Global Information Assurance Certification Paper

Security Hardening Techniques for HP-UX Core O/S

Jeffrey A. Malacko

October 31, 2004

SANS Security Essentials GSEC Practical Assignment
Version 1.4c
Option 1

Abstract

Computers and network elements connected to networks are vulnerable to any number of attacks, including (Nortel Networks, 2003):

1. Backdoor programs
2. Sniffing programs
3. Password grabber and cracking tools
4. Exploitation of defects in operating system services
5. Denial of service

Some of these attacks are based on well-publicized techniques, with scripts and other tools available to make it possible for less knowledgeable crackers to apply exploits against systems. Once a system has been compromised, an intruder can do a number of things, among which are (Nortel Networks, 2003):

1. Modify or destroy information
2. Disclose sensitive information
3. Install malicious code to gather information
4. Use the compromised server to attack other systems

The process of installing a computer into a production environment, whether a server or workstation, involves various steps regarding both hardware and software in which one step that is a critical component is installing the operating system. The operating system is the software foundation for which other services and applications are layered upon to ultimately deem it operational. Unfortunately, operating system vendors usually provide or sell an “out of the box” solution that is lacking bug fixes, security updates, and enablement functions. A big problem facing companies is the lack of knowledge to enhance security for the operating system beyond the base install before promoting a system into production. More often than not, once in production, the system is on the network and available for both internal and external attacks with little or no security measures taken into account from the operating system perspective. The system may service the end user just fine, but from a security viewpoint, it is vulnerable against various attacks. Hopefully, once support personnel realize this, measures can be taken to secure and harden the system to reduce and eliminate vulnerabilities that exist through change management. However, it is more desirable to “harden” the system prior to being placed on the network and promoted to production.

This paper will discuss common security hardening tools, techniques, recommendations, and methods that are used for HP-UX, Hewlett-Packard’s UNIX based operating system. The foundation is the HP-UX Core O/S Version 11i or, in other words, the out of the box release. The key word here is “common” since the recommendations will apply in a fairly generic fashion. Although the

main focus is on HP-UX, some portions are applicable to other operating systems as well.

This paper is not intended as a guide for setting up a bastion host, although many of the considerations are also used for setting up one. In addition, this is not a step-by-step guide as the detail level is more medium to high level. While no system is 100% secure, an increase in confidence by following these guidelines will result in systems that are harder for hackers to compromise. Ongoing vigilance is required to keep systems secure.

Installation Overview

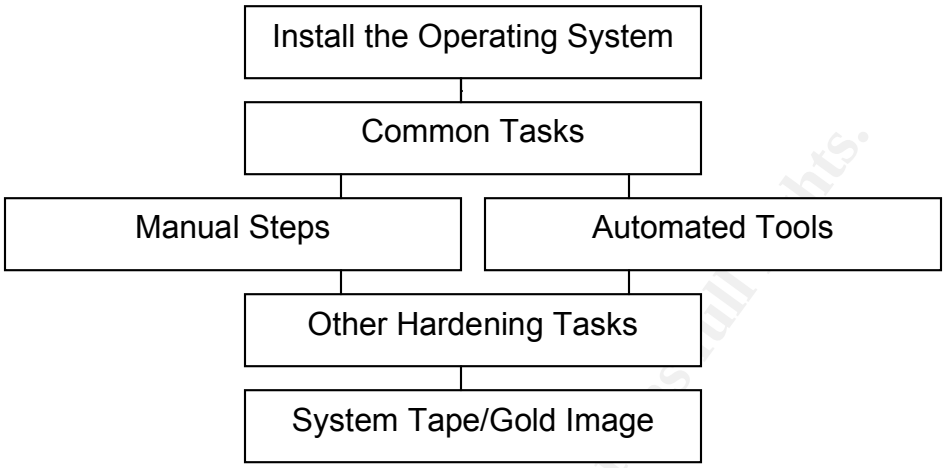
What you don't know can hurt you, and what you think that you might know you may not trust. Thus a cautious approach should be taken when installing the operating system which should come from a known good source. So, given this, the installation has to have a plan based on a sequential step by step approach. The following is a high level plan that gives a roadmap to accomplish this (Steves, 2002):

1. Install HP-UX 11i
2. Install Additional Products
3. Install the Support Plus Bundle
4. Install Security Patches
5. Install Software and Test Configuration
6. Re-install application patches for the newly installed applications
7. First Steps
8. Disable Network Services
9. Disable Other Daemons
10. Examine Set-id Programs
11. Examine File Permissions
12. Security Network Tuning
13. Create a System Recovery Tape

Keep in mind that this is a sample starting configuration, and you will need to make changes specific to your planned use of the system. If you're installing a future HP-UX version like 11.31, some things may be different. You may also choose to reorder things slightly for various reasons. Document your configuration steps as their being performed because you may discover later that the change made caused unforeseen problems. It may take several install iterations to get everything working correctly as well as multiple reboots.

The following simplified plan leads through hardening an HP-UX system and offers two possible paths to follow: manual or automated. Usually, a combination of the two paths is used for the best results. The recommendations discussed will

concentrate on both select manual methods and briefly on the automated. The following depicts the flow:



Install the Operating System

It is important to start with a clean slate when installing the operating system. It should be installed from a known good source which in this case would be CD media from Hewlett-Packard. Too many times the operating system is installed by cloning another system that has been in production for quite a while which really deems this type of source unreliable since it is not truly known what has and has not been done to the system. There is a chance that a virus or vulnerability could be propagated to one or many servers if this approach is taken.

It takes a little over one hour to install a minimal HP-UX 11i configuration from CD media. The security benefit of starting with a clean operating system install, and knowing exactly what the state is, far exceed the minor cost in time when using the cloning approach. Of course, the main benefit of cloning systems is the time savings since patches, kernel settings, drivers, and other configurations are carried over. However, with good procedures and documented practices to build the system, security will prevail over time. Even if the system is new and was shipped from the factory integration center with HP-UX 11i preinstalled, it is still a good idea to reinstall from scratch. In addition, during the initial installation and configuration, make sure the system is not connected to any network, whether it is trusted or possibly not. It is a good practice to only connect the system to a network after the configuration steps have been completed.

After the final reboot the base operating system has now been installed, it is a good idea to remove any unneeded software with the swremove command. It will depend on what the systems purpose is to decide what products are candidates for removal and could vary from other machines previously installed. As an example, if it is known that the system will not be a web server or other applications do not have a dependency for it, remove the Apache software that gets installed with the operating system. For some application vendors, it is common practice to bundle Apache with their product which uses a non-well known port. By default when Apache Web Server is installed on HP-UX 11i, port 80 is used which is fairly common knowledge and opens up a security concern in itself not to mention the many vulnerabilities that can exist in the Apache application. Another example is the need to disable sendmail, assuming the system is not a mail server, as there are a lot of vulnerabilities associated with it and the port 25 it listens on. Although this is just one example of the need to remove a software product from a security standpoint, there are many other products that get installed by default and should be removed when they will not be needed for the function of the system and thus will be one less thing to worry about when it gets promoted to production.

On the inverse, it may be necessary to install or upgrade additional applications that were not included with the operating system install. It is important to understand that whether the added software is supplied by Hewlett-Packard or another third party vendor, vulnerabilities may exist. The version selected and maintainability issues will have to be considered so the application is installed in a secure manner.

Once the operating system is installed and software has been removed and added, it is critical to patch the system from not only a functional aspect but a security perspective as well. Hewlett-Packard has various strategies for patching their system and applications. Some customers may take a conservative approach where patching is done per the rating and age of each individual patch or entire patch bundle, others may take an aggressive approach in which the latest is always installed. One of the most common approaches used is a combination of the two where pre-tested general patch bundles are installed and then, by using a tool supplied by HP called security_patch_check, security patches are identified by referencing a catalog. A security patch bundle can be developed from the list that is compiled by the tool (HP Software Portal, 2004). After the operating system, applications, and security vulnerability are patched, another tool supplied by HP called check_patches can be run to check the validity and checksums of all the patches that are installed on the system. After a clean check_patches output is obtained, it is important to commit all of the patches that have been superceded to prevent them from being available for use in attacks. The commands show_patches and show_patches –s lists active and superceded patches installed on a system respectively. For applications that are installed on the system that are from a vendor other than Hewlett-Packard, it is important to contact the support channels to obtain the patches and procedures

necessary to patch those applications as well. In addition, the application vendor may recommend operating system patches that enhance their application from a functional and security point of view in which a vendor specific patch bundle may be created.

Common Tasks

It is crucial to examine the security bulletins that affect HP-UX 11i on a regular basis, especially since not all bulletins result in a patch. Through a subscription service to the HP Security Bulletin mailing list, a notification of all HP-UX related advisories can be received via email on an ongoing basis (HP-UX Security Bulletin Listings, 2004). It is also valuable to subscribe to other security bulletin advisory sites like the CERT Coordination Center (Software Engineering Institute, 2004) or the SANS @Risk Consensus Security Vulnerability Alert (SANS Institute, 2004).

To obtain C2 level security, an HP-UX 11i system can be converted to a trusted system. Some of the enhanced security features are password aging, password generation, access time restrictions, serial port restrictions, terminal restrictions, auditing, enhanced login configuration, support for long passwords, protected password database, and the ability to require a password at single-user mode (Klein, 2004). When a trusted system is implemented, the encrypted passwords are removed from the /etc/passwd file and relocated in a series of files that are only readable by root.

Although not directly related to security hardening, enabling system accounting and kernel level auditing can be a true ally. System accounting gathers baseline system data such as CPU utilization and disk I/O every 20 minutes and archives this data for one month or alternatively administrators may archive the /var/adm/sa directory on a monthly basis to preserve this data for longer periods. The data may be accessed for reference when needed. To enable accounting, modify the /etc/rc.config.d/acct file to contain the following line: START_ACCT=1. Kernel-level auditing provides information on commands and system calls that are executed on the system. The audit trail may be reviewed with the audisp command. Note that auditing only works if the system has been converted to a trusted system and the AUDITING variable is set to 1 in the configuration file /etc/rc.config.d/auditing.

The /etc/default/security file is used to customize the system defaults. Some of the following entries that can be added to the security file to alter defaults are ABORT_LOGIN_ON_MISSING_HOMEDIR=1 if no home directory exists the login will fail, MIN_PASSWORD_LENGTH=8 where 8 is the minimum password length, NOLOGIN=1 if /etc/nologin exists all logons are disabled, PASSWORD_HISTORY_DEPTH=5 the system will remember the last 5 passwords and can't be re-used, SU_ROOT_GROUP=subgroup this group is the

only one allowed to run su to root (Wong, 2002). Beware that a good user-management is required for this, SU_DEFAULT_PATH=/usr/sbin:/usr/bin:/sbin this path is applied to all non-root users that switch user to the root id. There are other customizable options for the security file and can be referenced by accessing the manual page.

After the standard operating system is installed, the root user id has the file system root directory / as its home directory. For security reasons, this has to be moved to another location. If no further constraints exist, this should be /root or /home/root for COE systems. It is important that the root id's home directory should be on the "/" logical volume. After the root id's home directory is moved, the location should be reflected in the password file as well.

Automated Tools

HP-UX Bastille

HP-UX Bastille is a security hardening and lockdown tool which can be used to enhance the security of the HP-UX 11i operating system. It provides customized lockdown on a system by answering a series of questions which in turn will make modifications similar to a manual procedure using a checklist. The key features and benefits of Bastille are that it configures daemons and system settings to be more secure, turns off unneeded services, helps create chroot jails that partially limit the vulnerability of common Internet services such as web servers and DNS, configures the conversion to trusted systems, configures the security_patch_check tool to run automatically, the user interface is designed to educate users, and the ability to return the security configuration to the state before Bastille was run with a revert switch (Bastille Software, 2004). This revert switch has a big advantage over manual methods when it comes time to backing out the changes since Bastille creates and stores a file called action-log that contains the modifications that were made. This can eliminate human error and be a huge time savings for performing a back out plan.

Bastille can be used two ways, interactively through an X-Windows user interface, or non-interactively. If used interactively, the user interface has been designed to educate users by guiding them through a series of questions. Each question explains a security issue and describes the resulting action needed to lockdown the HP-UX machine. Each question describes, at a high-level, the cost/benefit of each decision, and then it lets the user decide how the tool should handle the issues. After answering all of the questions, Bastille provides automated support in performing each lockdown step. It performs the actions it can automatically perform, and then it produces a list of the remaining manual actions the administrator must perform. These actions must be performed to complete Bastille's lockdown process. Used non-interactively, the configuration engine is used directly without the aid of an interface. This makes the tool useful

for duplicating a particular security configuration onto multiple systems (Bastille Software, 2004).

Install Time Security Considerations

Beginning with HP-UX 11i Version 2, there is a choice of four predefined security configuration bundles that can be run with HP-UX Bastille to lock down a system. Each bundle represents a different level of security that can be applied. The four bundles are Sec00Tools, Sec10Host, Sec20MngDMZ, and Sec30DMZ with each bundle providing incrementally higher security respectively. The Sec00Tools security bundle is the default that is installed on the system and does not implement any security changes but checks to make sure that the required software is installed. The required software products are Perl, HP-UX Bastille, IPFilter, Security Patch Check, OpenSSL, and HP-UX Secure Shell. The Sec10Host Security Bundle performs a host based lockdown and disables some common clear text services but excludes firewall configuration and leaves telnet and ftp enabled. The Sec20MngDMZ Security Bundle performs a lockdown with secure management utilizing IPFilter firewall to block all incoming connections except common secured management protocols. The Sec30DMZ Security Bundle performs a network DMZ lockdown where IPFilter is configured to block all incoming connections except HP-UX Secure Shell to ensure network traffic packets are encrypted (HP-UX 11i v2 Install Guide, 2004).

Manual Steps

Although automated methods are convenient and robust, they are not programmed to account for all security hardening considerations. Depending on an organization's security policy, hardening may only consist of using automated tools; others may prefer to use manual configuration which at least equals and can go beyond automated, and some may deploy a combination. Some of the steps discussed here can only be accomplished manually and others can be implemented either manually or with an automated tool such as HP-UX Bastille or tsconvert to convert to a trusted system, but the most common security considerations will be covered.

System security can be enhanced by enabling a feature that prevents stack execution. A common method of breaking into systems is by maliciously overflowing buffers on a program's stack. Malicious unprivileged users often use this method to trick a privileged program into starting a super user root shell or similar unauthorized actions. By setting the kernel tunable parameter executable_stack to 0, systems can be configured to execute protect program stacks, providing significant protection from many common buffer overflow attacks. In most cases, enabling this feature will not affect compatibility of any legitimate applications or degrade performance. A setting of 1 causes all program

stacks to be executable, and is safest from a compatibility perspective but is the least secure setting for this parameter. A setting of 2 is equivalent to 0, except that it gives non-fatal warnings instead of terminating a process that is trying to execute from its stack. Using this setting is helpful for gaining confidence that using a value of 0 will not affect applications (Wong, 2002), (June 2004 Release Notes, 2004).

In some cases it is possible to disable the Internet Services daemon inetd. In most cases though, it will be required since the services offered are needed by certain applications. A good approach to take is to identify each TCP and UDP process that are not needed or cannot be secured should be disabled. By using the netstat command and inetd logging, processes that are not being used can be identified for being turned off. An alternate method is to start with an empty inetd.conf file, which some bastion hosts have, and determine and enable which services are needed per the applications the system will be running. If the inetd.conf file ends up empty, then it is best not to run inetd at all.

Many ARPA and Berkeley services should not be required on a hardened system or can be substituted for a more secure alternative. In some configurations an ftpd is required. In this case a better alternative is to run the wu-ftpd special release (HP Software Portal, 2004). Unless absolutely needed, the telnet service should not be run on a system as it can be an open door to hackers. The tftp service is required for an Ignite-UX server and frequently used by network appliances to up and down load configurations. Any other situation should disable trivial ftp and any lines associated with it as the entry is usually broken into multiple lines. Other services that are good candidates for disablement are bootps for dhcp and remotely booting diskless clients, finger for the remote finger service, login for rlogin services, shell for remsh services, and exec for running rexec's on the local system. The services uucp and ntalk are hardly needed anymore per application requirements, and ident should be looked at to verify if another system needs to know identification information of the local host. As for the printer service, a system usually does not need to receive print jobs and act as a print-server and therefore should be disabled if this is the case (Burton, 2004).

The inetd internal services are old default services and there is no valid reason to have these enabled on the system. The following should all be disabled for both udp and tcp: daytime, time, echo, discard, and chargen. If Kerberos ticketing systems for logins are not used turn the kshell and klogin services off. Not too many environments really use kerberos or have an MIT-Kerberos based system (Klein, 2004).

If a system does not need to run a CDE graphical user interface for remote logon and X-windows is required, ssh should be used to receive the X environment. The CDE related services dtspcd, rpc.ttdbserver, and recserv should be disabled.

The instl_boots is the service for remote booting off an ignite server, therefore if a system is not one, this service should be turned off. The CIFS service swat is the web front end to samba used for administration. If it has to be run, run it reduced to the local host only within the inetd.sec file. In many cases, samba is configured manually and swat can be disabled permanently. The calendar manager service rpc.cmsd is very rarely used and is a good practice to disable it as it shows up quite often on security scan reports. Finally, once the inetd.conf file is set per the security policy and application requirements, run inetd –c to commit the changes which will be logged to the syslog.log file for verification and reference. Lastly, if inetd will remain enabled, inetd logging should be turned on by adding –l (minus lower case ell) to the INETD_ARGS environment variable in the netdaemons configuration file (Indiana University, 2004).

One of the changes that take place when converting to a trusted system is that the default umask of user id 0 is changed to 07077 which is the most secure setting for this. If the system was not converted to a trusted system, then the umask needs to be tightened up by setting it to 07022 in the /etc/profile file and to 07077 in user root's .profile. Both of these settings have an affect on file and directory permissions when they are created.

The /etc/securetty file is used by the login command and contains the device names on which root are allowed to login. It is recommended to only allow root to logon to the console. By adding the word "console" to this file and setting the permissions so it is readable only by the root id, logon to the system by root is restricted to the console only.

If the system will allow the file transfer protocol daemon to be enabled, create an FTP user restriction file called /etc/ftpd/ftpusers where the permissions are root readable and writeable only. This file contains a list of users who are not allowed to access the system via FTP. Generally, only normal users should ever access the system via FTP and there should be no reason for system accounts to be transferring information via this protocol. The root account should never be allowed to transfer files directly via FTP. In addition, more fine grained FTP access controls can be placed in the /etc/ftpd/ftpaccess file.

Other Hardening Considerations

If a system contains sensitive data or is internet visible it is advised to consider examining set-id programs as well as a performing audits on a regular basis, especially after patching. Many UNIX systems, including HP-UX, are initially installed with numerous programs that are set-uid or set-gid. Many of these programs are not used or are only used by the root user. Many of the vulnerabilities that are discovered in UNIX utilities rely on the set-uid root bit to raise privilege. The security of a system can be improved by removing these programs or by removing the set-id bit. The find command can be used to obtain

a list, which can be lengthy, of all files with either the set-uid or set-gid bit set (Pipkin, 2003). Another strategy is to remove the set-id bits from all files, then selectively add it back to just a few programs that need to be run by non-root users. The commands selected to leave set-id depends on the specific usage and policies of the system. Additionally, a number of commands will function fine without privilege using default or commonly used options; however some functionality may be lost for non-root users. Another good tool to use related to examining file permissions is hp_checkperms from the Center for Internet Security. This is a non-invasive script that checks file permissions on a system against those in the software installed product database (CIS, 2004).

Security network tuning is one task that is often overlooked. On HP-UX 11i, the ndd command is used to perform network tuning and ndd –h produces a list of help text for each supported and unsupported parameter that can be changed. Although the number of tunable parameters is quite long with sections for IP, TCP, UDP, IPV6, and others, the following are candidates for being changed for enhancing security on a system with a fresh operating system install (HP Online Docs ndd man page, Sep 2004): ip_forward_directed_broadcasts to 0, ip_forward_src_routed to 0, ip_forwarding to 0, ip_ire_gw_probe to 0, ip_pmtu_strategy to 1, ip_send_source_quench to 0, tcp_conn_request_max to 4096, and tcp_syn_rcvd_max to 1000.

Utilizing security scanning tools to check systems security robustness is a necessity today. As a result, the number and quality of scanners on the market has increased significantly. There are many commercial and free products available to perform scans that are host based, web based, or network centric. Popular host based scanners that are available for free are HP-UX Bastille and Cops. Popular network scanners that are available for free are Nessus and Nmap (Insecure.org, 2004). Host based scanners examine local configuration files to identify potential vulnerabilities. Network scanners scan remote host ports for vulnerable services. After a system has been hardened and configured, it is a good idea to scan the system to identify any weaknesses or vulnerabilities. The output or reports from the scanners vary, but the end result is consistent in that recommendations are given. Nessus includes are very flexible reporting mechanism in which the results can be viewed with a Graphical User Interface, web browser, or ASCII text. The HTML reports even include charts and graphs which creates a good overall presentation to be shared with management (Nessus Project, 2004).

Sophos is an anti-virus solution that is available on the HP-UX 11i operating system and runs as an on-demand virus scanner (Sophos Plc, 2004). It is strongly recommended that Sophos or an alternative be used on all systems regardless of their purpose. Virus scanning should be mandatory for any system that stores Microsoft Windows files, such as Samba/CIFS 9000 servers. Although there are few viruses that attack UNIX operating systems, HP-UX servers are often used as Windows file servers with Samba software. This means that

although HP-UX 11i servers are not vulnerable to Windows viruses, they can be virus carriers. Therefore, all HP-UX servers that store or transport Windows files must be scanned for viruses to reduce this risk.

System checks and testing is a step that, though time consuming, is crucial to ensure the build meets the overall requirements. There are many methods to performing this task, but one constant is that a defined and documented procedure should be used that covers all areas per the system security policy. A common term for this type of document is the promotion to production checklist with the goal of ensuring that the system is ready to be promoted into a production environment. The promotion to production checklist is a living document that is added to and removed from as technology advancements deem the modifications necessary.

Capture the Image for Recovery and Cloning – Gold Image

It is important to create a bootable system recovery tape of the system build. To accomplish this, the Ignite-UX product must be installed. This tape can also be used to clone the system to other hardware that is supported with the same software configuration. The make_tape_recovery command can be executed online to create the image, although it is best to have the system in a quiescent state which can be achieved in single user mode. Alternatively, the make_net_recovery command can be used to create an image on disk if an Ignite Server is available. The make_sys_image command can be used to create a transferable image that can be burned on a CD as a bootable gold image for cloning or recovering systems (Hewlett-Packard Ignite-UX Online Docs, 2004).

Conclusion

Installing a system into a production environment is an everyday occurrence. The installation and configuration of the operating system is the base of all systems and needs to be built with security in mind. The practice of ensuring that the system being installed is as secure as possible is of utmost importance given today's distributed global computing environment as well as the presence of malicious software. One thing is for sure, no system is 100% resistant to security attacks and vulnerabilities, but with solid practices and procedures, the risk of a successful attack can be minimized. As discussed in this paper, a plan should be in place to build a system to meet security policies and application requirements of an organization.

The focus of this paper was on the HP-UX 11i operating system and addressed techniques and recommendations that can be applied to harden a system for preparation in promoting it to production. By following the overall flow and applying the applicable considerations, a system being built with a known good

source operating system will be much more secure than an out of the box or factory install and gives the best chance to defend against security vulnerability and attacks.

References

1. "HP-UX 11 Operating System Hardening Guideline Document", Nortel Networks, November 2003. URL:http://www.nortelnetworks.com/solutions/securenet/collateral/hp-ux_hardening_guide_v1.pdf

2. Steves, Kevin. "Building a Bastion Host Using HP-UX 11", 16 October 2002. URL:http://secinf.net/unix_security/Building_a_Bastion_Host_Using_HPUX_11.html

3. Wong, Chris. "hp-ux 11i security", Prentice Hall, 2002.

4. "HP-UX Security Bulletins Listing", Hewlett-Packard Company, 2004. URL:http://itrc.hp.com/

5. Burton, Doug. "/etc/inetd.conf Notes", 01 March 2004. URL: http://home.tampabay.rr.com/batcave/inetd_conf.htm

6. "CERT/CC Advisories and Incident Notes", Software Engineering Institute, 2004. URL: http://www.cert.org/

7. "@Risk: The Consensus Security Vulnerability Alert", SANS Institute, 2004. URL: http://www.sans.org/

8. D. Paul, Klein. "Practical Unix and Network Security", Hewlett-Packard Development Company, L.P., 2004

9. "HP Online Documentation", Hewlett-Packard Development Company, L.P. URL: http://docs.hp.com/

10. "HP-UX Bastille Software Depot", Hewlett-Packard Company, 2004. URL:http://software.hp.com/portal/swdepot/displayProductInfo.do?productNumber=B6849AA

11. "HP-UX 11i Version 2 Installation and Update Guide", Hewlett-Packard Development Company, L.P., September 2004, Edition 3. URL: http://docs.hp.com/hpux/onlinedocs/5990-8144/5990-8144.html

12. "HP-UX 11i June 2004 Release Notes", Hewlett-Packard Development Company, L.P., June 2004, Edition 13. URL: http://docs.hp.com/hpux/onlinedocs/5990-7288/5990-7288.html

13. "Security and Manageability Product List", Hewlett-Packard Software Portal, October 2004.

URL: http://software.hp.com/ISS_products_list.html

14. "Understanding inetd", Indiana University Unix System Support Group, 2004.
URL: http://www.uwsg.iu.edu/usail/network/services/inetd.html

15. "CIS Level-1 Benchmark and Scoring Tool for HP-UX", the Center for Internet Security, October 2004.
URL: http://www.cisecurity.org/bench_hpux.html

16. "Ignite-UX Administration Guide", Hewlett-Packard Development Company, L.P., September 2004, Edition 17.
URL: http://docs.hp.com/hpux/pdf/B2355-90849.pdf

17. "Nmap Network Exploration and Security Auditing", Insecure.org, 2004.
URL: http://www.insecure.org/nmap

18. "Nessus Remote Security Scanner", The Nessus Project, 2004.
URL: http://www.nessus.org/

19. Pipkin, Donald L. "Halting the Hacker", Prentice Hall, 2003.

20. "Sophos Anti-Virus", SOPHOS Plc. Software, 2004.
URL: http://www.sophos.com/