Logwatch

Logwatch is a log analysis tool designed for Linux and Unix systems. It makes the process of reviewing logs easier and somewhat faster. To install log watch in Linux, use the following command: sudo apt install logwatch -y. This is illustrated below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ sudo apt install logwatch -y
[sudo] password for tee:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdate-manip-perl postfix ssl-cert
Suggested packages:
  libsys-cpu-perl libsys-meminfo-perl procmail postfix-mysql postfix-pgsql postfix-ldap
  postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin | dovecot-common resolvconf postfix-cdb
  mail-reader postfix-mta-sts-resolver postfix-doc
The following NEW packages will be installed:
  libdate-manip-perl logwatch postfix ssl-cert
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.
Need to get 2,589 kB of archives.
After this operation, 19.3 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 postfix amd64 3.6.4-1ubuntu1.3 [1,
248 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libdate-manip-perl all 6.86-1 [946 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 logwatch all 7.5.6-1ubuntu1 [378 kB]
Fetched 2,589 kB in 2s (1,497 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ssl-cert.
(Reading database ... 128037 files and directories currently installed.)
Preparing to unpack .../ssl-cert_1.1.2_all.deb ...
Unpacking ssl-cert (1.1.2) ...
Selecting previously unselected package postfix.
Preparing to unpack .../postfix_3.6.4-1ubuntu1.3_amd64.deb ...
Unpacking postfix (3.6.4-1ubuntu1.3) ...
Selecting previously unselected package libdate-manip-perl.
Preparing to unpack .../libdate-manip-perl_6.86-1_all.deb ...
Unpacking libdate-manip-perl (6.86-1) ...
Selecting previously unselected package logwatch.
Preparing to unpack .../logwatch_7.5.6-1ubuntu1_all.deb ...
```

In logwatch, analysts can investigate for a particular service such as Secure Shell (SSH), and can also tailor scans to return logs from a particular day or timeframe.

- The '--range' argument can be used to select the desired range for analysis.
- The '--detail' argument specifies how much information is included in the report.

These arguments can be used at the same time as desired. Examples of these arguments are displayed below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ logwatch --range today

 ################### Logwatch 7.5.6 (07/23/21) ####################
        Processing Initiated: Sat Jul 20 16:47:12 2024
        Date Range Processed: today
                              ( 2024-Jul-20 )
                              Period is day.
        Detail Level of Output: 0
        Type of Output/Format: stdout / text
        Logfiles for Host: ubuntu3
 ##################################################################

 ---------------------- dpkg status changes Begin ------------------------

 Installed:
    libdate-manip-perl:all 6.86-1
    logwatch:all 7.5.6-1ubuntu1
    postfix:amd64 3.6.4-1ubuntu1.3
    ssl-cert:all 1.1.2

 ---------------------- dpkg status changes End ------------------------


 ---------------------- Kernel Begin ------------------------


 WARNING:  Kernel Errors Present
    WARNING: Spectre v2 mitigation leaves CPU vulner ...:  1 Time(s)
    [drm:vmw_host_printf [vmwgfx]] *ERROR* Failed to send  ...:  1 Time(s)

 ---------------------- Kernel End ------------------------


 ---------------------- pam_unix Begin ------------------------
```

```
tee@ubuntu3: ~

tee@ubuntu3:~$ logwatch --range yesterday

 ################### Logwatch 7.5.6 (07/23/21) ####################
        Processing Initiated: Sat Jul 20 16:52:10 2024
        Date Range Processed: yesterday
                              ( 2024-Jul-19 )
                              Period is day.
        Detail Level of Output: 0
        Type of Output/Format: stdout / text
        Logfiles for Host: ubuntu3
 ##################################################################

 ---------------------- SSHD Begin ------------------------

 Negotiation failed:
    no matching host key type found: 27 Times

 **Unmatched Entries**
 drop connection #12 from [192.168.1.254]:45914 on [192.168.1.91]:22 past MaxStartups : 1 Time
 error: beginning MaxStartups throttling : 1 Time
 error: kex_exchange_identification: Connection closed by remote host : 1 Time
 exited MaxStartups throttling after 00:00:00, 7 connections dropped : 1 Time

 ---------------------- SSHD End ------------------------


 ---------------------- Disk Space Begin ------------------------

 Filesystem                          Size  Used Avail Use% Mounted on
 /dev/mapper/ubuntu--vg-ubuntu--lv    30G   15G   14G  52% /
 /dev/sda2                           2.0G  254M  1.6G  14% /boot


 ---------------------- Disk Space End ------------------------
```

For this demo, a log file named *auth.log* will be used. To view the contents of this file, the *less* or *tail* command can be used. The *less* command is used to view the contents of a file, one page at a time, while the *tail* command is used to display the last part of a file. *Sudo* might be required if elevated privileges are needed to execute these commands. The result of the *less* command on the *auth.log* file is illustrated below:

```
tee@ubuntu3: ~

Mar 10 00:00:55 server1 sshd[4422]: Received disconnect from 10.0.2.2 port 60950:11: disconnected by
user
Mar 10 00:00:55 server1 sshd[4422]: Disconnected from user ajay 10.0.2.2 port 60950
Mar 10 00:00:55 server1 sshd[4366]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Session 13 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 13.
Mar 10 00:00:55 server1 sshd[3878]: Received disconnect from 10.0.2.2 port 61128:11: disconnected by
user
Mar 10 00:00:55 server1 sshd[3878]: Disconnected from user ajay 10.0.2.2 port 61128
Mar 10 00:00:55 server1 sshd[3822]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Session 5 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 sshd[4043]: Received disconnect from 10.0.2.2 port 62494:11: disconnected by
user
Mar 10 00:00:55 server1 sshd[4043]: Disconnected from user ajay 10.0.2.2 port 62494
Mar 10 00:00:55 server1 sshd[3987]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 5.
Mar 10 00:00:55 server1 systemd-logind[710]: Session 7 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 7.
Mar 10 00:01:00 server1 systemd-logind[710]: Power key pressed.
Mar 10 00:01:00 server1 systemd-logind[710]: Powering Off...
Mar 10 00:01:00 server1 systemd-logind[710]: System is powering down.
Mar 10 00:23:02 server1 systemd-logind[714]: New seat seat0.
Mar 10 00:23:02 server1 sshd[764]: Server listening on 0.0.0.0 port 22.
Mar 10 00:23:02 server1 sshd[764]: Server listening on :: port 22.
Mar 10 00:23:02 server1 systemd-logind[714]: Watching system buttons on /dev/input/event0 (Power Butt
on)
Mar 10 00:23:02 server1 systemd-logind[714]: Watching system buttons on /dev/input/event1 (Sleep Butt
on)
Mar 10 00:23:02 server1 systemd-logind[714]: Watching system buttons on /dev/input/event2 (AT Transla
ted Set 2 keyboard)
Mar 10 01:17:01 server1 CRON[1035]: pam_unix(cron:session): session opened for user root(uid=0) by (u
id=0)
Mar 10 01:17:01 server1 CRON[1035]: pam_unix(cron:session): session closed for user root
Mar 10 02:17:01 server1 CRON[1101]: pam_unix(cron:session): session opened for user root(uid=0) by (u
:
```

The result for the *tail* command is also shown below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ tail authy.log
Mar 18 03:50:40 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by nigel(ui
d=1004)
Mar 18 03:51:25 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 03:52:08 server1 sshd[960]: pam_unix(sshd:session): session closed for nigel
Mar 18 03:52:52 server1 sshd[970]: Accepted password for michelle from 10.0.0.19 port 53060 ssh2
Mar 18 03:52:52 server1 sshd[970]: pam_unix(sshd:session): session opened for michelle(uid=1005) by (
uid=0)
Mar 18 03:53:36 server1 sudo: michelle : TTY=pts/5 ; PWD=/home/michelle ; USER=root ; COMMAND=/usr/bi
n/htop
Mar 18 03:53:36 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by michelle
(uid=1005)
Mar 18 03:54:21 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 03:55:03 server1 sshd[975]: pam_unix(sshd:session): session closed for michelle

tee@ubuntu3:~$
```

The *-n* argument can also be used to specify the number of lines to be displayed by the command. For example, the following command; *tail -n 25 auth.log*. The *tail* command combined with the *-n 25*

arguments displays the last 25 lines of the *auth.log* file. This is illustrated below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ tail -n 25 authy.log
Mar 18 03:44:03 server1 sshd[925]: Accepted password for tara from 10.0.0.16 port 53025 ssh2
Mar 18 03:44:03 server1 sshd[925]: pam_unix(sshd:session): session opened for tara(uid=1002) by (uid=
0)
Mar 18 03:44:56 server1 sudo: tara : TTY=pts/2 ; PWD=/home/tara ; USER=root ; COMMAND=/usr/bin/nano /
etc/hosts
Mar 18 03:44:56 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by tara(uid
=1002)
Mar 18 03:45:30 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 03:46:07 server1 sshd[930]: pam_unix(sshd:session): session closed for tara
Mar 18 03:46:51 server1 sshd[940]: Accepted password for robert from 10.0.0.17 port 53040 ssh2
Mar 18 03:46:51 server1 sshd[940]: pam_unix(sshd:session): session opened for robert(uid=1003) by (ui
d=0)
Mar 18 03:47:35 server1 sudo: robert : TTY=pts/3 ; PWD=/home/robert ; USER=root ; COMMAND=/usr/bin/ta
il -f /var/log/syslog
Mar 18 03:47:35 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by robert(u
id=1003)
Mar 18 03:48:20 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 03:49:12 server1 sshd[945]: pam_unix(sshd:session): session closed for robert
Mar 18 03:49:55 server1 sshd[955]: Accepted password for nigel from 10.0.0.18 port 53050 ssh2
Mar 18 03:49:55 server1 sshd[955]: pam_unix(sshd:session): session opened for nigel(uid=1004) by (uid
=0)
Mar 18 03:50:40 server1 sudo: nigel : TTY=pts/4 ; PWD=/home/nigel ; USER=root ; COMMAND=/bin/dmesg
Mar 18 03:50:40 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by nigel(ui
d=1004)
Mar 18 03:51:25 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 03:52:08 server1 sshd[960]: pam_unix(sshd:session): session closed for nigel
Mar 18 03:52:52 server1 sshd[970]: Accepted password for michelle from 10.0.0.19 port 53060 ssh2
Mar 18 03:52:52 server1 sshd[970]: pam_unix(sshd:session): session opened for michelle(uid=1005) by (
uid=0)
Mar 18 03:53:36 server1 sudo: michelle : TTY=pts/5 ; PWD=/home/michelle ; USER=root ; COMMAND=/usr/bi
n/htop
Mar 18 03:53:36 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by michelle
(uid=1005)
Mar 18 03:54:21 server1 sudo: pam_unix(sudo:session): session closed for user root
```

To generate a Logwatch report that includes the authentication logs, the following command can be used: *sudo logwatch --service sshd --range today --detail High --output stdout.* The *sudo* command indicates administrative privilege, *logwatch* is the tool being called upon, *sshd* is the service being scanned for, the desired range is today's logs, comprehensive detail is desired, and the output should be displayed on the screen. This command is illustrated below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ sudo logwatch --service sshd --range today --detail High --output stdout
[sudo] password for tee:

 ################### Logwatch 7.5.6 (07/23/21) ####################
        Processing Initiated: Sat Jul 20 17:36:19 2024
        Date Range Processed: today
                              ( 2024-Jul-20 )
                              Period is day.
        Detail Level of Output: 10
        Type of Output/Format: stdout / text
        Logfiles for Host: ubuntu3
 ##################################################################

 ---------------------- SSHD Begin ------------------------

 SSHD Started: 2 Times

 Users logging in through sshd:
    tee:
       192.168.1.66 (Tabs-iMac.attlocal.net): 1 Time

 ---------------------- SSHD End ------------------------


 ######################## Logwatch End #########################

tee@ubuntu3:~$
```

The parameters for the arguments can be adjusted as needed to get the desired output.

TCPDump

TCPDump is a CLI tool used to analyze network packets. It provides the option to tailor the packet scan to suit the operator's needs. Hence, tcpdump captures, filters, and analyzes network packets. Basic usages of tcpdump includes packet capture on an interface using the *tcpdump -i eth0* command. However, it is imperative to use the right network interface. To check for the right network interface, use the following command as illustrated below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group defa
ult qlen 1000
    link/ether 08:00:27:ea:c8:7c brd ff:ff:ff:ff:ff:ff
tee@ubuntu3:~$
```

The command above shows that the network interface is running on enp0s3. Hence the command to capture packets on this interface will be: *tcpdump -i enp0s3.* This is illustrated below:

```
tee@ubuntu3: ~

tee@ubuntu3:~$ sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:39:21.504837 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 1286228739:128622
8847, ack 1417734058, win 501, options [nop,nop,TS val 1257004680 ecr 3513928852], length 108
18:39:21.505043 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 108, win 2046, opt
ions [nop,nop,TS val 3513928860 ecr 1257004680], length 0
18:39:21.505729 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 108:144, ack 1, w
in 501, options [nop,nop,TS val 1257004681 ecr 3513928860], length 36
18:39:21.505901 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 144, win 2047, opt
ions [nop,nop,TS val 3513928860 ecr 1257004681], length 0
18:39:21.506034 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 144:204, ack 1, w
in 501, options [nop,nop,TS val 1257004681 ecr 3513928860], length 60
18:39:21.506139 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 204, win 2047, opt
ions [nop,nop,TS val 3513928862 ecr 1257004681], length 0
18:39:21.506216 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 204:272, ack 1, w
in 501, options [nop,nop,TS val 1257004681 ecr 3513928862], length 68
18:39:21.506375 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 272, win 2046, opt
ions [nop,nop,TS val 3513928862 ecr 1257004681], length 0
18:39:21.506459 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 272:340, ack 1, w
in 501, options [nop,nop,TS val 1257004681 ecr 3513928862], length 68
18:39:21.506566 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 340, win 2046, opt
ions [nop,nop,TS val 3513928862 ecr 1257004681], length 0
18:39:21.506637 IP Tabs-iMac.attlocal.net.ssh > 192.168.1.66.52567: Flags [P.], seq 340:376, ack 1, w
in 501, options [nop,nop,TS val 1257004682 ecr 3513928862], length 36
18:39:21.507349 IP 192.168.1.66.52567 > Tabs-iMac.attlocal.net.ssh: Flags [.], ack 376, win 2047, opt
ions [nop,nop,TS val 3513928862 ecr 1257004682], length 0
18:39:21.602963 IP Tabs-iMac.attlocal.net.34023 > dsldevice.attlocal.net.domain: 38667+ [1au] PTR? 66
.1.168.192.in-addr.arpa. (54)
18:39:21.607091 IP dsldevice.attlocal.net.domain > Tabs-iMac.attlocal.net.34023: 38667 NXDomain* 0/0/
1 (54)
18:39:21.607206 IP Tabs-iMac.attlocal.net.34023 > dsldevice.attlocal.net.domain: 38667+ PTR? 66.1.168
.192.in-addr.arpa. (43)
18:39:21.609084 IP dsldevice.attlocal.net.domain > Tabs-iMac.attlocal.net.34023: 38667 NXDomain* 0/0/
0 (43)
```

To capture only network packets, use the following command: *tcpdump -c N -i eth0* (Be sure to use the right network interface.
To display captured data in verbose mode, include the *-v* argument as follows: *tcpdump -v -i eth0*. This provides more comprehensive detail about the scan.
To write captured files to a file, use the following command: *tcpdump -w file.pcap -i eth0.* The *-w* argument writes the file into the desired destination. Tcpdump files are typically saved in the .pcap format.
To read packets from a file, include the *-r* argument as shown: *tcpdump -r file.pcap.*

Capturing Syslog Messages

Tcpdumps can be used to capture packets to validate that a system is sending or receiving syslog messages. Since syslogs are often sent over UDP to port 514, the following command can be used to validate if syslogs are being received: *tcpdump -i eth0 port 514 -vv.* If scanning for just syslog traffic, then the following command will suffice: *tcpdump -port514*.

To capture http traffic, use the following command: *tcpdump port 80 -i enp0s3*
To capture packets from specific Ips: *tcpdump src host 192.168.1.1 -i enp0s3*
To filter by protocol: *tcpdump tcp/udp -i enp0s3*

The main configuration file for system logs is saved in the /etc/rsyslog.conf file. This is shown below:

```
tee@ubuntu3: /etc

tee@ubuntu3:/etc$ cat rsyslog.conf
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf
*.* @192.168.1.86

################
#### MODULES ####
################

module(load="imuxsock") # provides support for local system logging
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
```

Forwarding Logs

rsyslog can be forwarded to a remote syslog server. This is done by adding a line to the configuration, indicating the desired destination IP address. The command will read as follows: *.* @IPAddress:514. This directs the rsyslog to the assigned IP address using UDP on port 514.

To start the rsyslog service: *sudo systemctl start rsyslog*
To stop the rsyslog service: *sudo systemctl stop rsyslog*
To enable the servie to start at boot: *sudo systemctl enable rsyslog*
To check the status of the service: *sudo systemctl status rsyslog*

Viewing Logs

Logs gotten by rsyslog are saved in the /var/log folder. They can be viewed with any text viewer.

Configuring Syslogs

TCPdumps can be used to detect syslog traffic by running the following command: *sudo tcpdump -i any udp port 514 -w syslog_traffic.pcap.* This is illustrated below:

```
tee@ubuntu3: /

tee@ubuntu3:/$ sudo tcpdump -i any udp port 514 -w syslog_traffic.pcap
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
^C0 packets captured
4 packets received by filter
0 packets dropped by kernel
tee@ubuntu3:/$
```

The captured logs are sent to syslog_trafic.pcap. To view the contents of this file, any of the text viewers could be used. This is shown below:

```
tee@ubuntu3: /etc

tee@ubuntu3:/$ sudo find -name syslog_traffic.pcap
./syslog_traffic.pcap
./etc/syslog_traffic.pcap
tee@ubuntu3:/$ cd /etc
tee@ubuntu3:/etc$ less syslog_traffic.pcap
"syslog_traffic.pcap" may be a binary file.  See it anyway?
tee@ubuntu3:/etc$
```