run-time and permanently over a system boot. The candidate should also recognize how to modify a read-only MIB.

Practical:

sysctl(8), sysctl.conf(5)

3.5.7 View the status of a software RAID mirror or stripe.

Concept:

In addition to providing drivers for hardware RAID devices, BSD systems also provide built-in mechanisms for configuring software RAID systems. The BSDA candidate should know the difference between RAID levels 0, 1, 3 and 5 and recognize which utilities are available to configure software RAID on each BSD system.

Practical:

vinum(8), gmirror(8), gstripe(8), graid3(8), raidctl(8), ccdconfig(8)

3.5.8 Determine which MTA is being used on the system.

Concept:

The BSDA candidate should recognize the role of the MTA, recognize which MTA(s) are available during each BSD's operating system install routine and which configuration file indicates the MTA in use on the system. The candidate should also recognize the difference between the mbox or maildir mail destination file format type.

Practical:

mailer.conf(5)

3.5.9 Configure system logging.

Concept:

The BSDA candidate must understand that the system automatically maintains the creation and maintenance of many different logs. The candidate should be able to configure log rotation by either time or size, understand logging facilities and priorities, as well as view compressed logs.

Practical:

Note that the newsyslog(8) implementations vary by BSD.

newsyslog(8), newsyslog.conf(5), syslog.conf(5), zmore(1), bzip(1)

3.5.10 Review log files to troubleshoot and monitor system behavior.

Concept:

Candidates should be aware of the importance of reviewing log files on a regular basis as well as how to watch a log file when troubleshooting.

Practical:

tail(1), /var/log/*, syslog.conf(5), grep(1), dmesg(8)

3.5.11 Understand basic printer troubleshooting.

Concept:

The BSDA candidate should be able to view the print queue and manipulate the jobs within the queue. The candidate should also be able to recognize the meaning of the first two fields in an /etc/printcap entry.

Practical:

lpc(8), lpq(1), lprm(1), printcap(5)

3.5.12 Create or modify email aliases for Sendmail or Postfix.

Concept:

The BSDA candidate should understand when to create an email alias and how to do so for either Sendmail or Postfix.

Practical:

newaliases(1), aliases(5), postalias(1)

3.5.13 Halt, reboot, or bring the system to single-user mode.

Concept:

The BSDA candidate needs to understand the ramifications associated with halting, rebooting, or bringing a system to single-user mode, recognize when it may be necessary to do so and how to minimize the impact on a server system.

Practical:

shutdown(8)

3.5.14 Recognize the difference between hard and soft limits and modify existing resource limits.

Concept:

Candidates need to understand that resource limits are inherited by the shell as well as how to view their limits and change them both temporarily and permanently. In addition, candidates should understand the difference between soft and hard limits.

Practical:

limit(1), limits(1), login.conf(5); sysctl(8) on NetBSD

3.5.15 Recognize the BSD utilities that shape traffic or control bandwidth.

Concept:

Candidates should understand when it is advantageous to create policies controlling the amount of bandwidth available to specified services. In addition, candidates should recognize the utilities available on BSD systems to create bandwidth policies.

Practical:

ipfw(8), altq(4), dumynet(4), altq(9), altqd(8), altq.conf(5)

3.5.16 Recognize common, possibly third-party, server configuration files.

Concept:

BSD systems are often used to provide Internet services. The BSDA candidate may be asked to view or make a specified change to a service's configuration file and needs to recognize the names of the most commonly used configuration files and which

applications they are associated with.

Practical:

httpd.conf(5), sendmail.cf, master.cf, dhcpd.conf(5), named.conf(5), smb.conf(5)

3.5.17 Configure a service to start at boot time.**Concept:**

The BSDA candidate should recognize that the BSD boot process does not use runlevels. The candidate must be able to configure essential services to start at boot time to minimize the impact of a system reboot.

Practical:

rc.conf(5), rc(8), inetd(8)

3.5.18 Configure the scripts that run periodically to perform various system maintenance tasks.**Concept:**

BSD systems provide many scripts that are used to maintain and verify the integrity of the system. The BSDA candidate should be able to locate and run these scripts manually as required as well as configure which scripts run daily, weekly and monthly on each BSD system.

Practical:

periodic.conf(5) and periodic(8) on Dragonfly BSD and FreeBSD; security.conf(5), daily.conf(5), weekly.conf(5), and monthly.conf(5) on NetBSD; daily(8), weekly(8), and monthly(8) on OpenBSD

3.5.19 View the Sendmail or Postfix mail queue.**Concept:**

The BSDA candidate should be able to view the mail queue to determine if any mail is stuck in the queue, and if necessary, ask the MTA to reprocess or flush the queue.

Practical:

mailq(1), postqueue(1)

3.5.20 Determine the last system boot time and the workload on the system.**Concept:**

The BSDA candidate should be able to monitor the system's workload using the time since last system reboot, as well as the system load over the last 1, 5 and 15 minutes in order to determine operation parameters.

Practical:

uptime(1), w(1), top(1)

3.5.21 Monitor disk input/output.

Concept:

A system's disk input/output can have a dramatic impact on performance. The BSDA candidate should know how to use the utilities available on BSD systems to monitor disk I/O and interpret their results.

Practical:

iostat(8), systat(1), vmstat(1), nfsstat(1)

3.5.22 Deal with busy devices.

Concept:

The BSDA candidate should understand what can cause a process to hang, how to detect related processes and how to fix the situation.

Practical:

ps(1), fstat(1), kill(1), umount(8) and the third-party lsof utility

3.5.23 Determine information regarding the operating system.

Concept:

The BSDA candidate should be able to determine the type and version of the operating system installed.

Practical:

uname(1), sysctl(8); /etc/release on NetBSD

3.5.24 Understand the advantages of using a BSD license.

Concept:

The BSDA candidate should recognize the 2-clause BSD license and how the license does not place restrictions on whether BSD licensed code remains Open Source or becomes integrated into a commercial product.

3.6 Domain 6: Network Administration

TCP/IP was originally implemented on BSD systems and BSD systems continue to provide core networking services for a substantial portion of the Internet. The BSDA is expected to demonstrate a strong understanding of both IPv4 and IPv6 addressing as

well as basic networking theory. Trainers and material providers should provide conceptual depth similar to that found in Network+ or in the networking theory section of CCNA.

3.6.1 Determine the current TCP/IP settings on a system.

Concept:

The BSDA candidate should be able to determine a system's IP address(es), subnet mask, default gateway, primary and secondary DNS servers and hostname.

Practical:

ifconfig(8), netstat(1), resolv.conf(5), route(8), hostname(1)

3.6.2 Set a system's TCP/IP settings.

Concept:

Candidates should be able to modify required TCP/IP settings both temporarily and permanently in order to remain after a reboot.

Practical:

hostname (1), ifconfig(8), route(8), resolv.conf(5), rc.conf(5), hosts(5), hostname.if(5), myname(5), mygate(5), netstart(8)

3.6.3 Determine which TCP or UDP ports are open on a system.

Concept:

The BSDA candidate should be able to use the utilities found on BSD systems as well as third-party programs to determine which ports are open on a system and which ports are being seen through a firewall.

Practical:

netstat(1), services(5) and fstat(1); sockstat(1) and third-party nmap and lsof

3.6.4 Verify the availability of a TCP/IP service.

Concept:

The BSDA candidate should be able to determine if a remote system is available via TCP/IP, and if so, telnet(1) to a particular TCP service to determine if it is responding to client requests.

Practical:

ping(8), traceroute(8), telnet(1); nc(1) on FreeBSD and OpenBSD

3.6.5 Query a DNS server.

Concept:

The BSDA candidate should understand basic DNS theory, including types of resource records, types of DNS servers, reverse lookups and zone transfers. The candidate should be able to query a DNS server for a particular type of resource record, understand which servers are authoritative for a zone and determine if a DNS server is willing to do a zone transfer.

Practical:

dig(1), host(1), nslookup(1), ping(8), telnet(1)

3.6.6 Determine who is responsible for a DNS zone.

Concept:

The BSDA candidate should be able to perform a reverse DNS lookup to determine the network associated with an IP address and gather contact information regarding that network.

Practical:

dig(1) and whois(1)

3.6.7 Change the order of name resolution.

Concept:

Candidates should be able to determine the default order of host name resolution on BSD systems and recognize which configuration file controls the order of host name resolution.

Practical:

ping(8), telnet(1), nsswitch.conf(5), resolv.conf(5), host.conf(5)

3.6.8 Convert a subnet mask between dotted decimal, hexadecimal or CIDR notation.

Concept:

The BSDA candidate should be familiar with IPv4 addressing and how to convert a subnet mask from a given notation to another specified notation.

3.6.9 Gather information using an IP address and subnet mask.

Concept:

Given an IPv4 address and subnet mask, the BSDA candidate should be able to determine the subnet address, broadcast address and the valid host addresses available on that subnet address.

3.6.10 Understand IPv6 address theory.

Concept:

The BSDA candidate should be able to recognize basic IPv6 addressing theory including: the components of an IPv6 address; the support for multiple addresses (link, local, global) per interface; address and prefix representation (aaaa:bbbb::dddd/17) and the address format (48bit prefix, 16bit subnet, 64 hostbits). In addition, the candidate should understand the autoconfiguration process where the router sends its prefix or gets queried and the host adds its 64 host-bits which are derived from its MAC address. Finally, the candidate should be able to troubleshoot basic IPv6 connectivity.

Practical:

ifconfig(8), ping6(8), rtsol(8)

3.6.11 Demonstrate basic tcpdump(1) skills.

Concept:

Given some tcpdump(1) output, the BSDA candidate should be able to answer basic network connectivity questions. This requires the candidate to recognize common TCP and UDP port numbers, the difference between a TCP/IP server and a TCP/IP client, and the TCP three-way handshake.

Practical:

tcpdump(1)

3.6.12 Manipulate ARP and neighbor discovery caches.

Concept:

Candidates should understand basic ARP theory as well as the neighbor discovery cache used on IPv6 networks. Candidates should be able to view, modify and clear these caches and recognize when it is necessary to do so.

Practical:

arp(8), ndp(8)

3.6.13 Configure a system to use NTP.**Concept:**

The BSDA candidate should be familiar with the concepts in RFC 868, the importance of synchronizing time on server systems and which services in particular are time sensitive. The candidate should also be able to configure NTP and manually synchronize with a time server as required.

Practical:

ntpd(8), ntpd.conf(5), rc.conf(5), rdate(8)

3.6.14 View and renew a DHCP lease.**Concept:**

The BSDA candidate should have a basic understanding of DHCP leases and how to configure a client to override the settings received from a DHCP server. In addition, the candidate should be able to view the current lease, release it and renew a lease. Since the DHCP client used varies, the candidate should be familiar with using the DHCP client commands on each BSD.

Practical:

dhclient(8), dhclient.leases(5), dhclient.conf(5), rc.conf(5)

3.6.15 Recognize when and how to set or remove an interface alias.**Concept:**

The BSDA candidate should recognize when it is appropriate to set or remove an interface alias and the available commands on each of the BSDs.

Practical:

ifconfig(8), rc.conf(5), ifaliases(5), hostname.if(5)

3.7 Domain 7: Basic Unix Skills

BSD has its roots in Unix and many Unix utilities were originally developed on BSD systems. The BSDA candidate needs to demonstrate proficiency in the most commonly used Unix command line utilities.

3.7.1 Demonstrate proficiency in using redirection, pipes and tees.

Concept:

The BSDA candidate should be able to to redirect standard input, output or error, use a pipe to send the output of one command to another command or file, and use a tee to copy standard input to standard output.

Practical:

<, >, |, tee(1), >& and |&

3.7.2 Recognize, view and modify environmental variables.

Concept:

The BSDA candidate should be able to view and modify environmental variables both temporarily and permanently for each of the default shells found on BSD systems.

Practical:

env(1), sh(1), csh(1), tcsh(1), environ(7)

3.7.3 Be familiar with the vi(1) editor.

Concept:

The default editor on BSD systems is often vi(1) and many system utilities require familiarity with vi(1) commands. The BSDA candidate should be able to edit files using this editor, as well as modify a read-only file or exit vi(1) without saving any edits to the file.

Practical:

vi(1) including: :w, :wq, :wq!, :q!, dd, y, p, x, i, a, /, :, :r, ZZ, :set number, :set list

3.7.4 Determine if a file is a binary, text, or data file.

Concept:

While BSD systems use naming conventions to help determine the type of file, the BSDA candidate should be aware that these are conventions only and that there is a magic database to help determine file type.

Practical:

file(1), magic(5)

3.7.5 Locate files and binaries on a system.

Concept:

The BSDA candidate should be able to quickly find the location of any file on the system as needed and know which utilities can be used to find binaries, source, manpages and files. In addition, the candidate should be able to update the locate(1) database.

Practical:

whatis(1); whereis(1); which(1); locate(1); find(1); sh(1) including "type" built-in, -v and -V; locate.updatedb(8) or locate.conf(5)

3.7.6 Find a file with a given set of attributes.

Concept:

The find(1) utility is invaluable when searching for files matching a specific set of attributes. The BSDA candidate should be comfortable in using this utility and may be asked to locate files according to last modification time, size, type, file flags, UID or GID, permissions or by a text pattern.

Practical:

find(1)

3.7.7 Create a simple Bourne shell script.

Concept:

Most system administration tasks can be automated with shell scripts. The BSDA candidate should be aware of the advantages and disadvantages of using a Bourne shell script rather than a csh(1) or bash(1) shell script. The candidate should be able to recognize a shebang, comments, positional parameters and special parameters, wildcards, the proper use of quotes and backslashes and: for, while, if, case, and exec. In addition, the candidate should know how to make a script executable and how to troubleshoot a script.

Practical:

sh(1), chmod(1)

3.7.8 Find appropriate documentation.

Concept:

BSD systems are well documented and there are many detailed resources available to the system administrator. The BSDA candidate should be able to use the documentation found on the system itself as well as be aware of the resources available on the Internet.

Practical:

apropos(1), man(1), man.conf(5), whatis(1), and info(1); share/doc/ and share/examples/; in addition, each BSD project maintains an online handbook and several mailing lists

3.7.9 Recognize the different sections of the manual.

Concept:

The BSDA candidate needs to recognize what type of information is found in each section of the manual. In addition, the candidate should be able to specify a specific section of the manual, ask to see all sections of the manual, and do a search query within the manual.

Practical:

man (1), intro(1) to intro(9), "/"

3.7.10 Verify a file's message digest fingerprint (checksum).

Concept:

The BSDA candidate should be familiar with the theory behind a message digest fingerprint and why it is important to verify a file's fingerprint. In addition, candidates should be able to create a fingerprint as well as verify an existing fingerprint.

Practical:

md5(1), openssl(1), sha1(1), cksum(1)

3.7.11 Demonstrate familiarity with the default shell.

Concept:

The BSDA candidate should be comfortable using the sh(1), csh(1) or tcsh(1) shells. The candidate should be able to modify shell behavior both temporarily and permanently including: prevent the shell from clobbering existing files, use history substitution, and set command aliases to save time at the command line. The candidate should also know how to temporarily bypass a command alias.

Practical:

sh(1), csh(1), and tcsh(1) including: !, !!, \$, 0, h, t, r, p, \

3.7.12 Read mail on the local system.

Concept:

The BSDA candidate should be aware that by default, system messages may be emailed to the root user on the local system and that a third-party MUA may not be installed. The candidate should be able to both read and send mail using the built-in mail(1) command. The candidate should also know the location of user mailbox files.

Practical:

mail(1), /var/mail/\$USER

3.7.13 Use job control.

Concept:

The BSDA candidate should know how to start a process in the background, place an existing process into the background, and return a background process to the foreground. The candidate should also be able to verify if any jobs are currently in the

background and be aware of the difference between kill(1) and the shell built-in "kill".

Practical:

&, ^Z, jobs, bg, fg, and "kill" which are all built-in to the shell

3.7.14 Demonstrate proficiency with regular expressions.**Concept:**

Regular expressions are part of the daily life of a system administrator and the BSDA candidate needs to be able to match text patterns when analyzing program output or searching through files. The candidate should be able to specify a range of characters within [], specify a literal, use a repetition operator, recognize a metacharacter and create an inverse filter.

Practical:

grep(1), egrep(1), fgrep(1), re_format(7)

3.7.15 Overcome command line length limitations.**Concept:**

The command line length is limited, and often a command should be applied to more arguments than fit on a command line. The BSDA candidate should understand how to run the command multiple times with different arguments for each call using xargs(1) or a shell "while" read loop.

Practical:

xargs(1), find(1)

3.7.16 Understand various domain contexts.**Concept:**

The term "domain" is used in Unix for several facilities. The candidate should understand the meaning of the term in the context of the Network Information System (NIS), the Domain Name System (DNS), Kerberos, and NTLM domains.

Practical:

domainname(1), resolv.conf(5), krb5.conf(5), smb.conf(5)

3.7.17 Configure an action to be scheduled by cron(8).

Concept:

The BSDA candidate should understand the difference between the system crontab and user crontabs. In addition, the candidate should be familiar with using the crontab editor, be able to recognize the time fields seen in a crontab, and understand the importance of testing scripts before scheduling their execution through cron(8). Candidates should also recognize that the files `/var/cron/allow` and `/var/cron/deny` can be created to control which users can create their own crontabs.

Practical:

crontab(1), cron(8), crontab(5)

4 Conclusion

The BSD Associate examination covers a lot of ground. The candidate is expected to know several concepts and details from all four major BSD variants. While this may place an initial burden on some candidates, the BSD Certification Group believes that having some knowledge of all four branches strengthens candidates and places them in the best position to recommend and maintain the right BSD for the job.

By mastering a broad coverage of topics, the candidate absorbs knowledge and information they may not yet have encountered in their use of BSD Unix. The result of engaging these topics for study is two fold- better qualified BSD Unix users and administrators, and more people knowledgeable of the range of BSD choices available. This strengthens the case for BSD Unix deployment worldwide.

Appendix A

To assist the testing candidate in his studies, Table 1 provides an alphabetized list of the commands and files found in the "practical" sections of the exam objectives.

The candidate should note that the availability of a command either with a version of the operating system or as a third-party application may change over time. Candidates who find that a command is now available where the Table indicates it is not, are encouraged to contact the BSDCG through the contact form at the website, indicating the name of the command and the version of the operating system. An up-to-date version of the Table will be maintained at www.bsdcertification.org.

Key:

- X can be installed with operating system and has an associated man page
- * can be installed with operating system but has no associated man page
- P available as a third party package or port

Table 1: Command Reference

Command or File	Dragonfly BSD	FreeBSD	NetBSD	OpenBSD
/etc/release			X	
/etc/ssh/*	X	X	X	X
/kern/msgbuf			X	
/var/log/*	X	X	X	X
/var/mail/\$USER	X	X	X	X
/var/run/dmesg.boot	X	X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
ac(8)	X	X	X	X
accton(8)	X	X	X	X
acl(3)	X	X		
adduser.conf(5)	X	X		
adduser(8)	X	X		X
afterboot(8)			X	X
aliases(5)	X	X	X	X
altq.conf(5)			X	
altq(4) or (9)		X	X	X
altqd(8)			X	
apropos(1)	X	X	X	X
arp(8)	X	X	X	X
atacontrol(8)	X	X		
atactl(8)			X	X
audit-packages	P		P	
auth.conf(5)	X	X		
bsd.port.mk(5)				X
build.sh			X	
bzcat(1) (bzip)	X	X	X	P
camcontrol(8)	X	X		
ccdconfig(8)	X	X	X	X
cgd(4)			X	
chflags(1)	X	X	X	X
chfn(1)	X	X	X	X
chgrp(1)	X	X	X	X
chmod(1)	X	X	X	X
chown(8)	X	X	X	X
chpass(1)	X	X		X
chroot(8)	X	X	X	X
chsh(1)	X	X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
cksum(1)	X	X	X	X
cp(1)	X	X	X	X
cpdup(1)	X	P		
cpio(1) or (1L)	X	X	X	X
cron(8)	X	X	X	X
crontab(1)	X	X	X	X
crontab(5)	X	X	X	X
csh(1)	X	X	X	X
cvs(1)	X	X	X	X
cvsup(1)	P	P	P	P
cvsync(1)	P	P	P	P
daily.conf(5)			X	
daily(8)				X
dd(1)	X	X	X	X
df(1)	X	X	X	X
dhclient.conf(5)	X	X	X	X
dhclient.leases(5)	X	X	X	X
dhclient(8)	X	X	X	X
dig(1)	X	X	X	X
dmesg(8)	X	X	X	X
domainname(1)	X	X	X	X
du(1)	X	X	X	X
dump(8)	X	X	X	X
egrep(1)	X	X	X	X
env(1)	X	X	X	X
environ(7)	X	X	X	X
etcupdate(8)			X	
exports(5)	X	X	X	X
fetch(1)	X	X		
fgrep(1)	X	X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
file(1)	X	X	X	X
find(1)	X	X	X	X
finger(1)	X	X	X	X
firewall(7)	X	X		
fsck(8)	X	X	X	X
fsdb(8)	X	X	X	X
fstab(5)	X	X	X	X
fstat(1)	X	X	X	X
ftp(1)	X	X	X	X
ftpusers(5)			X	
gbde(4)(8)		X		
getfacl(1)		X		
gettytab(5)	X	X	X	X
gmirror(8)		X		
graid3(8)		X		
grep(1)	X	X	X	X
groups(1)	X	X	X	X
gstripe(8)		X		
hier(7)	X	X	X	X
host(1)	X	X	X	X
hostname.if(5)				X
hostname(1)	X	X	X	X
host.conf(5)	X			
hosts(5)	X	X	X	X
id(1)	X	X	X	X
ifaliases(5)			X	
ifconfig(8)	X	X	X	X
inetd(8)	X	X	X	X
info(1)	X	X	X	X
init(8)	X	X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
intro(1) to intro(9)	X	X	X	X
iostat(8)	X	X	X	X
ipf(8)	X	X	X	
ipfstat(8)	X	X	X	
ipfw(8)	X	X		
jail(8)	X	X		
kill(1)	X	X	X	X
killall(1)	X	X		
kldload(8)	X	X		
kldstat(8)	X	X		
kldunload(8)	X	X		
krb5.conf(5)	X	X	X	X
last(1)	X	X	X	X
lastcomm(1)	X	X	X	X
lastlog(5)	X	X	X	X
lastlogin(8)	X	X	X	
limit(1)	X	X	X	X
limits(1)	X	X	X	
lkm.conf(5)			X	
ln(1)	X	X	X	X
loader.conf(5)	X	X		
locate.conf(5)			X	
locate.updatedb(8)	X	X	X	X
locate(1)	X	X	X	X
login.conf(5)	X	X	X	X
lpc(8)	X	X	X	X
lpq(1)	X	X	X	X
lprm(1)	X	X	X	X
ls(1)	X	X	X	X
lsof(8)		P	P	P

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
mac(4)		X		
magic(5)	X	X	X	X
mail(1)	X	X	X	X
mailer.conf(5)	X	X	X	X
mailq(1) or (8)	X	X	X	X
make.conf(5)	X	X		
make(1)	X	X	X	X
make buildworld	X	X	X	X
make installworld	X	X	X	X
make replace	X		X	
make show-downlevel	X		X	
make update	X		X	
make quickworld	X			
man.conf(5)			X	X
man(1)	X	X	X	X
master.cf (postfix)		P	X	P
md5(1)	X	X	X	X
mergemaster(8)	X	X		P
mk.conf(5)			X	X
modload(8)			X	X
modstat(8)			X	X
modunload(8)			X	X
monthly.conf(5)			X	
monthly(8)				X
motd(5)	X	X	X	X
mount(8)	X	X	X	X
mountd(8)	X	X	X	X
mount_nfs(8)	X	X	X	X
mtree(8)	X	X	X	X
mygate(5)			*	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
myname(5)			*	X
nc(1) (netcat)	P	X	P	X
ndp(8)	X	X	X	X
netstart(8)				X
netstat(1)	X	X	X	X
newaliases(1) or (8)	X	X	X	X
newsyslog.conf(5)	*	X	X	X
newsyslog(8)	X	X	X	X
nfsd(8)	X	X	X	X
nfsstat(1)	X	X	X	X
nice(1)	X	X	X	X
nmap(1)	P	P	P	P
nologin(8)	X	X	X	X
nslookup(1) or (8)	X	X	X	X
nsswitch.conf(5)	X	X	X	
ntpd.conf(5)	X	X	X	X
ntpd(8)	X	X	X	X
openssl(1)	X	X	X	X
packages(7) or ports(7)		X	X	
passwd.conf(5)			X	
passwd(1) and (5)	X	X	X	X
patch(1)	X	X	X	X
pax(1)	X	X	X	X
pciconf(8)	X	X		
periodic.conf(5)	X	X		
periodic(8)	X	X		
pf(4)	X	X	X	X
pfctl(8)	X	X	X	X
pgrep(1)	X	X	X	X
ping(8)	X	X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
ping6(8)	X	X	X	X
pkg_add(1)	X	X	X	X
pkg_chk	P		P	
pkg_comp	P		P	
pkg_delete(1)	X	X	X	X
pkg_info(1)	X	X	X	X
pkgtools.conf(5)		X		
pkg_version(1)	X	X		
pkill(1)	X	X	X	X
portaudit(1)		P		
portmap(8)	X	X		X
portupgrade(1)		P		
postalias(1) (postfix)		P	X	P
postinstall(8)			X	
postqueue(1) (postfix)		P	X	P
printcap(5)	X	X	X	X
ps(1)	X	X	X	X
pstat(8)	X	X	X	X
pw(8)	X	X		
pwd_mkdb(8)	X	X	X	X
raidctl(8)			X	X
rc.conf(5) or (8)	X	X	X	X
rc(8)	X	X	X	X
rdate(8)	X	P	X	X
re_format(7)	X	X	X	X
renice(1) or (8)	X	X	X	X
resolv.conf(5)	X	X	X	X
restore(8)	X	X	X	X
rm(1)	X	X	X	X
rmuser(8)	X	X		X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
route(8)	X	X	X	X
rpc.lockd(8)	X	X	X	X
rpc.statd(8)	X	X	X	
rpcbind(8)		X	X	
rtsol(8)	X	X	X	X
sa(8)	X	X	X	X
scsi(4) or (8)		X	X	X
scsictl(8)			X	
security.conf(5)			X	
security(7) or (8)	X	X		X
sendmail.cf	X	X	X	X
services(5)	X	X	X	X
sh(1)	X	X	X	X
sha1(1)	X	X	X	X
share/doc	X	X	X	X
share/examples	X	X	X	
shutdown(8)	X	X	X	X
sockstat(1)	X	X	X	
sort(1)	X	X	X	X
src/BUILDING			X	
src/UPDATING		X	X	
ssh-keygen(1)	X	X	X	X
sshd_config(5)	X	X	X	X
sshd(8)	X	X	X	X
stat(1) or stat(2)	X	X	X	X
su(1)	X	X	X	X
sudo(8)	P	P	P	X
sudoedit(8) (sudo)	P	P	P	X
sudoers(5) (sudo)	P	P	P	X
swapctl(8)		X	X	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
swapinfo(8)	X	X		
sysctl.conf(5)	X	X	X	X
sysctl(8)	X	X	X	X
sysinstall(8)		X		
syslog.conf(5)	X	X	X	X
systat(1)	X	X	X	X
systrace(1)			X	X
tail(1)	X	X	X	X
tar(1)	X	X	X	X
tcsch(1)	X	X	P	P
tee(1)	X	X	X	X
telnet(1)	X	X	X	X
top(1)	X	X	X	X
traceroute(8)	X	X	X	X
ttys(5)	X	X	X	X
umask(1) or (2)	X	X	X	X
umount(8)	X	X	X	X
uptime(1)	X	X	X	X
user(8)			X	X
useradd(8)			X	X
userdel(8)			X	X
usermgmt.conf(5)			X	X
usermod(8)			X	X
users(1)	X	X	X	X
utmp(5)	X	X	X	X
verixecctl(8)			X	
vi(1)	X	X	X	X
vinum(8)	X	X		
vipw(8)	X	X	X	X
visudo(5) (sudo)	P	P	P	X

<i>Command or File</i>	<i>Dragonfly BSD</i>	<i>FreeBSD</i>	<i>NetBSD</i>	<i>OpenBSD</i>
vmstat(1) or (8)	X	X	X	X
vnd(4)			X	X
vuxml		P		
w(1)	X	X	X	X
weekly.conf(5)			X	
weekly(8)				X
whatis(1)	X	X	X	X
whereis(1)	X	X	X	X
which(1)	X	X	X	X
who(1)	X	X	X	X
whoami(1)	X	X	X	X
whois(1)	X	X	X	X
wtmp(5)	X	X	X	X
xargs(1)	X	X	X	X
Xen			P	
zmore(1)	X	X	X	X