Blacklist'd

A daemon to manage network attacks

Christos Zoulas
2015-01-31

Lately my servers have been feeling like

There were no successful break-ins
But my logs were getting pretty large...

```
Oct 14 20:05:40 broadway sshd[8161]: SSH: Server;Ltype: Version;Remote: 23.88.119.25-40466;Protocol: 2.0;Client: libssh-0.2
Oct 14 20:05:40 broadway sshd[8161]: SSH: Server;Ltype: Kex;Remote: 23.88.119.25-40466;Enc: aes128-cbc;MAC: hmac-sha1;Comp: none [preauth]
Oct 14 20:05:40 broadway sshd[8161]: SSH: Server;Ltype: Authname;Remote: 23.88.119.25-40466;Name: root [preauth]
Oct 14 20:05:41 broadway sshd[8161]: reverse mapping checking getaddrinfo for 25.119-88-23.rdns.scalabledns.com [23.88.119.25] failed -
POSSIBLE BREAK-IN ATTEMPT!
Oct 14 20:05:41 broadway sshd[8161]: error: PAM: authentication error for root from 23.88.119.25
Oct 14 20:05:41 broadway syslogd[174]: last message repeated 2 times
Oct 14 20:05:41 broadway sshd[8161]: Postponed keyboard-interactive for root from 23.88.119.25 port 40466 ssh2 [preauth]
view external: query (cache) 'degler.net/A/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#24603 (xhdscfoenjpj.degler.net): view external: query (cache)
'xhdscfoenjpj.degler.net/A/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#50983 (lsbbdzbzsbq.degler.net): view external: query (cache)
'lsbbdzbzsbq.degler.net/AAAA/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#28385 (lzfotsvdu.degler.net): view external: query (cache)
'lzfotsvdu.degler.net/AAAA/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#48416 (xhdscfoenjpj.degler.net): view external: query (cache)
'xhdscfoenjpj.degler.net/AAAA/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#47161 (lsbbdzbzsbq.degler.net): view external: query (cache)
'lsbbdzbzsbq.degler.net/A/IN' denied
Jan  9 01:24:19 broadway named[10929]: client 96.249.29.182#61202 (lzfotsvdu.degler.net): view external: query (cache)
'lzfotsvdu.degler.net/A/IN' denied
Jan  9 01:24:22 broadway named[10929]: client 96.249.29.182#42952 (gcvovxarcafgn.degler.net): view external: query (cache)
'gcvovxarcafgn.degler.net/AAAA/IN' denied
Jan  9 01:24:22 broadway named[10929]: client 96.249.29.182#54648 (eiirjfkhzsud.degler.net): view external: query (cache)
'eiirjfkhzsud.degler.net/AAAA/IN' denied
Jan  9 01:24:22 broadway named[10929]: client 96.249.29.182#63560 (gcvovxarcafgn.degler.net): view external: query (cache)
'gcvovxarcafgn.degler.net/A/IN' denied
```

How do you know you are under attack?

- Watch the logs, disk space
- Performance drops (not really)
  - Network
  - CPU
- Sniff the network (snort/tcpdump)

Sometimes the attack is silent

- Telnetd
  - Never reports auth failure just delays
- Ntpd
- identd
- Imap/dovecot/etc
- Rpc.*
- Postfix

I looked around for ways to fix the problem

- Don't listen to port 22 for ssh
- Reconfigure your name server
- Only allow connections from “known” places
- **Use one of the popular packages out there to dynamically manage firewall rules to prevent abuse**

Best solutions

- sshguard
- fail2ban
- denyhosts

All scan logs to figure out what to do…

There is no standard way for a daemon to report network attacks

Comparison Matrix

	sshguard	**fail2ban**	**denyhosts**
Version	1.5	0.9.1	2.6
URL	http://www.sshguard.com	http://www.failtoban.org	http://denyhosts.sourceforge.net
language	c	python	python
filters	ipf, pf, iptables	iptables, tcp-wrappers, shorewall, mail	tcp-wrappers, shorewall
daemons	many	many	sshd
how	builtin lex/yacc grammar with regex	Scripts configured statically or by the daemon	locally detected and distributed
advantages	fast	dynamic, popular	proactive
minuses	static	slow	slow, only sshd
notes	most efficient, kqueue, whitelists	client-server, unit-tests, inotify	distributed database

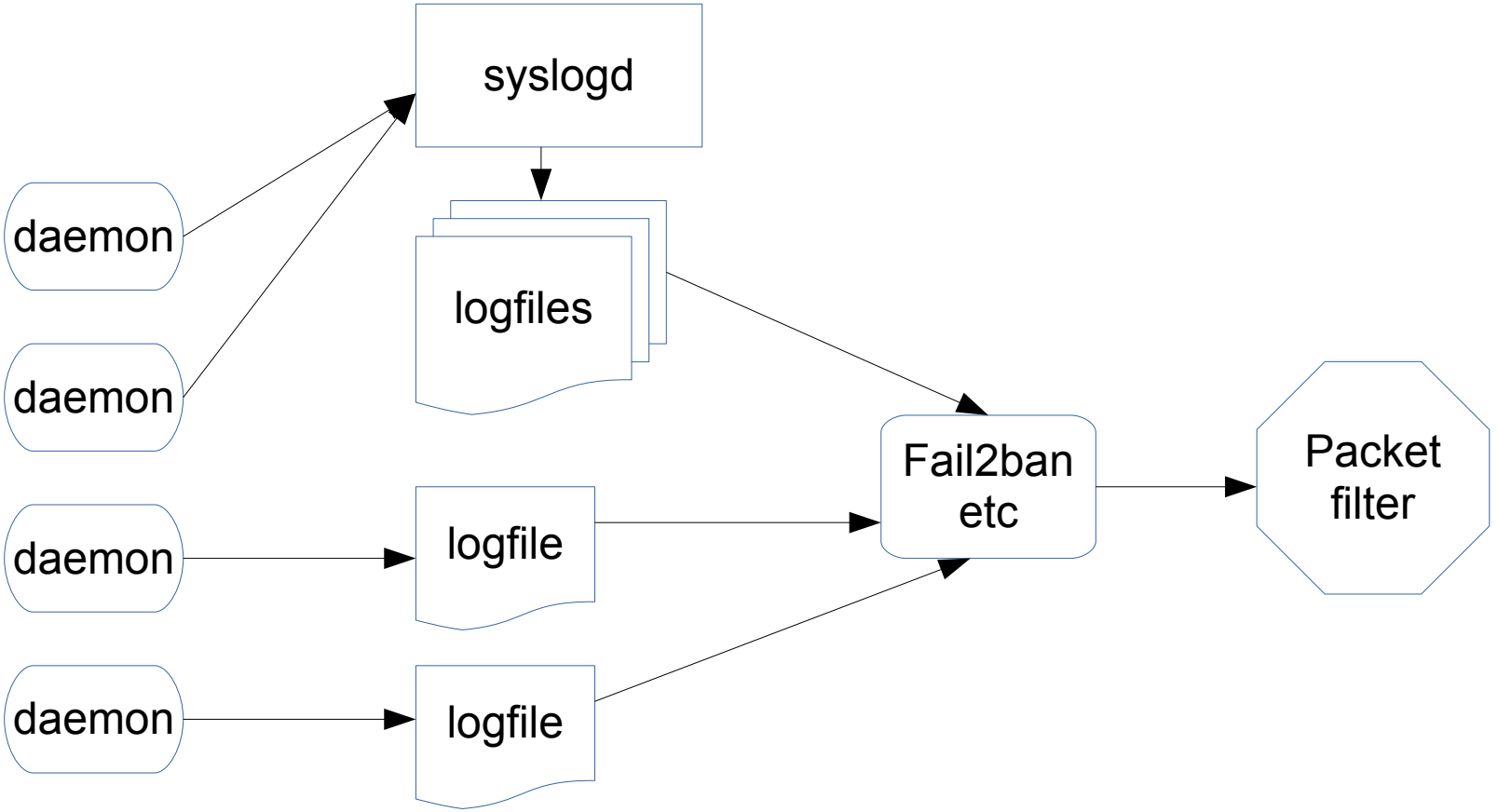
Parsing logs is not ideal

- Expensive to keep/process:
  - require logging (unbuffered logging)
  - require address translation (sometimes)
  - regex DoS injection?
- Fragile, format changes, files move around
- Many logs and formats to parse; complicated
- Other programs/users can log and confuse the parser
- Not secure enough:
  - logger -p auth.warning -t 'sshd[666]' 'Illegal user root from 1.2.3.4'
- Don't give enough information or are not precise enough
  - Which interface did the packet come from?

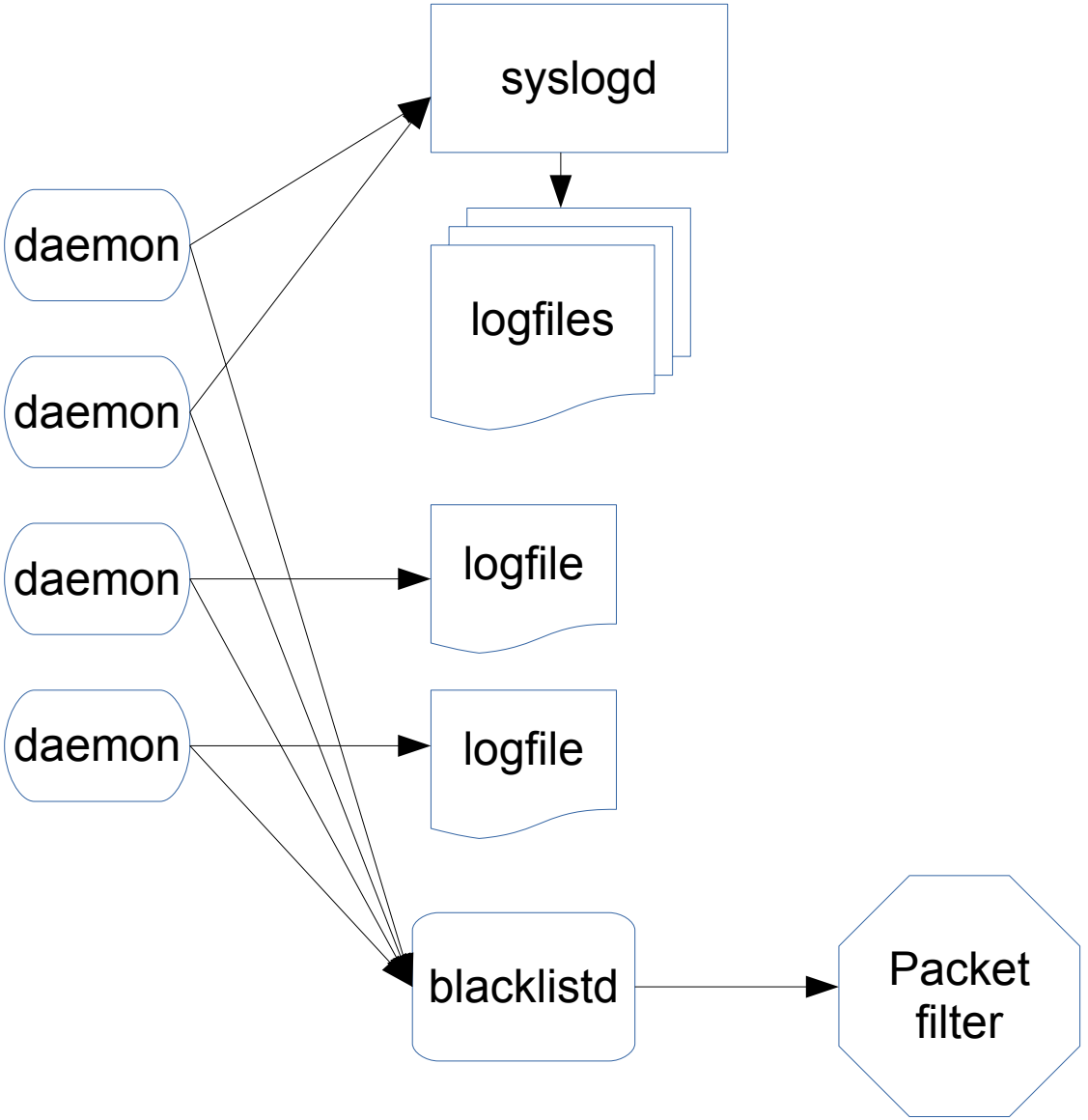
What if we could start over?

- Programs log already…
- What if we added specialized logging that:
  - has the structured information we need
  - is secure
- That would require us to modify every daemon
- How bad is it?
- The change must be simple and transparent

Current process



New process


syslogd
daemon
daemon
daemon
daemon
logfiles
logfile
logfile
blacklistd
Packet
filter

Communication Protocol

Use syslog-like unix socket

Advantages	**Disadvantages**
Non-blocking	Can lose packets
Simple to use	Need multiple endpoints (chroot)
Security (cred passing)	No security (fs security)

API

```
int blacklist(
    int block,
    const struct sockaddr *local,
    socklen_t localLen,
    const struct sockaddr *rem,
    socklen_t remLen,
    const char *msg);
```

API

```
int blacklist(
    int block,
    int fd,
    const char *msg);
```

```
getsockname(fd, &local, &localLen);
getpeername(fd, &rem, &remLen);
```

API

```
int blacklist(int block, int fd, const char *msg);
// Statefull
struct blacklist *blacklist_open(void);
void blacklist_close(struct blacklist *c);
int blacklist_r(struct blacklist *c, int block, int fd,
                const char *msg);
// UDP
int blacklist_sa(int block, int fd,
                 const struct sockaddr *rem, socklen_t remLen,
                 const char *msg);
int blacklist_sa_r(struct blacklist *c, int block, int fd,
                   const struct sockaddr *rem, socklen_t remLen,
                   const char *msg);
```

Configuration

Much like inetd…

```
# Blacklist rule
# addr/mask:port type    proto owner   name    nfail   disable
ssh             stream  *     *       *       3       6h
ftp             stream  *     *       *       3       6h
domain          *       *     named   *       3       12h
```

Configuration

```
# Blacklist rule
# adr/mask:port type    proto owner   name    nfail   disable
ssh             stream  *     *       *       3       6h
fxp0:ssh        stream  *     *       *       *       *
ftp             stream  *     *       *       3       6h
domain          *       *     named   *       3       12h
```

Configuration

```
# Blacklist rule
# addr/mask:port type    proto owner   name    nfail   disable
[local]
ssh             stream  *     *       *       3       6h
fxp0:ssh        stream  *     *       *       *       *
ftp             stream  *     *       *       3       6h
domain          *       *     named   *       3       12h
[remote]
38.17.134.0/24  *       *     *       *       *       *
25.6.24.0/24:ssh stream  *     *       /24     =       1d
[2001:34:43::1]:ssh  *   *     *       v6rule  =       =
```

Code modification

```
--- /dev/null   2015-01-20 21:14:44.000000000 -0500
+++ dist/pfilter.h      2015-01-20 20:16:20.000000000 -0500
@@ -0,0 +1,2 @@
+void pfilter_notify(int);
+void pfilter_init(void);
```

Code modification

[same for auth1, auth2.. for complete diff see $blacklistd/diff/ssh.diff]

```
--- /dev/null   2015-01-22 23:10:33.000000000 -0500
+++ dist/pfilter.c      2015-01-22 23:46:03.000000000 -0500
@@ -0,0 +1,26 @@
+#include "packet.h"
+#include "log.h"
+#include "pfilter.h"
+#include <blacklist.h>
+
+static struct blacklist *blstate;
+
+void
+pfilter_init(void)
+{
+       blstate = blacklist_open();
+}
+
+void
+pfilter_notify(int a)
+{
+       int fd;
+       if (blstate == NULL)
+               pfilter_init();
+       if (blstate == NULL)
+               return;
+       fd = packet_connection_is_on_socket() ?
+            packet_get_connection_in() : 3;
+       blacklist_r(blstate, a, fd, "ssh");
+}
```

Code modification

```
--- dist/auth.c 19 Oct 2014 16:30:58 -0000      1.10
+++ dist/auth.c 22 Jan 2015 21:39:22 -0000
@@ -62,6 +62,7 @@
 #include "monitor_wrap.h"
 #include "krl.h"
 #include "compat.h"
+#include "pfilter.h"

 #ifdef HAVE_LOGIN_CAP
 #include <login_cap.h>
@@ -362,6 +363,8 @@
            compat20 ? "ssh2" : "ssh1",
            authctxt->info != NULL ? ": " : "",
            authctxt->info != NULL ? authctxt->info : "");
+       if (!authctxt->postponed)
+               pfilter_notify(!authenticated);
        free(authctxt->info);
        authctxt->info = NULL;
 }
```

Code modification

```
--- dist/sshd.c 28 Oct 2014 21:36:16 -0000      1.15
+++ dist/sshd.c 22 Jan 2015 21:39:22 -0000
@@ -109,4 +109,5 @@
 #include "ssh-sandbox.h"
 #include "version.h"
+#include "pfilter.h"

 #ifdef LIBWRAP
@@ -364,5 +365,6 @@
         }

+       pfilter_notify(1);
         /* Log error and exit. */
         sigdie("Timeout before authentication for %s", get_remote_ipaddr());
 }
@@ -1160,5 +1162,6 @@
         for (i = 0; i < options.max_startups; i++)
                 startup_pipes[i] = -1;

+       pfilter_init();
         /*
          * Stay listening for connections until the system crashes or
```

Daemon

- Saves and restores state (db file)
- Parses configuration file
- Handles requests
- Runs script on state changes

Script

```
#!/bin/sh
# $1 command
# $2 rulename
# $3 protocol
# $4 address
# $5 mask
# $6 port
# $7 id
case "$1" in
add)
        exec /sbin/npfctl rule $2 add block in final proto $3 from $4/$5 to any port
$6
        ;;
rem)
        exec /sbin/npfctl rule $2 rem-id $7
        ;;
flush)
        exec /sbin/npfctl rule $2 flush
        ;;
*)
        echo "$0: Unknown command '$1'" 1>&2
        exit 1
        ;;
esac
```

NPF configuration example

```
# Transparent firewall example for blacklistd

$ext_if = "bnx0"

set bpf.jit on;
alg "icmp"

group "external" on $ext_if {
        ruleset "blacklistd"
        pass final all
}

group default {
        pass final all
}
```

Monitoring

```
$ blacklistctl dump -ar
        address/ma:port id      nfail   remaining time
    59.98.32.50/32:53   4fc     3/3     10h55m57s
    39.59.58.20/32:53   4e7     3/3     2h29m6s
  103.41.124.31/32:22   4f3     3/3     2h21m1s
 61.168.229.114/32:22           2/3     1h22m1s
$ npfctl rule blacklistd list
ruleset block in final family inet4 proto udp from 49.204.53.184/32 port 53
ruleset block in final family inet4 proto udp from 66.94.67.114/32 port 53
ruleset block in final family inet4 proto udp from 39.59.58.20/32 port 53
ruleset block in final family inet4 proto udp from 103.239.147.81/32 port 53
ruleset block in final family inet4 proto udp from 59.98.32.50/32 port 53
ruleset block in final family inet6 proto udp from 2604:2000:1481:6177:222:4dff:feaf:442c/128 port
53
ruleset block in final family inet4 proto tcp from 103.41.124.104/32 port 22
$ grep blacklistd /var/log/messages
Feb 16 14:22:57 broadway blacklistd[13521]: blocked 94.136.35.11/32:22 for 21600 seconds
Feb 16 14:30:47 broadway blacklistd[13521]: blocked 84.14.252.137/32:22 for 21600 seconds
Feb 16 14:34:50 broadway blacklistd[13521]: blocked 59.98.32.50/32:53 for 43200 seconds
Feb 16 14:47:27 broadway blacklistd[13521]: released 103.41.124.12/32:22 after 21600 seconds
Feb 16 14:51:24 broadway blacklistd[13521]: blocked 2604:2000:1481:6177:222:4dff:feaf:442c/128:53
for 43200 seconds
```

Status

- Code and documentation complete
- On NetBSD/current
  - /usr/src/external/bsd/blacklist
- Compiles and runs on Linux and MacOS/X
- Supports only npf, others easy to add
- Patches for bind, ftpd, sshd