Security and the e-smith server and gateway

A technical white paper

by Dan York, e-smith, inc.

1. Introduction

This white paper discusses security as it relates to the e-smith server and gateway version 4.1.*x*. It is aimed at system administrators and those with an interest in the technical aspects of system and network security.

While no product can be 100% secure, we have configured the e-smith server and gateway to provide the highest possible level of security "out of the box". We recognize that security is important in businesses of all sizes and we want to ensure that your system and network are protected. In order to do so we take actions such as disabling non-essential services, configuring services to operate in their most secure mode, automatically providing packet filtering, providing encrypted means of remote access and taking other steps that are part of an effective firewall solution.

Beyond these steps, we strive to automate *"best practices"* in Linux system administration so that tasks that an expert system administrator would normally do are done automatically for you. Regardless of your experience level, this automation relieves a great deal of pain that is normally associated with security and system administration and also reduces the possibility of accidental mis-configuration.

In this document, we will discuss the major components of the e-smith server and gateway as they relate to security and the technical steps that we have taken to ensure the integrity and security of the system.

2. The Role of the e-smith server and gateway

The e-smith server and gateway builds on the widely acknowledged reliability of the Linux platform, and addresses the complexity problem through a very simple installation process. Once installed, the e-smith server and gateway provides security and a rich set of Internet services including e-mail, web hosting, webmail, secure remote access, ftp hosting, file and print-sharing and many other features. e-smith's software runs on any commodity PC, works with virtually any Internet provider, and allows customers to use their choice of desktop platform.

The e-smith server can be configured in either of two modes of operation. In *server-only* mode, the e-smith server operates as a standalone server on a local network and provides file and network services to all systems on that network. In *server and gateway* mode, the e-smith server is configured with one network connection to the local network and a second connection to the Internet. In addition to providing file and network services to the local network, it also acts as a gateway allowing the entire local network to access the Internet.

3. Elimination of Non-essential Services

The developers of a generic Linux distribution such as Red Hat do not know exactly how that computer system will be used. For that reason, they usually configure the operating system to be able to support many different types of server and workstation uses. Many services are included on the system that may not ever be used. When someone installs such a generic distribution, those services may open up security holes into that computer system.

A cardinal rule of system security is "*only run the absolute minimum number of services necessary for your operations*". With the e-smith server and gateway, we know precisely how the vast majority of users are going to use our server and therefore can simply remove all other services. Unnecessary services such as `NFS`, `NIS`, the Berkeley "r" suite of programs (such as `rsh` and `rwhod`) and many others have not been included.

Beyond services, we do not include many of the packages that are included with a standard Red Hat installation. As an example, the e-smith server and gateway does not include any of the standard compilers or development libraries, as these development tools have been used in the past to install compromised programs during network attacks. Additionally, we eliminate most of the user tools that would be used in a Linux workstation configuration. For instance, the X Window System and associated programs are not included.

Essentially, if we could not identify a *need* to include a package or service in the product that would benefit a small to mid-size business with their daily operations, we eliminated it. Through this simplicity, we are able to minimize the security risks to the system.

If you are interested in specifically what services are enabled, the following list indicates the services started in a default configuration. They are listed in the order in which they are started. A definitive listing of running processes can be found in Appendix A.

```
syslogd, klogd, crond, xinetd, ntpd, lpd, dhcpd, ldap, qmail, smtpfwdd, httpd,
sshd, mysqld, squid, atalkd, papd, afpd, smbd, nmbd, named, mingetty
```

While this list may seem lengthy, experienced Linux/UNIX system administrators will recognize the list as being small compared to a typical Linux/UNIX system.

4. Replacement of Services With More Secure Alternatives

As part of our security review of the base Linux operating system, we also evaluate services that are included in the standard Red Hat release to determine if there are more secure alternatives. For instance, we have replaced the following services from Red Hat 7.0:

- **sendmail** - Given the number of security vulnerabilities reported in sendmail over the years, we have instead opted to use `qmail` and `obtuse-smtpd`, both of which have been designed from the beginning with security in mind.
- **wu-ftpd** - Like sendmail, wu-ftpd has suffered from security flaws over the past years. We chose `proftpd` as a replacement because of its focus on security as well as our ability to more easily configure it to limit access.

These services will be discussed in more detail in other sections of this document.

5. Open Source Code Review

One of the greatest strengths of the e-smith server and gateway is that our product source code is completely open and available to anyone to examine and review. At first glance, this statement might seem counter-intuitive. If you let people see all of your code, how is it secure? The traditional view in software development has been to closely guard your product source code so that no one can examine it and find potential security holes. Why do we say that our open source process is in fact *more secure* than traditional "closed source" or "proprietary" development models?

The answer lies in the fact that our code undergoes a strenuous peer review by people working with our product. Given that anyone can look at our code, we benefit from a large and constantly growing developer community. For instance, on our e-smith developers mailing list, we have over 400 developers who are working with our product on an ongoing basis. Many more participate in our web forums and still others simply download and use our product. When someone suspects a problem, they (or someone else with the technical knowledge) can look directly at our source code and find out how we are implementing a particular item. If they see that there is an issue, they even have the option of implementing their own fix.

The value of open source peer review was dramatically illustrated in the year 2000 with the discovery that for *eight years* the Interbase database had contained a hard-coded username and password that would allow anyone knowing that user/password combination the ability to access *any* Interbase database. The developers had put this back door in the product to solve a particular authentication problem. For eight years this security vulnerability had been there. It is impossible to know how many databases might have been compromised. How was the hole found? In mid-2000, Interbase opened their source code and made it available on the Internet. A developer in Germany was looking through the source code and found this back door. He immediately alerted the product developers who quickly put out a fix and disabled this back door in new versions of the product. Had the code not been available, who knows how long this might have continued and who might have had access to the back door?

We believe, too, that the fact that our code *is* open encourages a much stronger internal code review process as well. All our staff developers know that others *will* see the code, and for that reason we need to be sure that the code is as secure and tight as possible. Our developer community is extremely active, strong, diverse and not at all hesitant to test our code and push the edges of what we say it can and should do. We welcome that scrutiny as we firmly believe that through that process our product becomes that much more secure!

6. Network Security

For a server functioning as a network gateway, the security related to the underlying basic network connection is of critical concern. We take this extremely seriously and use multiple tools and layers to restrict access. It starts with the fundamental distinction that in server and gateway mode, we have an *internal* network interface card connected to the local network and an *external* connection to the outside Internet, through either another network interface card or a dial-up modem. The internal card will allow most connections from the local network, but connections coming into the external interface are subject to very tight controls.

In this arrangement we use network address translation (NAT) to masquerade the entire internal network behind a single external IP address. In our recommended (and default) configuration, all internal systems have non-routable private IP addresses (per RFC1918) and there is therefore no possible way for a connection to be made from the external Internet to any internal machines. This allows us to concentrate all network security resources on protecting the server and external interface.

When we speak of network security, we wish to secure the server so all actions taken by the server in response to external network packets are defined, and all responses to external network packets are authorized and conform to a defined policy.

We achieve this restriction of server response to external network packets by multiple restrictive layers described below.

6.1. Packet filtering

A linux kernel packet filter is configured, using the `ipchains` command, which implements the general policy on the external interface that "all incoming packets are denied except those which are explicitly allowed". Packets which are denied may be logged (there is a configuration parameter which defines whether all, none or some of any denied packets are logged), but otherwise elicit no response from the server - they are simply discarded.

The set of explicitly allowed incoming packets is configurable. Each of the set of services offered by the server can be enabled or disabled in a configuration database. A subset of the services may be available as a public service (for example, a web server is a public service) - each of these services is optional, and can be restricted to private access (i.e. only on the local network). Whenever this configuration database is changed, the packet filter is reconfigured to enforce the changed access policy. An annotated listing of the default ipchains rules is provided in Appendix C.

6.2. Disabled Services Do Not Run

The e-smith server and gateway is configured to run only those services which are enabled in its configuration database.

6.3. Selective Address or Port Binding

Some applications offer the option to selectively bind to network interfaces. For services which are configured to offer access only to the internal network, this feature can be used to prevent any possibility of external connection to the network service program independent of the packet filter configuration. This feature is used, for example, in the SMB (Samba) daemon configuration. Listings of exactly what programs are bound to which ports in a default installation are provided in Appendices D and E.

6.4. Application-level Access Control Lists

Each network service program is configured to provide selective service based on the IP address of the originating request. Selective service configuration is mediated by three different mechanisms:

1. by use of the TCP wrappers daemon `tcpd`, which acts as a gatekeeper application, dropping the network connection without any data transfer if the request is not from a permitted address. Accesses from permitted addresses result in the TCP wrappers daemon executing the network service program (for example, the IMAP daemon). The access restrictions are specified in the files `/etc/hosts.allow` and `/etc/hosts.deny`, and these files in turn are configured to comply with the policy recorded in the e-smith configuration database. Examples of these files from a default installation are provided in Appendix F.
2. by the application itself, through use of the TCP wrappers *library*, which naturally is also used by the TCP wrappers daemon. The application uses the TCP wrappers library to evaluate access rules specified in `/etc/hosts.allow` and `/etc/hosts.deny`, and then drops the network connection without any data transfer, or provides the network service, as appropriate. An example of this class of application is `sshd`, the OpenSSH daemon, which implements the SSH protocol for secure remote access.
3. by the application itself, using application specific access control mechanisms. A further class of applications has its own set of mechanisms to restrict service availability according to originating network address of the request. These applications have their own mechanism for specifying and enforcing access restrictions. Examples of this class of application are `mysql` and `squid`.

6.5. Authentication and Authorization Mechanisms

Once the connection has passed through all these layers, it must finally be authenticated by the appropriate mechanism. In most cases this involves checking that the user does in fact have a valid user account and password. In some cases, such as `pptpd` and `sshd`, encryption initialization will also occur.

7. User Accounts, Groups and Passwords

In order for a user to access most services on the server, a user account is required. As a server administrator, you create the user accounts through the e-smith web manager. Each user account name can be up to twelve characters in length and each must be unique on the server. With an account, a user can login to receive e-mail and may also access private portions of the e-smith server by various file transfer methods.

Before a user can use their account, the administrator must first assign the user a password. *User accounts are locked out and cannot be used until you set this initial password.* Accounts without set passwords appear in red italics in the e-smith web manager. Passwords are allowed to contain upper and lowercase letters, numbers and punctuation. They are not restricted in length.

Once the initial password has been provided to the user, the user has total control over changing that password. At any time, they can use their web browser to go to `http://servername/e-smith-password`, enter their old password, and then enter a new password. The administrator does *not* have the ability to view the user's password in any form. If the user forgets their password, the administrator cannot retrieve that password. Instead, the administrator can *reset* the password and communicate that new password to the user, but they can never see the actual passwords used by the users.

Once user accounts have been created, they can be put into *group* accounts to ease administration issues. The group can in turn be assigned specific permissions. For instance, the ability to load files into an information bay can be restricted to members of a certain group. Users can be members of multiple groups.

Note that as described in the next section, only one user, `root`, is configured by default to be able to login to the e-smith server and access the Linux shell prompt.

8. Remote Access

In any network configuration, users typically want access to the network from remote locations. Examples may include employees who also want to work from home or who are traveling and want to connect from hotel rooms. The local system integrator or e-smith partner who installed your system may also wish to connect to your network. The challenge, of course, is to allow this access while ensuring that your system is secure.

To meet this need, e-smith allows several forms of remote access. The services, such as e-mail access, the web server, ftp, etc. can be configured individually to allow private (local network only) or public (entire Internet) access. In the case of some of those services, you may even allow public access but require a password. These services will be discussed in more detail below.

However, there are three services specifically targeted at allowing remote login *to the server* (ssh and telnet) and *to the network* (PPTP). These will be covered in this section. It should be noted that all of these services are *disabled* by default and a server administrator must specifically enable them for such access to occur.

If either of the services that allow remote login to the server are enabled, by default only one account is able to login to the server. Logging in as the `admin` user will bring you to the e-smith management console. For both ssh and telnet, the server administrator also has the option to *allow administrative access*. While this feature is *disabled* by default, enabling it will allow the `root` user to login and access the standard Linux command prompt.

By default, no other user accounts are configured to allow a login to the server. If you wish for a user account to be able to login to the server remotely, you must first login as the root user and then change the user account `shell` to be `/bin/bash` using the `chsh` command. Without this change, the user account will not be able to login to the server using either ssh or telnet.

8.1. Secure Shell (ssh)

The `secure shell (ssh)` command provides a secure, encrypted method of communication between a client and server. Unlike telnet, passwords are encrypted in transit and a secure session key is used to encrypt all packets sent between the client and the server. `ssh` and its companion program `scp` (secure copy) are available for Linux/UNIX, Windows, Macintosh and other client operating systems. In its simplest form, use of ssh merely involves initiating the command and entering the user account password. The user will then see a Linux command prompt and can begin entering Linux commands.

The implementation of ssh used by e-smith is OpenSSH available from http://www.openssh.com/. It supports both the SSH1 and SSH2 protocols as well as both DSA and RSA authentication.

When ssh access is enabled, a user may connect and enter their user account password to gain access to the system. The only issue in this default configuration is that if they do not provide the correct password, they are offered another chance to enter the password. They may continue to do so, which does allow for someone trying to crack into the system to sit there and simply keep trying to guess passwords. Note that there is a time delay between login prompts that makes brute force attacks on good passwords impractical.

To get around this issue, for systems where security is very high yet secure remote access is desired, the e-smith server also supports ssh using *RSA authentication.* In this mode, an ssh key is generated on the client computer system and then added to the list of allowed keys on the server. Only users connecting from a system with an authorized key will be allowed to login to the system.

Note that all ssh access is *disabled* by default.

8.2. Telnet

Because telnet has traditionally been used for remote access to Linux systems, we have included telnet access. However, it is *disabled by default*. The primary security problem with telnet is that it transmits all user names and passwords over the network **without any form of encryption**. Someone operating "packet sniffing" software and connected to any network between your e-smith server and the remote machine may be able to intercept and read any user names and passwords and thus could gain access to your system. For that reason we **strongly** discourage the use of telnet and encourage users to use `ssh` instead.

Within the e-smith web manager, telnet access can be restricted to only the local network or allowed from the entire network. Administrative access (the ability to telnet in as ’root’ or ’admin’) is disabled by default and can be enabled for the local network or the Internet. Again, because of the security implications, this is **strongly** discouraged.

8.3. Point-to-Point Tunnelling Protocol (PPTP)

While ssh meets many remote access needs, many users simply want to connect to their remote network and then access e-mail or use their file manager (such as the Windows Explorer or Network Neigborhood) to view and access files across the network.

To do this, the e-smith server and gateway provides network access using the *Point-to-Point Tunnelling Protocol (PPTP)*. PPTP allows you to provide a *virtual private network* between your remote client computer and your e-smith server and internal network. It creates a secure encrypted channel between your client and the server. As far as the server is concerned, the client computer appears to be on the local *internal* network and can access all resources that the user would normally see if they were connected to the internal LAN.

When PPTP was first introduced by Microsoft, many implementations suffered from poor security and the protocol gained a reputation of being insecure. Since that time, the quality of the protocol definition and its implementations has improved dramatically and it now does offer a reliable and highly secure connection.

PPTP clients are available for Microsoft Windows and may already be installed in recent versions of Windows. Users typically launch a PPTP connection by double-clicking an icon on their desktop and then entering a password.

Because PPTP allows a remote client to appear as if they are a local user and can therefore access anything on the local network, the security of the passwords in transit is paramount. For that reason, we require client systems to use *128-bit encryption.* The e-smith server will *not* accept connections from PPTP clients that use 40-bit encryption. This may require an upgrade to the PPTP component of some of your client systems.

Like ssh and telnet, PPTP access is specifically *disabled* and must be enabled through the e-smith web manager. At the time you enable access, you may configure how many PPTP clients you will allow to access the system at any given time.

9. File Transfer

One of the main reasons users want remote access is to be able to retrieve or use files located on the server. The e-smith server and gateway allows users to access files via the FTP protocol or through standard Windows and Macintosh networking.

9.1. File Transfer Protocol (FTP)

To allow users to upload or download files to and from the e-smith server and gateway, we provide FTP access. The specific FTP server we use is `proftpd` (http://www.proftpd.net/), configured with the latest security updates and patches.

In the default configuration, ftp access for users is allowed on the local (internal) network, but not from the external network. A user must login with a valid user name and password combination in order to access files.

Through the e-smith web manager, it is possible to configure the server to allow *public (external)* FTP access for users. We **strongly** discourage this action, though, because `ftp`, like `telnet`, transmits passwords from the client to the server as *clear, unencrypted text.* Instead we suggest users use the `scp` (secure copy) program provided with the `ssh` family of tools.

Anonymous ftp access is allowed on the local network for *read access only*. Information bays also may have a username and password associated with them and this information can be used for read access only. In no event can either anonymous or i-bay users *upload* information. The ability to write to the server is restricted to users with a valid user account on the e-smith server and gateway.

FTP access for i-bays is a setting that can be configured for each individual i-bay and can be configured to be allowed for either *private* or *public* access. However in the e-smith web manager Remote Access panel you have the ability to set a *FTP access policy* that will override all other FTP settings. You have the option of disabling *public (external)* FTP access and in fact completely disabling all FTP access from both the internal and external networks.

9.2. Windows Networking (SMB)

Users may access files on the e-smith server using standard Windows networking through such tools as the Windows Explorer, Network Neighborhood or My Network. They may connect to either their home directory on the server or to one of the information bay directories. Connections occur through the standard Server Message Block (SMB) protocol used within Microsoft networking.

Each user’s home directory is protected so that *only* the user may read and write to that "share". In the process of doing so, the user must enter the password for their user account on the server.

Shares for information bays are also visible in the browse list. Access to those shares is controlled by the configuration for that specific i-bay. An i-bay is assigned a group ownership and the default configuration limits read and write access only to group members. A user would therefore need to be a member of the appropriate group in order to access the i-bay.

The specific tool we use on the e-smith server to allow Windows users to access server directories via the SMB protocol is `samba` (http://www.samba.org/). The security-related portions of the default Samba configuration file are provided in Appendix G.

9.3. Macintosh Networking

For file transfer to and from Macintosh systems we use an implementation of the AppleTalk file sharing protocols called `netatalk` (http://sourceforge.net/projects/netatalk/). As with Windows networking, users must supply a valid username and password to access private directories on the server. Macintosh users connect to the folders using the regular Chooser within the Macintosh operating system.

10. E-mail

Because e-mail is mission-critical for almost all organizations, we view it as extremely important that the mail server component of the e-smith server and gateway be extremely secure. For this reason, we replaced the standard Linux `sendmail` program with the highly secure `qmail` server. qmail (http://www.qmail.org/) was designed from the very beginning with security in mind. The entire architecture of qmail is designed to minimize any possibilities for security exploits.

While qmail alone is extremely secure, we went one step further and chose a more flexible yet secure program called `obtuse-smtpd` for the mail server to which all SMTP mail connections (both outbound and inbound) are made. `obtuse-smtpd` allows us to further restrict exactly who can send in e-mail (for use in, for instance, blocking spam) and also provides hooks that we can use for filters such as content-filtering and virus scanning.

While `qmail` and `obtuse-smtpd` provide the mail server functionality, most users will be using the POP3 or IMAP protocols to *read* their e-mail. The e-smith server supports both protocols. By default, POP3/IMAP access is only available to users *on the local network*. If you wish to allow users to access their e-mail via POP3 or IMAP from remote systems outside your network, the server administrator must specifically enable such "Public" access through the e-smith web manager.

Note that under *no* circumstances are any users outside of your local network allowed to *send* mail to other external users through your mail server. The mail server (`obtuse-smtpd`) will only accept mail messages from external sources that are for *local* users. Allowing such access would open your server to abuse by spammers as a mail relay. External users who want to send mail to other external users will either need to connect to their ISP's mail server or use PPTP to make a VPN connection to the internal side of the e-smith server (at which point the e-smith server *will* accept mail for other external users because as far as it is concerned, the PPTP-connected user *is* on the local network).

11. DNS

As with most Linux systems, we use the industry standard BIND server to provide Domain Name Service (DNS) access to the local network. Because of recent security exploits related to BIND, we are constantly monitoring BIND security mailing lists and regularly update our system to have the most current and secure version of the BIND program.

We take additional steps, too, to ensure that the system is safe. For instance, the BIND daemon (called `named`) is set to run as the local user `dns` instead of the normal configuration of having named run as the `root` user. In addition to running named as an unprivileged user, named is further restricted by being forced to run in a "chrooted jail". Essentially, this means that the program has an extremely restricted view of the system, which it *thinks* is in fact the entire system. In the event that somehow the named daemon was compromised, the attacker would not be able to see anything outside of the limited area in which the named daemon is confined.

Further, DNS access is only allowed from the internal local network. Users on the external Internet are *not* able to connect to the DNS server because it is configured to listen only on the internal network.

12. Web Services

Web services on the e-smith server and gateway are provided through the open source Apache web server (http://www.apache.org/ (http://www.apache.org)). As the most highly-used web server on the Internet, Apache is constantly undergoing extensive source code peer review and has a solid record of being secure. The e-smith server supports both standard HTTP connections as well as secure HTTPS connections that use the Secure Socket Layer (SSL) protocol.

There are actually two Apache server daemons running on the e-smith server. One runs on ports 80 (HTTP) and 443 (HTTPS) and provides standard user access to the primary web information, i-bays or webmail. The second operates on port 980 and provides access to the e-smith web manager. A user can only connect to it *on the local network* and only if they know the password to the `admin` user account.

To further secure the server, the main Apache web server runs on the e-smith server as the user `www` in a severely restricted environment. The administrative Apache server runs as the user `admin` which also has a restricted shell (the e-smith console). For more information, you can view the security-related sections of the configuration file for the main Apache web server in Appendix H.

12.1. SSL Support

As mentioned above, the primary Apache server supports SSL authentication and listens on the standard port 443 for HTTPS connections. During the installation of the software, a set of 128-bit RSA keys are generated by the `openssl` command. The public RSA key is then placed in a self-signed X.509 certificate. This certificate is presented to all browsers attempting to connect via HTTPS.

The OpenSSL Project (http://www.openssl.org/) has developed an open source, commercial-grade implementation of the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. OpenSSL is very widely used throughout the Linux and BSD UNIX world and is also used on other versions of UNIX. As we have done on the e-smith server, OpenSSL is very often used in conjunction with the Apache web server.

12.2. CGI Scripts

When speaking of web server security, one of the primary concerns is often with the execution of CGI scripts on the server. On the e-smith server and gateway, CGI scripts are *disabled* by default and must be specifically enabled by the server administrator for the i-bay in which the scripts are to be executed. Once enabled, CGI scripts *can* be executed if they are placed in the `cgi-bin` directory found in each information bay or as part of the primary directory used for the main web site. As this separate directory is not viewable by regular HTTP requests, users cannot see the contents of the actual CGI scripts.

12.3. PHP

In order to allow the webmail component of our server to function, we needed to enable the PHP (http://www.php.net/) scripting module for the Apache web server. PHP is a robust language that allows people to very quickly and easily create dynamically-generated content for their web site. For example, they could create a discussion forum or a catalog that linked to a database to provide the actual data. PHP is similar conceptually to Microsoft's Active Server Pages (ASP) and provides similar functionality.

As of e-smith version 4.1.2, users may choose to enable the use of PHP in each i-bay through the *"Enable dynamic content"* checkbox in the i-bay configuration screen. PHP is *disabled* by default and must be specifically enabled by the server administrator.

While this allows users to easily add dynamic content to their web site and may be a great benefit to your users, you should be aware of an inherent weakness within server-based active content such as that of PHP. With PHP enabled, a knowledgeable user could upload a PHP script to the i-bay and then call that script to read any file that the Apache userid has access to. For instance, if a user did not have access to an i-bay called `sales`, but did have *write* access to an i-bay called `research` where PHP was enabled, the user could upload a PHP script into the research i-bay that, when called through a web browser, would open up and display files found in the sales i-bay.

As this is part of PHP's basic functionality, and is also possible through the use of the CGI scripts mentioned earlier, we recommend that you only enable dynamic content for those i-bays where either write access is restricted to the administrator or to a group of users that you trust will not upload scripts that could potentially compromise your security.

12.4. Webmail

The webmail functionality of your e-smith server is provided through the `Internet Messaging Program (IMP)` open source software package created by the Horde Project (http://www.horde.org/imp/). It is *disabled* by default and must be specifically enabled by the server administrator before users can access the server. Once enabled, the IMP software uses PHP scripts to connect to a MySQL (http://www.mysql.com/) database which starts running on the server.

When the server administrator enables webmail, it can be set to use secure connections via Secure Socket Layer (SSL) connections (commmonly called "HTTPS" access) or to allow both standard HTTP and HTTPS connections. Because user account names and passwords will be transmitted across the network or Internet, we *strongly* recommend enabling webmail in only the secure HTTPS mode. This will ensure that all communication between the client web browser and the e-smith server will be encrypted during transit.

12.5. Web Proxy Server

In addition to the standard Apache web server, the e-smith server and gateway comes automatically configured with a fully functional proxy and caching web server. The specific program we use is called `squid` (http://www.squid-cache.org/). It runs on the standard port 3128 and can be utilized by any web browsers on the internal network.

13. e-smith Console

The e-smith console program is one of the means by which someone can administer the e-smith server and gateway. In the default configuration, the e-smith console appears on the monitor of the e-smith server after the initial reboot during the installation process. It can also be activated by logging into the e-smith server locally or remotely as the user `admin`. From the e-smith console, a user can also use a text-based browser to access the e-smith web manager. Between the console and web manager, you have full access to the entire e-smith server and gateway configuration. Note that you need to know the `admin` password to access the e-smith web manager via a browser.

There is a physical security issue with regards to the console. If the console is set to automatically display on the system, anyone with physical access to the server will be able to use the console and web manager. If you are unable to guarantee the physical security of the monitor and keyboard of your e-smith server, we recommend you configure your system to *require a login* before the console can be viewed.

14. Information Bays

Information bays, or i-bays, are areas on the server in which data can be stored and made accessible through a variety of methods. Files in an i-bay can be access through the web server, Windows and Macintosh file sharing and public or private FTP access. While settings for i-bays have been discussed in previous sections, they can be summarized as follows:

- **Group ownership** - each i-bay is owned by a specific group account
- **User access via file sharing and ftp** - the default configuration of an i-bay is to restrict read and write access to only members of the assigned group. Access can also be restricted so that only the "Admin" group can write to the i-bay, or opened up so that everyone can read or write to the i-bay.
- **Public access via web or anonymous ftp** - by default this is configured for *No access*. However, the server administrator can allow such access to either the local network or the larger Internet, and can allow open access or require a password.
- **Dynamic content** - by default, this is *disabled*, but if enabled will allow users with write access to the i-bay to upload CGI or PHP scripts that can create web pages dynamically. While a powerful tool, be aware of the security implications mentioned earlier.

You should note that if a password is required for public access via web or anonymous ftp, the password is the one assigned to the *i-bay* and not to a user's regular user account. As with user accounts, the i-bay password is *not* set by default and users will not be able to access the i-bay until the administrator sets the password for the i-bay. Until that time all attempts to access the i-bay where the password is required will be refused.

15. Ongoing Security Updates

The e-smith technical staff constantly monitors industry sources of security information to be sure that no emerging issues will impact our product. Because our product is based on the Red Hat Linux distribution, we pay careful attention to notices originating from Red Hat's offices or mailing lists. Each notice is carefully evaluated to determine if there is a security impact on the e-smith server and gateway. Because we ship *without* many of the standard Linux services installed, many security alerts that apply to a generic Red Hat Linux installation do *not* apply to the e-smith server and gateway. Regardless, each alert is examined in detail.

We also continually monitor our support forums both on our web bulletin boards and our *devinfo* mailing list. Many of the developers and administrators using those forums are among the first to identify any potential security issues. They also are often connected to additional security mailing lists and forward announcements and warnings from those sources.

Finally, each and every release of our product undergoes constant intensive scrutiny by our development team. As part of the release cycle, we extensively test all services and packages.

In the event that security holes are identified by our own staff or by other parties, we rapidly make fixes available through our public FTP site and *http://www.e-smith.org/* web site and through automatic updates to registered customers.

16. Conclusion

While no system can be 100% secure, we have created a product that provides an extremely secure computing environment. Through the elimination of non-essential services, the replacement of other services with secure alternatives, and the general tightening of all possible security parameters, the e-smith server and gateway comes "out of the box" ready to protect your network and server. While sophisticated administrators can adjust the configuration to meet particular needs, both they and others will benefit from the e-smith approach.

Appendix A. List of Processes

The following processes are running on an e-smith server after a default installation. The list was generated by the command `ps fauxw`. Note that the `f` option, also available as `-forest`, shows the child processes underneath their parent, providing a clearer view of which processes are launched from the parent processes.

```
USER       PID %CPU %MEM    VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  1.7  1.1   1344  548 ?        S    03:40   0:06 init [7]
root         2  0.0  0.0      0    0 ?        SW   03:40   0:00 [kflushd]
root         3  0.0  0.0      0    0 ?        SW   03:40   0:00 [kupdate]
root         4  0.0  0.0      0    0 ?        SW   03:40   0:00 [kpiod]
root         5  0.0  0.0      0    0 ?        SW   03:40   0:00 [kswapd]
root         6  0.0  0.0      0    0 ?        SW<  03:40   0:00 [mdrecoveryd]
root       531  0.0  1.8   2116  888 ?        S    03:42   0:00 /usr/sbin/slapd
root       852  0.0  1.4   1420  656 ?        S    03:42   0:00 syslogd -m 0 -a /home/dns/dev/log
root       862  0.0  1.8   1680  872 ?        S    03:42   0:00 klogd -c 1
root       887  0.0  1.2   1388  580 ?        S    03:42   0:00 crond
root       933  0.0  1.6   2108  784 ?        S    03:42   0:00 xinetd -reuse -pidfile /var/run/xinetd.pid
lp         987  0.0  1.1   2392  552 ?        S    03:42   0:00 lpd Waiting
root      1012  0.0  0.5   1552  260 ?        S    03:43   0:00 /usr/sbin/dhcpd eth0
root      1059  0.0  0.7   1312  352 ?        S    03:43   0:00 supervise /var/lock/qmail qmail-start ./Maildir/
accustamp qmail
qmails    1062  0.0  0.8   1368  416 ?        S    03:43   0:00  \_ qmail-send
qmaill    1091  0.0  0.6   1308  316 ?        S    03:43   0:00      \_ accustamp qmail
root      1092  0.0  0.7   1324  364 ?        S    03:43   0:00      \_ qmail-lspawn ./Maildir/
qmailr    1093  0.0  0.7   1324  364 ?        S    03:43   0:00      \_ qmail-rspawn
qmailq    1094  0.0  0.8   1316  376 ?        S    03:43   0:00      \_ qmail-clean
qmaill    1060  0.0  0.7   1316  372 ?        S    03:43   0:00 cyclog -s 3500000 /var/log/qmail
mail      1082  0.0  0.7   1444  348 ?        S    03:43   0:00 smtpfwdd -d /var/spool/smtpd/spool
root      1193  0.0  8.2   8604 3868 ?        S    03:43   0:00 httpd
www       1201  0.0  4.7   8604 2236 ?        S    03:43   0:00  \_ httpd
www       1202  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1203  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1204  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1205  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1206  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1207  0.0  4.6   8604 2184 ?        S    03:43   0:00  \_ httpd
www       1213  0.0  7.5   8604 3512 ?        S    03:43   0:00  \_ httpd
www       1214  0.0  8.4   8604 3972 ?        S    03:43   0:00  \_ httpd
www       1215  0.0  8.4   8604 3972 ?        S    03:43   0:00  \_ httpd
root      1370  0.0  3.3   3444 1564 ?        S    03:43   0:00 /usr/sbin/httpd-admin -f /etc/httpd/admin-conf/
httpd.conf -D HAVE_PER
admin     1376  0.0  3.3   3464 1556 ?        S    03:43   0:00  \_ /usr/sbin/httpd-admin -f /etc/httpd/admin-conf/
httpd.conf -D HAVE
root      1389  0.0  1.9   1912  932 ?        S    03:43   0:00 sh /usr/bin/safe_mysqld -defaults-file=/etc/my.cnf
mysql     1434  0.0  3.9  12160 1856 ?        S    03:43   0:00  \_ /usr/libexec/mysqld -defaults-file=/etc/my.cnf
-basedir=/usr -
mysql     1436  0.0  3.9  12160 1856 ?        S    03:43   0:00      \_ /usr/libexec/mysqld -defaults-file=/etc/my.cnf
-basedir=/us
mysql     1437  0.0  3.9  12160 1856 ?        S    03:43   0:00          \_ /usr/libexec/mysqld -defaults-file=
/etc/my.cnf -basedir
root      1440  0.0  2.1   3660 1000 ?        S    03:43   0:00 squid -D
squid     1444  0.2  8.9   6016 4168 ?        S    03:43   0:00  \_ (squid) -D
squid     1472  0.0  0.7   1304  344 ?        S    03:43   0:00      \_ (unlinkd)
root      1486  0.0  1.2   1416  608 ?        S    03:43   0:00 atalkd
root      1487  0.0  3.5   4296 1672 ?        S    03:43   0:00 smbd -D
root      1497  0.0  3.5   3852 1680 ?        S    03:43   0:00 nmbd -D
root      1500  0.0  2.9   3780 1384 ?        S    03:43   0:00  \_ nmbd -D
root      1520  0.5  8.9   5332 4176 tty1     S    03:43   0:01 perl -wT /sbin/e-smith/console tty1
root      1524  0.1  0.0      0    0 ?        Z    03:43   0:00  \_ [rpmq <defunct>]
root      1525  0.0  0.9   1312  424 tty1     S    03:43   0:00  \_ /usr/bin/logger -p local1.info -t console
root      1526  0.0  1.5   1936  728 tty1     S    03:43   0:00  \_ /usr/bin/whiptail -clear -backtitle e-smith
server and gateway
root      1521  0.1  2.1   2196 1028 tty2     S    03:43   0:00 login - root
root      1551  0.1  2.6   2160 1224 tty2     S    03:44   0:00  \_ -bash
```

```
root      1572  0.0  1.6  2588  756 tty2     R    03:47   0:00      \_ ps auxw -forest
root      1522  0.0  0.9  1316  444 tty3     S    03:43   0:00 /sbin/mingetty tty3
dns       1523  0.0  3.6  2792 1716 ?        S    03:43   0:00 /usr/sbin/named -f -u dns -g dns -t /home/dns
root      1539  0.0  1.0  1368  488 ?        S    03:44   0:00 papd
root      1549  0.0  1.6  1704  776 ?        S    03:44   0:00 afpd -c 20 -n e-smith-server
```

Appendix B. List of Loaded Kernel Modules

The is the minimal list of kernel modules loaded after a default installation (created by the `lsmod` command). The `pcnet32` module is for the network interface card used in the sample machine. As the sample machines was installed in *server and gateway mode* the `ip_masq_*` modules are loaded. In *server-only* mode, these modules are not loaded. Note that our default policy is to provide as much power and flexibility for users as possible, while maintaining high security. As a consequence, our default gateway configuration includes a large set of masquerading modules, including popular games. Naturally, this can be customized for sites who want restricted functionality.

Note that this list is highly dependent upon the exact hardware configuration of your system. Additional modules may be loaded to support USB controllers, tape drives and certain types of disk drives and video boards.

```
Module                  Size  Used by
appletalk-fixed        20960  12  (autoclean)
ip_masq_vdolive         1376   0  (unused)
ip_masq_raudio          3008   0  (unused)
ip_masq_quake           1392   0  (unused)
ip_masq_irc             1632   0  (unused)
ip_masq_icq            10144   0  (unused)
ip_masq_h323            3600   0  (unused)
ip_masq_ftp             4256   0  (unused)
ip_masq_cuseeme         1120   0  (unused)
pcnet32                10736   2  (autoclean)
```

Appendix C. Packet Filtering Rules

The following packet filtering rules are established by the `ipchains` command for the default installation in *server and gateway* mode for version 4.1.1. In this example, the IP address for the *internal* network interface card is 192.168.1.1 and the IP address for the *external* network interface card is 192.168.65.17. This listing was generated by `ipchains -L -n`.

Annotations are provided below to explain specific blocks of rules.

The first chain, `input`, specifies what packets will be accepted into the server. At the very beginning, all inbound ICMP packets are immediately routed to a separate chain, `icmpIn` for processing. Note that the second line, starting with "ACCEPT", applies to the loopback interface, `lo`, and accepts all incoming packets (which can only originate on the local machine).

```
Chain input (policy DENY):
target     prot opt     source               destination          ports
icmpIn     icmp ----  0.0.0.0/0            0.0.0.0/0            * ->   *
ACCEPT     all  ----  0.0.0.0/0            0.0.0.0/0            n/a
denylog    tcp  ----  0.0.0.0/0            0.0.0.0/0            0:19 ->   *
denylog    udp  ----  0.0.0.0/0            0.0.0.0/0            0:19 ->   *
denylog    tcp  ----  0.0.0.0/0            0.0.0.0/0            * ->   0:19
denylog    udp  ----  0.0.0.0/0            0.0.0.0/0            * ->   0:19
DENY       all  ----  224.0.0.0/3          0.0.0.0/0            n/a
DENY       all  ----  0.0.0.0/0            224.0.0.0/3          n/a
ACCEPT     all  ----  192.168.1.0/24       0.0.0.0/0            n/a
ACCEPT     tcp  !y---  0.0.0.0/0            0.0.0.0/0            * ->   *
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   113
ACCEPT     udp  ----  0.0.0.0/0            0.0.0.0/0            67:68 ->   *
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   20
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   21
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   80
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   443
ACCEPT     tcp  ----  0.0.0.0/0            192.168.65.17        * ->   25
ACCEPT     tcp  ----  0.0.0.0/0            0.0.0.0/0            * ->   1023:65535
ACCEPT     udp  ----  0.0.0.0/0            0.0.0.0/0            * ->   1023:65535
```

```
denylog    all  ----  0.0.0.0/0            0.0.0.0/0           n/a
```

The `forward` chain specifies that any packets from the internal network will be masqueraded and passed on to the `output` chain for processing before being sent out to the external Internet.

```
Chain forward (policy DENY):
target     prot opt     source               destination          ports
ACCEPT     all  ----  192.168.1.0/24       192.168.1.0/24       n/a
MASQ       all  ----  192.168.1.0/24       0.0.0.0/0            n/a
DENY       all  ----  0.0.0.0/0            0.0.0.0/0            n/a
```

The `output` chain specifies which packets will be allowed to leave the server. Note again that outbound ICMP packets are sent to the separate `icmpOut` chain for processing.

```
Chain output (policy ACCEPT):
target     prot opt     source               destination          ports
icmpOut    icmp ----  0.0.0.0/0            0.0.0.0/0           * ->   *
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   80
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   22
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   23
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   21
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   110
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   25
-          tcp  ----  0.0.0.0/0            0.0.0.0/0           * ->   20
ACCEPT     all  ----  0.0.0.0/0            0.0.0.0/0           n/a
DENY       all  ----  224.0.0.0/3          0.0.0.0/0           n/a
DENY       all  ----  0.0.0.0/0            224.0.0.0/3         n/a
ACCEPT     icmp ----  192.168.1.0/24       0.0.0.0/0           * ->   *
ACCEPT     all  ----  0.0.0.0/0            192.168.1.0/24      n/a
ACCEPT     tcp  !y---  192.168.65.17        0.0.0.0/0            20 ->   *
ACCEPT     tcp  !y---  192.168.65.17        0.0.0.0/0            21 ->   *
ACCEPT     tcp  !y---  192.168.65.17        0.0.0.0/0            80 ->   *
ACCEPT     tcp  !y---  192.168.65.17        0.0.0.0/0            443 ->   *
ACCEPT     tcp  !y---  192.168.65.17        0.0.0.0/0            25 ->   *
ACCEPT     all  ----  0.0.0.0/0            0.0.0.0/0           n/a
```

The `denylog` chain is referenced by several of the other chains and just denies a packet while logging the packet denial. This chain is used simply to avoid cluttering up individual rules with logging options.

```
Chain denylog (7 references):
target     prot opt     source               destination          ports
DENY       all  ---l-  0.0.0.0/0            0.0.0.0/0            n/a
```

This `icmpIn` chain specifies which types of ICMP packets we will allow into the system.

```
Chain icmpIn (1 references):
target     prot opt     source               destination          ports
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           8 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           0 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           3 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           4 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           11 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           12 ->   *
denylog    all  ----  0.0.0.0/0            0.0.0.0/0           n/a
```

The `icmpOut` chain specifies which types of ICMP packets we will allow to leave the system.

```
Chain icmpOut (1 references):
target     prot opt     source               destination          ports
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           8 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           0 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           3 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           4 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           11 ->   *
ACCEPT     icmp ----  0.0.0.0/0            0.0.0.0/0           12 ->   *
```

```
denylog    all  ----  0.0.0.0/0           0.0.0.0/0           n/a
```

Appendix D. List of Open Network Ports

Programs are bound to and listening on these TCP or UDP ports. This output was generated by `netstat -anp | egrep "LISTEN|udp" | grep -v "^unix"` and then sorted by port number and divided into sections. The last column lists the process that is bound to the specific port number.

The following processes are bound to the internal network interface card and/or the localhost. They are for DNS (`named`), NetBIOS naming a.k.a. WINS (`nmbd`) and SMB file sharing (`smbd`).

```
tcp        0      0 192.168.1.1:53          0.0.0.0:*               LISTEN      1523/named
udp        0      0 192.168.1.1:53          0.0.0.0:*                           1149/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1523/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           1149/named
udp        0      0 192.168.1.1:137         0.0.0.0:*                           1128/nmbd
udp        0      0 192.168.1.1:138         0.0.0.0:*                           1128/nmbd
tcp        0      0 127.0.0.1:139           0.0.0.0:*               LISTEN      1487/smbd
tcp        0      0 192.168.1.1:139         0.0.0.0:*               LISTEN      1487/smbd
```

The following processes are bound to the external interface. Most of these processes should be familiar to Linux/UNIX administrators, but two may not be. `afpd` is the AppleTalk file sharing component of `netatalk`. `slapd` is the LDAP directory server that we are using. Note that external connections to all of these processes except for the two with an asterisk in the first column are blocked by the packet filtering rules mentioned previously. These processes must be blocked by the `ipchains` rules because they do not provide the option to selectively bind to specific interfaces.

```
  udp        0      0 0.0.0.0:67              0.0.0.0:*                           615/dhcpd
* tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      1193/httpd
  udp        0      0 0.0.0.0:137             0.0.0.0:*                           1128/nmbd
  udp        0      0 0.0.0.0:138             0.0.0.0:*                           1128/nmbd
  tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN      531/slapd
* tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      1193/httpd
  tcp        0      0 0.0.0.0:515             0.0.0.0:*               LISTEN      987/lpd Waiting
  tcp        0      0 0.0.0.0:548             0.0.0.0:*               LISTEN      1549/afpd
  tcp        0      0 0.0.0.0:980             0.0.0.0:*               LISTEN      1370/httpd-admin
  udp        0      0 0.0.0.0:1024            0.0.0.0:*                           1149/named
  tcp        0      0 0.0.0.0:3128            0.0.0.0:*               LISTEN      1444/(squid)
  udp        0      0 0.0.0.0:3130            0.0.0.0:*                           1082/(squid)
  tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      1434/mysqld
  udp        0      0 0.0.0.0:3401            0.0.0.0:*                           1082/(squid)
```

In addition, a number of processes do not run as separate daemons but instead are launched by the `inetd` daemon. With version 7.0 of their Linux distribution, Red Hat began using `xinetd`, a newer and more flexible implementation of the traditional `inetd` daemon. As shown below, `xinetd` is listening on ports 21 (FTP), 25 (SMTP), 110 (POP3), 113 (ident) and 143 (IMAP).

```
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      933/xinetd
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      933/xinetd
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN      933/xinetd
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN      933/xinetd
tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN      933/xinetd
```

Appendix E. List of Open UNIX Domain Sockets

These ports are UNIX sockets open for access by programs on the e-smith *server* only. This output was generated by `netstat -anp | grep unix`. The final column shows the socket file name to which the process is bound.

```
unix  10     [ ]         DGRAM                    1230   852/syslogd         /dev/log
unix  0      [ ACC ]     STREAM     LISTENING     6169   1523/named          /var/run/ndc
unix  0      [ ACC ]     STREAM     LISTENING     3516   1434/mysqld         /var/lib/mysql/mysql.sock
unix  1      [ ]         DGRAM                    1232   852/syslogd         /home/dns/dev/log
unix  0      [ ]         DGRAM                    6233   1521/login - root
unix  0      [ ]         DGRAM                    6218   1549/afpd
unix  0      [ ]         DGRAM                    6208   1539/papd
```

```
unix  0      [ ]         DGRAM                    6167   1523/named
unix  0      [ ]         DGRAM                    6094   1486/atalkd
unix  0      [ ]         DGRAM                    6011   1440/squid
unix  0      [ ]         DGRAM                    1556   1082/smtpfwdd
unix  0      [ ]         DGRAM                    1446   1012/dhcpd
unix  0      [ ]         DGRAM                    1350   933/xinetd
unix  0      [ ]         DGRAM                    1286   887/crond
unix  0      [ ]         DGRAM                    1250   862/klogd
unix  0      [ ]         DGRAM                    888    531/slapd
```

Appendix F. Configuration of tcp_wrappers

The primary application level access control lists are implemented via a mechanism called `tcp_wrappers`. This involves both an explicit daemon `tcpd` as well as a set of libraries called by some applications (such as `sshd`). Both the daemon and the libraries first check `/etc/hosts.deny` to see if a connection is denied and then check `/etc/hosts.allow` to see if the connection is *explicitly* allowed.

As you can see from the listings below, our default policy is to deny *all* connections to the server except for those *explicitly* allowed to specific ports.

Appendix F.1. /etc/hosts.deny

```
ALL: ALL
```

Appendix F.2. /etc/hosts.allow

Note that `/etc/hosts.allow` is dynamically changed whenever a subnet is added or the IP address of the server or its netmask is changed.

```
# appletalk services
afpd : 127.0.0.1, 192.168.1.0/255.255.255.0
papd : 127.0.0.1, 192.168.1.0/255.255.255.0

# identification server
in.identd : ALL

# IMAP server
imapd : 127.0.0.1, 192.168.1.0/255.255.255.0

# LDAP servers
slapd : 127.0.0.1, 192.168.1.0/255.255.255.0

# obtuse-smtpd - also see smtpd_check_rules
smtpd : ALL

# ftp daemon
in.proftpd : ALL

# pop3 server
qmail-popup : 127.0.0.1, 192.168.1.0/255.255.255.0

# sshd access is currently disabled
# telnet access is currently disabled
```

Appendix G. SMB Security

The following information is from the `/etc/smb.conf` configuration file that controls Samba's operations. Note that only the relevant security-related sections of the file have been included here.

```
# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
```

```
  # following example restricts access to two C class networks and
  # the "loopback" interface. For more examples of the syntax see
  # the smb.conf man page

     hosts allow = 127.0.0.1 192.168.1.0/255.255.255.0

  # Configure Samba to use multiple interfaces
  # If you have multiple network interfaces then you must list them
  # here. See the man page for details.
     interfaces = 127.0.0.1 192.168.1.1/255.255.255.0

  # Security mode. Most people will want user level security. See
  # security_level.txt for details.
     security = user
  # Use password server option only with security = server
  ;   password server = <NT-Server-Name>

  # Password Level allows matching of _n_ characters of the password for
  # all combinations of upper and lower case.
  ;  password level = 8
  ;  username level = 8

  # If this parameter is 'yes' for a service, then no password is
  # required to connect to the service.
     guest ok = yes

  # This is a username which will be used for access to services which
  # are specified as 'guest ok'.
     guest account = public

  # If unknown user logs in, treat as guest. (In older versions of
  # Samba this was a compile-time option.)
     map to guest = bad user

  # You may wish to use password encryption. Please read
  # ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
  # Do not enable this option unless you have read those documents
    encrypt passwords = yes
    smb passwd file = /etc/smbpasswd

  # The following are needed to allow password changing from Windows to
  # update the Linux sytsem password also.
  # NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
  # NOTE2: You do NOT need these to allow workstations to change only
  #        the encrypted SMB passwords. They allow the Unix password
  #        to be kept in sync with the SMB password.
  ;  unix password sync = Yes
  ;  passwd program = /usr/bin/passwd %u
  ;  passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
  *passwd:*all*authentication*tokens*updated*successfully*

  # Unix users can map to different SMB User names
  ;  username map = /etc/smbusers

  # This global parameter allows the Samba admin to limit what
  # interfaces on a machine will serve smb requests.
      bind interfaces only = yes

  # Browser Control Options:
  # set local master to no if you don't want Samba to become a master
  # browser on your network. Otherwise the normal election rules apply
     local master = yes

  # Domain Master specifies Samba to be the Domain Master Browser. This
  # allows Samba to collate browse lists between subnets. Don't use this
```

```
# if you already have a Windows NT domain controller doing this job
   domain master = yes

# Preferred Master causes Samba to force a local browser election on startup
# and gives it a slightly higher chance of winning the election
   preferred master = yes

# Use only if you have an NT server on your network that has been
# configured at install time to be a primary domain controller.
;   domain controller = <NT-Domain-Controller-SMBName>

[homes]
   comment = Home directory
   browseable = no
   guest ok = no
   read only = no
   writable = yes
   printable = no
   create mode = 0660
   force create mode = 0660
   directory mode = 0770
   force directory mode = 0770
   path = /home/e-smith/files/users/%S/home

[Primary]
   comment = Primary site
   path = /home/e-smith/files/primary
   read only = no
   writable = yes
   printable = no
   create mode = 0640
   force create mode = 0640
   directory mode = 0750
   force directory mode = 0750

[netlogon]
   comment = Network Logon Service
   path = /home/netlogon
   guest ok = yes
   writable = yes
   browseable = no
   share modes = no
```

Appendix H. Apache Security

The following lines are the security-related sections of the Apache configuration file found at `/etc/httpd/conf/http.conf`. The first block of code determines what port the server will run on, where the root of the files are and what user and group the server will run as.

```
Port 80
ServerAdmin admin@tofu-dog.com
ServerRoot /etc/httpd
User www
Group www
```

The next relevant block lists all the modules that are loaded into the Apache web server. Apache provides a modular structure that allows additional functionality to be be loaded into the program through compiled modules. In keeping with UNIX/Linux conventions, the # character indicates that a given line is "commented out" and that module is therefore not loaded.

```
# Dynamic Shared Object (DSO) Support
#
#LoadModule mmap_static_module modules/mod_mmap_static.so
LoadModule php4_module /usr/lib/apache/libphp4.so
```

```
LoadModule env_module         modules/mod_env.so
LoadModule config_log_module  modules/mod_log_config.so
LoadModule agent_log_module   modules/mod_log_agent.so
LoadModule referer_log_module modules/mod_log_referer.so
#LoadModule mime_magic_module  modules/mod_mime_magic.so
LoadModule mime_module        modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule status_module      modules/mod_status.so
LoadModule info_module        modules/mod_info.so
LoadModule includes_module    modules/mod_include.so
LoadModule autoindex_module   modules/mod_autoindex.so
LoadModule dir_module         modules/mod_dir.so
LoadModule cgi_module         modules/mod_cgi.so
LoadModule asis_module        modules/mod_asis.so
LoadModule imap_module        modules/mod_imap.so
LoadModule action_module      modules/mod_actions.so
#LoadModule speling_module     modules/mod_speling.so
LoadModule userdir_module     modules/mod_userdir.so
LoadModule proxy_module       modules/libproxy.so
LoadModule alias_module       modules/mod_alias.so
LoadModule rewrite_module     modules/mod_rewrite.so
LoadModule access_module      modules/mod_access.so
LoadModule auth_module        modules/mod_auth.so
LoadModule anon_auth_module   modules/mod_auth_anon.so
#LoadModule dbm_auth_module    modules/mod_auth_dbm.so
LoadModule db_auth_module     modules/mod_auth_db.so
LoadModule digest_module      modules/mod_digest.so
#LoadModule cern_meta_module   modules/mod_cern_meta.so
LoadModule expires_module     modules/mod_expires.so
LoadModule headers_module     modules/mod_headers.so
LoadModule usertrack_module   modules/mod_usertrack.so
#LoadModule example_module     modules/mod_example.so
#LoadModule unique_id_module   modules/mod_unique_id.so
LoadModule setenvif_module    modules/mod_setenvif.so

# Extra Modules
#LoadModule php_module         modules/mod_php.so
#LoadModule php3_module        modules/libphp3.so
#LoadModule perl_module        modules/libperl.so
LoadModule external_auth_module modules/mod_auth_external.so

LoadModule ssl_module        /usr/lib/apache/libssl.so
#  Reconstruction of the complete module list from all available modules
#  (static and shared ones) to achieve correct module execution order.
#  [WHENEVER YOU CHANGE THE LOADMODULE SECTION ABOVE UPDATE THIS, TOO]
ClearModuleList
AddModule mod_php4.c

#AddModule mod_mmap_static.c
AddModule mod_env.c
AddModule mod_log_config.c
AddModule mod_log_agent.c
AddModule mod_log_referer.c
#AddModule mod_mime_magic.c
AddModule mod_mime.c
AddModule mod_negotiation.c
AddModule mod_status.c
AddModule mod_info.c
AddModule mod_include.c
AddModule mod_autoindex.c
AddModule mod_dir.c
AddModule mod_cgi.c
AddModule mod_asis.c
AddModule mod_imap.c
AddModule mod_actions.c
```

```
#AddModule mod_speling.c
AddModule mod_userdir.c
AddModule mod_proxy.c
AddModule mod_alias.c
AddModule mod_rewrite.c
AddModule mod_access.c
AddModule mod_auth.c
AddModule mod_auth_anon.c
#AddModule mod_auth_dbm.c
AddModule mod_auth_db.c
AddModule mod_auth_external.c
AddModule mod_digest.c
#AddModule mod_cern_meta.c
AddModule mod_expires.c
AddModule mod_headers.c
AddModule mod_usertrack.c
#AddModule mod_example.c
#AddModule mod_unique_id.c
AddModule mod_so.c
AddModule mod_setenvif.c

# Extra Modules
#AddModule mod_php.c
#AddModule mod_php3.c
#AddModule mod_perl.c

AddModule mod_ssl.c
AddExternalAuth pwauth /usr/lib/apache/pwauth
SetExternalAuthMethod pwauth pipe
```

The next block of code configures Secure Socket Layer support within Apache.

```
##############################################################
##  SSL Global Context Configuration
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts
## (unless overridden by virtual hosts)
##
<IfModule mod_ssl.c>
##  SSL Support
##  When we also provide SSL we have to listen to the
##  standard HTTPS port - 443
##
Listen 443
Listen 80

SSLEngine off

SSLPassPhraseDialog  builtin

SSLSessionCache         dbm:logs/ssl_scache

SSLSessionCacheTimeout  300

SSLMutex  file:logs/ssl_mutex

SSLRandomSeed startup file:/dev/urandom 512
SSLRandomSeed connect builtin

SSLLog      logs/ssl_engine_log


SSLLogLevel info
```

```
SSLProtocol all
</IfModule>
```

Next, the configuration file specifies how many clients can connect at any one time, where the documents are stored and other content-related configuration parameters.

```
MaxClients 150
MaxRequestsPerChild 100
#ProxyRequests On
ServerName www.tofu-dog.com
MinSpareServers 8
MaxSpareServers 20
Timeout 300
DirectoryIndex index.htm index.html index.shtml index.cgi index.php index.php3
DocumentRoot /home/e-smith/files/primary/html
FancyIndexing on
# UserDir public_html
AccessFileName .htaccess
DefaultType text/plain
TypesConfig /etc/mime.types

Redirect /e-smith-manager http://192.168.1.1:980
Redirect /e-smith-password http://192.168.1.1:980/e-smith-password

# To use CGI scripts:
AddHandler cgi-script .cgi

AddHandler server-parsed .shtml
```

Next are listed the *aliases* for the primary web site and any virtual domains. These allow the settings above to be overriden for each of the individual virtual domains. Note that this information is generated automatically by e-smith management software.

```
NameVirtualHost 127.0.0.1:80

<VirtualHost 127.0.0.1:80>

    ServerName www.tofu-dog.com
    ServerAlias tofu-dog.com

    # primary content

    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files
    Alias       /common  /etc/e-smith/web/common

    # alias for Apache icons

    Alias /icons/ /var/www/icons/

</VirtualHost>

NameVirtualHost 192.168.1.1:80

<VirtualHost 192.168.1.1:80>

    ServerName www.tofu-dog.com
    ServerAlias tofu-dog.com

    # primary content

    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files
    Alias       /common  /etc/e-smith/web/common
```

```
    # alias for Apache icons

    Alias /icons/ /var/www/icons/

</VirtualHost>

NameVirtualHost 192.168.65.17:80

<VirtualHost 192.168.65.17:80>

    ServerName www.tofu-dog.com
    ServerAlias tofu-dog.com

    # primary content

    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files
    Alias       /common  /etc/e-smith/web/common

    # alias for Apache icons

    Alias /icons/ /var/www/icons/

</VirtualHost>
```

Now the aliases are again defined, but this time for SSL (port 443) connections with additional SSL-related information.

```
NameVirtualHost 127.0.0.1:443

<VirtualHost 127.0.0.1:443>

    ServerName secure.tofu-dog.com
    ServerAlias tofu-dog.com www.tofu-dog.com

    # primary content

    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files

    # alias for Apache icons

    Alias /icons/ /var/www/icons/

    # SSL Directives

    SSLEngine on
    SSLCertificateFile /home/e-smith/ssl.crt/secure.tofu-dog.com.crt
    SSLCertificateKeyFile /home/e-smith/ssl.key/secure.tofu-dog.com.key
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive  ssl-unclean-shutdown  downgrade-1.0 force-response-1.0

</VirtualHost>

NameVirtualHost 192.168.1.1:443

<VirtualHost 192.168.1.1:443>

    ServerName secure.tofu-dog.com
    ServerAlias tofu-dog.com www.tofu-dog.com

    # primary content
```

```
    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files

    # alias for Apache icons

    Alias /icons/ /var/www/icons/

    # SSL Directives

    SSLEngine on
    SSLCertificateFile /home/e-smith/ssl.crt/secure.tofu-dog.com.crt
    SSLCertificateKeyFile /home/e-smith/ssl.key/secure.tofu-dog.com.key
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive  ssl-unclean-shutdown  downgrade-1.0 force-response-1.0

</VirtualHost>

NameVirtualHost 192.168.65.17:443

<VirtualHost 192.168.65.17:443>

    ServerName secure.tofu-dog.com
    ServerAlias tofu-dog.com www.tofu-dog.com

    # primary content

    DocumentRoot         /home/e-smith/files/primary/html
    ScriptAlias /cgi-bin /home/e-smith/files/primary/cgi-bin
    Alias       /files   /home/e-smith/files/primary/files

    # alias for Apache icons

    Alias /icons/ /var/www/icons/

    # SSL Directives

    SSLEngine on
    SSLCertificateFile /home/e-smith/ssl.crt/secure.tofu-dog.com.crt
    SSLCertificateKeyFile /home/e-smith/ssl.key/secure.tofu-dog.com.key
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive  ssl-unclean-shutdown  downgrade-1.0 force-response-1.0

</VirtualHost>
```

The final security-related section of the Apache configuration deals with configuring the security of the various directories whose content will be served by the Apache web server. Once information bays are added to the system, they will each have entries listed here.

```
# First, we configure the "default" to be a very restrictive set of
# permissions.

<Directory />
    Options None
    AllowOverride None
    order deny,allow
    deny from all
    allow from none
</Directory>

# horde not configured as it is disabled in the config db

# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
```

```
#---------------------------------------
# primary directories
#---------------------------------------

<Directory /etc/e-smith/web/common>
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /home/e-smith/files/primary/html>
    Options Indexes Includes
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /home/e-smith/files/manager/html>
    Options Indexes Includes ExecCGI
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /var/www/icons>
    Options Indexes Includes
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /home/e-smith/files/primary/cgi-bin>
    Options ExecCGI
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /home/e-smith/files/manager/cgi-bin>
    Options ExecCGI
    AllowOverride None
    order deny,allow
    deny from all
    allow from all
</Directory>

<Directory /home/e-smith/files/primary/files>
    AllowOverride None
    ForceType application/octet-stream
    order deny,allow
    deny from all
    allow from all
</Directory>
```

Appendix I. Rules For Mail Relaying

The rules listed below are from `/var/spool/smtpd/etc/smtpd_check_rules`. They are used by the `smtpd` program to determine whether mail should be accepted for delivery based on the sender's IP address as well as the envelope recipient address. This example uses our sample domain `tofu-dog.com` and blocks all relaying of e-mail except from users on the internal network. Note that it also blocks anyone from the Internet from being able to send to *everyone@tofu-dog.com* or *shared@tofu-dog.com*.

```
# Don't allow bang paths via us
noto:ALL:ALL:*!*@*:551 Sorry %H (%I), I don't allow unauthorized relaying. You can't use me to send
mail from %F to %T.

# Don't allow two @s (equivalent to %hack) via us
noto:ALL:ALL:*@*@*:551 Sorry %H (%I), I don't allow unauthorized relaying. You can't use me to send
mail from %F to %T.

# Don't allow %hack relay via us
noto:ALL:ALL:*%*@*:551 Sorry %H (%I), I don't allow unauthorized relaying. You can't use me to send
mail from %F to %T.

# Allow relaying from the local network
allow:127.0.0.1:ALL:ALL
allow:192.168.1.0/24:ALL:ALL

# Prohibit access to these addresses from the outside world
noto:ALL:ALL:everyone@*.tofu-dog.com everyone@tofu-dog.com:551 Sorry %H (%I), you cannot send
mail to %T from outside our local network.
noto:ALL:ALL:shared@*.tofu-dog.com shared@tofu-dog.com:551 Sorry %H (%I), you cannot send
mail to %T from outside our local network.

# Allow any of our domains
allow:ALL:ALL:*.tofu-dog.com *@tofu-dog.com

# Just say no to anything else, we won't relay for people we don't know.
noto:ALL:ALL:ALL:551 Sorry %H(%I), I don't allow unauthorized relaying. Please use another SMTP host
to mail from %F to %T
```

Appendix J. List of setuid Files and Directories

The following files have the `setuid` bit set which, in all but one case, means that these commands are executed as if by the `root` user. Note that the e-smith `setuid` programs can only be executed by the root user or members of the `admin` group. This list was generated by the command `find / -path /proc -prune -o perm +4000 -exec ls -ld {} \;` which lists all `setuid` files except for those found in the virtual `/proc` filesystem.

```
-rws-x-x    1 qmailq   qmail       12680 Feb  8 21:13 /var/qmail/bin/qmail-queue
-rwsr-x--    1 root     admin       32632 Feb 15 15:04 /etc/e-smith/web/functions/backup
-rwsr-x--    1 root     admin       20997 Jan 15 17:11 /etc/e-smith/web/functions/groups
-rwsr-x--    1 root     admin       32761 Jan 25 19:04 /etc/e-smith/web/functions/ibays
-rwsr-x--    1 root     admin       13868 Jan 15 17:11 /etc/e-smith/web/functions/localnetworks
-rwsr-x--    1 root     admin        7899 Jan 15 17:11 /etc/e-smith/web/functions/navigation
-rwsr-x--    1 root     admin        7497 Jan 15 17:11 /etc/e-smith/web/functions/noframes
-rwsr-x--    1 root     admin        2684 Jan 15 17:11 /etc/e-smith/web/functions/online-manual
-rwsr-x--    1 root     admin        6354 Jan 15 17:11 /etc/e-smith/web/functions/password
-rwsr-x--    1 root     admin         965 Aug 21  2000 /etc/e-smith/web/functions/pleasewait
-rwsr-x--    1 root     admin        4679 Jan 15 17:11 /etc/e-smith/web/functions/reboot
-rwsr-x--    1 root     admin       13232 Feb 15 17:40 /etc/e-smith/web/functions/remoteaccess
-rwsr-x--    1 root     admin        9011 Feb 15 17:40 /etc/e-smith/web/functions/review
-rwsr-x--    1 root     admin        9640 Jan 15 17:11 /etc/e-smith/web/functions/starterwebsite
-rwsr-x--    1 root     admin       24131 Feb 15 17:40 /etc/e-smith/web/functions/support
-rwsr-x--    1 root     admin       43540 Feb 15 17:40 /etc/e-smith/web/functions/useraccounts
-rwsr-x--    1 root     admin       16924 Feb 15 17:40 /etc/e-smith/web/functions/virtualdomains
-rwsr-x--    1 root     admin        7437 Jan 25 19:04 /etc/e-smith/web/functions/workgroup
-rwsr-x--    1 root     admin       12550 Feb 12 19:59 /etc/e-smith/web/functions/emailretrieval
-rwsr-x--    1 root     admin       10700 Feb 12 19:59 /etc/e-smith/web/functions/otheremail
-rwsr-x--    1 root     admin       17763 Feb 12 19:59 /etc/e-smith/web/functions/pseudonyms
```

```
-rwsr-x--    1 root     admin       18858 Jan 25 17:04 /etc/e-smith/web/functions/hostentries
-rwsr-x--    1 root     admin        9925 Jan 25 16:50 /etc/e-smith/web/functions/directory
-rwsr-x--    1 root     admin       12421 Jan 25 17:36 /etc/e-smith/web/functions/printers
-rwsr-x--    1 root     admin       17501 Feb 12 14:11 /etc/e-smith/web/functions/datetime
-rwsr-x--    1 root     admin        5778 Feb  8 19:49 /etc/e-smith/web/functions/qmailanalog
-rwsr-x--    1 root     admin        4921 Jan 25 16:40 /etc/e-smith/web/functions/reinstall
-r-sr-s--    1 root     admin        5914 Jan 15 17:11 /etc/e-smith/web/panels/password/cgi-bin/userpassword
-rwsr-xr-x    1 root     root        14184 Jul 12  2000 /bin/su
-rwsr-xr-x    1 root     root        22108 Oct 10 16:18 /bin/ping
-rwsr-xr-x    1 root     root        55420 Oct  4  2000 /bin/mount
-rwsr-xr-x    1 root     root        25404 Oct  4  2000 /bin/umount
-r-sr-xr-x    1 root     root        14784 Nov 30 13:16 /sbin/pwdb_chkpwd
-r-sr-xr-x    1 root     root        15360 Nov 30 13:16 /sbin/unix_chkpwd
-r-sr-sr-    1 root     root       102973 Feb 15 17:40 /sbin/e-smith/console
-rwsr-xr-x    1 root     root        34220 Aug  8  2000 /usr/bin/chage
-rwsr-xr-x    1 root     root        36344 Aug  8  2000 /usr/bin/gpasswd
-rwsr-xr-x    1 root     root        35964 Aug 23  2000 /usr/bin/at
-rwsr-xr-x    1 root     root        21248 Aug 24  2000 /usr/bin/crontab
-rws-x-x    2 root     root       793603 Aug  7  2000 /usr/bin/suidperl
-rws-x-x    2 root     root       793603 Aug  7  2000 /usr/bin/sperl5.6.0
-rwsr-xr-x    1 root     root       176932 Nov 21 17:53 /usr/bin/ssh
-r-s-x-x    1 root     root        13536 Jul 12  2000 /usr/bin/passwd
-rwsr-sr-x    1 root     mail        63772 Aug 11  2000 /usr/bin/procmail
-rws-x-x    1 root     root        13184 Aug 30  2000 /usr/bin/chfn
-rws-x-x    1 root     root        12640 Aug 30  2000 /usr/bin/chsh
-rws-x-x    1 root     root         5464 Aug 30  2000 /usr/bin/newgrp
-r-sr-x--    1 root     www          4656 Feb  8 20:53 /usr/lib/apache/pwauth
-rwsr-xr-x    1 root     root         6316 Feb  8 21:32 /usr/sbin/usernetctl
-rwsr-xr-x    1 root     root        16992 Jul 19  2000 /usr/sbin/traceroute
```

Appendix K. List of setgid Files

The following files have the `setgid` bit set which means that these commands are executed by a user with the specific group permissions rather than the user's normal group settings. Note that some of these files will also be found earlier in Appendix J. This list was generated by the command `find / -path /proc -prune -o perm +2000 -exec ls -ld {} \;` which lists all `setgid` files except for those found in the virtual `/proc` filesystem. The listing was further piped through `grep -v "^d"` to remove extraneous directory entries.

```
-r-sr-s--    1 root     admin        5914 Jan 15 17:11 /etc/e-smith/web/panels/password/cgi-bin/userpassword
-rwxr-sr-x    1 root     root         4144 Feb  8 21:32 /sbin/netreport
-r-sr-sr-    1 root     root       102973 Feb 15 17:40 /sbin/e-smith/console
-rwxr-sr-x    1 root     man         35260 Aug 23  2000 /usr/bin/man
-rwxr-sr-x    1 root     uucp       171452 Aug 24  2000 /usr/bin/minicom
-rwxr-sr-x    1 root     mail        10932 Aug 11  2000 /usr/bin/lockfile
-rwsr-sr-x    1 root     mail        63772 Aug 11  2000 /usr/bin/procmail
-rwxr-sr-x    1 root     slocate     19932 Dec 18 12:15 /usr/bin/slocate
-r-xr-sr-x    1 root     tty          6524 Aug  8  2000 /usr/bin/wall
-rwxr-sr-x    1 root     tty          8500 Aug 30  2000 /usr/bin/write
-rwxr-sr-x    1 root     utmp         6584 Jul 13  2000 /usr/sbin/utempter
```

Appendix L. Sample ssh Server Configuration File

Although not enabled by default, many users enable the `sshd` daemon to provide login capabilities through `ssh` programs. The example below shows `/etc/ssh/sshd_config` in the most secure combination of *Private* (internal network only) access, with no `root` login allowed and requiring RSA authentication. The system is configured with the default IP address of 192.168.1.1.

```
Port 22
ListenAddress 192.168.1.1
HostKey /etc/ssh/ssh_host_key
KeyRegenerationInterval 3600
LoginGraceTime 600
ServerKeyBits 768
```

```
IgnoreRhosts yes
PasswordAuthentication no
PermitEmptyPasswords no
PermitRootLogin no
RSAAuthentication yes
RhostsAuthentication no
RhostsRSAAuthentication no
StrictModes yes
X11DisplayOffset 10
X11Forwarding no
CheckMail no
KeepAlive yes
PrintMotd yes
SyslogFacility AUTH
```

Appendix M. Listing of Software Packages

The following listing is a sorted list of all RPM software packages installed in an e-smith server and gateway for version 4.1.1. This list was generated by the command `rpm -qa | sort -f`.

```
anacron-2.3-9
apache-1.3.14-3
apmd-3.0final-18
appletalk-fixed-0.1-6
ash-0.2-26
at-3.1.8-12
authconfig-4.0.16-4
autopassword-2.0-3
basesystem-7.0-2
bash-2.04-11
bc-1.05a-13
bdflush-1.5-14
bind-8.2.3-1
bind-utils-8.2.3-1
binutils-2.10.0.18-1
buffer-1.19-5
bzip2-1.0.1-3
checkpassword-0.76-1
chkconfig-1.2.16-1
console-tools-19990829-25
cpio-2.4.2-20
cracklib-2.7-8
cracklib-dicts-2.7-8
crontabs-1.8-1
daemontools-0.53-1
db1-1.85-4
db2-2.4.14-4
db3-3.1.14-6
dev-3.0.6-5
dhcp-2.0-12
dhcpcd-1.3.18pl8-6
diald-0.99.4-2
diffutils-2.7-21
dot-forward-0.71-02
dump-0.4b19-5es
e2fsprogs-1.18-16
ed-0.2-19
eject-2.0.2-6
e-smith-4.2.0-02
e-smith-backup-1.2.0-09
e-smith-base-4.2.0-29
e-smith-boot-image-1.2.0-03
e-smith-devtools-1.2.0-03
findutils-4.1.5-4
flexbackup-0.9.8-06es
freetype-1.3.1-7
ftp-0.17-6
gawk-3.0.6-1
gd-1.8.3-4
gdbm-1.8.0-5
getty_ps-2.0.7j-12
glib-1.2.8-4
glibc-2.2-9
gmp-3.0.1-5
gnupg-1.0.4-9
gpm-1.19.3-4
grep-2.4.2-4
groff-1.16-7
gzip-1.3-6
hdparm-3.9-6
horde-1.2.4-3es
horde-mysql-1.2.4-3es
imap-4.7-1mdir4
imp-2.2.4-4es
indexhtml-7.0-2
info-4.0-15
initscripts-5.49-07es
ipchains-1.3.9-17
ipmasqadm-0.4.2-4
ip_masq_h323-1.0beta-5
ip_masq_icq-0.56-6
iproute-2.2.4-7
iptables-1.1.1-2
iputils-20001010-1
isapnptools-1.22-2
ispell-3.1.20-26es
kbdconfig-1.9.7-3
kernel-2.2.16-22
kernel-utils-2.2.16-22
krb5-libs-1.2.1-8
kudzu-0.72-4es
less-358-7
libjpeg-6b-13
libpng-1.0.8-1
libsmbpw-1.1-3
libstdc++-2.96-69
openssh-2.3.0p1-4
openssh-clients-2.3.0p1-4
openssh-server-2.3.0p1-4
openssl-0.9.5a-14
pam-0.72-37
passwd-0.64.1-4
patch-2.5.4-4
pciutils-2.1.8-8
perl-5.6.0-9
perl-Text-Template-1.20-2
php-4.0.3pl1-1
php-imap-4.0.3pl1-1
php-ldap-4.0.3pl1-1
php-mysql-4.0.3pl1-1
pidentd-2.8.5-3+masq
pine-4.30-2
popt-1.6-4
ppp-2.4.0-7
pptpd-1.1.2-2
procmail-3.14-5
procps-2.0.7-3
proftpd-1.2.0rc3-2es
psmisc-19-4
pump-0.8.3-2
pwdb-0.61.1-1
python-1.5.2-27
qmail-1.03-06
qmailanalog-0.70-2
qmail-initscripts-1.03.1-3
quota-2.00pre3-7
raidtools-0.90-13
readline-4.1-5
redhat-logos-1.1.2-3
rmt-0.4b19-4
rootfiles-7.0-4
rpm-4.0-4
rpm-build-4.0-4
rp-pppoe-2.5-1
rsync-2.4.4-1
samba-2.0.7-21ssl
samba-client-2.0.7-21ssl
samba-common-2.0.7-21ssl
sash-3.4-8
```

```
e-smith-dynamicdns-dyndns-1.2.0-02
e-smith-dynamicdns-dyndns.org-1.2.0-02
e-smith-dynamicdns-tzo-1.2.0-02
e-smith-dynamicdns-yi-1.2.0-02
e-smith-email-4.4.0-15
e-smith-flexbackup-1.2.0-05
e-smith-horde-1.2.0-08
e-smith-hosts-1.2.0-04
e-smith-imp-1.2.0-05
e-smith-ipmasq-1.2.0-02
e-smith-ldap-4.2.0-03
e-smith-lib-1.4.0-04
e-smith-lilo-1.2.0-03
e-smith-LPRng-1.2.0-07
e-smith-mod_ssl-1.6.0-03
e-smith-mysql-1.2.0-05
e-smith-named-1.4.0-08
e-smith-netatalk-1.2.0-04
e-smith-netlogon-1.2.0-03
e-smith-ntp-1.2.0-12
e-smith-obtuse-smtpd-1.2.0-02
e-smith-openssh-1.2.0-02
e-smith-packetfilter-1.2.0-02
e-smith-php-1.2.0-03
e-smith-pptpd-1.2.0-03
e-smith-proftpd-1.2.0-06
e-smith-proxy-4.2.0-02
e-smith-qmailanalog-1.4.0-03
e-smith-register-server-1.2.0-02
e-smith-reinstall-floppy-1.2.0-03
e-smith-release-4.1.1-1
e-smith-rp-pppoe-1.2.0-05
e-smith-telnet-1.2.0-03
e-smith-wu-imap-1.2.0-02
fastforward-0.51-03
fetchmail-5.5.0-3
file-3.30-7
filesystem-2.0.7-1
fileutils-4.0x-3
libtermcap-2.0.8-25
lilo-21.4.4-10
logrotate-3.5.2-1
losetup-2.10m-5
LPRng-3.6.24-2
lynx-2.8.4-4es
mailcap-2.0.9-2
mailx-8.1.1-20
MAKEDEV-3.0.6-5
man-1.5h1-10
mc-4.5.51-18
mingetty-0.9.4-13
minicom-1.83.1-4
mkbootdisk-1.2.8-2
mkinitrd-2.6-1
mktemp-1.5-5
mod_auth_external-2.1.2-6
mod_perl-1.24-6
mod_php-4.0.3pl1-1
mod_ssl-2.7.1-3
modutils-2.3.21-1
mount-2.10m-6
mtools-3.9.7-3
mt-st-0.5b-10
mutt-1.2.5i-3
mysql-3.23.32-1.7
mysqlclient9-3.23.22-3
mysql-server-3.23.32-1.7
ncompress-4.2.4-20
ncurses-5.2-2
netatalk-1.4b2+asun2.1.3-6es
net-tools-1.56-2
newt-0.50.17-3es
ntp-4.0.99j-7
ntsysv-1.2.16-1
obtuse-smtpd-2.0-23
openldap-1.2.11-15
openldap-clients-1.2.11-15
openldap-servers-1.2.11-15
sed-3.02-8
setserial-2.17-2
setup-2.3.4-1
shadow-utils-19990827-18
shapecfg-2.2.12-5
sharutils-4.2.1-7
sh-utils-2.0-11
slang-1.4.1-5
slocate-2.4-1
squid-2.3.STABLE4-1
sshell-2.0-3
stat-2.2-1
stunnel-3.10-2
sysklogd-1.3.33-8es3
SysVinit-2.78-10
taper-6.9b-3
tar-1.13.17-8
tcp_wrappers-7.6-15
tcsh-6.10-1
telnet-0.17-7
telnet-server-0.17-7
termcap-11.0.1-3
textutils-2.0e-8
time-1.7-12
tmpwatch-2.6.2-1.7
traceroute-1.4a5-23
utempter-0.5.2-4
util-linux-2.10m-12
vim-common-5.7-6
vim-minimal-5.7-6
vixie-cron-3.0.1-56
wget-1.5.3-10
which-2.11-4
whois-1.0.3-2
words-2-16
xinetd-2.1.8.9pre11-1
zlib-1.1.3-12
```