# Logging

# Log Files – why bother

- We care, we really care, about  “What happened and when it happened”
- Log files are an administrator’s best friend when debugging
- Application developers will often ask for log files when trouble shooting

# Logging Options

- Application specific
- Common logging facility (syslog/rsyslog, Window$ Event Log)

# General Truths about logs

- Developers spend considerable time and energy writing messages to the log file – to help **you**
- Most logging mechanisms share these traits
  - Time stamp
  - Severity level (debug, info, warn, error)
  - Ability to limit messages to one level and ‘above’
  - Description text

# syslog and rsyslog

- Centralized logging system
- Used by most *nix system daemons
- Can be used by most other daemons
- Supports remote logging (very important for security)
- Log ‘level’ and ‘destination(s)’ can be controlled ‘centrally’

# Why is remote logging important for security?

Consider these two facts:

1) Security events, like failed login attempts, are logged to a file only root can access
2) Intruders know fact 1.

# [r]syslog Message Structure

Facility.Severity Message

**Facility**: “Who” sent the message
**Severity**: How “important” is the message
**Message**: What the developer wanted to say to you.

# [r]syslog Facilities *(from syslog.h)*

```
CODE facilitynames[] =
  {
    { "auth", LOG_AUTH },
    { "authpriv", LOG_AUTHPRIV },
    { "cron", LOG_CRON },
    { "daemon", LOG_DAEMON },
    { "ftp", LOG_FTP },
    { "kern", LOG_KERN },
    { "lpr", LOG_LPR },
    { "mail", LOG_MAIL },
    { "mark", INTERNAL_MARK },          /* INTERNAL */
    { "news", LOG_NEWS },
    { "security", LOG_AUTH },           /* DEPRECATED */
    { "syslog", LOG_SYSLOG },
    { "user", LOG_USER },
    { "uucp", LOG_UUCP },
    { "local0", LOG_LOCAL0 },
    { "local1", LOG_LOCAL1 },
    { "local2", LOG_LOCAL2 },
    { "local3", LOG_LOCAL3 },
    { "local4", LOG_LOCAL4 },
    { "local5", LOG_LOCAL5 },
    { "local6", LOG_LOCAL6 },
    { "local7", LOG_LOCAL7 },
    { NULL, -1 }
  };
```

# [r]syslog Priorities

| KEYWORD | DESCRIPTION |
|---|---|
| emerg | System is unusable |
| alert | Should be corrected immediately |
| crit | Critical conditions |
| err | Error conditions |
| warning | May indicate that an error will occur if action is not taken. |
| notice | Events that are unusual, but not error conditions. |
| info | Normal operational messages that require no action. |
| debug | Information useful to developers for debugging the application. |

# Parting Thoughts

- Save your logs
- When you write admin scripts – log
- Have a look at Apache httpd logs – they’re great!
- Investigate `logrotate`
- Listen to the master:

  https://youtu.be/fewUSu_QZAY