LinuxFocus article number 245
http://linuxfocus.org

Securing a heterogeneous network with free software tools

by Georges Tarbouriech
<georges.t(at)linuxfocus.org>

About the author:

Georges is a long time Unix user. He thanks the Free Software community for providing us with lots of great security tools.

Translated to English by:
Georges Tarbouriech
<georges.t(at)linuxfocus.org>

Abstract:

This article was first published in a Linux Magazine France special issue focusing on security. The editor, the authors, the translators kindly allowed LinuxFocus to publish every article from this special issue. Accordingly, LinuxFocus will bring them to you as soon as they are translated to English. Thanks to all the people involved in this work. This abstract will be reproduced for each article having the same origin.

Preamble

Security in computers networks is probably one of the biggest technology challenges of the 21st century.

However, like for many worrying fields, everybody talks about it, but the ones who should feel the most affected do not seem to have detected the scale of the potential disaster. The "most affected" of course, are the main software or system designers. The best example, once again, comes from Redmond, where security seems to be a word, at least much less "under control" than marketing, for instance.
Fortunately, the two last decades of the 20th century have seen the birth of Free Software and the philosophy going with it. If you "wish" to improve the security of your machines, your systems, your networks... this is where you will have to look for. The Free Software community has done much more about security than all the big software companies together.
That said, tools don’t make it all, and securing a network, for instance, is an almost permanent job: new

changes all the time!
This means you will never be able to say that a network is 100% secure. You can only reduce the risks. What we show here, is only a small part of what you can do to limit these risks. After reading this special issue (Author's note: remember, this article was part of a Linux Magazine France special issue focusing on security), you will know a bit more about security, but in no way will you be able to say that your network is secure. You have been warned.
Last but not least: such an article can't be exhaustive. There is a lot of literature on the matter and it is far from having gone round the problem. Accordingly, don't expect from this article to mention everything, as far as OSes, tools, configuration, use... are concerned.
To end with this preamble, let's add that some parts of this article are borrowed from LinuxFocus, but don't worry, with the author's agreement: it turns out to be one and the same person!

Presentation

First, we will talk about the structure of a very heterogeneous network, containing systems more or less widespread. The more OSes, the worse the complexity since not all systems are not equal in front of adversity. Furthermore, the machines used as servers should have different functions in a network: we will have a diversified network.
Next, we will go through a range of tools essential to improve security. The choice will be arbitrary: they are far too numerous to mention all of them. Obviously, we will explain how to secure machines and networks with these tools. The following chapter will review the features of different systems during the securing stage.
The conclusion will try to explain the "relativity" of the securing processes, to show why this a long way, without "diving" into futurology.

Example of an heterogeneous network

As a first advantage, the TCP/IP protocol is "spoken" by every OSes on earth. With it, very different systems are able to communicate with each other. Accordingly in the network we will use as an example, TCP/IP will always be present. In other words, we will not mention proprietary protocols, the least widespread nor the outdated ones. Neither will we talk about the physical structure, that is the type of connection, the category, etc.
So, in this network, we will put a bit of everything. Of course, we will find Unix, proprietary or free: for instance, a drop of Solaris 2.6, or SunOS 5.6, if you prefer, a drop of Irix 6.5, Linux (RH 6.2), MacOS X. We could have added a little bit of QNX or NeXTSTEP, or NetBSD or OpenBSD. On the "conventional" side we will include the one and lonely Not Terminated 4.0 (no, not any other, they are worse). Here too, we could have added OS2 which is less worse. Last, we will add a drop of "unconventional", let's say BeOS and AmigaOS (yes, it does exist... well, not much really!)
Of course, some of you are already complaining: what, no AIX, no HP-UX ? No! If we would like to mention every Unix, it would be a ten volumes article. However, the basic security rules are applicable to all the systems.

Now, what will we ask them ?

For example, let’s say Solaris will be an applications server. Irix will manage the backups. NT will be another applications server. Linux will be a gateway. Another Linux box will be an http server or a database server. All the other machines are clients. We will consider that this network contains about 30 machines using password file authentication. We could have selected a more sophisticated authentication: NIS (Yellow Pages) or LDAP or Kerberos... Let’s make things simple! Neither will we use NFS. Even if it can be helpful, when security is a concern, you better forget it, despite some improvement. In France, elderly people, used to say "don’t put all your eggs in the same basket". Then, the "uncertain" but required, services or protocols will be present only once, on machines doing nothing else. For instance, only one ftp server, one http server, preferably on Unix machines. Some Unix machines will be SSH servers and the other ones will be SSH clients. Back on this later. We will use static IP addresses: no DHCP. In other words, we will stay basic! This of course this can be applied to a 50 machines network: with many more machines, it could become a nightmare.

Tools and how to use them

As usual, there is more than one way to do it (TIMTOWDI). The ideal case would be to start from scratch, with machines to install and network to setup. But this is only true in films! Accordingly, let’s consider a network grown up over time, with machines moving from one place to another, new ones coming, and so on. Due to the Mhz "race", for instance, today Intel machines don’t last long. After about 3 years, it becomes quite difficult to find spare parts. Thus, either you recycle the machines to subsidiary tasks or you get rid of them: sad but true! Fortunately, some others last much longer and deserve to be improved. Don’t believe this is off topic: an administrator must work with high availability in mind.

The basics

We could call "generalities" the first step of the job. It consists in removing everything useless on every machine: not a "light" task! Each OS, Unix included, installs an incredible number of services, protocols, that you will never use. The master word is: throw them away! Under Unix, a simple... and rough way is to comment out everything in /etc/inetd.conf. That makes a few services less. Of course, this is a bit exaggerated, but on many machines it is perfectly acceptable. It depends on your needs. Under Linux and a few others you can also use the *chkconfig* command to deactivate some services.
Also check the SUID/SGID files and don’t hesitate in removing the "faulty" bit or consider deactivating the program. A command like: *find / -user root -a \(-perm -4000 -o -perm -2000 \) -print* will give you the list of those files. To remove the "s" bit, type *chmod a-s programname* (note: of course you loose some functionality by removing the "s" bit. It has its purpose after all).
Remove "dangerous" programs or the ones known as "risky": the remote commands such as rsh, rlogin, rcp... for instance. SSH will very well replace them.
Check the permissions for directories such as /etc, /var... The more restrictive the better. For instance, a command such as *chmod -R 700* on the directory containing the startup files (/etc/rc.d/init.d on many Unixes) is not a bad idea. The same rule applies to all the systems being part of the network: remove what you don’t use or, at least, deactivate. For NT, fell free to stop a maximum of services from the configuration panel. There are many basic "things" to do and there is a lot of literature on the subject out there.

The tools

Let’s begin with Unix, since it is the only one to really take security problems into account. Next, there is a huge quantity of free tools and most of them work on (almost) every Unix flavors.

For now, we will work on the individual machines since securing a network means, before everything, to secure its elements. Installing these tools is quite simple, that is why we will not spend time on the matter. Their parameters also depend on the systems, the needs... Up to you on how to apply this to your own case. The first required tool is called *shadow utils*. It is a means to do password encryption. Fortunately, it is part of many Unix distributions. The /etc/shadow file is then "created" from /etc/passwd.

Even better, *PAM* (Pluggable Authentication Modules) allows to restrict user access by service. Everything is managed from the directory containing the configuration files for each concerned service, usually /etc/pam.d. Many services can be PAM "driven", such as ftp, login, xdm, etc, allowing the administrator to choose who has right to do what.

The next tool is a must have: *TCPWrapper* . It also works on every Unix flavor or almost every. To make it short, it allows to restrict the access to services to some hosts. These hosts are allowed or denied using two files: /etc/hosts.allow and /etc/hosts.deny. TCPWrapper can be configured in two ways: either moving the daemons or changing the /etc/inetd.conf file. Later, we will see that TCPWrapper works fine in conjunction with other tools. You will find TCPWrapper at ftp://ftp.porcupine.org/pub/security

Another interesting tool is xinetd. Again, to make things short, xinetd is a replacement for inetd with much more features. According to what we above said about inetd, we will not insist. If you are interested, you will find it at http://www.xinetd.org.

Under Linux, there is one tool you can’t live without: it is called Bastille-Linux. You will find it at http://www.bastille-linux.org. This tool, written in Perl, is not only didactic but also very efficient. After running a script, you answer many questions and Bastille-Linux acts accordingly. Every question is explained and default answers are provided. You can undo the changes, start a new configuration, check what has been done... Everything is there! It also offers a firewall configuration: back on this later. At the time of this writing, Bastille-Linux is at version 1.1.1, but the version 1.2.0 is already available as release candidate. It is much improved, and provides a GUI based on Tk and its Perl module. (Author’s note: this article was written many months ago. As a matter of fact, the present version of Bastille-Linux is 1.3.0).

Intrusion detection systems are also essential. The two "heavywheight" are called snort and portsentry. The first one can be downloaded from http://www.snort.org and the second one from the Abacus website, http://www.psionic.com. Those tools should not be compared: the first one is an NIDS (Network Intrusion Detection System) mainly providing with information, while the second one can be considered host oriented and more active. snort has a lot of options to supervise the network traffic. You can listen to everything you want: incoming, outgoing, inside the firewall, outside the firewall. Of course, it then can create huge logs, but you must know what you want! A Win 32 version is available, it is important if we consider the number of free tools available on these "systems".

portsentry has a very interesting feature: it can block the scanned ports according to your choice. Either you redirect the attacker to an unused address or you redirect to the firewall. Of course, you can select who to block and who not to block. Now we can go back to TCPWrapper: portsentry is able to write into the /etc/hosts.deny file if you want to. Thus, portsentry becomes quite efficient. We will not get into the

debate about portsentry philosophy using port binding. It’s up to you: make your choice after going deeper into the subject. Also be advised that portsentry can make a machine "invisible", what is not bad! Last, portsentry can use different operating modes, the most advanced being "reserved" for Linux (at least for now).

We cannot talk about security without mentioning encryption. However, the law about it, is different from one country to another and sometimes it is completely forbidden to use encryption.
Author’s note: the following section has been removed from the English version of this article since it only concerns French law.
Conclusion: if your country allows encryption, install ssh clients and servers on your Unix machines (well, according to the needs!).

To finish with Unix tools, let’s mention the ones belonging to proprietary Unixes. Under Solaris, you have ndd, aset; under Irix, you can use ipfilterd. MacOS X provides you with some free tools: ssh, ipfwadm...
Back on this later.

Now, let’s talk about the one and lonely (fortunately!) Not Terminated 4.0. Here we cannot speak about free tools... however, the man from Redmond provides us with "free" stuff to improve the system features (it has nothing to do with bug corrections since there are no bugs!). Concerning security, NT 4.0 is a model... of absurdity. It’s a bit like a sieve! Never mind. Accordingly, you just have to download the latest service pack (6 at the time of this writing) and the HotFixes... which are security patches. Next... you can get some free tools (in the meaning of freely available and without the source code). That’s all.

For other systems you will have to search. For AmigaOS, development doesn’t seem to motivate much people and the TCP/IP layer is a bit old. However, Public Domain is still there to keep you busy.
Concerning BeOS, things are not better: this great OS seems to have a very compromised future and the network layer called Bone is still in the works.
(Author’s note: unfortunately, now BeOS is dead. A few people try to keep it alive as a free software product... and they do a very good job.)
But there too, you will find some tools from the Unix world to improve things.

Securing the hosts

Now, you will have to configure all this! Again, let’s consider that every Unix machine is "equipped" with shadow-utils, PAM, TCPWrapper, that every useless service has been stopped or removed, that permissions have been hardened on the "sensitive" directories, etc.

On the Linux machines, it’s time to launch Bastille-Linux. (This tool should work on most of the Linux distros, however, originally it has been designed for RedHat and Mandrake). Feel free to answer the questions in a very restrictive way.
On the Linux machine used as a gateway, the system must be "minimalist". You can remove most of the servers: http, ftp, etc. Remove X11 : you don’t need it! Remove the not needed software... that is, almost everything. Stop the useless daemons. You should get a system where a *ps ax* command won’t even fill the console screen. If you use IP Masquerading, the *lsof -i* command should display one line: the one concerning the listening server (we suppose that it is not a permanent connection).
Arbitrarily, we will install portsentry on the Linux machines and it will be launched at startup time, using the "advanced" mode (reserved for Linux, that is with -atcp and -audp options). This implies that

TCPWrapper and a firewall have been installed. Back on this later.

For Solaris, we will use the *aset* and *ndd* commands. More on this later too. portsentry will be installed as well. We could add IP Filter and repace the standard version of RPCbind with version 2.1 available from porcupine.org. For Irix, we will choose ipfilterd for packet filtering as the name says. It is part of Irix distributions but it is not installed by default.

Concerning NT, things get a bit more complicated... The "fascist" solution consists in blocking ports 137 and 139, that is the famous NetBIOS (or even better removing NetBIOS)... but then no network is left (that is Windos network) it can be a small problem when it concerns an applications server! You can also install snort but it will not prevent those machines from being like sieves. Accordingly, you will have to be very restrictive about partition access, directory access... as soon as you work with NTFS partitions, of course. There is a freely available program to get rid of the guest account but the source code is not available. Then, install all the security patches you can find! Last but not least, roll-up your sleeves and try to make that thing less vulnerable. It is a bit like to go round an assault course but it is compulsory.

For the "exotic" OSes, you will have to search and choose. As usual, and before all, the basic rules should be applied: the less active services, the better.

Protecting the network

If the hosts have been properly "prepared", you are half the way. But you will need to go further. Since we are talking about free software, we will choose a free firewall for the gateway: well, it is the machine allowing you to access the "wild" world. Arbitrarily (again!) we use a Linux box: so we can use the Bastille-Linux firewall. It works with ipchains or ipfwadm according your kernel version. If you use a 2.4 kernel, it will work with iptables.

A small digression: it is not a good idea to have all the initial problems to put up with, when security is a concern. The "race" to the latest kernel version may lead to a very negative situation. This does not mean that the work on new kernel is not a good, however, the "marriage" with existing tools, not designed to work that way can be a big mistake. An advice: be patient! The new firewall tool, part of the 2.4 kernel is very promising but probably a bit "young". That said, it is up to you...

So, the Bastille-Linux firewall is both simple and efficient. However, there is a much more elaborated tool, a bit like a "gas factory", called T.REX. It is available from http://www.opensourcefirewall.com. If you look for a very sophisticated free tool, here it is.

Other solutions exist, such as proxys, however they are not always better. Another digression: proxys are often called "firewalls". Nevertheless, they are two very different things. The firewalls we are talking about use packet filtering and do not provide authentication method. There are two types of proxy servers: applications or socks. In short, an application proxy does the job for you managing the entire communication and it allows for user authentication. This is why it needs much more resources than a firewall. But, again and again, this sort of tool only protects for a short lapse of time. A firewall can be "cracked" in about 15 minutes. Good to know, isn’t it ? Hence the need to properly secure the hosts in your network: deciding to secure a network only relying on a firewall or a proxy is an heresy!

Another method to reduce the risks in a network is encryption. For example, using telnet is like making crackers walk on a red carpet. It is a way to give them the keys of the shop. Not only can they see the circulating data, but even better, that get the password in clear text: nice, isn't it ? Accordingly, feel free to use ssh with the "uncertain" protocols (or instead of). If you MUST (?) use telnet, send the data through a secure connection. In other words, redirect the telnet port to a secure one. You will find more on this in the article titled "Through the tunnel" (LinuxFocus, May2001, article 202). (Free ads!)

OK, we tried to improve security, but now we should check our work. To do this, let's become "crackers", sort of: we will use their tools. Ugly, isn't it ? In this area too, there is a nice collection of programs, then, again arbitrarily, we will choose two of them: nmap and nessus. There is no redundancy, since, for instance, the second one requires the first one. These tools are port scanners, even if nessus is much more than this. Nessus informs you about system vulnerabilities, comparing the scan results to its vulnerabilities database. Running these tools in a network will allow you to discover each host's weaknesses, whatever the OS is. The results are quite revealing thus making these tools a must have. You will find nmap at http://www.insecure.org and nessus at http://www.nessus.org.

From the beginning of this article we are talking about securing a local network in which some machines are opened to the external world. An Internet Service Provider case obviously would be quite different and we will not get into the many details of the subject. Let's say that all we mentioned is still available but you will have to use much more elaborated methods, such as VPN (Virtual Private Network), LDAP for authentication (for example), etc. It is almost another subject since constraints are much more numerous according to the case. Let's not talk about e-business sites, where things are reckless. Secured sites they say! Don't tell me... Do you send your credit card number through the Internet ? If yes, you are very courageous. Suggestion: if you can read French have a look at this website http://www.kitetoa.com, it is worth it.

Systems particularity

As already mentioned, systems are not equal when in front of the enemy. Some have very good abilities while others are sieves. Paradoxically (well, not really!), free OSes are among the better. The different BSD's (OpenBSD, NetBSD, FreeBSD...), the different Linuxes are quite ahead when security is a concern. Again, it is the result of the great work from the free software community . The others, even Unix labeled, are a bit less advanced. When they are not Unix, it is much worse!

All the tools mentioned in this article have been developed for free OSes. Most of the proprietary Unix systems can benefit from them. However, these proprietaries OSes often have their own tools. For instance, concerning Solaris, we mentioned *ndd* and *aset*. Despite a widespread idea, Sun systems are not security models. A tool such as aset, allows to improve things as far as access rights are concerned. aset offers three protection levels: low, medium and high. You can run it from a shell or from a cron task. In a running network the situation changes, what was true at 5pm may become false at 5.30pm. Hence the interest to run commands periodically to keep some homogeneity. This is why aset has the ability to be cron managed. Thus, it will check every 30 minutes or every hour, or whatever you want, the permissions of directories, files...
ndd, allows to change the IP-stack parameters. For instance, it can be used to hide the system fingerprints. An identified system is a more vulnerable one, since the crackers know better where to

"strike". With ndd, you can change the TCP Maximum Segment Size (MSS). By default, this size is 536 under Solaris 2.6. The *ndd -set /dev/tcp tcp_mss_def 546* command changes it to 546. The higher MSS is, the better (not too much!). Nmap, for instance is able to find this weakness. Using ndd, you cut the ground from under its feet. If you have machines running Solaris, feel free to use ndd. There are many options: check the man page.
You can also use IP Filter, a packet filtering tool. It is available from ftp://coombs.anu.edu/pub/net/ip-filter.

Concerning Irix, the situation is again different. SGI (ex Silicon Graphics) , as the name says, designed its systems for graphics. Security was not the main concern. Necessity knowing no law, it became compulsory to provide ways to reduce the risks. ipfilterd was then provided in Irix distributions, but it is not installed by default: you will have to look for it ! ipfilterd, is of course used for packet filtering, thus allowing to deny access to who you want. It relies on a configuration file called ipfilterd.conf and this is where things become a bit tricky. The syntax of this file is rather peculiar and does not like unexpected spaces or empty lines. Thus, to allow the machine called "mars" to talk to the machine called "jupiter" (which is the SGI workstation), you will have to type a line looking like:
accept -i ec0 between jupiter mars
The machines not listed in this file will not be able to access jupiter. Even worse: if you do not change the *ipfilterd_inactive_behavior* parameter using systune, nobody will access the machine! Efficient, isn't it? This parameter defaults to 1, and you will need to change it to 0 using the *systune -i ipfilterd_inactive_behavior 0* command.
Another well known thing, better to remind, Irix has a "great" vulnerability, called fam (File Alteration Monitor). This program is in charge of a very nice feature, the communication between various daemons. For example, it is the one allowing to get beautiful icons in the file manager. Nevertheless, there is only one thing to do: deactivate it! Sad, but it is like that.

To end with Unix systems, let's mention that QNX is very vulnerable but it can of course benefit from free tools. Mac OS X already provides some of these tools.

We must talk a bit of the absolute reference among network systems: the one and lonely NT 4.0. Securing that thing is an utopian view, despite what the King of Redmond (and many others) says. Simulating an attack with nessus, for instance, will be a nightmare. As far as NetBIOS is active, nessus will provide you with the names of every machine in the domain with their corresponding users, including administrators. The answer is: get rid of NetBIOS! Right, but as already mentioned, no NetBIOS, no network... You will have to choose your side.
nessus will kindly inform you that it can login as the guest user with a NULL session (that is with a NULL username and a NULL password). Remove it, then ! Yes, but how ? And it is all like that!
So, reduce the access to partitions (NTFS), to directories. For FAT partition... no solution. However, according to the software you use you may need FAT partitions: some software will not work on NTFS.
To end with it, avoid the great IIS, especially as ftp server. In fact, don't install it. If today, so many ISP are mad enough to use that thing, we just can suggest them to use Apache instead, but... Don't we spend too much time on IIS, there is a lot of literature on the subject.
As a matter of fact, there is a way to make the sieve become a filter (holes are smaller!). The problem is that it is rather a long way and the whole magazine would not be enough. Let's only mention the most important. Obviously, the point is not to secure with free software: we are talking about the Microsoft world! The first suggestion is to use MSCE (Microsoft Security Configuration Editor) available from ServicePack 4 with MMC (Microsoft Management Console). However, be extremely careful! If you make a mistake, you have won. Of course, this software is an English version. If you use a foreign (not

English) version of the system, be advised that the mixup of languages never gave very good results in the Redmond world. You have been warned. Next, among the required measures, you must "secure" the administrator account, or even deactivate it. Have a look at *passprop* available from the SP 3. You can also harden the passwords using the *passfilt* dll through the registry (I always thought that people who invented that thing were under LSD influence...). Deactivate the famous guest account. It is not very useful (see above), but it makes things less worse. But, you can restrict its access to the logs from the registry. In "HKEY_LOCAL_MACHINE", create the keys *System\CurrentControlSet\Services\EventLog\Application*, Security and System (these two last should replace Application). Their name is "RestrictGuestAccess", the type is REG_SZ and the value is 1. You can encrypt the passwords with *syskey*. Careful, it is an irreversible operation! At least, some good news: you can restrict the guest access. Again, let's play with the registry, still in "HKEY_LOCAL_MACHINE". This time the key is called *System\CurrentControlSet\Control\Lsa*. The name is "RestrictAnonymous", the type is "REG_DWORD" and the value is 1. However, Microsoft world is a teaser: be advised that this change may alter some network services... Among the important things, you can restrict the access to some ports, using the Network application in the configuration panel. From the TCP/IP properties, select "Advanced" and check the "Activate security" box (I believe that this its name, but I don't have this kind of thing at home to be able to check). From the "Security" window, check "Allow only" and select the ports you want to activate. Here too, be careful. You should know what you are doing, otherwise some services will not work anymore.
A lot more can be done, but these are the essential. To learn more, you can visit sans.org: tons of documents are available.

The unbearable lightness of things

Well, you have done all this. You run nessus to scan the whole network and you still get security holes. We will not say where they come from... we already know! Try to delude these system substitutes. It will not remove the holes "provided" by NetBIOS, but it will limit the damage. Create subdomains. Don't login as an administrator. Apply patches. Last, try to hide all this behind Unix machines used as gateways. Unfortunately, the relativity of security doesn't only come from products made in Redmond. A network is alive: there is always something going on. A good administrator is a "paranoid" one, accordingly, often check the "inventory of fixtures". Write scripts to automate the checks. For instance to control on a regular basis the SUID/SGID programs, the critical files, the logs... To get a few more friends, lock the users floppy or CDROM devices. Don't accept that users download software without your agreement, especially when this software is executable like always in the Microsoft world. Prevent your users from opening attached documents like those in Word or Excel format using a mail filtering system. Yes, I know it is like fascism, but what can you do against macro-viruses ? Do not use products such as Outlook. Once again, you must know what you want! I know, what I say is useless, but can you talk about security with such products ? The famous "I love you" did not teach any lesson.
Concerning Unix, downloads must be controlled as well. Checksums have not been provided by accident.

Get the habit of controlling your network on a regular basis with logs, scripts, scans... You will notice: things change quite fast and not only in the good way.
Last, we did not say a word about it, but don't forget backups. The strategy is unchanging: daily, weekly and monthly. An Unix machine can also have problems, even if it is unusual. And, sometimes the users make mistakes... but not very often. It is well known that the problems come from the machines or from

the department in charge of them:-(

At least, it is over!

If you reached this section is that you are courageous. The problem is that we only skimmed over the subject! Security has no end and doesn't only concern networks. Vulnerable applications can compromise a network. A badly configured firewall is far more dangerous than no firewall at all. An Unix machine often holds thousands of files. Who can be sure that none of them is vulnerable ? Who thinks a cracker will try to break a 128 bits key ? Don't be fooled: he will try to find a door behind the house. Again and again, you can install all the security tools available, if you leave a very small hole, this is where the "bad" will go through.

Security is also a behavior: follow what is going on. For example, visit the security websites on a regular basis, same for the websites of your OSes editors... For example, Sun publishes recommended patches every month. SGI releases a new Irix version every three months. Microsoft frequently provides ServicePacks or HotFixes. Linux distributors publish erratas for each newly discovered vulnerability. Same for the different BSD's. If you don't use the products corresponding to a patch remove them from your hard disk. And so on: the list of things to be done is a very, very long one. In short, this job should not know lay-offs.

Last, let's say it again, all this will only contribute to make your network a bit less vulnerable. Don't expect you will get a 100% secure network, even at a given time (well, may be, if all the machines are stopped). That said, it is not a requirement to be paranoid to do this job... but it helps ! But don't be like like that in your everyday life, it will be much nicer for the people around you...

References

- http://www.linuxsecurity.com
- http://www.sans.org
- http://www.infosyssec.org
- http://www.securityfocus.com
- http://www.cs.purdue.edu/coast/hotlist/

Life is sad: let's have some fun!

Another way to do the job ;-)

Translation information:
fr --> -- : Georges Tarbouriech <georges.t(at)linuxfocus.org>
fr --> en: Georges Tarbouriech <georges.t(at)linuxfocus.org>

2005-01-14, generated by lfparser_pdf version 2.51