SSPE: Managing security policies for multiple firewalls

CENTRAL CONTROL

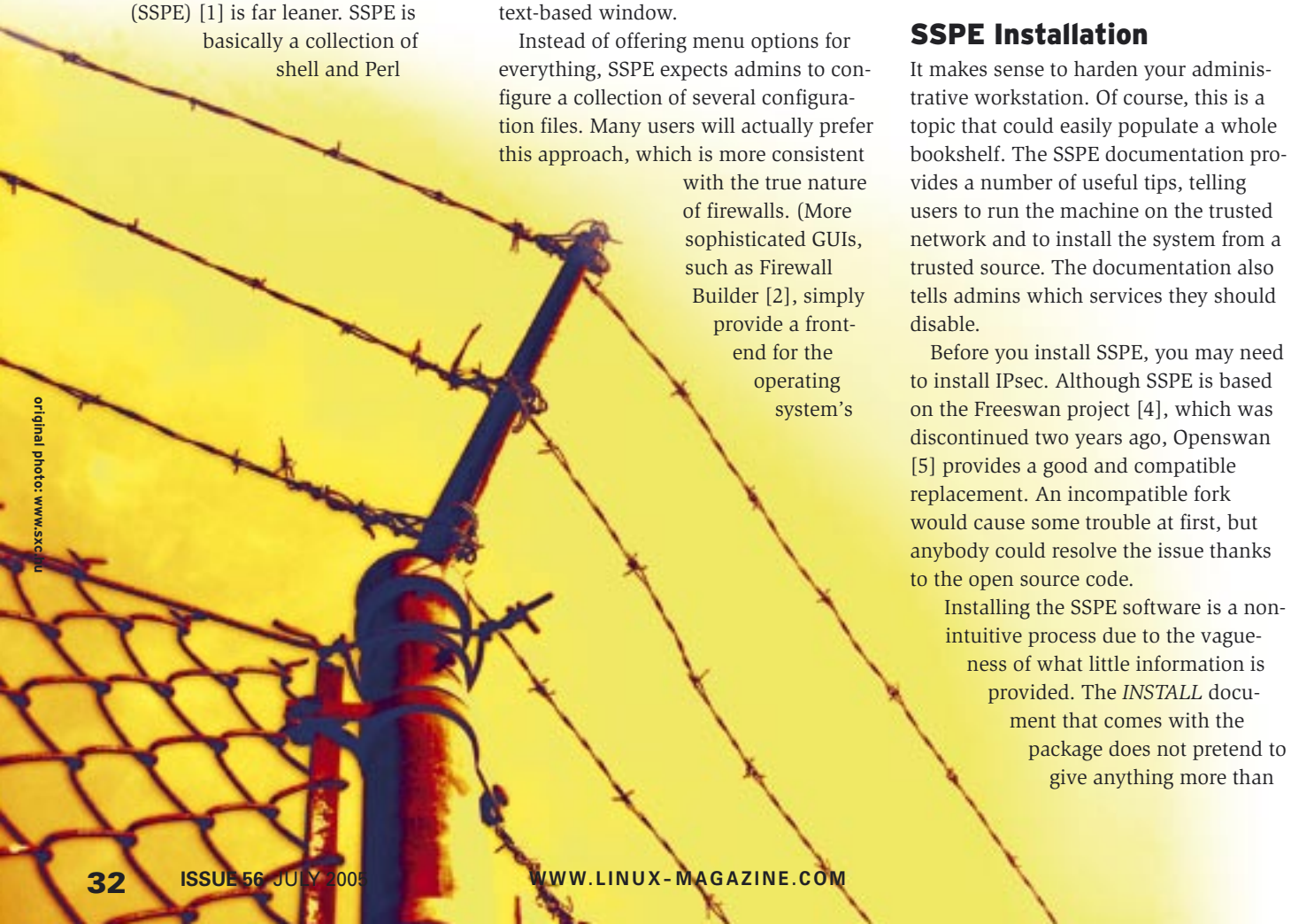
The Simple Security Policy Editor (SSPE) helps you organize your network and keep track of security policies across multiple firewalls. You can reference a central policy to generate rulesets for packet filters and VPN gateways. **BY CHRISTIAN NEY**

Many networks require multiple firewalls. Firewalls guard the Internet gateway, separate the departments within an enterprise, and help to connect remote sites through VPN technology. In this scenario, it helps to have a central tool for managing firewall security policies. The pioneer in this field was the Checkpoint Firewall-1 GUI [3]. But this commercial package tends to be overcomplicated, with an overabundance of features – not all of them useful – adding to the tool's complexity.

The free Simple Security Policy Editor (SSPE) [1] is far leaner. SSPE is basically a collection of shell and Perl scripts that help admins manage a distributed firewall environment.

When first launched, SSPE comes up with a plain dialog-based interface ([6] and Figure 1), but the deliberately spartan front-end conceals a surprisingly powerful tool. The text-based interface includes support for remote administration – you can use a modem or cellphone to dial in, although the tool does struggle with lame X11 forwarding. This text-based interface does not mean you have to do without a mouse, however; depending on your terminal emulation, you may be able to point and click in the text-based window.

Instead of offering menu options for everything, SSPE expects admins to configure a collection of several configuration files. Many users will actually prefer this approach, which is more consistent with the true nature of firewalls. (More sophisticated GUIs, such as Firewall Builder [2], simply provide a front-end for the operating system's command line tools, hiding their complexity from the user.)



original photo: www.sxc.hu

Holistic Approach

Because every additional feature is a potential security hazard, doing without a complex GUI improves the security of the administrative workstation. Even if a potential security problem does not affect the firewall directly, the danger is only too real, as an attacker could manipulate the rules that the admin later uploads to the firewalls.

SSPE's author stipulates a minimal Debian installation, which includes the packages SSPE needs: Bash, Dialog and Perl. But the minimal requirements mean that any Linux should be fine as the basic system. Our tests showed that Debian Woody, the upcoming Sarge release, the Debian derivative Ubuntu, Gentoo, and Red Hat Linux were perfectly suitable.

SSPE Installation

It makes sense to harden your administrative workstation. Of course, this is a topic that could easily populate a whole bookshelf. The SSPE documentation provides a number of useful tips, telling users to run the machine on the trusted network and to install the system from a trusted source. The documentation also tells admins which services they should disable.

Before you install SSPE, you may need to install IPsec. Although SSPE is based on the Freeswan project [4], which was discontinued two years ago, Openswan [5] provides a good and compatible replacement. An incompatible fork would cause some trouble at first, but anybody could resolve the issue thanks to the open source code.

Installing the SSPE software is a non-intuitive process due to the vagueness of what little information is provided. The *INSTALL* document that comes with the package does not pretend to give anything more than

Figure 1: SSPE's main menu is uncluttered. The simple text-based interface allows admins to remotely control the tool.

installation tips and expects security administrators to know what they are doing. When the tips run out, you may find yourself sitting in front of an empty screen and not getting the kind of descriptive error messages you need for troubleshooting. The cause can be as simple as a missing directory, or you may just be in the wrong path. But the scripts do not catch these error conditions.

Admin Account

You will want to avoid running any kind of security-critical software as the root user, and SSPE only needs a normal user account. You need to add a directory for SSPE below the user's home directory; the documentation refers to this directory as *adm*. As some scripts expect to find the directory below *$HOME/adm*, you should keep to the recommendation. The next chore is to create a directory tree below the administrative directory; you can do this quite easily by copying the subdirectories from the SSPE tarball. SSPE expects shell and Perl scripts to be located below *adm/bin*; the scripts manipulate the generic configuration files in *adm/etc*.

The *adm/desc* directory has another subdirectory for each machine that SSPE will be configuring. This is where rules, routing tables, and other details for the machine

Listing 1: Profile File

```
01 find sspe-0.2.5 -type d -exec cp {}
02   adm \;
01 # adduser sspe
02 # su - sspe
03 $ mkdir adm
04 $ for DIR in bin etc desc software ;
05   do cp -r sspe-0.2.5/$DIR adm/ ; done
06 $ mkdir adm/tmp
07 $ cp sspe-0.2.5/config adm/.config
08 $ vi adm/.config # BASEDIR anpassen
09 $ echo "export ADMROOT=/home/sspe/adm"
10   >> ~/.profile
11 $ source ~/.profile
```

are stored. *adm/software* has programs that SSPE distributes to its gateways. Temporary files are placed in *adm/tmp*, and you also need an *adm/.config* file, which defines the variables that SSPE needs at runtime. *BASEDIR* is a particularly interesting variable – many scripts will point at a black hole if the *BASEDIR* variable is defined incorrectly. So make sure it points at the *adm* directory.

Besides the variables defined in *.config*, some auxiliary scripts need a shell variable called *ADMROOT*, which points to the directory created in the first step. The best idea is to use a profile file to set this up. You could handle the whole procedure as shown in Listing 1.

As SSPE addresses target machines by host name, it makes sense to add the machines to your */etc/hosts* file. Although you can omit this step if you have a working DNS environment, using */etc/hosts* is preferable to DNS for security reasons, as it mitigates the risk of DNS poisoning attacks.

As communication with the firewall gateways relies entirely on SSH with Public Key authentication, the new user account also needs an SSH keypair. To remove the need to keep typing the passphrase, admins can assign an empty passphrase to the key or use an SSH agent. Of course, you will need to transfer the public part of the keypair to the target machine where SSPE will be logging on as root. The SSPE documentation mentions this but does not give any real advice.

Basic Configuration

Several files handle the configuration details, each one of them managing a specific item, and examples are provided. The documentation is a lot more detailed at this point to the extent of explaining the relationships between the individual components.

The central file is called *hostnet*. It resides in *adm/etc* and groups machines that will be governed by similar policies.

Listing 2: Hosts and Networks

```
01 #Name           Network address # Comment
02 ##################################################################################
   ######
03 any             0.0.0.0/0              # anything not explicitly
   assigned
04
05 # Internal networks
06 lan-dtm             192.168.0.0/24          # Work LAN DTM
07 lan-muc             192.168.1.0/24          # Work LAN MUC
08 dmz             192.168.2.0/24          # DMZ
09
10 # The boss has more privileges than others
11 boss            192.168.0.15/32         # The boss' PC
12
13 # The administrative machines need SSH access
14 admin           192.168.0.10/32         # sysadmin PC
15
16 # The SSPE workstation needs special rules
17 sspe            192.168.0.2/32          # SSPE administrative machine
18
19 # Internal and external gateway definitions
20 def-gw             192.168.0.1/32          # Internal NIC of firewall
21 gw-all             192.168.0.1/32          # Firewall DTM location
22 gw-all             1.2.3.4/32              # Firewall external DTM
   location
23 gw-all             2.3.4.5/32              # Firewall external MUC
   location
```

Groups make it easier to deploy a highly granular security design. It is normal to have a common policy for the internal network, where access is only allowed for a few special clients. Access rules for the firewall itself are typically identical for most machines, with only the administrative machine needing special attention. Listing 2 gives an example.

NAT and IPsec

If your network uses NAT (Network Address Translation), it is important to define the private networks for internal use (typically based on RFC 1918 [7]) beforehand to ensure that the ruleset is applied correctly. The *privates* configuration file (Listing 3) is used for this purpose. NAT mappings for network and IP addresses are defined in *nathosts* (Listing 4). If you additionally require SSPE to handle your IPsec configuration, you will also need to modify the *ipsecs* file.

Another three configuration files define the complete firewall ruleset. The rules for administrative access in *rules.admin* and IPsec *rules.ipsec* are mostly static and do not vary much between the machines involved. The ruleset itself is stored in *rules.user*. The syntax for all these files is quite simple (Listing 5).

The rules applied by our sample files would allow the *admin* machine to establish an SSH connection with the target machine *sspe* (both machines are defined in *hostnet*). The *Oneway* keyword ensures that the connection is unidirectional. SSPE applies these rules to every gateway, so the ruleset you define should be as universal as possible.

Individual Firewalls

We can now use the SSPE interface, which is launched by typing *adm/bin/adm*. Of course, there is nothing to stop you from modifying the configuration files in the directories below *desc* manually, but you will probably agree that the *machine administration | add* option (Figures 1 and 2) gives you an easier approach. The machines need to use the same names as those defined in your *hostnet* file. Unfortunately, the tool does not resolve names to IP addresses and insists on asking the admin for the details.

SSPE generates the required directory with three files: *desc* contains the description, *ip* the machine address, and

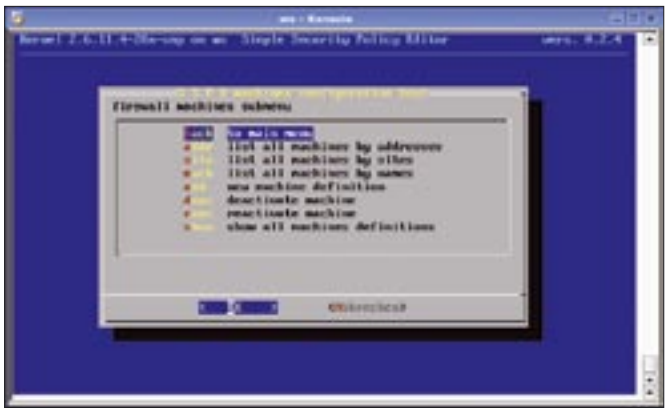


Figure 2: Firewalls that need an individual configuration are added to the machines menu. SSPE adds the individual configurations to the basic settings, which apply to the whole network.

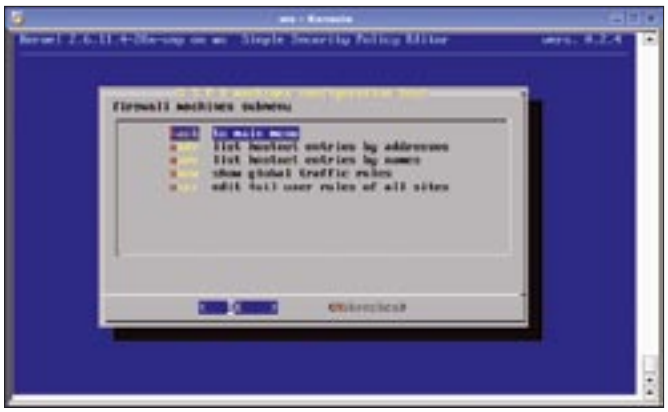


Figure 3: The rules menu confuses users with an item called machines configuration. At this time of writing, users can only view the configuration or call vi for manual editing.

hw identifies Unix-style and Cisco operating systems. This has an influence on how SSPE discovers a gateway's routes, for example.

Besides these three independent files, there are three symlinks to the global *rules.** files. The symlinks ensure that each firewall also honors the global ruleset. In a simple scenario with a single firewall, you would only need the global rules; more complex cases dictate the use of a separate *rules.user* file for each firewall. The rules in this file take priority over the global file.

Five Course Menu

The main SSPE program is primarily designed to help visualize configurations. The *rules administration* item (Figure 3) lists the contents of *hostnet* and allows admins to call a text editor to manually edit the global ruleset. An individual policy is required for the specific rules of individual sites.

Calling the *apply* menu item (Figure 1) rolls out the security design to all the target machines, or to a selection of machines. This is where you find out if the settings are correct.

The *ipsecs administration* item simply reveals the contents of the *ipsecs* configuration file, but without the comments. If the administrative workstation does not have a */etc/ipsecs* file, SSPE issues an error message and quits. If successful, the tool waits for the admin to press [Enter]. It then goes on to generate and distribute the IPsec configurations and preshared secrets.

machine configuration (Figure 2) is not only used to add new gateways and list existing ones; you can also disable individual machines that are temporarily unreachable.

Conclusions

SSPE achieves its goal of providing simpler and more effective administration for distributed firewall systems. The design might be hard to get used to, but it is well thought out and capable of handling quite complex scenarios.

The project has not had the exposure it deserves thus far, but as its popularity increases, one can only hope that more developers will recognize its potential and contribute innovations. ■

Listing 3: Private Networks

```
01 #Name          Network address # Comment
02 ##############################################################################
03 lan-dtm            192.168.0.0/24         # Work LAN DTM
04 lan-muc            192.168.1.0/24         # Work LAN MUC
05 dmz             192.168.2.0/24          # DMZ
```

Listing 4: NAT Configuration

```
01 #Local network  NAT Address            # Comment
02 ##############################################################################
03 192.168.0.0/24  1.2.3.4                # DTM
04 192.168.1.0/24  2.3.4.5                # MUC
```

Listing 5: Ruleset

```
01 #Source  Target    Direction  Protocol  Port  Policy  Options
02 ##############################################################################
03 admin    sspe     Oneway     TCP       ssh   accept  LOG
```

INFO

[1] SSPE: *http://sspe.sourceforge.net*

[2] FW-Builder: *http://www.fwbuilder.org*

[3] Checkpoint Firewall-1: *http://www.checkpoint.com/products/firewall-1/*

[4] Freeswan: *http://www.freeswan.org*

[5] Openswan: *http://www.openswan.org*

[6] Dialog: *http://hightek.org/dialog/*

[7] RFC 1918, "Address Allocation for Private Internets": *http://www.ietf.org/rfc/rfc1918.txt*

THE AUTHOR

Christian Ney is a Unix and firewall administrator employed by a regional airline. On his leisure time, Christian runs a wiki at *RootieWiki.de* and contributes to a number of open source projects.