The Feasibility of Implementing Linux in a Small Business Environment

An Honors Thesis (HONRS 499)

By

Peter Mathias

Thesis Advisor
David Hua

David Hua

Ball State University
Muncie, Indiana

January 2006

Expected Date of Graduation
May 2006

Abstract

With small businesses being the backbone of the American economy, software developers need to be sure their products are suitable for use in a small business environment. While many server operating systems do exist, they tend to be exceedingly expensive. Free alternatives such as SUSE 10.0, a version of Linux, are available as alternatives. This document covers four common services offered by servers: Dynamic Host Configuration Protocol, Domain Name System, Samba, and Lightweight Directory Access Protocol. Researching and testing these four services shows that while SUSE 10.0 can be used in place of more expensive mainstream server operating systems, improvements need to be made, especially to the user interface. All of the services are explained in terms of purpose, function, testing results, and a conclusion.

Acknowledgements

-I would like to thank Prof. David Hua for taking time out of his busy schedule to advise me through this project. His knowledge and experience have been valuable throughout my entire college career.

-I would also like to thank Dr. Laurie Lindberg for checking this document for errors and showing me that Honors classes can be interesting and enjoyable.

Small businesses are the backbone of the American economy, providing a multitude of services in varying sectors. According to the United States Small Business Administration, based on data provided by the United States Census Bureau, approximately 22.9 million small businesses existed in the United States in 2002, representing 99.7 percent of all employers ("Small Business Statistics"). Small businesses often do not have access to the same resources as large businesses: a large business may have a separate department or branch with full-time professional employees dedicated to the computer and network infrastructure of the business, while many small businesses have one person, or even nobody, who is solely tasked with computer and network issues. The definition of a small business varies, but a common definition is one hundred employees or less. For the purposes of this document, the definition of a small business is an organization that has limited or no personnel dedicated to the computer and network infrastructure. In today's digital world, basic small business needs often include at least a simple computer network with services offered by servers.

Servers are available in a wide variety and offer a plethora of different services, from e-mail to file sharing to security. Microsoft Server 2003 is currently one of the most popular operating system platforms used on servers, but alternatives are readily available. One such alternative is SUSE 10.0, a distribution of Linux created by Novell. SUSE 10.0 is freely available to the public, meaning anybody can download and use it for free. For a small business, or even individuals, using SUSE 10.0 can save a considerable amount of money as compared to Microsoft Server 2003. The feasibility of utilizing Novell's SUSE 10.0 as a server operating system instead of Microsoft Server 2003 in a small business environment is the concept behind this document.

Four basic network services will be discussed: Dynamic Host Configuration Protocol, Domain Name Service and Berkeley Internet Name Domain, Samba, and Lightweight Directory Access Protocol. It is important to note that unlike Microsoft Server 2003, SUSE 10.0 is not meant to be a server operating system. Specifically, Microsoft Server 2003 is designed from the ground up to offer network services, while SUSE 10.0 is simply versatile enough to fill multiple roles, including the role of a server. In other words, directly comparing Microsoft Server 2003 to SUSE 10.0 is like comparing apples to oranges. The goal of this document is simply to determine if SUSE 10.0 is a feasible substitute for Microsoft Server 2003 in a small business, not a replacement.

Dynamic Host Configuration Protocol (DHCP) is used in computer networks large and small to make the lives of both the network administrator and the device users much easier. The purpose of a DHCP server is to automatically handle the assigning and releasing of dynamic IP addresses to devices on a network. If DHCP is configured and operating properly, all the device user needs to do is plug the device into the network and make sure it is set to use DHCP. Typically the device will contact and negotiate a deal with the closest DHCP server within seconds and will then be connected to the network and able to communicate. Without DHCP, a network administrator would have to go to each device that connects to the computer network and manually assign it an Internet Protocol (IP) address. A unique IP address is required in order for a device to communicate with other devices on a network. DHCP provides other information about the network to devices as well, such as the subnet mask, default gateway, DNS servers, and more. The IP address and other information are leased from the DHCP server, meaning that the lease must either be renewed or the device will lose its network access upon the expiration of the DHCP lease. The DHCP process is fairly simple.

When a device is first connected to a network, it checks to see if it has a proper IP address and subnet mask. If the device is not manually, or statically, configured with the necessary IP information and is instead set to use DHCP, the device sends out a broadcast message including its MAC address called a DHCPDISCOVER. If a DHCP server receives the initial broadcast message, it formulates a reply called a DHCPOFFER. The DHCPOFFER contains a valid IP address and subnet mask for the network as well as all other pertinent information, including the MAC address of the original device. Since the device still does not have an IP address to identify it, the DHCP server must broadcast the DHCPOFFER on the network. All devices that receive the broadcast must examine it and look at the destination MAC address. The device that broadcast the original DHCPDISCOVER will receive the responding broadcast, recognize the MAC address as its own, and accept the data in the DHCPOFFER. The device then broadcasts a DHCPREQUEST with the IP information it will use. The DHCPREQUEST is broadcast so that all DHCP servers that may have responded to the initial DHCPDISCOVER can learn whether their DHCPOFFER has been accepted or rejected. The chosen DHCP server responds with a final broadcast, the DHCPACK, to the device. The DHCPACK is broadcast because the device has not started to use the IP information it received yet. Upon receiving the DHCPACK, the device begins using the IP information and is now a communicating device on the network. Keep in mind that this whole process usually takes a few seconds at most depending on other variables (Eckert 100).

DHCP is a standardized protocol, meaning it operates in the same manner on all software and hardware. The DHCP client is installed with SUSE 10.0 by default, and installing the DHCP server is a simple process that involves retrieving a few files from the installation media. The process is automated and user-friendly. Configuring the DHCP options is an easy process that involves setting the range of IP addresses to be given to clients, the default gateway, DNS servers, WINS server, and other options. A setup utility is provided to facilitate the creation and configuration of a DHCP server within SUSE 10.0.

However, the utility may be too simple. There is no way to set up exclusions or scopes through the setup utility. Exclusions are IP addresses that the DHCP service is not allowed to lease out to clients. Exclusions are often used for devices that need to have the same IP indefinitely, such as servers and network printers. In the Microsoft Server 2003 operating system, scopes are the available IP address ranges to be given out by DHCP per subnet. A subnet is a smaller subset of a bigger network. Subnets are often used to group network devices based on physical or logical location, access privileges, and much more. Depending on the size of a business and its particular needs, it can be one simple network or a spider web of networks and subnets. One DHCP server can service multiple subnets by identifying the subnet of the source of the DHCP request and responding with an IP address from the corresponding scope. Since the SUSE 10.0 setup utility allows the entry of only a single IP address range for DHCP, the setup utility does not support the concept of scopes.

To use scopes in SUSE 10.0, the setup utility must be left behind. The specific configuration file must be manually edited with the required information. Unfortunately, the file to be edited does not contain any notes or explanation about its format, so outside research is necessary. The DHCP server must also have an interface in all of the subnet ranges that it will be servicing, meaning that the server will require multiple IP addresses, one from each subnet. This can be accomplished if there are enough network cards to provide one interface for each subnet, but this is not always feasible. If the server has only one network card that must be used for all of the subnets, multiple IP addresses must be bound to the real interface. Giving a single physical network card multiple IP addresses is achieved through the network card configuration utility. SUSE 10.0 can be used as a DHCP server, but the process to create and configure the server needs to be streamlined and standardized.

The Domain Name System (DNS) is the most widely used system for translating IP addresses to host names and vice versa. Even though most people probably have no idea what DNS is and what function it serves, DNS plays an integral part in allowing people to easily navigate the Internet. Almost all businesses have an Internet website to better serve their customers, and even individual people maintain personal websites on the Internet. People advertising or giving out the location of their website do not often use the IP address, however.

Most people have trouble remembering strings of numbers, such as an IP address, especially if they are trying to remember several. The use of a fully qualified domain name (FQDN) allows for a more memorable website name instead of an IP address. An FQDN is a host name, or computer name, followed by a DNS suffix, or domain. The main FQDN for Ball State University is www.bsu.edu, which is composed of the host name *www* and the DNS suffix *bsu.edu*. Without DNS, typing in www.bsu.edu in the address bar of an Internet browser would do absolutely nothing. The IP address of the server that hosts the Ball State University main website is 147.226.7.15. Opening a browser and typing that string of numbers into the address bar will open the same page as www.bsu.edu. DNS is responsible for the translation of www.bsu.edu into the IP address 147.226.7.15. The translation of an FQDN to an IP address is called a forward lookup, and is the most common service of DNS. The opposite function, finding an FQDN from an IP address, is called a reverse lookup (Eckert 158).

Here is an example of how DNS works: A student at an on-campus computer lab at Ball State University types in the Internet address www.bsu.edu in an Internet browser. The computer does not know the IP address for www.bsu.edu, so it sends a DNS query to the DNS server it is configured to use. Ball State University currently has a primary and secondary DNS server that all on-campus computers contact first. The local bsu.edu DNS server would certainly know the IP address for the web server at www.bsu.edu, so a DNS reply is sent back to the student's computer with the IP address of the *www* server at the domain *bsu.edu*. The computer contacts the server using the IP address it has just learned, and the student should now be viewing the Ball State University homepage. The above scenario is a simple forward lookup, but the process can be more complicated.

Now imagine the same student is at home in Indianapolis for a weekend and is trying to reach the www.bsu.edu website again. The student's computer does not know the IP address for www.bsu.edu, so it must send out a DNS query. The computer is configured with the Internet Service Provider's DNS server, which is often the case. The computer sends a DNS query for the IP address of www.bsu.edu to the ISP DNS server. If the ISP DNS server does not contain an entry for www.bsu.edu in its records, it contacts a root DNS server on the Internet and asks for the location of a root DNS server that can help resolve *.edu* domains. The root DNS server responds to the ISP DNS server with the IP address of the root DNS server that handles *.edu* domains. The ISP DNS queries the new root DNS server about the bsu.edu domain; the root DNS server responds with the IP address of the *bsu.edu* DNS server. The ISP DNS server sends a query about the host at www.bsu.edu to the bsu.edu DNS server, which responds with the IP address for www.bsu.edu. Finally, the ISP DNS server responds to the student's computer with the IP address for www.bsu.edu. The process of contacting other DNS servers when the local DNS server does not hold the needed information is called a recursive lookup. A recursive lookup is usually completed within a few seconds but can take longer.

SUSE 10.0 includes a setup utility for creating a BIND server. The Berkeley Internet Name Domain (BIND) is an implementation of the DNS protocol. BIND is used on a majority of the name resolution servers that support the Internet (Bodammer). BIND and DNS are compatible, though there are a few minor variations. The standard options are included, such as creating the forward and reverse lookup zones and the associated records, allowing and restricting zone transfers, starting and stopping the DNS server process, and more. Each page of the DNS server console includes a separate column on the left side that has examples and explanations of what to enter for each field. The explanations are helpful, but at the same time they can be confusing. The user must also enter the data precisely formatted or else the DNS server will experience an error. While allowing a granular level of control, users with little experience with DNS will most likely find the process puzzling. The DNS MMC found in Microsoft Server 2003 is comparable in its ability to confound an unknowing user, but Server 2003 also automatically formats entered values for the required fields.

Researching the task of creating a DNS server in SUSE 10.0, or any other flavor of Linux, leads to the conclusion that most system administrators do not use the configuration utility. As with DHCP, the general trend with DNS is to do the entire configuration at the command line and in the configuration files. For a person who is unaccustomed to using a command line interface and editing configuration files with frustratingly particular syntax and structure, this can be an intimidating step. For example, a single period out of place will cause the whole lookup zone to fail. Troubleshooting the problem can be time consuming, although there are utilities available on the Internet to aid with this function.

Once the DNS function is successfully running on the server, the end devices work as expected. BIND can handle DNS requests from end devices running the Microsoft Windows operating system with no extra configuration. The suggested method for using DNS servers is to have a mix of DNS and BIND servers in the network. Each type of name resolution protocol has its own weaknesses and vulnerabilities, so having some of both can lessen the impact of a vulnerability. For example, if all of the DNS servers fail because of a bug or attack, it is very possible the BIND servers will still be available for service because they are not susceptible to the same bug or attack. Further ways to protect the network from DNS and BIND attacks are to use secure zone transfers, passwords, and encryption. Although these methods and others are available, they are beyond the scope of the topic covered in this paper. A system administrator for a small business will not have use for such features in most cases. SUSE 10.0 can be used as a DNS server, but, as with DHCP, the process needs to be simplified for the everyday user.

Linux and Windows use different file systems by default, which makes interoperability a problem. With Linux controlling a fair share of the operating system market, especially outside of the United States, a means of allowing Windows and all of the various Linux distributions to communicate and share resources is a necessity. Samba is a popular piece of software that allows a Linux server to share files and other resources with Microsoft Windows end devices. Like Linux distributions, Samba is an Open Source project, meaning the application is free to the public and the program code is openly available.

Microsoft adopted the Common Internet File System (CIFS), which is based on the Server Message Block (SMB) protocol. In a nutshell, the purpose of CIFS is to share network resources among many end devices. Samba works by simply simulating the CIFS protocol on a Linux server. The Windows client devices are then able to share resources with file and print services, authentication and authorization, name resolution, and service announcements (Hertel).

File services allow multiple computers to access the same file space. For example, a file that multiple users need to share can be placed on the server so that it is accessible to all users. Print services let multiple devices share a local printer. Authentication and authorization handle security while also making it easier for clients to access all of their resources with a single log in. Name resolution takes on the role of a Windows Internet Name Service (WINS) server by accepting the computer name and IP address of devices on the network and answering name-to-IP or IP-to-name resolutions. Service announcements basically compile a list of the network resources available to clients. In Microsoft Windows, the list of available network resources is traditionally viewed through the My Network Places icon on the desktop (Hertel).

A configuration utility is included in SUSE 10.0 for Samba, but is not installed by default. Upon trying to access the utility, the system asks for the installation media to gather files necessary for the installation. The utility installs itself automatically and quickly. After installing the Samba configuration utility, a system administrator can set when to start the Samba service and specify the shared resources with detailed options for each shared resource. Creating the shares through the utility automatically edits the underlying configuration text files with the appropriate values. However, only the bare minimum of options is automatically added. Extra parameters can be defined for shares through the configuration utility, but explanations are not provided in the utility for each parameter. Once again, outside research is necessary. Other manual changes must be made as well. Each folder, file, and resource must be edited with the appropriate permissions, depending on who should have access to them. System administrators unfamiliar to the structure of Linux file permissions may encounter difficulties setting the permissions correctly.

While not as easy to create and configure as it should be, Samba is definitely a step in the right direction in terms of being user-friendly. With a minor amount of outside research, the Samba service was successfully running in far less time than either the DHCP or DNS services. System administrators comfortable with editing configuration files may still prefer to skip the configuration utility, but the utility does provide a working graphical alternative for novice system administrators. With all of the options set, the Samba service worked perfectly with a test end device using Windows XP. The test shares created were visible and accessible. Setting limiting permissions for users and shares also correctly affected access. No errors were encountered while installing or starting the Samba service. While not perfect, Samba is an excellent utility for sharing resources in a hybrid operating system environment.

Within an organization or business, the amount of important information to be stored, manipulated, and searched can be staggering. For example, keeping track of all of the employees' names, departments, e-mail addresses, phone and fax numbers, and much more can quickly become a complex and time-consuming task. As the organization or business grows, the task only grows more difficult. Microsoft's solution is the Active Directory system, which maintains a directory of information in a hierarchical structure. Multiple programs are able to access the information with the Active Directory for their own purposes, such as the search and retrieval of particular information, adding data, or even using the information for authentication and determining access privileges. At Ball State University, a large Active Directory system is used to store information for the thousands of students, faculty, and staff. The username and password each Ball State University student, faculty, and staff is given is actually a record within Active Directory; this is how all students, faculty, and staff are able to access various resources with a single username and password, such as e-mail, grades, schedules, and more. In all of these cases, Active Directory authenticates the user using the supplied credentials and either permits or denies access accordingly.

Active Directory is loosely based on another protocol, the Lightweight Directory Access Protocol (LDAP). SUSE 10.0 uses the OpenLDAP implementation of LDAP, which is a recognized standard. In line with Linux, OpenLDAP is freely available to the public both in use and in the availability of the source code. OpenLDAP serves mostly the same role as Active Directory, except that it is compatible with multiple operating system platforms, including Microsoft Windows, UNIX, and Linux.

The LDAP directory of information is stored in an organized, hierarchical manner, which allows for the timely retrieval of information. Besides a diagram, an example is probably the most effective way of explaining the structure of an LDAP tree. The topmost level of the tree is the most general category; for this example, that will be Ball State University. Underneath the topmost level are connected components of the higher level. Within Ball State University there are several colleges, such as the College of Applied Sciences and Technology, College of Sciences and Humanities, College of Architecture and Planning, and others. The example LDAP tree has Ball State University at the top with the individual colleges beneath it. The process of splitting a level into subcomponents continues as far as necessary. Within each college at Ball State University are several departments. For example, the departments of Industry & Technology and Nursing are two parts of the College of Applied Sciences and Technology. The process can continue, as there are specific majors and minors within each department, and even options within each major. Within the LDAP tree for Ball State University are entries for the faculty and staff. Each entry, called a Distinguished Name (DN), is composed of various defined attributes and their appropriate values. For example, the LDAP entry for the fictitious Professor

Jane Doe may include attributes for her telephone number, e-mail address, and office location. The values for those attributes may be something like (765) 555-1234, jdoe@bsu.edu, and CA 521, respectively. The Distinguished Name for Jane Doe includes all of the levels of the LDAP tree that relate to her, so the DN for Jane Doe in this example may be as follows: Jane Doe, Department of Industry & Technology, College of Applied Sciences and Technology, Ball State University. The actual syntax within LDAP is different, but the concept is the same. Once established, the LDAP tree can be modified and searched as necessary for pertinent information. If the president of Ball State University wanted to contact Professor Jane Doe, but knew only her department and could not remember her name, a search of the LDAP tree could be done to show all entries within a specific department. If Professor Jane Doe left Ball State University for another job, her LDAP entry could be removed and a new one created for her replacement. Because LDAP is an open protocol, various programs from different vendors are able to add, modify, and extract information from the LDAP tree. Done properly, LDAP can greatly increase the efficiency of managing information for businesses of all sizes (OpenLDAP).

The installation process of the LDAP modules in SUSE 10.0 is as simple as retrieving a few packages from the source media. The process is automated and painless. However, the ease of use ends there. LDAP is a complicated system that takes a great deal of research and time to understand. The necessary information is well-documented on the Internet, but the process is not inherently intuitive or straightforward. SUSE 10.0 does not include a built-in control panel for LDAP; all of the necessary commands related to LDAP must be run from a command line, which can be a frightening place for the unknowing. All of the commands and configurations worked without error, but the syntax of the commands and inputs are sometimes awkward. OpenLDAP includes several organizational schemes that cover basic templates for most structured organizations. Custom schemes can also be defined and included in the LDAP process, but defining a custom scheme takes extensive knowledge most likely well beyond the technical expertise of the average small business owner. Similar to a few of the other network services covered in this document, the SUSE 10.0 implementation of LDAP is functional yet in need of polish and an easy interface.

A more in-depth review of Linux as a possible alternative to Microsoft Windows as a server platform would involve many more aspects, but due to time restrictions, this document is limited in scope to the following services: Dynamic Host Configuration Protocol, Domain Name System, Samba, and Lightweight Directory Access Protocol. These services are some of the more common protocols used, but are by no means the only protocols used in a small business environment. All of the services tested operated correctly, but in general needed to be polished and made more user-friendly. The DHCP service ran fine once configured properly, but configuration was slightly troubling. Also, configuring the DHCP service to handle scopes was not intuitive and required extra research. The DNS service, like the DHCP service, operated as expected once configured. However, configuring the DNS service proved to be even more of a challenge than with DHCP. The slightest mistake in syntax could bring the entire name translation service to a complete stop. Samba was by far the easiest service to configure and operated without error. LDAP probably requires the most research to understand and implement, but it is a very powerful tool. Like DHCP and DNS, LDAP operated fine once configured, but configuring LDAP was the greatest trial. SUSE 10.0 does not include a graphic interface for LDAP, so configuration must be done from the command line.

Since all of the services provided worked correctly, no changes or improvements need to be made to their functionality. The biggest weakness that needs to be overcome in SUSE 10.0 is the interface for the services. While a graphical interface does exist for many of the services, they tend to be simplistic and of little aid for a novice user. Mountains of documentation are available for all of these services, but that can require hours of outside research. While Linux in general has made leaps and bounds towards becoming more accessible to the average user, it still needs more work.

Whether SUSE 10.0 can be used as an alternative to Microsoft Server 2003 and other pricey server-oriented platforms is a moot point. After researching various topics for this document, it became apparent that many people are using SUSE 10.0 as a server operating system, so indeed SUSE 10.0 can effectively be used as an alternative. The question addressed in this document is whether SUSE 10.0 is a feasible alternative in a small business environment where the in-house technical skills available may be lacking. A simple yes or no answer will not suffice. SUSE 10.0 handled all of the services tested well, but the designers must improve the interface if they hope to appeal to a wider population. In many cases, the best choice may be to contract an outside consultant to handle the installation and maintenance of the server if the technical skills are not available within the small business. However, this is probably true for any server platform, not just SUSE 10.0.

SUSE 10.0 is just one version of Linux; drawing a conclusion about Linux in general from this document would be inappropriate. There are literally dozens of other versions of Linux, all of which can vary greatly. Many Linux developers, including the creators of SUSE 10.0, also offer a server operating system based on Linux. While an enterprise version of SUSE 10.0 does exist, it is not free and therefore does not coincide with the premise of reducing costs in a small business by using alternatives to the mainstream server platforms.

Works Cited

Bodammer, Frank, et al. "SUSE Linux Administration Guide." Novell. Ed. Jorg Arndt, et al. 12 Feb. 2006 <http://www.novell.com/documentation/suse91/suselinux-adminguide/html/index.html>.

Eckert, Jason W., and M. John Schitka. MCSE Guide to Managing a Microsoft Windows Server 2003 Network, Enhanced. Canada: Thomson Course Technology, 2006.

Hertel, Chris. "Samba: An Introduction." Samba. 27 Nov. 2001. 8 March 2006 <http://us1.samba.org/samba/docs/SambaIntro.html>.

LinuxQuestions.org. 2006. 12 Feb. 2006 <http://www.linuxquestions.org>.

"OpenLDAP Software 2.3 Administrator's Guide." OpenLDAP. 9 Aug. 2005. 18 March 2006 <http://www.openldap.org>.

"Small Business Statistics." United States Small Business Administration. 26 March 2006 <http://www.sba.gov/aboutsba/sbastats.html>.

Vernooij, Jelmer R., John H. Terpstra, and Gerald Carter, eds. "The Official Samba-3 HOWTO and Reference Guide." Samba. 8 March 2006 <http://us1.samba.org/samba/docs/man/Samba-HOWTO-Collection>.