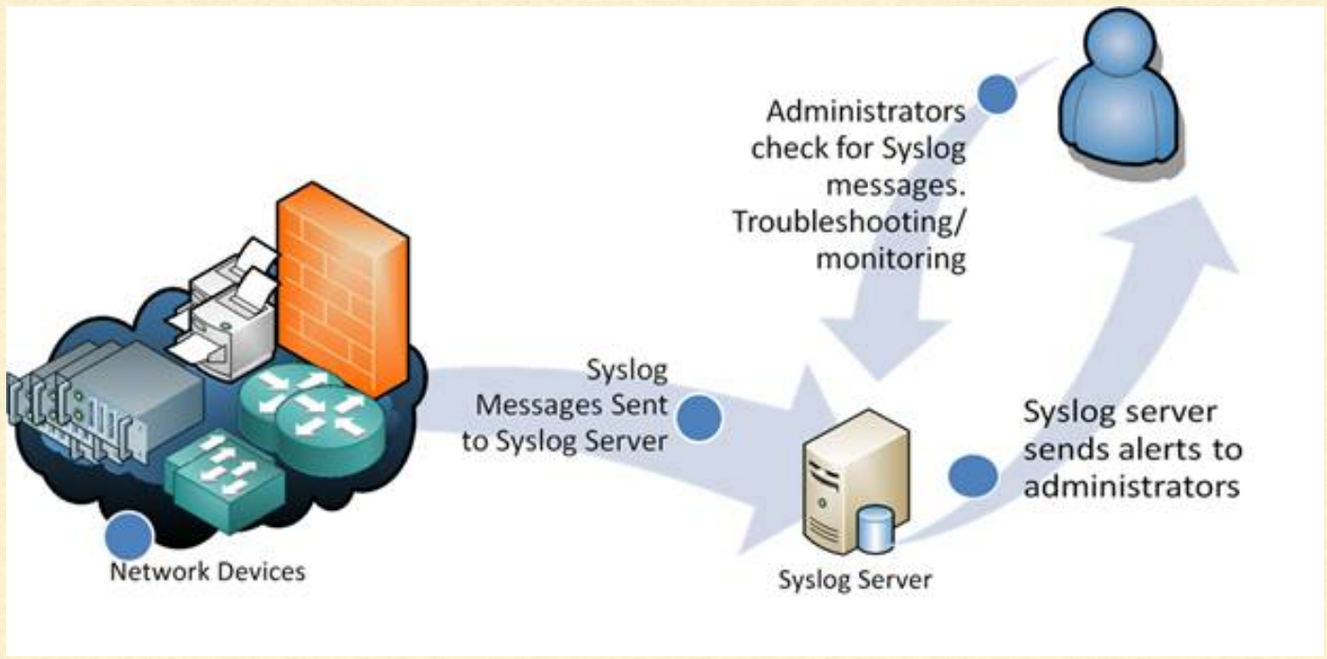
MONITORING AND LOG MANAGEMENT

Message Capture

NETS1037 - Winter 2020

MESSAGE CAPTURE

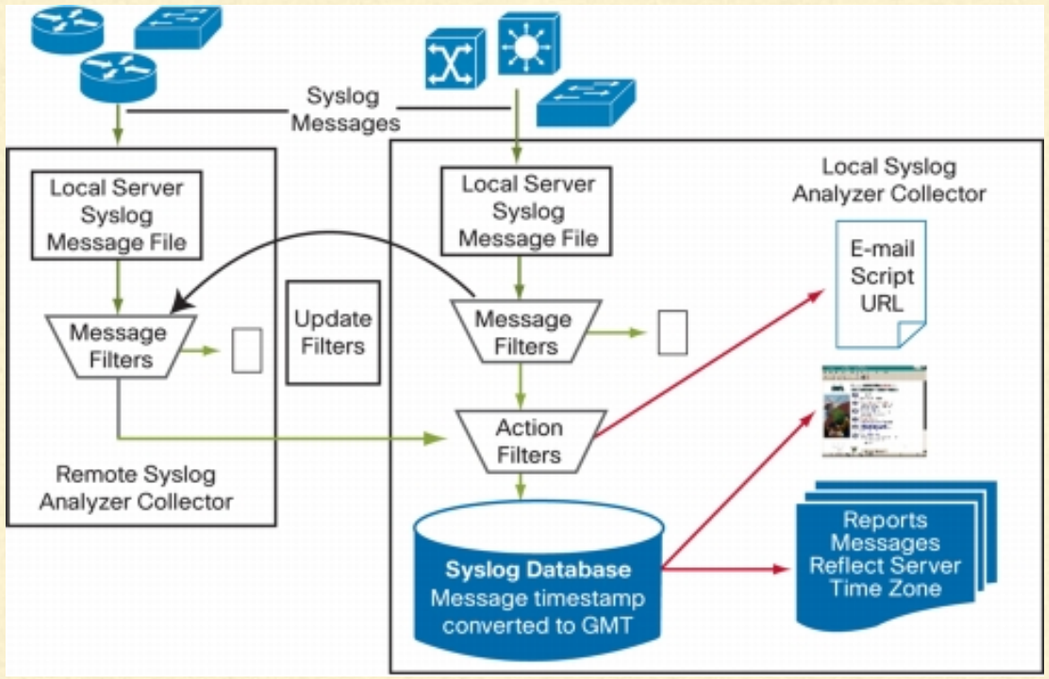
- Log messages are one of the primary ways we find out what has been happening on otherwise unattended servers
- Different operating systems produce various kinds of messages and log them according to their configuration
- Many operating systems include support for syslog-style message logging



https://www.networkmanagementsoftware.com/what-is-syslog/

NETWORK DEVICE LOGS

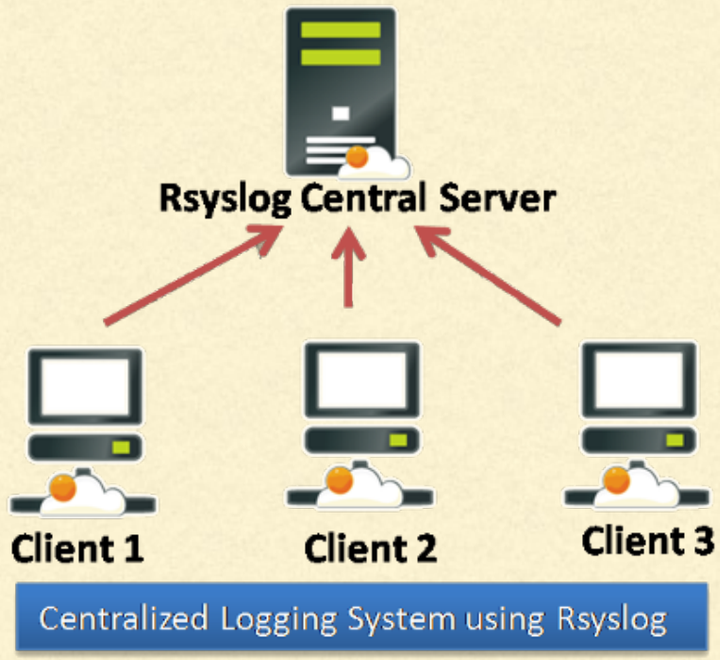
- Most network devices have built-in logging which may or may not be enabled by default
- The logs may or may not persist through reboots, /var may be a ramdisk on your device
- The logs are typically viewed through a web interface, Status->Package Logs and Status->System Logs for pfsense
- The logging may or may not be configurable with respect to what is logged, Status->System Logs->Settings for pfsense
- Remote logging is often preferred



https://www.cisco.com/c/en/us/products/collateral/services/high-availability/white_paper_c11-557812.html

NETWORK DEVICE REMOTE LOGGING

- Set the log host IP address, and optionally tcp or udp
- Verify the loghost is receiving the log entries
- Disable local logging if you don't want local logs to consume space on the device
- Ideally only log messages relevant to device management to a remote server



SERVER LOGS

- Servers record logs in multiple ways
- The kernel puts messages in a ring buffer, which can be viewed using the dmesg command, and klogd may be used to write those messages to rsyslog
- The system init scripts or startup services daemons write the contents of the ring buffer to /var/log/dmesg prior to starting rsyslog in order to preserve boot messages from the kernel, unless klogd is running

SERVER LOG LOCATIONS

- Some distros use /var/log/dmesg, some use /var/log/kern.log

- Various daemons directly write log files (e.g. apache, postfix), some are configurable

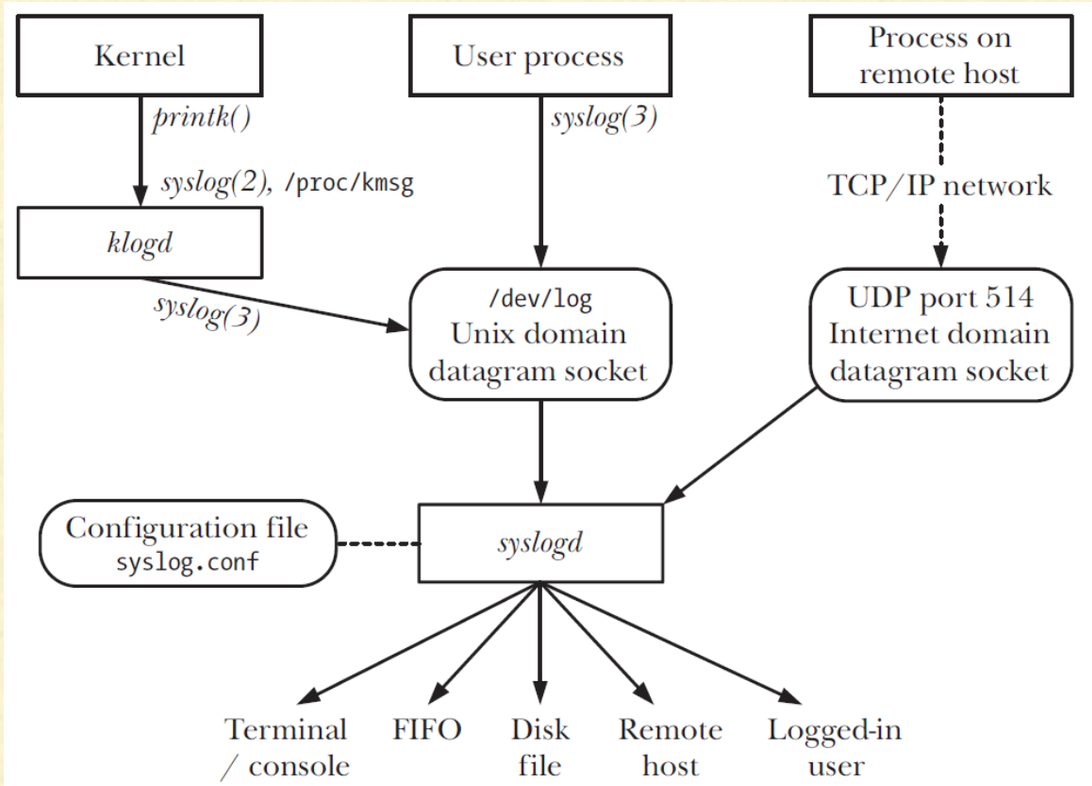


Figure 37-1: Overview of system logging

SERVER DAEMON LOGS

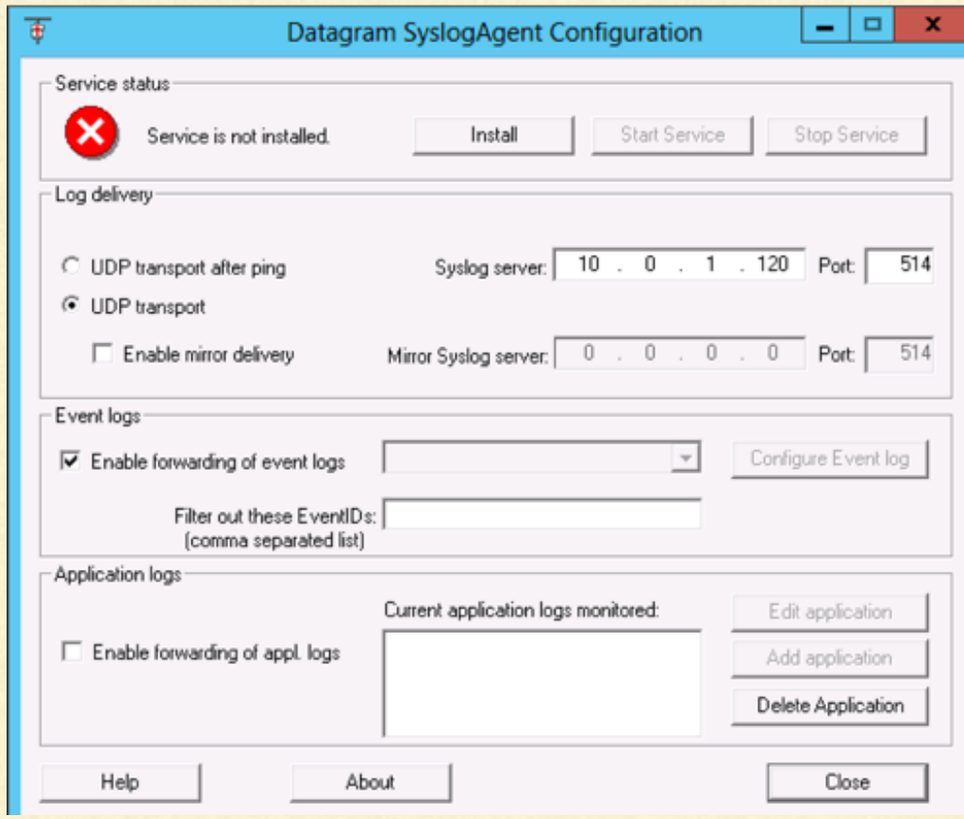
- Various functions in the system are provided by daemons
- Many of them log status changes, warnings, and errors
- Services running on the server typically directly write log files, but also send messages to the standard syslog service, either through /dev/log, or using UDP or TCP to connect to port 514

SERVER APPLICATION LOGS

- Services on a server are provided by application packages
- Applications produce log messages for status changes, warnings, and errors
- Applications are configured individually with respect to logging
- Applications often support multiple message formats
- Remote syslogging can be accomplished in the system syslog configuration file even if the service software doesn't provide that capability

WINDOWS EVENT LOGS

- Windows event logs are not directly compatible with UNIX-style logging
- Various solutions are available to send Windows event logs to a syslog server, they involve a service watching the Windows log files and sending new entries to syslog
- rsyslog.com has a windows agent, datagram syslogagent has been popular, correlog.com has a free agent, newer monitoring systems use tools like beats to feed elasticsearch logging systems



MESSAGE CONTENTS

- Messages are plain text, by default just sent as the data in a udp packet, or via tcp connection

- Messages contain properties, some interesting properties include:

message content - 1024 printable ascii characters by convention
tag - name of program that generated the message
source - hostname or IP address or other identifier for host that generated the message
priority - number composed of a facility and severity to categorize the message

- Message properties can be used to sort and format messages, the PRI (priority - facility.level) is used for typical rsyslog.conf rules

Syslog Message Sources

Syslog identifies various log sources as "Facilities," which makes it easy for you to assign monitoring to different groups.

FACILITY	ORIGIN OF THE MESSAGE
Kern	System kernel
User	User processes or applications
Auth	Login system or authentication service
Security	Security processes
Syslog	Syslog service
Cron	Schedule service
<any service>	FTP, mail, news can generate messages

Syslog Message Severity

Syslog provides a range of severity messages, from general information to emergency alerts. You can configure these messages to reflect what is most critical to your environment.

SEVERITY LEVEL	DESCRIPTION
Info	General informational messages
Notice	Event that might require some additional analysis or attention
Debug	Messages used for troubleshooting and to provide additional details on an issue
Warning	Warning on potential issue but usually something that is minor
Err	Error condition that occurs when a service malfunctions
Crit	Critical condition that could be a system failure
Alert	Requires immediate attention
Emerg	Highest priority and indicates that there is the highest level possible event

https://searchsecurity.techtarget.com/magazineContent/How-to-setup-and-configure-syslog-to-view-and-filter-data

RELIABLE MESSAGING FOR RSYSLOG

- The original log transport, UDP, was chosen for simplicity and speed
- TCP is an alternative which guarantees delivery, but only if the remote host is reachable, messages can still be lost in the case of down servers and connection losses
- Reliable Event Logging Protocol (RELP) is a protocol which can solve this problem, but it has to be enabled on the server (not the default) and the client must be configured to use it (not the default)
- A server can listen for RELP connections without interfering with listening for traditional connections, RELP uses port 2514 by default
- See http://www.rsyslog.com/doc/relp.html for more information about RELP

RSYSLOG

- rsyslog is one of the 3 main packages providing syslog-compatible logging services
- rsyslog is the default for redhat and debian-derived distros and many others
- /etc/rsyslog.conf is the config file, entries describe a pattern match and rule to use

```
facility.level      /var/log/somefile.log
mail.*              /var/log/mail.log
*.=crit,*.=emerg    @loghost
*.*                 @@loghost
kern.*              ~
$IncludeConfig      /etc/rsyslog.d/*.conf
```

- rsyslog addons are typically configured in their own files under /etc/rsyslog.d

RSYSLOG CONFIGURATION

- Modules can be dynamically loaded by the rsyslog daemon to add functionality to the server

```
$Modload ommysql
*.*       :ommysql:DBHost,DBName,rsysloguser name,rsyslogpassword
```

- Remote message reception is enabled in the rsyslog.conf file by turning on the imudp module and starting the udpserver, or the imtcp module and starting tcpserver - don't forget to allow the traffic through your firewall

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

- You can restrict access to clients specified with the AllowedSender directive, to help avoid log overflow attacks, TCP and RELP can help with this as well but have lower throughput

```
$AllowedSender UDP, 172.16.209.3
$AllowedSender UDP, 192.168.2.0/24
$AllowedSender UDP, *.mycompany.com
```

RSYSLOG CONFIGURATION

- To enable RELP, the server must load the module and start the module listening on a port

```
$ModLoad imrelp
$InputRELPServerRun 2514
```

- To use RELP, a client must use the module for RELP to send the messages of interest to the server's RELP port

```
*.* :omrelp:192.168.0.1:2514
```

LOGGING TO A DATABASE

- Database storage of log data allows for much more flexibility in information retrieval and analysis, as well as access control
- This comes at a significant performance cost when writing records, but often provides significant gains when retrieving records
- The rsyslog-mysql package provides a canned configuration which connects rsyslog to mysql, be sure to assess the default config to ensure you send messages you want (the default is to send all messages to the database)
- There are similar packages supporting other database packages (e.g. rsyslog-pgsql)
- The creation of the database and its tables is separate from the rsyslog-mysql package install

RSYSLOG-MYSQL INSTALLATION

- These instructions are part of the lab
- Run mysql with the createDB.sql SQL to create a new database ready for rsyslog, this sets up the Adiscon Monitorware schema for rsyslog
  i.e. run the mysql command with the SQL from https://github.com/rsyslog/rsyslog/blob/master/plugins/ommysql/createDB.sql
- Install the rsyslog-mysql package, this will add a mysql.conf file to your /etc/rsyslog.d directory
- Restart rsyslog

  - Note that the default installation still stores logs in plain text files as well as the database, so consider whether that is what you want
  - Also consider only sending some messages to the database

https://www.thissmarthouse.net/consolidating-iot-logs-into-mysql-using-rsyslog/