Hands-on Lab Description



2020 Copyright Notice: The lab materials are only used for education purpose. Copy and redistribution is prohibited or need to get authors’ consent.

Please contact Professor Dijiang Huang: [Dijiang.Huang@asu.edu](mailto:Dijiang.Huang@asu.edu)

*CS-SYS-00008 –*

*Syslog (rsyslogd) Local Logging on Linux*

## CONTENTS

[1 Selectors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4](#_TOC_250008)

[1.1 Facility . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4](#_TOC_250007)

[1.2 Severity Levels . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4](#_TOC_250006)

[1.3 Syslogd Configuration Examples . . . . . . . . . . . . . . . . . . . . . . . . 5](#_TOC_250005)

[2 Support for Remote Logging . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6](#_TOC_250004)

[3 Task 1 Install rsyslogd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6](#_TOC_250003)

1. [Task 2 Log iptables messages to a local log file . . . . . . . . . . . . . . . . . 7](#_TOC_250002)
2. [Task 3 Additional Practice on rsyslog . . . . . . . . . . . . . . . . . . . . . . . . . . . 9](#_TOC_250001)

[6 Related Information and Resource . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9](#_TOC_250000)

1

### Category:

CS-SYS: Computer System

### Objectives:

1

Learn System Logging Service basis

Learn how to set up rsyslog services on a Linux machine

2

### Estimated Lab Duration:

1

Expert: 20 minutes

Novice: 100 minutes

2

### Difficulty Diagram:

Implementation



Design 5

4

3

2

1

1

Time

**Difficulty Table.**

Time 1

Measurements Values (0-5)

Design 0

Implementation 0

Configuration 2

Knowledge 2

Score (Average) 1

Configuration

Knowledge

### Required OS:

Linux: Ubuntu 18.04 LTS

### Lab Running Environment:

ThoTh Lab: https://thothlab.org



Client

Server



Network

1

Client: Linux (Ubuntu 18.04 LTS)

Server: Linux (Ubuntu 18.04 LTS)

Network Setup: connected through a local network

2

3

**Lab Preparations:**

Initial setup: basic Ubuntu 18.04 LTS is required for this lab

Basic Linux knowledge and operations. Reference Lab: CS-SYS-00001.

**Lab Overview**

Syslog is one of the most important standards used in Linux as it is the key file which helps you determine the different level of logs which are getting generated and stored every second while you are working on your Linux box. Syslog can be taken as “System Log”. Benefits of using *syslog*:

* Helps analyze the root cause for any trouble or problem caused.
* Reduce overall downtime helping to troubleshoot issues faster with all the logs.
* Improves incident management by active detection of issues.
* Self-determination of incidents along with auto resolution.
* Simplified architecture with different level of severity like error,info,warning, etc.

The *syslog.conf* file is the main configuration file for the *syslogd* which *logs* system messages on Unix/Linux Systems. This file specifies rules for logging. However, in this lab, we use syslog terms to illustrate the basic concepts and use rsyslog as the working logging system, which is the default system log on Ubuntu Linux distribution.

# Selectors

When configure syslog (syslog.conf is the one for syslogd and rsyslog configures are set up in the /etc/rsyslog.d/ folder), every rule consists of two fields, a *selector* field and an *action* field. These two fields are separated by one or more spaces or tabs. The selector field specifies a pattern of facilities and priorities belonging to the specified action.

The selector field itself again consists of two parts, a *facility* and a *priority*, separated by a period (“.”).

Both parts are case insensitive. For example,

Kern.none, mail.info, etc.

Here,

Kern = Facility

None = severity or priority

## Facility

The facility is one of the following keywords: *auth*, *authpriv*, *cron*, *daemon*, *kern*, *lpr*, *mail*, *mark*, *news*, *security* (same as *auth*), *syslog*, *user*, *uucp* and *local0* through *local7*. The keyword *security* should not be used anymore and mark is only for internal use and therefore should not be used in applications.

You may want to specify and redirect these messages here. The facility specifies the subsystem that pro- duced the message, i.e., all mail programs *log* with the *mail* facility (*LOG MAIL*) if they log using *syslog*. The facility numbers are given in Table CS-SYS-00008.1.

## Severity Levels

The priority is one of the following keywords, in ascending order: *debug, info, notice, warning, warn* (same as *warning*), *err, error* (same as *err*), *crit, alert, emerg, panic* (same as *emerg*). The keywords *error, warn* and *panic* are deprecated and should not be used anymore. The priority defines the severity of the message, which is presented in Table CS-SYS-00008.2.

You can specify multiple facilities with the same priority pattern in one statement using the comma (“,”)

### Table CS-SYS-00008.1

Facility Number.

|  |  |  |
| --- | --- | --- |
| Keyword | Facility | Description |
| 0 | kern | kernel messages |
| 1 | user | user level messages |
| 2 | mail | mail system |
| 3 | daemon | system daemons |
| 4 | auth | security/authorization messages |
| 5 | syslog | messages generated internally by syslogd |
| 6 | lpr | line printer subsystem |
| 7 | news | network news subsystem |
| 8 | uucp | UUCP subsystem |
| 9 | clock daemon |  |
| 10 | authpriv | security/authorization messages |
| 11 | ftp | FTP daemon |
| 12 | - | NTP susbsystem |
| 13 | - | log audit |
| 14 | - | log alert |
| 15 | cron | clock daemon |
| 16 | local0 | local use 0 (local0) |
| 17 | local1 | local use 1 (local1) |
| 18 | local2 | local use 2 (local2) |
| 19 | local3 | local use 3 (local3) |
| 20 | local4 | local use 4 (local4) |
| 21 | local5 | local use 5 (local5) |
| 22 | local6 | local use 6 (local6) |
| 23 | local7 | local use 7 (local7) |

operator. You may specify as much facilities as you want. Multiple selectors may be specified for a single action using the semicolon (“;”) separator. Remember that each selector in the selector field is capable to overwrite the preceding ones. Using this behavior you can exclude some priorities from the pattern.

## Syslogd Configuration Examples

The following presented modifications provide network support to the *syslogd* facility. For examples, if we want to log all the critical events on your Linux machine in a separate log file inside */var/log* with a name of critical.log Append the following line inside */etc/syslog.conf* :

\*.=crit /var/log/critical.log

### Table CS-SYS-00008.2

Facility Number.

|  |  |
| --- | --- |
| Integer | Facility |
| 0 | Emergency: System is unusable |
| 1 | Alert: Action must be taken immediately |
| 2 | Critical: critical conditions |
| 3 | Error: Error conditions |
| 4 | Warning: Warning conditions |
| 5 | Notice: Normal but significant conditions |
| 6 | Informational: Informational messages |
| 7 | Debug: Debug level messages |

Log all the kernel related messages in separate log file inside */var/log/firewall.log*:

# Add a new line Kern.\*

/var/log/firewall.log

# Add a new entry at the end of the below line

# Log anything (except mail) of level info or higher. # don’t log private authentication messages!

# don’t log kernel related events and messages

\*.info;mail.none;authpriv.none;cron.none;kern.none /var/log/messages

Redirect all the error logs to a remote user *root* and *dijiang* on their terminals:

# Messages of the priority alert will be directed

# to the operator #

\*.err root,dijiang

Log all the firewall warning level messages inside */var/log/firewall-warning.log*:

Kern.warn /var/log/firewall-warning.log

# Support for Remote Logging

Network support means that messages can be forwarded from one node running *syslogd* to another node running *syslogd* where they will be actually logged to a disk file.

The strategy is to have *syslogd* listen on a Unix/Linux domain socket for locally generated log messages. This behavior will allow *syslogd* to inter-operate with the *syslog* found in the standard *C* library. At the same time *syslogd* listens on the standard *syslog* port for messages forwarded from other hosts. To have this work correctly the */etc/services* file must have the following entry:

Syslog 514/udp

If this entry is missing *syslogd* neither can receive remote messages nor send them, because the UDP port cannot be opened. Instead, *syslogd* will die immediately, blowing out an error message. For example, to forward ALL messages to a remote host uses the following *syslog.conf* entry:

# Sample syslogd configuration files to

# Messages to a remote host forward all.

\*.\* @hostname

To forward all kernel messages to a remote host the configuration file would be as follows:

# Sample configuration files to forward all kernels # messages to a remote host.

kern.\*

@hostname

# Task 1 Install rsyslogd

The implementation of *syslog* on Unbuntu is *rsyslogd*. To determine if *rsyslogd* is installed on Ubuntu, you can issue the following command:

$ rsyslogd -v

$ sudo systemctl status rsyslog % check rsyslog running status

A common *rsyslogd* version reply will be looked like the following

rsyslogd 8.32.0, compiled with:

PLATFORM: x86\_64-pc-linux-gnu PLATFORM (lsb\_release -d): FEATURE\_REGEXP: Yes

GSSAPI Kerberos 5 support: Yes FEATURE\_DEBUG (debug build, slow code): No 32bit Atomic operations supported: Yes 64bit Atomic operations supported: Yes memory allocator: system default Runtime Instrumentation (slow code): No uuid support: Yes

systemd support: Yes

Number of Bits in RainerScript integers: 64

See [http://www.rsyslog.com](http://www.rsyslog.com/) for more information.

To check rsyslog running status:

$ sudo systemctl status rsyslog

A common *rsyslogd* status checking reply will be looked like the following

See [http://www.rsyslog.com](http://www.rsyslog.com/) for more information.

rsyslog.service - System Logging Service

Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)

Active: active (running) since Tue 2020-09-29 15:54:05 UTC; 1h 0min ago Docs: man:rsyslogd(8)

<http://www.rsyslog.com/doc/> Main PID: 21503 (rsyslogd)

Tasks: 4 (limit: 2317)

CGroup: /system.slice/rsyslog.service

--21503 /usr/sbin/rsyslogd -n

Sep 29 15:54:05 ubuntu systemd[1]: Starting System Logging Service... Sep 29 15:54:05 ubuntu systemd[1]: Started System Logging Service.

If not, you can install *rsyslogd*. At a terminal prompt, you can enter the following command:

$ sudo apt update && apt install rsyslog

$ sudo systemctl enable rsyslog

$ sudo systemctl start rsyslog

# Task 2 Log iptables messages to a local log file

* How to modify the iptables rules to let it log at the appropriate level?
* How to configure syslog to log the iptables messages to a different log file?
* To stop iptables messages to get logged into /var/log/messages ?

Now, let’s first configure an *rsyslog* version, and on the machine go to */etc/rsyslog.d* and reate a *01- client.conf* file.

$ sudo touch 01-client.conf % create a new configuration file with a small value

1. Assume that you are currently logging all the firewall warning level messages inside */var/log/firewall- warning.log*. Make the following addition in */etc/rsyslogd/01-client.conf* :

Kern.warning /var/log/firewall-warning.log

1. Restart the *rsyslog*, and make sure that there are no errors.

$

$

sudo systemctl restart rsyslog journalctl -f -u rsyslog

1. Make sure the iptables rule is logging at the appropriate level. This can be done by using the log-level switch. Default log-level is warning. Please refer to Table CS-SYS-00008.2 regarding to log levels. Below example will log ssh attempts:

$ iptables -I INPUT -p tcp --dport 22 -j LOG --log-level 4

Below example use to log ping and add the prefix “#### Firewall ####”.

$

iptables -I INPUT -p icmp --icmp-type ping -j LOG --log-level 4

--log-prefix "#### Firewall ####"

1. Run ssh and ping to test your firewall logs at */var/log/firewall-warning.log*. Open a new terminal and run the following command to view updated log information:

$ tail -f /var/log/firewall-warning.log

Edit the file and provide the following inputs:

\*.\* /var/log/syslog.log

If you want to print your iptables logs on your monitoring console, you can follow the steps below:

**Step 1** Add below entry in /etc/sysctl.conf

kernel.printk = 4 1 1 7 % refer to the reference link on printk

**Step 2** Run below command to make changes effectively at runtime.

$ sudo /sbin/sysctl -p /etc/sysctl.conf

**Step 3** Check the changes at below file

$ cat /proc/sys/kernel/printk

# Task 3 Additional Practice on rsyslog

Here, a few more example on how to add and modify logging facilities and corresponding logging files.

Append the following line inside */etc/rsyslogd/01-client.conf* :

\*.=crit /var/log/critical.log

Assume that you are currently logging all the kernel related messages in separate log file inside */var/log/- firewall.log*. Make the following addition in */etc/rsyslogd/01-client.conf* :

# Add a new line Kern.\*

/var/log/firewall.log

# Add a new entry at the end of the below line

# Log anything (except mail) of level info or higher. # don’t log private authentication messages!

# don’t log kernel related events and messages

\*.info;mail.none;authpriv.none;cron.none;kern.none /var/log/message.log

Restart the *rsyslog* server, and make sure that there are no errors on the client-side.

$ sudo systemctl restart rsyslog

$ journalctl -f -u rsyslog

# Related Information and Resource

Syslog Linux Man page: https://linux.die.net/man/8/syslogd

Syslog Wiki: https://en.wikipedia.org/wiki/Syslog Message logging with printk:

https://[www.kernel.org/doc/html/latest/core-api/printk-basics.html](http://www.kernel.org/doc/html/latest/core-api/printk-basics.html)