Syslog Server Configurations

Syslog Server Configurations

This application note describes the configuration and setup of a syslog server for use with the EdgeWave ePrism mail exchanger. This scenario configures the ePrism mail exchanger to log remotely to a properly configured syslog server.

What is Syslog?

Syslog is a client-server protocol for IP networks, as defined in RFC 3164. Syslog uses UDP on port number 514. Syslog clients are implemented in devices ranging from home wireless routers to mainframes. Syslog has become the de facto standard for logging. The protocol defines 24 facilities and 8 severities, as shown in the following tables:

Numerical Code	Facility
0	kernel messages
1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon
16	local use 0 (local0)

17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Numerical Code	Severity
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 condition
6	Informational: informational messages
7	Debug: debug-level messages

Security Concerns

The syslog service was initially designed as a networked logging service using an unauthenticated UDP socket for message delivery. Before long, Denial of Service (DOS) attacks on the syslog service began to surface, and vendors began to change the default configurations to use (host-only) UNIX domain sockets.

This document discusses transmission of log messages from a ePrism mail exchanger to a remote logging host, and requires allowing incoming network connections from this remote logging host. Be aware of the dangers of doing this, and take steps to protect the logging host. The following methods supply adequate protection:

- A firewall that allows only known logging clients to connect to your syslog daemon.
- A syslog daemon configured to only bind to an internal interface.

In addition to the DOS threat of inundating the logging host with huge numbers of nuisance messages, there is the threat of log file growth without bound. This can easily happen with legitimate logging messages, especially if logging verbosity is increased (for example, for debugging purposes) and then forgotten. To mitigate this threat:

- Use built-in log file size limiting directives.
- Use an external log file rotation facility, like logrotate (**`http://sourceforge.net/projects/logrotate`**).

Red Condor recommends logrotate to reduce the hazard of self-inflicted DOS.

UNIX/Linux Syslog Servers

syslogd

Syslogd is the traditional syslog server, originating in early BSD UNIX distributions. It is configured with a single file (syslog.conf) typically located in **`/etc`**. The standard syslogd server has very little filtering capability. It can direct messages of a given facility-severity tuple to an individual output file, or to a remote host. Multiple facility-severity tuples are commonly directed to the same destination.

After determining the facility that the ePrism mail exchanger is logging to, configuration of syslogd is straightforward. For example, assume the ePrism mail exchanger is logging to 'local7'. Simply add the line:

```
local7.*            /some/log/file
```

to **`/etc/syslog.conf`** and restart the logging daemon.

Configuring syslogd to accept connections from remote hosts, however, is typically not done in **`/etc/syslog.conf`**. Rather it is a command line option when the daemon is started. To ensure syslogd is started with the proper options, investigate the mechanism by which services are started. For example with:

- **Solaris:** start syslogd with a '-T' option to allow it to receive remote log messages.
- **HP-UX:** specify '-N' to not listen for log messages on a socket.
- **Linux:** run syslogd, a '-r' option to allow it to receive remote log messages.

On Solaris systems, syslog is started with the 'svc' subsystem. Most other platforms use some form of init. Consult your vendor documentation for the exact requirements.

syslog-ng

Syslog-ng is a second-generation logger intended to replace syslog on UNIX and Linux systems. It is also available with Cygwin for Windows systems. Red Condor recommends syslog-ng for logging mail exchanger messages.

Syslog-ng has flexible message filtering capabilities. All relevant configurations are done in the config file **`syslog-ng.conf`**. Some distributions place this file in **`/etc/syslog-ng/syslog-ng.conf`**, others put it in **`/etc/syslog-ng.conf`**. To configure syslog-ng to receive remote logging messages, add a line in the source block of the format:

```
udp(ip("0.0.0.0") port(514));
```

This will allow incoming log messages from anywhere. To restrict incoming messages to a specific interface, change the 0.0.0.0 to the IP address of the interface on the logging host that you want to receive messages.

Next, make a filter specification to select your Red Condor mail exchanger messages. For example:

```
filter f_RCI { facility(local7); };  #assuming Red Condor logs on local7
```

Finally, specify a logging destination. For example:

```
destination RCI_Log { file("/var/log/RCILog"); };
log { source(src); filter(f_RCI); destination(RCI_Log); };
```

Since syslog is an integral part of any system, these lines must be merged into an existing configuration. You can make further configurations to ensure proper file permissions, timestamp messages, and so forth.

Windows Syslog Servers

Perhaps in a nod to syslog's dominance, most Windows syslog software available is designed to send eventlog events to a remote syslog server. A few syslog servers exist. Configure one to forward syslog events to the Windows Event Log.

Microsoft Windows Services for UNIX

Windows Services for UNIX is distributed free of charge by Microsoft. It supplies a syslog daemon similar to syslogd discussed in *syslogd* (on page 5).

The syslogd supplied does not log to the Windows Event Log. Rather it is completely orthogonal to the traditional Windows logging facility. By using the supplied korn shell, and the supplied editor **/bin/vi** you can edit **/etc/syslog.conf** to send logging with the correct facility and severity to a log file. See *syslogd* (on page 5) for more information.

Cygwin syslog-ng

Cygwin's syslog-ng configures exactly the same as described above for UNIX/Linux systems. EdgeWave recommends using Cygwin and syslog-ng on Windows systems for syslog logging. See **http://www.cygwin.com** for more information about Cygwin.

After installing Cygwin and syslog-ng, the program **/usr/bin/syslog-ng-configure** will create a default setup that can then be further customized.

CodeProject Syslog Daemon for Windows Event Log

The CodeProject Syslog daemon for Windows Event Log, (**http://www.codeproject.com/KB/IP/Syslogd.aspx**) is a C# demonstration project that nonetheless might be interesting because it offers tight integration with the Windows Event Log.

Further Reading

For more information about the Windows Event Log/Event Viewer, visit: **http://www.syslog.org/wiki/Eventlog/EventLogWiki**.

Mac OS-X Syslog

Mac OS-X Syslog is similar in functionality to the traditional syslogd discussed in *syslogd* (on page 5). However, syslog-ng can easily replace it. Since OS-X is UNIX, configuration is similar to the methods outlined above.