Central logging Security

James Hunter
November, 25 2000

In todays distributing computing architecture consisting of numerous dedicated servers handling applications such as webservers, DNS, mail-relays, and ftp it’s not only beneficial but necessary to create a centralized logging system. Having a central log server is helpful for keeping records of failed attempts, who is spamming your servers, and a place for your firewalls to keep their logs. Syslogs are also extremely important in finding out the health and integrity of the systems you manage. Centralized logging is easy to install and configure, however security in transit is often overlooked. Figure 1.1 lists some of the types of syslogs that are generated, what they do, and where they log to locally.

Facility	Description	Typical local location on Redhat Linux 7.0
auth	Authentication	/var/log/secure
cron	Used for the cron and at systems	/var/log/cron
Mail	Mail log	/var/log/maillog
News	News system	/var/log/spooler
Uucp	Reserved for uucp system	/var/log/spooler

Figure 1.1

During this exercise in security I put together a lab consisting of two linux servers to simulate a log server and a log client. These servers were put on a hub along with a packet sniffer. I had the test client send all of its logs to the logserver. In Figures 1.2 and 1.3 is an example trace while performing a mail transaction on the log client. In bold are interesting items that someone sniffing the wire can determine.

Notice in Figure 1.2 and 1.3 that it easy to grep for the protocol, the type of mailer being used, what domain it’s sending to, and of course the to and from fields. If this was a corporation would you want someone to know that your CEO just sent a message to a stock firm in the middle of an IPO? Or that your CEO just sent a email to a lawfirm?

```
       TIME       Source          Destination       Protocol         Info
44 5.535396 192.168.1.201  192.168.1.200             Syslog       MAIL.INFO:
sendmail[788]: OAA00788: fro...
0000  00 90 27 f9 59 1f 00 d0  b7 27 73 b8 08 00 45 00   ..'ùY..Ð ·'s¸..E.
0010  00 b5 01 d8 00 00 40 11  f3 7e c0 a8 01 c9 c0 a8   .µ.Ø..@. ó~À¨.ÉÀ¨
0020  01 c8 02 02 02 02 00 a1  f7 f3 3c 32 32 3e 73 65   .È.....¡ ÷ó<22>se
0030  6e 64 6d 61 69 6c 5b 37  38 38 5d 3a 20 4f 41 41   ndmail[7 88]: OAA
0040  30 30 37 38 38 3a 20 66  72 6f 6d 3d 72 6f 6f 74   00788: f rom=root
0050  2c 20 73 69 7a 65 3d 32  39 2c 20 63 6c 61 73 73   , size=2 9, class
0060  3d 30 2c 20 70 72 69 3d  33 30 30 32 39 2c 20 6e   =0, pri= 30029, n
0070  72 63 70 74 73 3d 31 2c  20 6d 73 67 69 64 3d 3c   rcpts=1,  msgid=<
0080  32 30 30 30 31 31 32 32  31 39 32 39 2e 4f 41 41   20001122 1929.OAA
0090  30 30 37 38 38 40 6c 6f  63 61 6c 68 6f 73 74 2e   00788@lo calhost.
00a0  6c 6f 63 61 6c 64 6f 6d  61 69 6e 3e 2c 20 72 65   localdom ain>, re
00b0  6c 61 79 3d 72 6f 6f 74  40 6c 6f 63 61 6c 68 6f   lay=root @localho
00c0  73 74 0a                                           st.
```

Figure 1.2

```
   TIME       Source              Destination       Protocol         Info
45 5.565105 192.168.1.201  192.168.1.200 Syslog       MAIL.INFO:
sendmail[790]:OAA00788: to=...

0000  00 90 27 f9 59 1f 00 d0  b7 27 73 b8 08 00 45 00   ..'ùY..Ð ·'s¸..E.
0010  00 8f 01 db 00 00 40 11  f3 a1 c0 a8 01 c9 c0 a8   ...Û..@. ó¡À¨.ÉÀ¨
0020  01 c8 02 02 02 02 00 7b  8f f0 3c 32 32 3e 73 65   .È.....{ .ð<22>se
0030  6e 64 6d 61 69 6c 5b 37  39 30 5d 3a 20 4f 41 41   ndmail[7 90]: OAA
0040  30 30 37 38 38 3a 20 74  6f 3d 72 6f 6f 74 2c 20   00788: t o=root,
0050  63 74 6c 61 64 64 72 3d  72 6f 6f 74 20 28 30 2f   ctladdr= root (0/
0060  30 29 2c 20 64 65 6c 61  79 3d 30 30 3a 30 30 3a   0), dela y=00:00:
0070  30 30 2c 20 78 64 65 6c  61 79 3d 30 30 3a 30 30   00, xdel ay=00:00
0080  3a 30 30 2c 20 6d 61 69  6c 65 72 3d 6c 6f 63 61   :00, mai ler=loca
0090  6c 2c 20 73 74 61 74 3d  53 65 6e 74 0a            l, stat= Sent.
```

Figure 1.3

Knowing this information shows that it is important that all logging should find it's way to the log server in a safe way. This requires encrypting the information. Before encrypting this information it needs to be TCP instead of UDP. There are many packages made freely on the Internet to convert UDP to TCP including netcat from l0pht, Cryptcat (an encrypted version of netcat), Zebedee, or a VPN. Openssh has scripts to tunnel NFSv1 and NFSv2 that could be modified for this purpose.

For my lab I used a program called "Cotty" along with some scripts that were provided in the package to create a VPN so that I could encapsulate UDP over TCP. My lab network was a private network space 192.168.1.0/24 but could have been a public IP space. Inside the 192.168.1.0/24 I used a ppp link on one side so the IP address was 10.10.10.1 and the other side was 10.10.10.2. This was set up so that the logserver initiated all connections with its client/peers. Figure 1.4 is a diagram of what the established link looks like.

```
TCP/IP Link

Logserver 192.168.1.200                     Logclient
192.168.1.201
________________________________________________________________
|
|Logserver encapsulated link         Logclient encapsulated link
|10.10.10.1                          10.10.10.2
=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=
/=
=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=
/=
|
|
________________________________________________________________
```

1. Installed openssl and openssh on both client/server

```
[root@logserver /x]# rpm -i openssl-0.9.5-1.i386.rpm
[root@logserver /x]# rpm -i openssh-1.2.3-1.i386.rpm
[root@logserver /x]# rpm -i openssh-server-1.2.3-1.i386.rpm
[root@logserver /x]# rpm -i openssh-clients-1.2.3-1.i386.rpm
```

2. Put the IP addresses and names in the /etc/hosts table on both client and server machines.
3. On the client machine changed the sshd_config file so that there was a trust relationship. Made sure these parameters were set. After these options were set restarted the sshd process.

```
        PermitRootLogin yes
        IgnoreRhosts no
        RhostsRSAAuthentication yes
```

4. On the client side created a /etc/.shosts and added these lines.

```
        logserver.testing.com    root
```

5. Added the public key from the logserver in the /root/.ssh/known_hosts. Tested that everything was

working by using ssh from the server side to the client side to make sure it didn’t ask for a password.

6. Untarred the cotty package and ppp rpm package. After the extract was finished I used gcc to compile the source and moved the binary to /usr/sbin.

```
        [root@logserver /x]# tar xvf cotty.tar
        [root@logserver /x]# rpm –i ppp-2.3.11-
        4.i386.rpm
        [root@logserver /x]# gcc cotty-0.4.c
        [root@logserver /x]# mv cotty /usr/sbin
```

7. On the client machine it was necessary to change /etc/ppp/options and add a line at the beginning and add the option noauth.
8. On the logserver modified the script to bring up the ppp link as well as do the encrypting of traffic. As described this brings up the ppp session and creates the ip addresses to be 10.10.10.1 and 10.10.10.2.

```
        REMOTE_ACCOUNT=root@logclient.testing.com
        REMOTE_PPPD="pppd noauth ipcp-accept-local
        ipcp-accept-remote"
        ###pppd silent ip:ip are vpn who cares
        addresses(make non-routable)###
        LOCAL_PPPD="pppd silent
        10.10.10.1:10.10.10.2"
        ### -d option is used for passing pty to ssh -
        t ssh in terminal mode
        cotty -d -- $LOCAL_PPPD -- ssh -t
        $REMOTE_ACCOUNT $REMOTE_PPPD
```

9. Then I ran the script which created the ppp link and checked this with ifconfig to make sure the link was up.

```
[root@logserver /x]# ./bring-up-logclient
[root@logserver /x]# ifconfig –a

  ppp0      Link encap:Point-to-Point Protocol
  POINTOPOINT NOARP MULTICAST  MTU:1500  Metric:1
  RX packets:8 errors:1 dropped:0 overruns:0 frame:1
  TX packets:8 errors:0 dropped:0 overruns:0
  carrier:0
  collisions:0 txqueuelen:10
```

10. Change the remote log host on the logclient in

/etc/syslogd.conf to be the VPN link and restart the service.

```
*.*  @10.10.10.1
[root@logserver /x]# /etc/rc.d/init.d/syslog
restart
```

After I made the changes and brought up the VPN I sent an email exactly the same as I did before the VPN was up. My trace indicated that all the traffic was encrypted. This is a sample of the tcpdump trace.

```
TIME         Source              Destination        Protocol          Info
7 0.217970 192.168.1.201    192.168.1.200 TCP    ssh > npmp-local [PSH, ACK]
Seq=2650243640 Ack=1837051758 Win=32120 Len=100

0000  00 90 27 f9 59 1f 00 d0  b7 27 73 b8 08 00 45 10   ..'ùY..Ð ·'s¸..E.
0010  00 98 01 1b 40 00 40 06  b4 53 c0 a8 01 c9 c0 a8   ....@.@. ´SÀ¨.ÉÀ¨
0020  01 c8 00 16 02 62 9d f7  82 38 6d 7f 2f 6e 80 18   .È...b.÷ .8m./n..
0030  7d 78 9f cc 00 00 01 01  08 0a 00 00 33 f5 00 00   }x.Ì.... ....3õ..
0040  37 ba 00 00 00 58 9f 44  77 ac fb a5 cb 0e 0b d9   7º...X.D w¬û¥Ë..Ù
0050  b6 14 06 95 7d 14 f1 30  3d a6 5a ee 5b 17 24 59   ¶...}.ñ0 =¦Zî[.$Y
0060  66 e2 85 d9 aa 58 b4 c2  3c a3 ab 44 4d 1b 11 ac   fâ.ÙªX´Â <£«DM..¬
0070  0e ab 46 fe 46 95 aa 4a  fd 7b a7 78 27 2a e1 66   .«FþF.ªJ ý{§x'*áf
0080  76 da 04 23 c1 95 b5 5f  37 30 30 73 57 12 bc 48   vÚ.#Á.µ_ 700sW.¼H
0090  dc f6 0d 0d 93 a4 e4 c0  4f 92 2e 19 36 96 80 d3   Üö...¤äÀ O...6..Ó
00a0  38 4d 85 a4 33 23                                  8M.¤3#
```

The syslog facility is a valuable tool in finding out your servers health and other important information, but it isn’t secure or connection reliable. Anyone that has access to the ethernet cable has the ability to not only snoop all traffic going by but to also inject bogus information. Hopefully the example lab has shown how to make remote logging reliable as well as secure.

Silver, Geoff. “Remote Syslog” 24, Febuary 2000.
http://www.uslinux.net/HOWTO/remote-syslog.shtml

Conover, Matt and Zielinski, Mark. “SRS (Secure Remote Streaming) 1.0”
http://www.securityfocus.com/tools/1179

Beattie,Steven “How to do UDP wrapping (syslog via ssh)?” 17

January, 1999.
http://www.patoche.org/LTT/security/00000118.html (27, July 1999)

L0pht "Netcat 1.10"
http://www.l0pht.com/~weld/netcat/readme.html

Winton, Neil "Zebedee: Secure IP tunnel"(13 August 2000)
http://www.winton.org.uk/zebedee/manual.html

Waters, Stephen "RE: IPSEC usage to protect syslog"(22 August 2000)
http://www.mail-archive.com/syslog-sec%40employees.org/msg00479.html

Wilson, Matthew "VPN-HOWTO"(December 1999)
http://www.ibiblio.org/pub/Linux/docs/HOWTO/VPN-HOWTO