

Lazy SScripter v0.02

Carlo Contavalli <ccontavalli at commedia.it>

Mon Mar 7 03:02:26 CET 2000

This file documents lsc v0.02, a dumb perl script I wrote to help me out in writing firewalling rules with ipchains, iptables, ip or any other tools. This manual, the mentioned script and all the provided files are copyright © Carlo Contavalli 2000-2002. Please read the following sections for more details. **Note that this is free software and authors hold no responsibility for any damage or loss, direct or indirect, caused by using this software. Use it only on your OWN risk and AFTER carefully reading this documentation.** The latest version of this document can be found at <http://www.commedia.it/ccontavalli/>.

Contents

1 License, copyright and...	1
2 About this script...	1
3 Installation	2
4 Usage	3
4.1 Parameters	4
4.2 Input file format	4
5 Known bugs/problems...	5
6 Future...	5

1 License, copyright and...

This document and lsc were written by Carlo Contavalli <ccontavalli at commedia.it> and are thus Copyright © Carlo Contavalli 2001-2002.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts.

Any example of program code available in this document should be considered protected by the terms of the GNU General Public License.

lsc is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

lsc is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by their respective owners.

2 About this script...

LSC is quite a dumb script I wrote when I was just a perl beginner since I got tired of continuously writing ipchains or iptables in my firewalling scripts (never liked ipchains-save&Co.).

It is quite dumb since it just reads the input from a file and then pipes the “produced” commands to a shell.

The advantage of using lsc is that it accepts a nested format that allows you to write each “prefix” just once, without having to rewrite everything every time and avoiding such boring and hard to track problems due to the cut&paste philosophy (never happened of cutting and pasting firewalling rules and forgetting to change the chain name?). Here is an example input:

```
#!/bin/bash

lan=eth0
dmz=eth1

iptables {
  -P {
    OUTPUT ACCEPT
    FORWARD DROP
    INPUT DROP
  }

  -N {
    lan-dmz
    dmz-lan
  }

  -A FORWARD {
    -i $lan -o $dmz -j lan-dmz
    -i $dmz -o $lan -j dmz-lan
  }

  [ . . . ]
}
```

That would pass to the shell something like:

```
lan=eth0
dmz=eth1
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -N lan-dmz
iptables -N dmz-lan
iptables -A FORWARD -i $lan -o $dmz -j lan-dmz
iptables -A FORWARD -i $dmz -o $lan -j dmz-lan
[. . .]
```

As you can see, lsc is quite dumb. However, it has some big advantages... for example, as you can see in the example, you can use any shell facility since lsc is just a dumb filter, not an interpreter.

You can find the latest version of this document and the mentioned software at <http://www.commedia.it/ccontavalli/>. If you have troubles/suggestions/corrections feel free to mail me at <ccontavalli at commedia.it>.

3 Installation

I suggest you do three things to install lsc:

1. Unpack the provided tarball somewhere in your hard drive

```
$ tar -xvzf ./lsc.tar.gz
```

2. Copy "lsc" in some useful location with the right ownership and permissions set, for example:

```
# cp -a ./lsc /usr/bin/  
# chown root:root /usr/bin/lsc  
# chmod 0755 /usr/bin/lsc
```

3. Check the perl path. The script assumes the perl binary is /usr/bin/perl and this is true for the most part of the systems. To check this out, type "which perl" to have the location of the perl executable. If it is somewhere else, just edit lsc and change the first line to point to the right place.

4 Usage

As I told you, lsc is quite a dumb script. It reads a valid input file to produce a series of commands for the specified shell. For example, a file like this would be a valid input:

```
echo {  
  hello {  
    world  
    foo  
    bar }  
' for a in eth0 eth1 eth2  
' do  
' echo $a  
' done  
how {  
  are {  
    you doing?  
    they doing?  
  }  
  
  is {  
    he doing?  
    she doing?  
    Carlo doing?  
  }  
}  
}
```

in order to execute commands like

```
echo hello world
echo hello foo
echo hello bar
for a in eth0 eth1 eth2
do echo $a
done
echo how are they doing?
echo how are you doing?
echo how is he doing?
echo how is she doing?
echo how is Carlo doing?
```

Ok, so how do I use it?

4.1 Parameters

lsc wants the file to process as its last argument and has very few command line parameters.

- **-n** Tells lsc “don’t do what you are told to – just print those commands on the screen”
- **-i** Tells lsc “to print useful info to debug the input file” (the first column would be the command number, the second column the line it was found in the input file, while the third column the command that would be executed – **-i** must be used in combination with **-n**)

So, using a unixism, the syntax would be:

```
lsc [-n[i]] inputfile
```

4.2 Input file format

A input file must obey the following rules:

- The **#** character indicates the beginning of a comment. It can be put anywhere on a line.
- The **{** indicates the beginning of a block. Before each block there should be a string. Each command inside a block is preceded by all the previous strings and then executed by the specified shell.
- The **}** indicates the end of a block.
- The **##!** followed by the name of a shell indicates the shell that should be used to execute the generated commands. It can be specified anywhere in the file and affects only the commands following this statement.
- The **'** at the beginning of a line indicates that the line must be passed to the shell “as is”, without any change.
- To escape characters, you **MUST** use the **%** character. For example, **%{** is not interpreted by lsc and is passed to the shell as **{**. I decided not to use the **** character to avoid having to write commands like: `grep \\\\.\\.*` just to write `grep \.\\.*`. To write a **%**, you must use **%%**. Escaping any character beside **'**, **{**, **}**, **#**, **%** doesn’t make any sense, but works.

Anything that follows the indicated rules is processed by lsc and passed to the shell. This means that if you want to, you can use any bashism (or perlism, depending on the ##! shell specified) inside the input file and still get the expected output. This is quite useful if you want to maintain some kind of independence. For example, you could write something like:

```
##!/bin/bash

. /usr/lib/fwlib

fwall policy {
  of $input is $accept;
  of $output is $drop;
}

fwall drop $input {
  from localhost;
}
```

Given that /usr/lib/fwlib provides the right functions and variables.. Some dumb examples are provided in the lsc tarball.

5 Known bugs/problems...

1. I wrote this dumb script for my own use in an afternoon, so, don't expect anything good looking at the source code, I was just a perl beginner when I wrote this.
 2. I didn't even want to put this script on a web site. I don't want to waste too much time on it. I'm not going to work on it any longer.
 3. I know it has some troubles sometimes escaping characters.
 4. I know it has some troubles deciding the order of the commands when you use '. For example, this file with lsc
-

```
echo
{
  hello
' echo world
}
```

could have an output of "world hello" instead of the correct hello world.

6 Future...

I'm not going to do anything to lsc. It does exactly what I wanted it to do. If anybody is interested in it and wants to maintain/improve/correct some bugs, please contact me by email..