Towards a Least Privilege Desktop

Colin Walters <walters@verbum.org>
Red Hat, Inc.

Outline

- Goals and Motivation
- Traditional Unix server security and SELinux
- Information flow in Unix desktops (GNOME)
- D-BUS
- Proof of concept least-privilege D-BUS services (Imsep)
- Contrast with other techniques (Systrace)
- Conclusions and discussion

Goals and Motivation

- Many threats against a modern desktop
  - Worms, spyware, adware, phishing
  - Can propagate/succeed via different means
    - Flaws requiring no user interaction
    - Deception
    - Combination of above
- Focus on threats which require little to no user interaction and lead to arbitrary code execution

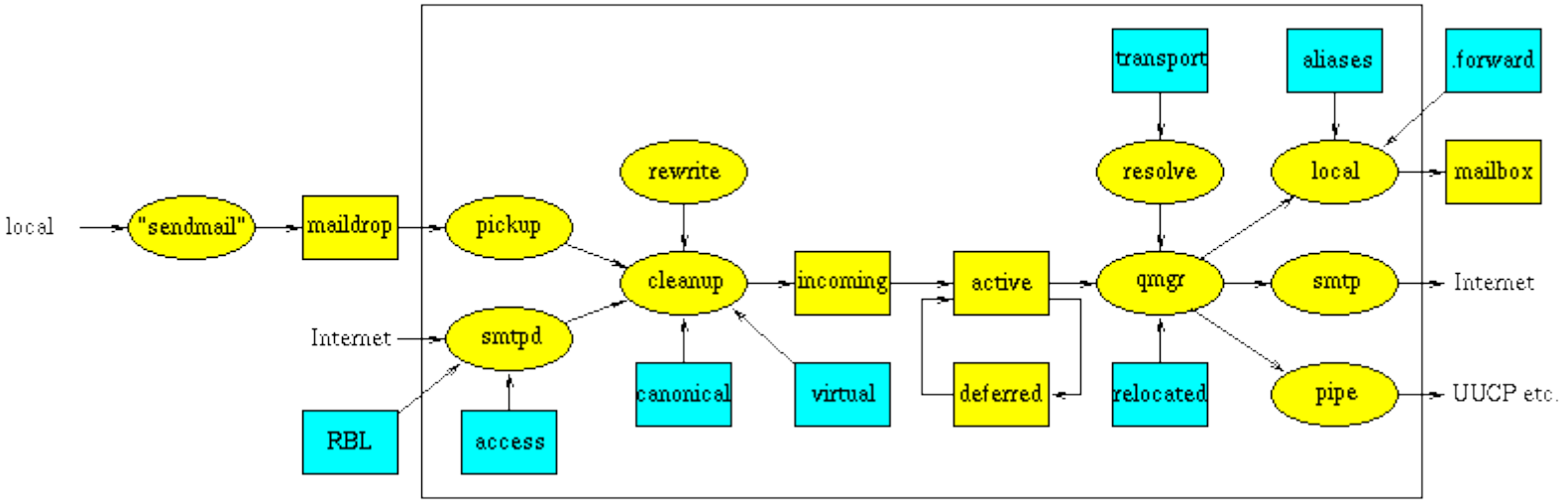
Goals and Motivation, contd.

- Concrete examples of flaws requiring little to no user interaction and could result in execution of arbitrary code:
  - PDF parser flaws (CAN-2004-888)
  - gdk-pixbuf flaws (CAN-2004-0782, CAN-2004-0783)
  - Mozilla parser flaws (http://lwn.net/Articles/107110/)
  - RealPlayer (SUSE-SA:2005-004)

Traditional Unix server security

- Unix server software is written to use discretionary uid-based mechanisms to isolate components (and other servers)
  - Typical pattern is a “master” server running as root, and “unprivileged” children doing other processing
- Goals
  - Protect “root”
  - Preserve integrity/confidentiality of data controlled by “unprivileged” uid
- Separate untrusted process design maps well to SELinux domains

Postfix Architecture



Linux Desktop Security

- Typically runs as “unprivileged” uid
- Protecting “root” is not the highest priority, as all of the interesting data on a desktop is owned by the user anyways
  - And “unprivileged” users can still send spam, worms, etc
- Many different applications communicating via a number of shared channels

Linux Desktop Information Flow

- Channels
  - X server (http://www.nsa.gov/selinux/papers/x11-abs.cfm)
    - Including extensions such as root window properties
  - ~/.recently-used
  - ORBit (CORBA)
    - GConf
    - gnome-keyring
  - D-BUS
- Historically not very integrated, but growing more so

D-BUS introduction

- IPC framework designed for two primary use cases
  - System <-> User session interaction (HAL, NetworkManager)
  - Intra-session communication (GConf, notification area, gnome-keyring)
    - Goal is to be acceptable replacement for ORBit in GNOME and DCOP in KDE
- Designed to expose high-level objects (methods, signals, properties)
- Low-overhead, low-latency, easy-to-use
- Designed with security in mind

D-BUS security

- Client library and bus implementation extensively validate messages
- Enhanced to act as a “userspace object manager”
  - Performs “send_msg” check based on domain of sending and receiving processes:
    - allow cupsd_t system_dbusd_t:dbus send_msg;
  - Performs “acquire_svc” check based on requestor domain and service name (e.g. org.freedesktop.NetworkManager)
  - Pending work will allow more control based on the “interface” of a message, and service activation

Imsep

- Image format loaders (png, tiff, etc) are complex code
  - Linked into and run from essentially every desktop program that reads images
- Composed of two programs (imsep-master and imsep-loader)
  - master gets images (png, tiff) over D-BUS and sends pixbuf back
  - imsep-loader is strictly confined via SELinux policy
- GTK+ plugin created which transparently sends image requests to Imsep

Musep/Mvsep

- Musep
  - Similar argument as for image loading – handling compressed music is complex
  - Factor music playback core into separate D-BUS service
  - Difficulty – need to allow access to local files and network (ideally these are mutually exclusive)
  - Work is pending D-BUS completion
- Mvsep
  - Similar to Musep, except for totem/HelixPlayer. Requires Security-Enhanced X work to allow video playback into a particular X window, but not permission to e.g. create new windows

Alternate approaches

Alternate Approaches, contd.

- Without trusted X, can be bypassed
- Binary pathname is not trustworthy
- Normal users are generally not going to understand

Conclusions

- Can not counter all threats using D-BUS and SELinux
  - But goes a long way towards mitigating a large class of potential flaws
- Much work remaining on D-BUS
- Obvious omission: Firefox