DOWNLOAD PDF SUSE LINUX 9 BIBLE

Chapter 1 : suse linux 9 bible - TÃ i liá»‡u text

SuSE Linux 9 Bible gives good detail on installation using YaST. The book goes through disk partitioning, the fundamentals of the Linux System, RPM packages, networking, administration and desktop customization.

Taking all of these into consideration, you can define a rule that describes a very specific sce- nario for a network connection. Putting numerous rules together, you can produce a very powerful firewall. With the introduction of iptables, we were given the godsend that was stateful firewalls. They are rela- tively simple to use and extremely powerful when done correctly. All kudos to Rusty Russell the lead iptables developer for implementing this feature as it allowed us to produce tight firewalls with fewer rules. We will talk about stateful firewalls and what they do in this chap- ter, as well as a few scenario-based iptables rules. Why Use a Firewall? A firewall, whether Linux-based or not, should always be used to protect machines connected to the Internet. A firewall, by its very nature, is designed to control what can be accomplished over the network, and it is very unlikely you want your Windows machines to be connected to the Internet in full view of any malicious person that comes along and bare Windows machines on the Internet are like drops of blood in a mile radius of a pack of sharks! Most people think that a firewall is there to stop crackers from the Internet, but the fact of the matter is that your users are untrusted, too. It is all well and good to trust your users when you have security checked them and have run psychoanalytical tests to see if they have a predisposition for breaking the rules you have imposed on them. Take the following example. For their internal users, there were no restrictions on connections to the Internet. All users were trusted and all good guys. Their email and operating system on the other hand were not, and they started receiving emails with viruses that arbitrarily scanned thousands of hosts on the Internet to carry on propagating throughout the ether. The customer found this out only because their Internet service provider ISP called them to say their connection would be closed if the scanning did not stop. This virus came through email to the user, and because Simple Mail Transport Protocol SMTP traffic was allowed through to the mail server, there was nothing to stop it. This is an important point. We used the logging facilities of iptables to track the source of these problems, and we pro- ceeded to remove the virus the customer subsequently installed virus scanners on all machines. To combat these internal problems in the future, we tightened the security of the organiza- tion from a network standpoint. We restricted what could be accessed on the Internet from the internal network apart from the essentials. This stopped port scans from exiting the net- work and stopped most incarnations of virus transmission over Internet protocols. It is used not only by crackers, but also by legitimate users who wish to see what services are available on a server. You should port scan only hosts that you have been allowed to interrogate. Port scanning a machine usually triggers alarms on a system, and you may get into trouble depending what the administrator is feeling like that day. This example fully illustrates that network security must be considered as a whole, not just as a threat from the Internet. Configuring a Firewall with iptables To configure a firewall on Linux, you need to get used to the iptables command, which is used to manipulate the kernel packet filtering settings from user space. The terms user space and kernel space are used a lot in the Unix community. When some- thing runs in kernel space, it is under the control and the constraints of the kernel. Something running in kernel space could be a kernel module or the packet filtering code. When something is in user space, it uses the system libraries and is not under the strict con- trol of the kernel. The kernel filtering code uses chains to signify where a packet is in the kernel. This also helps us to see how iptables interacts with these packets later in the chapter. The chains themselves represent the final destination of the packet: Consider these examples to show how the chains work in a normal firewall: When setting up a firewall appliance, you need to enable IP forwarding. IP forwarding allows packets to be routed from one network interface to another in the Linux machine. Most iptables firewalls that protect a network run on low-cost, low CPUâ€”powered hardware. When the firewall receives the packet, it analyzes it to find its destination. The important part of the scenario is that any non-local packets destined or originat- ing from the machine are passed to the forward chain for forwarding! In the same way that the packet will reach the firewall as in the for- warding example, the kernel analyzes the packet to see where it is destined. As my machine is the final destination for the packet, it is inserted into the INPUT chain for further processing. If the packet is allowed through, it is passed over to the kernel to be handed over to user space which is normal when no firewalling is used. Implementing an iptables firewall As a general rule of thumb when talking about network security, you should deny all and allow some. You can set this policy to drop all packets that do not trigger a rule that is, are not explicitly allowed. The Default Filtering Rules bible: In the past, it was up to the administrator to track all connections through the firewall, which produced a lot of rules that were difficult to manage. With a state- ful firewall, netfilter keeps a record of connection states. With this information, netfilter can track a connection initiation and match up related network traffic. For example, previously, if you wanted to allow an incoming connection to SSH on the fire- wall, you had to first allow the incoming connection and also the return traffic from the SSH server to the client. With stateful firewalls, you can tell the firewall to manage the subsequent outgoing connection automatically because it is aware that an incoming connection to the machine will produce traffic in the opposite direction. It does this by storing the state of a connection and acting upon it with connection tracking. To enable the stateful connection tracking, you need to enable states in the firewall. We dis- cuss this in a small firewall script later in the chapter. Setting your first rules Before you touch upon setting more specific rules, you need to set the default policy for the firewall and enable some state rules see Listing Setting Initial Firewall Rules bible: At this moment in time, all network connections, regardless of their originating address, will be dropped. If they have to wait for a timeout on each port, it will take them quite a few hours to complete the full scan. It provides a kind of tar pit for any malicious users. This is also true for internal connection, too. If your users are interested in what they can and cannot connect to, without reading the network rules, then making them wait will, one hopes, deter them from pushing the network too hard. You have also configured the stateful firewall with the -m state declaration. This tells the fire- wall that you will allow any established or related connections on the INPUT chain. This may seem like quite a big security hole, but bear in mind that it will allow only a connec- tion that has been established, not a new connection. For the stateful rules to kick in, you would have already had to allow a new connection through the chain. At this point, your firewall is locked down with the exception of allowing outgoing connections. Now, suppose you want to allow an incoming SSH connection to the firewall. At a minimum, you need the chain, protocol, and destination port. With just this information, you do not have a very good rule because it does not specify the interface you are allowing the SSH connection to. Another option that can be set is the connection type: When setting up a rule for connections, you really need to know how the protocol works. With this in mind, it is relatively easy to write a rule for it. When you do not specify something explicitly with an iptables rule, it is assumed that you want the default setting. For example, if you did not set the interface for the incoming connection, netfilter would have allowed an SSH connection on all network interfaces. This is indeed the same for the protocol type and the destination port. Be very careful how you write your rules, and make sure you explicitly set everything you wish to control; otherwise you will probably let in more than you think. If you wish to insert a rule at the top of the list that is, making it the first rule that is executed , you can use the -I insert parameter to iptables. If none of the rules fires off a packet to a target, that packet is dealt with by the default policy, which is to kill the packet in this case. Network Address Translation While one of the main uses of netfilter is its packet filtering functions, another very impor- tant aspect of netfilter is its NAT functions. Network Address Translation NAT is the process whereby the source or destination IP address of a packet is seamlessly changed when it passes through the firewall. Chapter 6 contains some more information about NAT. This drastically reduces the cost of acquiring public IP addresses and allows you to use non-routable addresses in your internal network. This includes any packets that are routed onto other destinations. Network using a netfilter firewall In this scenario, all of the machines are behind a netfilter firewall that not only protects the machines, but also provides SNAT for outgoing connections. This is a volatile opera- tion, and once your machine has been rebooted, IP forwarding will be turned off by default. It is a great feature, but unfortu- nately is not in widespread circulation and can stop your network traffic from traversing the Internet correctly if it goes through a router that does not support ECN. We have been on customer sites where their network just stopped working for certain sites for no reason. Turning off ECN fixed this. In the home network, you need to source NAT all the internal traffic This includes source and destination address translation. We have stated that any traffic from the In the example, note that we have tried to be as descriptive as possible concerning what traf- fic should be subject to the SNAT, detailing the source IP address specifying the network address with netmask and the network adaptor that the traffic will leave on. This can be through the eth1 interface only. Any traffic that is sent back to the machines behind the firewall for example, during the three-way handshake will be translated back by the firewall it remembers connection states and the destination address will automatically be set to the address of the machine on the private network that initiated the connection. To correct this, you need to allow the firewall to forward these packets before they can be manipulated by the SNAT rule. To do this, you need to enable forwarding for traffic from the private network to the Internet: Any traffic from the So, in this example, we have told netfilter that any traffic from the Again, we are relying on the fact that any traffic coming in on eth0 and leaving on eth1 that is from

Chapter 2 : Books - Museum | SUSE

SUSE Linux 9 Bible is that guide -- and, as a bonus, it comes with SUSE Linux Professional on DVD. To begin with, here's all you need to know about installation: from partitioning to boot options, changing runlevels to configuring hardware.

Chapter 3 : - SUSE Linux 9 Bible by Justin Davies; Roger Whittaker; William von Hagen

** SUSE is the leading Linux distribution in Europe, with a strong enterprise presence and reputation as the most secure Linux distribution * Written by two SUSE insiders, this book explains the best way to carry out a task while making full use of SUSE's configuration utilities and unique YaST modules.*

Chapter 4 : suse linux 10 bible - TÃ i liá»‡u text

Part I SUSE Linux Basics 1 --Part II The SUSE System 95 --Part III Using the Command Line in SUSE Linux --Part IV Implementing Network Services in SUSE Linux --Part V SUSE Linux in the Enterprise --Appendix B About SUSE Linux Professional Version --GNU General Public License

Chapter 5 : SUSE Linux 9 Bible : Justin Davies :

Email to friends Share on Facebook - opens in a new window or tab Share on Twitter - opens in a new window or tab Share on Pinterest - opens in a new window or tab.

Chapter 6 : SUSE Linux 9 Bible - PDF Free Download - Fox eBook

If SUSE Linux 9 can do it, you can do it, too SUSE is the oldest commercial Linux distribution, favored in Europe and rapidly gaining popularity in the United States. Whether you're just discovering Linux or switching from another distribution, this comprehensive reference gives you not only the.

Chapter 7 : wiley publishing suse linux 9 bible pháº§n 9 ppt

Download opensuse 11 0 and suse linux enterprise server bible ebook free in PDF and EPUB Format. opensuse 11 0 and suse linux enterprise server bible also available in docx and mobi. Read opensuse 11 0 and suse linux enterprise server bible online, read in mobile or Kindle.

Chapter 8 : [PDF] Opensuse 11 0 And Suse Linux Enterprise Server Bible Download eBook for Free

** SUSE is the leading Linux distribution in Europe, with a strong enterprise presence and reputation as the most secure Linux distribution * Written by two SUSE insiders, this book explains the best way to carry out a task while making full use of SUSE's configuration utilities and unique YaST.*

Chapter 9 : Red Hat Linux 9 Bible Simple Step Faster Received

trator post-SUSE, he joined the world of the value-added reseller and now works for SCC as an enterprise solutions architect, helping organizations realize that Linux is a viable business solution.