Protecting SSH at the Transport Layer

Amiel Heyde
Centre for Advanced Internet Architectures, Technical Report 090119A
Swinburne University of Technology
Melbourne, Australia
amiel@swin.edu.au

***Abstract*—SSH daemons are common targets for brute force attacks. Through log monitoring and firewalling, the impact of these attacks on both security and bandwidth consumption can be minimised. We consider a number of implementations and employ Stockade [1] as a backend to SSHGuard [2] for blocking attackers.**

I. Introduction

The SSH [3] daemon provides secure remote log in facilities and may be thought of as an encrypted version of telnet. It is often used to allow remote system administration. Despite the inherent security features of the SSH protocol, common user names combined with poor passwords leaves a system extremely vulnerable.

A. The Problem

A malicious user may script a dictionary or brute force attack which involves attempting to log in as well known users (particularly root) with a large selection of possible passwords. They hope to stumble upon a valid user name and password combination. In this manner they try (and sometimes succeed) in obtaining unauthorised access to the system. Any system that uses SSH with password based authentication is at risk. A typical machine may see many thousands of failed log in attempts per day. This also may generate many Megabytes of wasted bandwidth.

B. The Solution

SSH is generally configured to record all log in attempts via syslog, (such as in /var/log/auth.log under FreeBSD). This logfile can be monitored continuously checking for suspect activity such as is shown in listing 1. When suspicious events are observed, actions can be triggered which will mitigate the risk posed from these events. Usually, a firewall rule will be used to prevent any more log in attempts from the offending IP address. For more information on log monitoring see [4].

II. Comparing Implementations

There are a number of utilities available for *nix which manipulate firewall rules based on information gained from SSH logging output. Naturally, some implementations are more mature than others. We have considered the features of many and have decided to preset the following possible solutions: Blockhosts [5], Blocksshd [6], DenyHosts [7], SSHGuard [7], Sshit [8] and Sshutout [9]. Other solutions which were investigated but not reported on include ShellTer [10], Brute Force Blocker [11] (on which BlockSSHD was based), fail2ban [12] and SSHDfilter [13]. Programs that have been reported on generally provide better features, availability or documentation.

A. Overview

It is clear that there are a number of choices that are adequate for basic protection of SSHD. All solutions examine logs and keep a count of failed log in attempts for each IP address. If the number of failed log in attempts exceed a threshold in a specific amount of time, a firewall rule is added to block the offending address for a fixed time. Despite their similarities, implementation vary in their licence, language, platforms they support, and extra features. A feature matrix is shown in table I

The Input section considers the method with which the programs gain access to the logs. Some implementations take data in through standard input. In this case, syslog can be configured to start them and feed them with relevant log entries directly. Other programs run standalone and poll the default log file (such as auth.log) periodically.

The output section considers the method with which the programs prevent connections from addresses that are deemed to be mounting an attack. Options include firewalls such as PF, IP Filter and IP which operate in BSD environments, as well as the Linux firewall, IP Tables. TCPWrappers [14] is an application level approach to controlling access. It is available on FreeBSD, Linux

Listing 1 An extract from auth.log on FreeBSD showing suspect activity.

```
Aug 13 13:56:08 www sshd[26091]: Illegal user test from 162.21.103.20
Aug 13 13:56:11 www sshd[26093]: Illegal user guest from 162.21.103.20
Aug 13 13:56:15 www sshd[26096]: Illegal user admin from 162.21.103.20
Aug 13 13:56:18 www sshd[26103]: Illegal user admin from 162.21.103.20
Aug 13 13:56:21 www sshd[26105]: Illegal user user from 162.21.103.20
Aug 13 13:56:25 www sshd[26107]: Failed password for root from 162.21.103.20 port 39678 ssh2
Aug 13 13:56:28 www sshd[26109]: Failed password for root from 162.21.103.20 port 39760 ssh2
Aug 13 13:56:32 www sshd[26111]: Failed password for root from 162.21.103.20 port 39836 ssh2
Aug 13 13:56:35 www sshd[26113]: Illegal user test from 162.21.103.20
Aug 13 14:25:36 www sshd[26485]: Illegal user test from 202.28.120.57
Aug 13 14:25:41 www sshd[26487]: Illegal user guest from 202.28.120.57
```

and MacOSX. Null Routing is a basic technique where routes are added that send packets nowhere, and may be used a crude firewall.

The next section outlines extra features. Blacklists and whitelists specify addresses that should always be blocked and allowed, respectively. Logging through syslog and email notifications both provide better reporting for the system administrator. The remaining features will be discussed in the following sections which consider each application individually.

B. BlockHosts

Blockhosts is stand alone python script which is designed primarily to use TCP Wrappers but also supports IPtables. It is the only program to be released into the public domain. It support blacklists and whitelist as well as email notification. It is best suited to a Linux environment (IP Tables). It supports the most additional services and is the only program that supports the Postfix SMTP daemon.

C. BlockSSHD

Supporting both IP Tables and PF as firewall backends, BlockSSHD is at home on Linux or BSD. It is the only product available in the ports tree which provides both Syslog logging and email notifications.

D. DenyHosts

While limited to using TCPWrappers as a blocking mechanism, DenyHosts is an otherwise feature full program. As well as good reporting features, it sports two features that make it unique. Firstly, it can interact with an online database of known attackers which can be blacklisted automatically. Addresses which are observed as attackers are also submitted to this database. Secondly it provides independent settings for each class of failed log in attempt. For example, the block duration could be set differently for those who failed to enter a correct password and those who attempted to log in as an invalid user. Along with good online documentation, Deny hosts is an excellend candidate for those who are only interested in blocking SSH and are happy to employ TCP Wrappers to do so.

E. SSHGuard

SSHGuard is a well designed C program realeased under the BSD license. It supports all the common firewalls and TCPWrappers, providing ultimate platform flexibility. As well as good online documentation, it provides a simple interface for adding new backends. Unlike any of the other programs it has a mechanism which can been used to authenticate log messages, helping to reduce the possibility of a malicious local user sending faked messages to syslog in order to forcibly block addresses. With support for FTP and POP3, as well as direct piped connection to syslog, SSHGuard is perhaps the preferred application of those available in the FreeBSD port tree.

F. sshit

Sshit is a Perl script designed for the BSD environment, support both IPFW and PF firewalls. It has a easy to use configuration file and supports a direct log feed from syslog. This script is suitable for those who prefer simplicity over features.

G. sshutout

Sshutout only provides basic features (including whitelisting and syslog logging), but is included since it is a standalone C application that is compatible with both the Linux firewall (IP Tables), and also the GPL.

Name	Blockhosts	BlockSSHD	DenyHosts	SSHGuard	sshit	sshutout
Availability						
In FreeBSD Ports Tree		X		X	X	
Language	Python	Perl	Python	C	Perl	C
License	Public Domain	GPL	Unknown	BSD	Unknown	GPL
Input						
syslog				X	X	
File Polling	X	X	X		X	X
Output						
PF		X		X	X	
IPF				X		
IPFW				X	X	
IP Tables	X	X		X		X
TCP Wrappers	X		X	X		
Null Routing	X					
Features						
Blacklists	X	X	X			
Whitelists	X	X		X		X
Internet Database			X			
Syslog Logging		X	X	X		X
Email Notification	X	X	X			
Class based settings			X			
Log Message Auth				X		
Additional Services						
FTP	X	X		X	X	
POP3	X			X		
SMTP	X					

TABLE I
FEATURE MATRIX FOR A SELECTION OF SSH PROTECTION SOFTWARE.

This program may therefore be a candidate for use in embedded Linux applications.

H. Comments

Overall, SSHGuard appears to be an excellent choice for the FreeBSD environment and DenyHosts provides the largest featureset for those who want SSH protection only without firewall support. Sshutout is uniquely placed as a good option for the embedded Linux platform.

III. CONFIGURING SSHGUARD FOR USE WITH STOCKADE

A. Stockade Overview

Stockade [1] is a TCP-layer tool for reducing the level of network traffic arriving at an SMTP server due to spammers. The Stockade blacklist server (blserv) operates at the network level as a filter. It sits 'in front' of a mailserver, rejecting incoming TCP connections from known (or suspected) spammers. The rejection is statistical in nature, based on the presumed likelyhood of a new connection's source being 'a spammer'.

A unique feature of Stockade is the auto-rehabilitation of IP addresses that have previously been declared to be spammers. Blacklisted IP addresses initially have new TCP connection attempts rejected with 100% probability. Then, over time, stockade slowly rehabilitates the IP address - reducing the connection rejection probability a few percent at a time. If the source sends new spam in the future, the rejection probability goes back to 100%. Otherwise, the source is eventually rehabilitated (allowed to resume sending us emails) without manual operator intervention.

Adding Stockade support to SSHGuard will give the

opportunity to explore the benefits of dynamic probabilistic blocking of SSH and possibly other services.

SSHGuard provides a simple method to add new backends with only a few commands that must be defined in a new header file. The Stockade distribution contains a standalone program register_spam, which can be used to control it from the command line. Given these two features, we can quickly and easily modify SSHGuard to block SSH attacks through Stockade.

B. Installing SSHGuard

Begin by ensuring your FreeBSD ports tree is up to date. Once this is complete procede to the sshguard port, and extract it.

```
cd /usr/ports/security/sshguard
make extract
```

Now a few modifications must be made to support Stockade as a backend. First, a new file defining the commands needed to control Stockade should be added. Listing 2 shows the contents of this file. Assuming you are using default paths, it should be created at /usr/ports/security/sshguard/work/sshguard-1.x/src/fwalls/command_stockade.h, where x is the version revision.

Secondly, the SSHGuard configure script should be modified to recognise Stockade as a valid backend (Listing 3). You may now return to the sshguard port directory and execute

```
make SSHGUARDFW=stockade
make install
```

After installation you must edit /etc/syslog.conf, adding the following on a single line near the top of the file.

```
 auth.info;authpriv.info
|exec /usr/local/sbin/sshguard
```

After a restart, syslog will pipe any authentication information messages into sshguard automatically.

C. Installing Stockade

Download Stockade from http://caia.swin.edu.au/stockade/. Extract the tarball in a location of your choice and modify make.conf as per the included documentation. Run `make install` to install the software.

A default configuration file is installed to /usr/local/etc/stockade.conf.dist. It should be copied to /usr/local/etc/stockade.conf and customised for use with SSHGuard as follows:

- Add line 'PIDFile /var/run/blserv.pid'
- Change the 'FilterPort' configuration option to port 22
- The 'ClientPassword' option should be set to a password of your choice.

D. Testing the System

Ensure you have IPFW running by entering `ipfw show`. If you receive a message says 'Protocol not available', the IPFW kernel module has not be loaded. To load it, type `kldload ipfw`. For details on configuring the firewall to allow other traffic please see the ipfw man page. If you need to allow all traffic other than traffic blocked by SSHGuard/Stockade, enter:

```
ipfw 65200 add allow ip from any to any
```

Now run `/usr/local/sbin/blserv`. Confirm it is running with

```
ps aux | grep blserv
```

Running `ipfw show` should now list a number of lines with different reset probabilities. You are now ready to restart syslog.

```
/etc/rc.d/syslogd restart
```

Attempt to log in to your machine with incorrect passwords a number of times. You should soon see a line in /var/log/messages from SSHGuard saying that the IP address has been blocked. Further attempts to connect should be blocked most of the time.

IV. Future Work

This report has shown a method to integrate Stockade and SSHGuard, although only at a basic level. The utility of this combination would be greatly increased with some additional features in both programs. It would be useful if Stockade included another command which could clear all blocked addresses so that the system could be reset if necessary. Multiple service support for Stockade would also be useful. In this way, Stockade could work as initially intended, blocking spam as well as blocking other ports such as SSH and FTP as directed by SSHGuard. SSHGuard could be extended to make better use of Stockade's capabilities by supporting an a variable blocking probability, whereby SSHGuard would command Stockade to block with low probability after only two failed attempts and sequentially increase that probability as more failed attempts are observed. This

Listing 2 Contents of new file src/fwalls/command_stockade.h to be placed in the SSHGuard distribution.

```
#ifndef COMMAND_H
#define COMMAND_H

/* user-define backend Stockade */
#include "../config.h"

#define COMMAND_INIT       ""
#define COMMAND_FIN        ""
#define COMMAND_BLOCK      "register_spam $SSHG_SERVICE $SSHG_ADDR .8"
#define COMMAND_RELEASE    "register_spam $SSHG_SERVICE $SSHG_ADDR .05"
#define COMMAND_FLUSH      "echo flush"

#endif
```

Listing 3 Patch to be applied to the SSHGuard configure script (/usr/ports/security/sshguard/work/sshguard-1.x/configure)

```
--- configure.orig      Wed Dec 17 09:59:09 2008
+++ configure   Wed Dec 17 09:59:16 2008
@@ -6482,6 +6482,10 @@
         cp $FWALLSDIR/command_null.h $FWALLSDIR/command.h
         usenull=true
         ;;
+     stockade)
+        usestockade=true
+        cp $FWALLSDIR/command_stockade.h $FWALLSDIR/command.h
+        ;;
      *)
         echo "Choose a valid firewall backend (see --help)"
         exit 1
```

would help reduce impact from attackers while reduce frustration for valid users who may have mistyped their password a number of times. (With SSHGuard alone, they would be completely blocked after a fixed number of failed attempts).

V. Conclusion

Employing log monitors to block attackers from daemons such as SSHD improve security while also reducing wasted bandwith. A selection of implementations have been explored which each have advantages, depending on the feature requirements, licensing constraints and the system environment. In addition, SSHGuard has been shown to operate with Stockade, giving an administrator a more powerful blocking mechanism.

References

[1] "Stockade - a network-level spam mitigation tool," March 16 2007, version 0.2, last accessed 19/01/09. [Online]. Available: http://caia.swin.edu.au/stockade

[2] "Ssh guard," version 1.3, accessed 19/01/09. [Online]. Available: http://sshguard.sourceforge.net/

[3] "Open secure shell," last accessed 19/01/09. [Online]. Available: http://www.openssh.org

[4] B. Glass, "Log monitors on bsd unix," February 11-14 2002, last accessed 17/12/08. [Online]. Available: http://www.usenix.org/event/bsdcon02/full_papers/glass/glass.pdf

[5] "Block hosts," version 2.4, last accessed 19/01/09. [Online]. Available: http://www.aczoom.com/cms/blockhosts/

[6] "Blocksshd," June 2008, version 1.3, accessed 19/01/09. [Online]. Available: http://blocksshd.sourceforge.net/

[7] "Deny hosts," version 2.7, last accessed 19/01/09. [Online]. Available: http://denyhosts.sourceforge.net/

[8] "sshit - the ssh/ftp brute force blocker," version 0.5, accessed 19/01/09. [Online]. Available: http://anp.ath.cx/sshit/

[9] "sshutout - a daemon to stop ssh dictionary attacks," December

2007, version 1.0.5, last accessed 19/01/09. [Online]. Available: http://www.techfinesse.com/sshutout/sshutout.html
[10] "Shellter," version 1.0, last accessed 19/01/09. [Online]. Available: http://shellter.sourceforge.net/
[11] "Brute force blocker," version 1.2.3, last accessed 19/01/09. [Online]. Available: http://danger.rulez.sk/index.php/bruteforceblocker/
[12] "fail2ban," July 2008, version 0.8.3, last accessed 19/01/09. [Online]. Available: http://www.fail2ban.org/
[13] "Sshd filter," May 2007, version 1.5.5, last accessed 19/01/09. [Online]. Available: http://www.csc.liv.ac.uk/~greg/sshdfilter/
[14] "Freebsd handbook: Tcp wrappers," last accessed 17/12/08. [Online]. Available: http://www.freebsd.org/doc/en/books/handbook/tcpwrappers.html