syslog

From Wikipedia, the free encyclopedia

In computing, **syslog** is a widely used standard for message logging. It permits separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them.

Computer system designers can use syslog for system management and security auditing as well as general informational, analysis, and debugging messages. A wide variety of devices (such as printers and routers) and message receivers across multiple platforms use the syslog standard. Because of this, system designers can use syslog to integrate log data from different types of systems in a central repository.

In the syslog standard, each messages is labeled with a facility code and assigned a severity label. The facility code indicates which of the following software types generated the message: `auth`, `authpriv`, `daemon`, `cron`, `ftp`, `lpr`, `kern`, `mail`, `news`, `syslog`, `user`, `uucp`, or `local0` ... `local7`. The severity designations, from most to least severe, are: Emergency, Alert, Critical, Error, Warning, Notice, Info, and Debug.

Implementations of syslog exist for many operating systems. Specific configuration may permit directing messages to various devices (e.g., console), files (e.g., `/var/log/`), or remote syslog servers. Most implementations provide a command line utility, often called *logger*, that can send messages to the log. Some implementations permit filtering and display of syslog messages.

In 2009, the Internet Engineering Task Force (IETF) standardized syslog in RFC 5424.

Contents

- 1 History
- 2 Outlook
- 3 Facility levels
- 4 Severity levels
- 5 Format of a Syslog packet
  - 5.1 Priority
    - 5.1.1 Calculating Priority Value
      - 5.1.1.1 Calculating Facility and Severity Values from a Priority Value
  - 5.2 Header
  - 5.3 Message
- 6 Limitations
- 7 Protocol
- 8 Internet standards
- 9 See also
- 10 References
- 11 External links

History

Syslog was developed in the 1980s by Eric Allman as part of the Sendmail project, and was initially used solely for Sendmail. It proved so valuable that other applications began using it as well. Syslog has since become the standard logging solution on Unix and Unix-like systems; there have also been a variety of syslog implementations on other operating systems and is commonly found in network devices such as routers.

Syslog functioned as a *de facto* standard, without any authoritative published specification, and many implementations existed, some of which were incompatible. The Internet Engineering Task Force documented the status quo in RFC 3164. It was made obsolete by subsequent additions in RFC 5424.[1]

At different points in time, various companies have attempted patent claims on syslog.[2][3] This had little effect on the use and standardization of the protocol.

Outlook

Various groups are working on draft standards detailing the use of syslog for more than just network and security event logging, such as its proposed application within the health care environment.

Regulations, such as SOX, PCI DSS, HIPAA, and many others are requiring organizations to implement comprehensive security measures, which often include collecting and analyzing logs from many different sources. Syslog has proven to be an effective format to consolidate logs, as there are many open source and proprietary tools for reporting and analysis. Converters exist from Windows Event Log as well as other log formats to syslog.

An emerging area of managed security services is the collection and analysis of syslog records for organizations. Companies calling themselves Managed Security Service Providers attempt to apply analytics techniques (and sometimes artificial intelligence algorithms) to detect patterns and alert customers to problems.

Facility levels

A facility level is used to specify what type of program is logging the message. This lets the configuration file specify that messages from different facilities will be handled differently.[4] The list of facilities available:[5] (defined by RFC 3164 (http://tools.ietf.org/html/rfc3164))

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 subsystem
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)

The mapping between Facility Number and Keyword is not uniform over different operating systems and different syslog implementations.[6]
For cron either 9 or 15 or both may be used.
The confusion is even greater regarding auth/authpriv. 4 and 10 are most common but 13 and 14 may also be used.

Severity levels

RFC 5424 (http://tools.ietf.org/html/rfc5424) defines eight severity levels:

Code	Severity	Keyword	Description	General Description
0	Emergency	emerg (panic)	System is unusable.	A "panic" condition usually affecting multiple apps/servers/sites. At this level it would usually notify all tech staff on call.
1	Alert	alert	Action must be taken immediately.	Should be corrected immediately, therefore notify staff who can fix the problem. An example would be the loss of a primary ISP connection.
2	Critical	crit	Critical conditions.	Should be corrected immediately, but indicates failure in a secondary system, an example is a loss of a backup ISP connection.
3	Error	err (error)	Error conditions.	Non-urgent failures, these should be relayed to developers or admins; each item must be resolved within a given time.
4	Warning	warning (warn)	Warning conditions.	Warning messages, not an error, but indication that an error will occur if action is not taken, e.g. file system 85% full - each item must be resolved within a given time.
5	Notice	notice	Normal but significant condition.	Events that are unusual but not error conditions - might be summarized in an email to developers or admins to spot potential problems - no immediate action required.
6	Informational	info	Informational messages.	Normal operational messages - may be harvested for reporting, measuring throughput, etc. - no action required.
7	Debug	debug	Debug-level messages.	Info useful to developers for debugging the application, not useful during operations.

A common mnemonic used to remember the syslog levels from bottom to top is: "Do I Notice When Evenings Come Around Early".

Format of a Syslog packet

The full format of a Syslog message seen on the wire has three distinct parts:

```
<PRI> HEADER MSG
```

The total length of the packet cannot exceed 1,024 bytes, and there is no minimum length.

Priority

The ***PRI*** part is a number that is enclosed in angle brackets. This represents both the Facility and Severity of the message. This number is an eight bit number. The first 3 least significant bits represent the Severity of the message (with 3 bits you can represent 8 different Severities) and the other 5 bits represent the Facility of the message. You can use the Facility and the Severity values to apply certain filters on the events in the Syslog Daemon.

Calculating Priority Value

The Priority value is calculated by first multiplying the Facility number by 8 and then adding the numerical value of the Severity. For example, a kernel message (Facility=0) with a Severity of Emergency (Severity=0) would have a Priority value of 0. Also, a "local use 4" message (Facility=20) with a Severity of Notice (Severity=5) would have a Priority value of 165. In the PRI part of a Syslog message, these values would be placed between the angle brackets as <0> and <165> respectively.

Calculating Facility and Severity Values from a Priority Value

This is a calculation derived from the previous one. To get the Facility number implied in a given Priority value, divide the Priority number by 8. The whole number part is the Facility. To get the Severity, multiply the Facility by 8 and subtract that number from the Priority.

For example:

Priority = 191

191/8 = 23.875

Facility = 23

Severity = 191 - (23 * 8) = 7

Work backword to check the formula: 23*8 = 184 + 7 = 191

Another Method:

To get the Facility number from a given priority value, divide priority by 8. The whole number is the Facility. Then to get Severity, take Priority mod 8.

For example:

Priority = 191

191 / 8 = 23.875

Facility = 23

Severity = 191 mod 8 = 7

Header

The ***HEADER*** part contains the following:

- Timestamp -- the date and time at which the message was generated. This is picked up from the sending system's system time which might differ from the receiving system's system time
- Hostname or IP address of the device.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.