CUPS Plenary

Michael Sweet, Apple Inc.
August 12, 2013
Introduction

- CUPS is the standards-based, open source printing system developed by Apple Inc. for OS X and other UNIX®-like operating systems.

- CUPS 1.7.x is the current stable branch
  - 1.7.5 released July 30, 2014
  - *Maybe* one more 1.7.x release after that

- CUPS 2.0.x is the current development branch
  - Beta testing started July 30, 2014
  - Probable 2.0.0 release this October
CUPS.org Web Site

- New Apple server finally went live December 17, 2013
  - Bug database restored
    - Working to add “cc” functionality back in (STR #4458)
  - Git mirror of internal Subversion repository
    - Provides read-only access to all development branches and tags
  - Some functionality will not be brought back:
    - PPD database (legal, security, lack of use)
    - Windows drivers (legal)
    - NNTP access to mailing lists (security)
CUPS 1.7 Release History

- CUPS 1.7.0 released October 24, 2013
- CUPS 1.7.1 released January 8, 2014
  - Security: potential lppasswd issue
- CUPS 1.7.2 released April 10, 2014
  - Security: web interface redirection issue
- CUPS 1.7.3 released May 27, 2014
  - General bug fix release
- CUPS 1.7.4 released July 14, 2014
  - Security: Potential symlink information disclosure
CUPS 1.7 Release History

- CUPS 1.7.5 released July 30, 2014
  - Security: More symlink information disclosure

- CUPS 1.7.6?
  - Maybe, depending on need
  - One functional regression for remote log file viewing (STR #4461) that will definitely be fixed for 2.0.0
CUPS 2.0

- Mainly a performance/security release
- See the announcement on CUPS.org or the CHANGES.txt file for a detailed list of changes
CUPS 2.0 - Security Improvements

- TLS/SSL (server/printer) certificate validation to prevent obvious man-in-the-middle attacks
  - OS X and Windows provide verification of CA-signed certs/chains for us
  - GNU TLS code will need some more platform-specific love to verify well-known CAs in the chain
    - ... but most hosts uses self-signed certs

- TLS/SSL policy settings in `client.conf`
  - Default policy like ssh - trust on first use, error if something changes
CUPS 2.0 - User-Visible Changes

- Man pages have been completely updated
  - Now conform to Linux man page guidelines
  - Please file bugs if there are still issues
  - Config file documentation now uses man as the primary source with HTML versions online
    - Files not meant for generate editing by users are now documented as such

- Default **AccessLogLevel** and **PageLogFormat** now disable generation of the access_log and page_log files
CUPS 2.0 - Scheduler

- Completed jobs are now reported in the correct newest-to-oldest order
- Support for the IPP “first-index” operation attribute (allows for retrieving smaller segments of job history data)
- Better caching of job history for completed jobs to eliminate re-loading of all the “c” files in most cases
- Support for run-as-root backends with group read+execute permissions
CUPS 2.0 - Scheduler

- (OS X only) New strict sandbox profile for filters and backends
- **Systemd** support for launch-on-demand and idle exit, unless sharing printers
  - OS X still uses **launchd** to provide the same functionality
- **Fallback rasterization**
  - The scheduler and IPP backend now support resubmission of PDF/PS jobs as (PWG) raster data if the PDF/PS is not printable
CUPS 2.0 - ippserver

- IPP Everywhere conformance (for 2.0.0)
- SSL/TLS support (via new CUPS server APIs)
CUPS 2.0 - ipptool

- Better collection attribute support
  - **ATTR** syntax ("{name=value},{name=value}")
  - **EXPECT** syntax ("EXPECT attribute/member")
- **PAUSE** directive
  - Wait for user to press key to continue test
- Simultaneous test and XML (plist) output
- TIFF file format support
CUPS 2.0 - Developer Features

- **Server-side TLS/SSL functions**
  - `cupsMakeServerCredentials`: creates a self-signed certificate/private key pair
  - `cupsSetServerCredentials`: specifies the location of certificates and private keys, the default server name, and whether to automatically create a self-signed cert/key pair
  - Existing `httpEncryption` function is used to negotiate TLS/SSL on the server side as well
CUPS 2.0 - Developer Features

- Client-side TLS/SSL functions
  - `httpCompareCredentials`: Compare two credentials (certificate chains)
  - `httpCredentialsAreValidForName`: Determine whether the credentials are valid for a given hostname
  - `httpCredentialsGetExpiration`: Gets the expiration date of the credentials
  - `httpCredentialsGetTrust`: Gets the level of trust for the credentials based on the current policy configuration
CUPS 2.0 - Developer Features

- **httpCredentialsString**: Returns a human-readable string representing the credentials (common name, expiration, hash)
- **httpLoadCredentials**: Loads credentials for a given name
- **httpSaveCredentials**: Saves credentials for a given name (typically used to “remember” self-signed certificates)
CUPS 2.0 - Developer Features

- **Destination APIs**
  - `cupsGetDestWithURI`: Create a destination from a printer URI
  - `cupsLocalizeDestMedia`: Generates a localized media size name/description ("8x10 Glossy Photo", etc.)

- **Server APIs**
  - Many accessor (`httpGetXxx`, `httpIsXxx`) and support functions needed to write a full-featured HTTP/IPP server like `cupsd`
CUPS 2.0 - Dropped Stuff

- OpenSSL support
- AIX, HP-UX, and OSF/1 (aka Digital UNIX) support
- Really-old 1.0 APIs (cupsGetClasses, cupsGetDefault, cupsGetPrinters)
  - Still have stubs for binary compatibility
  - These have been deprecated since CUPS 1.1
- HTTP Digest support
  - No more lppasswd program
CUPS Developer “Cheats”

- `#define _CUPS_NO_DEPRECATED 1`
  - Turns off compatibility defines/typedefs for enums
  - Marks deprecated functions and types as unavailable so you get a compile error instead of a warning

- `#define _IPP_PRIVATE_STRUCTURES 1`
  - Makes ipp_t structure public
  - Will be removed after CUPS 2.0.x

- `#define _PPD_DEPRECATED ""`
  - Turns off PPD warnings
CUPS Future

- Continue march to ubiquitous printing via IPP Everywhere
  - Focus on PDF, JPEG, and PWG Raster
  - Better status/state reporting

- Investigate alternate (per-user) spooler implementations that just do IPP Everywhere - no printer drivers
  - CUPS server APIs and ippserver sample code make this feasible
  - Run in parallel with traditional cupsd for driver-based queues
CUPS Future

- Better power support on Linux and others
  - Like we already do on OS X
  - Idle sleep, forced sleep

- Better network awareness on Linux and others
  - Like we already do on OS X
  - Detect network changes, default-printer-per-network
CUPS Future

- Additional discovery/directory service support
  - Bring back LDAP support, this time using the standard schema
  - DNS-SD/mDNS enhancements being discussed in the IETF

- Potential Cloud/infrastructure and release printing support
  - Based on IPP Shared Infrastructure Extensions (IPPSIX) in the PWG
  - Solves some of the harder network accessibility issues
CUPS Future

- Challenges

  - Can we make these changes transparent to applications, i.e., will we be able to stay binary compatible?

  - Can we provide a consistent user experience on all platforms, i.e., do we have all of the tools/libraries we need for networking, USB, graphics, etc?

  - Can we do this quickly enough?
Resources

- CUPS Web Site
  - http://www.cups.org/

- CUPS Roadmap
Q&A