

System Administration (2)

IT401

Prepared by
Dr. Ahmed El-Douh

Lecture 9



System Administration 2

Lecture 9

IT401

System Log Architecture

Prepared by
Dr. Ahmed El-Douh

Objectives

Upon completion of this lecture, you will be able to:

- Understand the System Log Architecture and the purpose of system logging.
- Analyze Syslog data to identify system and network events.
- Locate and view log files on the system.
- Configure remote logging for centralized log management.

Outlines

- System Logging
- Analyze a Syslog
- Finding Log
- Configuring a remote logging solution
- Testing Environment
- Firewall

System Log Architecture



System Log Architecture

❖ System Logging

- Processes and the operating system kernel need to record events that occur.
- These logs are useful for auditing the system and troubleshooting issues.
- By convention, log files are stored in **/var/log**. Linux uses a standard logging system based on the **Syslog protocol**.
- Many programs use this logging system to record events and organize them into log files.
- In Red Hat Enterprise Linux, syslog messages are handled by two services: **systemd-journald** and **rsyslog**.

System Log Architecture

- The **rsyslog service** then sorts the syslog messages by type (or facility) and priority and writes them to persistent files in the `/var/log` directory.
- The **`/var/log` directory** holds various system- and service-specific log files maintained by rsyslog:

Overview of system log files

LOG FILE	PURPOSE
<code>/var/log/messages</code>	Most syslog messages are logged here. The exceptions are messages related to authentication and email processing, that periodically run jobs, and those which are purely debugging-related.
<code>/var/log/secure</code>	The log file for security and authentication-related messages and errors.
<code>/var/log/maillog</code>	The log file with mail server-related messages.
<code>/var/log/cron</code>	The log file related to periodically executed tasks.
<code>/var/log/boot.log</code>	Messages related to system startup are logged here.

System Log Architecture

PURPOSE	LOG FILE
Most syslog messages are logged here. The exceptions are messages related to authentication, email processing, and that periodically run jobs, or those which are purely debugging-related.	/var/log/messages
The log file for security and authentication-related messages and errors.	/var/log/secure
The directory to which rsyslog is writing all the log files.	/var/log
The log file with mail server-related messages.	/var/log/maillog
The log file related to periodically executed tasks.	/var/log/cron
Messages related to system startup are logged here.	/var/log/boot.log

System Log Architecture

SYSLOG FILES

- Many programs use the syslog protocol to log events to the system. Each log message is categorized by a facility (the type of message) and a priority (the severity of the message).
- The facilities which are available are documented by the `rsyslog.conf(5)` man page. The eight priorities are also standardized and ranked as follows:

Overview of syslog priorities

CODE	PRIORITY	SEVERITY
0	emerg	System is unusable.
1	alert	Action must be taken immediately.
2	crit	Critical condition.
3	err	Non-critical error condition.

System Log Architecture

4	warning	Warning condition.
5	notice	Normal but significant event.
6	info	Informational event.
7	debug	Debugging-level message.

- The **rsyslogd** service uses the facility and priority of log messages to determine how to handle them.
- This is configured by the file `/etc/rsyslog.conf` and by ***.conf files** in `/etc/rsyslog.d`. Programs and administrators can change rsyslogd configuration in a way that will not be overwritten by updates to rsyslog by putting customized files with a .conf suffix in the **/etc/rsyslog.d** directory.

System Log Architecture

The ##### RULES

- section of **/etc/rsyslog.conf** contains directives that define where log messages are saved.
- The left side of each line indicates the facility and severity of the log message the directive matches.
- The **rsyslog.conf file** can contain the character * as a wild card in the facility and severity field, where it either stands for all facilities or all severities.
- The right side of each line indicates what file to save the log message in.
- Log messages are normally saved in files in the **/var/log directory**.
- A message handled by rsyslog can appear in multiple different log files.

System Log Architecture

- To prevent that, the severity field can be set to **none**, which means that none of the messages directed to this facility are added to the specified log file.
- Instead of logging syslog messages to a file, they can be printed to the terminals of all logged-in users.
- In the default rsyslog.conf file, this is done for all messages that have "**emerg**" priority.

SAMPLE RULES SECTION OF rsyslog.conf

RULES

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                /var/log/secure

# Log all the mail messages in one place.
mail.*                                      -/var/log/maillog

# Log cron stuff
cron.*                                    /var/log/cron

# Everybody gets emergency messages
*.emerg                                    :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                            /var/log/spooler

# Save boot messages also to boot.log
local7.*                                /var/log/boot.log
```

System Log Architecture

ANALYZE A SYSLOG ENTRY

- The system logs written by **rsyslog** start with the oldest message on top and the newest message at the end of the log file.
- All log entries in log files managed by **rsyslog** are recorded in a standard format.
- The following example will explain the anatomy of a log file message in the `/var/ log/secure` log file:

```
❶ Feb 11 20:11:48 ❷ localhost ❸ sshd[1433]: ❹ Failed password for student from  
172.25.0.10 port 59344 ssh2
```

- ❶ The time stamp when the log entry was recorded.
- ❷ The host from which the log message was sent.
- ❸ The program or process that sent the log message.
- ❹ The actual message sent.

System Log Architecture

MONITOR A LOG FILE WITH tail

- It is especially helpful for reproducing problems and issues to monitor one or more log files for events.
- The **tail -f /path/to/file** command outputs the **last 10 lines** of the file specified and continues to output new lines as they get written to the monitored file.
- To monitor for failed login attempts on one terminal, run **ssh as user root** while a user tries to log in to the **serverX machine**:

```
[root@serverX ~]$ tail -f /var/log/secure
...
Feb 10 09:01:13 localhost sshd[2712]: Accepted password for root from
172.25.254.254 port 56801 ssh2
Feb 10 09:01:13 localhost sshd[2712]: pam_unix(sshd:session): session opened for
user root by (uid=0)
```

System Log Architecture

❖ **FINDING LOG ENTRIES**, you will reconfigure rsyslog to write specific messages to a new log file.

- **OUTCOMES** The rsyslog service writes all messages with priority debug to the `/var/log/messagesdebug` log file for temporary troubleshooting purposes.

1. Configure rsyslog on serverX to log all messages with severity debug, or higher, for any service into a newly created log file `/var/log/messages-debug` by adding the rsyslog configuration file `/etc/rsyslog.d/debug.conf`. Verify that a generated debug log message with the logger command arrives in the `/var/log/messages-debug` log file.

- **Change the rsyslog configuration to log all messages with severity debug to `/var/log/messages-debug` on serverX by adding the `/etc/rsyslog.d/ debug.conf` file.**
- **Restart the rsyslog service on server.**

```
[root@serverX ~]#  
systemctl restart rsyslog
```

```
[root@serverX ~]#  
echo "*.debug /var/log/messages-debug" >/etc/rsyslog.d/debug.conf
```

System Log Architecture

2. Generate a debug log message with the `logger` command and verify that the message gets logged to the log file `/var/log/messages-debug` with the `tail` command on serverX.

- Monitor the `/var/log/messages-debug` with the `tail` command on serverX.

```
[root@serverX ~]#  
tail -f /var/log/messages-debug
```

- On a separate terminal window, use the `logger` command to generate a debug message on serverX.

```
[root@serverX ~]#  
logger -p user.debug "Debug Message Test"
```

- Switch back to the terminal still running the `tail -f /var/log/messages-debug` command and verify the message sent with the `logger` command shows up.

```
[root@serverX ~]# tail -f /var/log/messages-debug  
...  
Feb 13 10:37:44 localhost root: Debug Message Test
```

System Log Architecture

Configuring a remote logging solution

- To ensure that logs from various machines in your environment are recorded centrally on a logging server, you can configure the **Rsyslog** application to record logs that fit specific criteria from the client system to the server.
 - ✓ **The Rsyslog logging service**
- The Rsyslog application, in combination with the **systemd-journald service**, provides local and remote logging support in Red Hat Enterprise Linux.
- The rsyslogd daemon continuously reads syslog messages received by the systemd-journald service from the Journal. rsyslogd then filters and processes these syslog events and records them to **rsyslog log files** or forwards them to other services according to its configuration.

System Log Architecture

- The **rsyslogd** daemon also provides extended filtering, encryption protected relaying of messages, input and output modules, and support for transportation using the **TCP and UDP protocols**.
- In **/etc/rsyslog.conf**, which is the main configuration file for **rsyslog**, you can specify the rules according to which rsyslogd handles the messages.
- Generally, you can classify messages by their source and topic (**facility**) and urgency (**priority**) and then assign an action that should be performed when a message fits these criteria.
- In **/etc/rsyslog.conf**, you can also see a list of log files maintained by rsyslogd. Most log files are located in the **/var/log/** directory. Some applications, such as **httpd** and **samba**, store their log files in a subdirectory within **/var/log/**.

How to Set Up Remote Logging on Linux Using Rsyslog

- **Logs** are a critical component of any software or operating system. Logs usually record user's actions, system events, network activity and so much more, depending on what they are intended for. One of the most widely used logging systems on Linux systems is **rsyslog**.
- **Rsyslog** is a powerful, secure and high-performance log processing system which accepts data from different types of source (systems/applications) and outputs it into multiple formats.
- It has evolved from a regular **syslog** daemon to a fully-featured, enterprise level logging system. It is designed in a client/server model, therefore it can be configured as a client and/or as a central logging server for other servers, network devices, and remote applications.

System Log Architecture

Testing Environment

➤ For the purpose of this guide, we will use the following hosts:

Server: 192.168.241.140

Client: 172.31.21.58

How to Install and Configure Rsyslog Server

- Most Linux distributions come with the **rsyslog** package preinstalled.
- In case it's not installed, you can install it using your Linux package manager tool as shown.

```
$ sudo yum update && yum install rsyslog          #CentOS 7  
$ sudo apt update && apt install rsyslog          #Ubuntu 16.04, 18.04
```

System Log Architecture

- ❑ Once **rsyslog** installed, you need to start the service for now, enable it to auto-start at boot and check it's status with the systemctl command.

```
systemctl start rsyslog  
systemctl enable rsyslog  
systemctl status rsyslog
```

- ❑ The main rsyslog configuration file is located at **/etc/rsyslog.conf**, which loads modules, defines the global directives, contains rules for processing log messages and it also includes all config files in **/etc/rsyslog.d/** for various applications/services.

System Log Architecture

- ❑ By default, **rsyslog** uses the **imjournal** and **imuxsock** modules for importing structured log messages from **systemd journal** and for accepting syslog messages from applications running on the local system via Unix sockets, respectively.

```
#### MODULES ####  
  
# The imjournal module bellow is now used as a message source instead of imuxsock.  
$ModLoad imuxsock # provides support for local system logging (e.g. via logger command)  
$ModLoad imjournal # provides access to the systemd journal  
#$ModLoad imklog # reads kernel messages (the same are read from journald)  
$ModLoad immark # provides --MARK-- message capability
```

System Log Architecture

- ❖ To configure rsyslog as a network/central logging server, you need to set the protocol (either **UDP** or **TCP** or **both**) it will use for remote syslog reception as well as the port it listens on.
- ❖ If you want to use a **UDP** connection, which is faster but unreliable, search and uncomment the lines below (replace **514** with the port you want it to listen on, this should match the port address that the clients send messages to, we will look at this more when configuring a **rsyslog client**).

```
$ModLoad imudp  
$UDPServerRun 514
```

System Log Architecture

- ☐ To use **TCP** connection (which is slower but more reliable), search and uncomment the lines below.

```
$ModLoad imtcp  
$InputTCPServerRun 514
```

- ☐ In this case, we want to use both UDP and TCP connections at the same time.

```
# Provides UDP syslog reception  
$ModLoad imudp  
$UDPServerRun 514  
  
# Provides TCP syslog reception  
$ModLoad imtcp  
$InputTCPServerRun 514  
[ ]  
-- INSERT -- 21,1 14%
```

System Log Architecture

- ❑ Next, you need to define the **ruleset** for processing remote logs in the following format.

```
facility.severity_level destination (where to store log)
```

Where:

- **facility**: is type of process/application generating message, they include auth, cron, daemon, kernel, local0..local7. Using * means all facilities.
- **severity_level**: is type of log message: emerg-0, alert-1, crit-2, err-3, warn-4, notice-5, info-6, debug-7. Using * means all severity levels and none implies no severity level.
- **destination**: is either local file or remote rsyslog server (defined in the form IP:port).

System Log Architecture

- ❑ We will use the following **ruleset** for collecting logs from remote hosts, using the **RemoteLogs** template. Note that these rules must come before any rules for processing local messages, as shown in the screenshot.

```
$template RemoteLogs, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*. * ?RemoteLogs
& ~
```

```
$ModLoad imudp
$UDPServerRun 514

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

##rules for processing the remote logs
$template RemoteLogs, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*. * ?RemoteLogs
& ~

#### GLOBAL DIRECTIVES ####
# Where to place auxiliary files
$WorkDirectory /var/lib/rsyslog

# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
-- INSERT -- 15,1 18%
```

Define Ruleset for Rsyslog-Logging

System Log Architecture

- ❑ Looking at the above **ruleset**, the first rule is “**\$template RemoteLogs, “/var/log/%HOSTNAME%/PROGRAMNAME%.log”**”.
- ❑ The directive **\$template** tells rsyslog daemon to gather and write all of the received remote messages to distinct logs under **/var/log**, based on the **hostname** (client machine name) and remote client facility (program/application) that generated the messages as defined by the settings present in the template **RemoteLogs**.
- ❑ The second line “***.* ?RemoteLogs**” means record messages from all facilities at all severity levels using the **RemoteLogs** template configuration.

System Log Architecture

- ☐ The final line “& ~” instructs rsyslog to stop processing the messages once it is written to a file.
 - ☐ If you don't include “& ~”, messages will instead be written to the local files.
- There are many other templates that you can use, for more information, see the rsyslog configuration man page (**man rsyslog.conf**) or refer to the [Rsyslog online documentation](#). That's it with configuring the rsyslog server.
- ☐ Save and close the configuration file. To apply the recent changes, restart rsyslog daemon with the following command.

```
systemctl restart rsyslog
```

System Log Architecture

- ❑ Now verify the rsyslog network sockets. Use the **ss command** (or netstat with the same flags) command and pipe the output to **grep to filter** out **rsyslogd** connections.

```
grep "rsyslog"
```

```
[admin@vps1 ~]$ sudo ss -tulnp | grep "rsyslogd"
udp    UNCONN    0      0      *:514      *:~
udp    UNCONN    0      0      :::514     :::*
tcp    LISTEN    0      25     *:514      *:~
tcp    LISTEN    0      25     :::514     :::*
users: (("rsyslogd",pid=25863,fd=3))
users: (("rsyslogd",pid=25863,fd=4))
users: (("rsyslogd",pid=25863,fd=5))
users: (("rsyslogd",pid=25863,fd=6))
[admin@vps1 ~]$
```

Check Rsyslog Network Status

- ❑ If the system has firewall enabled, you need to open port **514** to allow both **UDP/TCP** connections to the rsyslog server, by running.

```
firewall-cmd --permanent --add-port=514/udp
firewall-cmd --permanent --add-port=514/tcp
firewall-cmd --reload
```

System Log Architecture

How to Configure Rsyslog Client to Send Logs to Rsyslog Server

- ☐ Now on the client system, check if the rsyslog service is running or not with the following command.

```
systemctl status rsyslog
```

- ☐ If it's not installed, install it and start the service as shown earlier on.

```
systemctl start rsyslog  
systemctl enable rsyslog  
systemctl status rsyslog
```

System Log Architecture

- ❑ Once the **rsyslog service** is up and running, open the main configuration file where you will perform changes to the default configuration.

```
vim /etc/rsyslog.conf
```

- ❑ To force the rsyslog daemon to act as a log client and forward all locally generated log messages to the remote rsyslog server, add this forwarding rule, at the end of the file as shown

```
# forwarding rules, duplicate the whole block!
# Remote Logging (we use TCP for reliable delivery)
#
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
#$ActionQueueFileName fwdRule1 # unique name prefix for spool files
#$ActionQueueMaxDiskSpace 1g    # 1gb space limit (use as much as possible)
#$ActionQueueSaveOnShutdown on  # save messages to disk on shutdown
#$ActionQueueType LinkedList    # run asynchronously
#$ActionResumeRetryCount -1     # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#
```

- ❑ Save the changes and close the configuration file. To apply the above settings, restart the rsyslog daemon.

```
systemctl restart rsyslog
```

System Log Architecture

How to Monitor Remote Logging on the Rsyslog Server

- ☐ The final step is to verify if the rsyslog is actually receiving and logging messages from the client, under **/var/log**, in the form **hostname/programname.log**.
- ☐ Run a ls command to long listing of the parent logs directory and check if there is a directory called **ip-172.31.21.58** (or whatever your client machine's hostname is).

```
$ ls -l /var/log/
```

- ☐ If the directory exists, check the log files inside it, by running.

```
ls -l /var/log/ip-172-31-21-58/
```

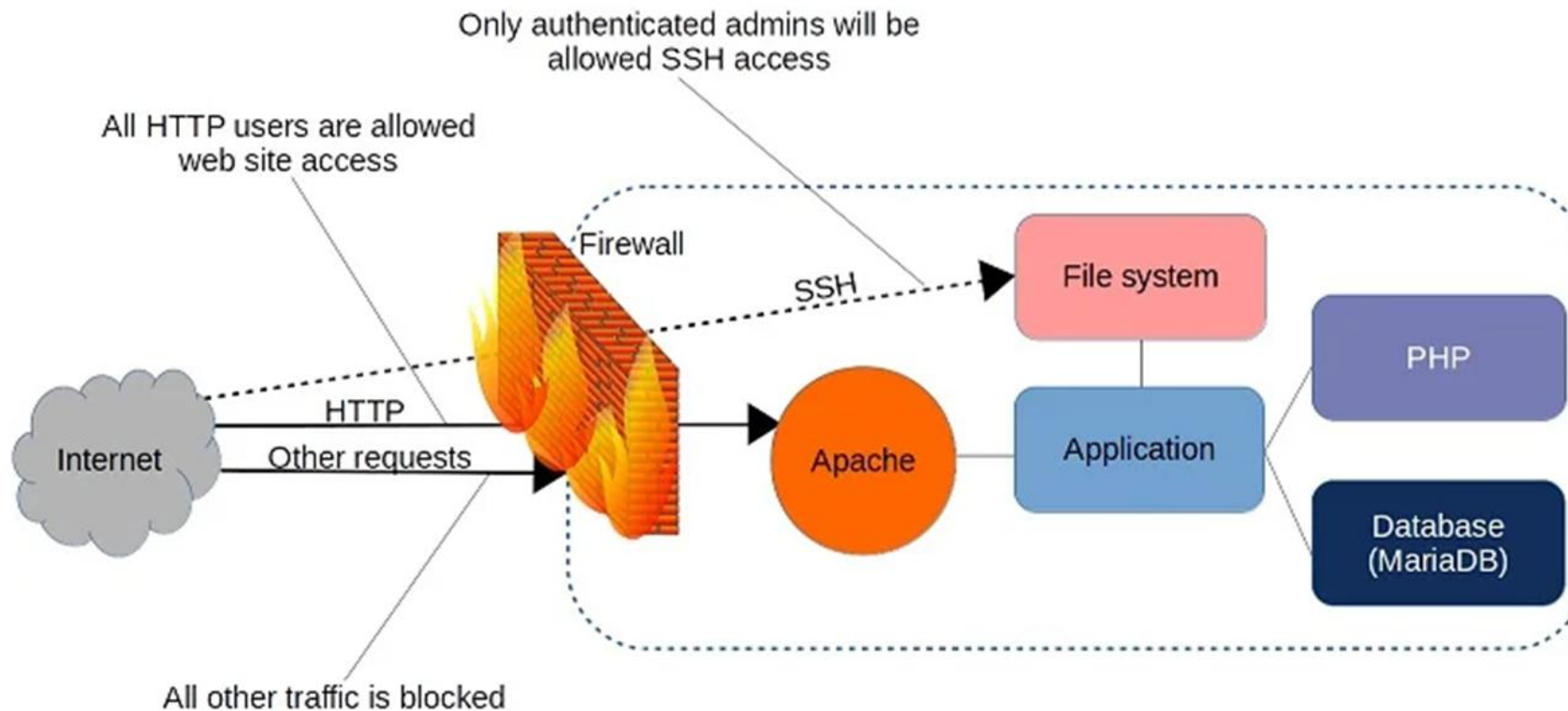
System Log Architecture

➤ What is a Firewall?

Firewall is a network security system that filters and controls the traffic on a predetermined set of rules. This is an intermediary system between the device and the internet.

- A firewall is a set of rules. When a data packet moves into or out of a protected network space, its contents (in particular, information about its origin, target, and the protocol it plans to use) are tested against the firewall rules to see if it should be allowed through. Here's a simple example:

System Log Architecture



A firewall can filter requests based on protocol or target-based rules.

System Log Architecture

Configure HTTP access using firewalld

- ❑ As you might have guessed from its name, firewalld is part of the [systemd](#) family. Firewalld can be installed on Debian/Ubuntu machines, but it's there by default on Red Hat and CentOS. If you've got a web server like Apache running on your machine, you can confirm that the firewall is working by browsing to your server's web root. If the site is unreachable, then firewalld is doing its job.
- ❑ You'll use the `firewall-cmd` tool to manage firewalld settings from the command line. Adding the `--state` argument returns the current firewall status:

```
# firewall-cmd --state  
running
```

System Log Architecture

- ☐ By default, firewalld will be active and will reject all incoming traffic with a couple of exceptions, **like SSH**.
- ☐ That means your website won't be getting too many visitors, which will certainly save you a lot of data transfer costs.
- ☐ As that's probably not what you had in mind for your web server, though, you'll want to open the HTTP and HTTPS ports that by convention are designated as 80 and 443, respectively.
- ☐ firewalld offers two ways to do that.

System Log Architecture

- ❑ One is through the `-add-port` argument that references the port number directly along with the network protocol it'll use (TCP in this case).
- ❑ The `-permanent` argument tells firewalld to load this rule each time the server boots:

```
# firewall-cmd --permanent --add-port=80/tcp  
# firewall-cmd --permanent --add-port=443/tcp
```

System Log Architecture

- ☐ Curious as to the current settings on your firewall? Run `-list-services`:

```
# firewall-cmd --list-services  
dhcpv6-client http https ssh
```

- ☐ The `-reload` argument will apply those rules to the current session:
`firewall-cmd -reload`



Thank You