The Bleeding Edge

or
How To Run Ubuntu Development Branches
And Not Get Cut
http://outflux.net/ul07/bleeding-edge.odp

Kees Cook kees@ubuntu.com
(and ghost-writer Colin Watson)

Ubuntu Live 2007

Why *not* to run the devel branch

- "I think it will make my computer faster"
  - probably not -- likely less reliable
  - upgrading remotely is scary
- "I just need one or two new applications"
  - https://help.ubuntu.com/community/UbuntuBackports
  - or rebuild the package yourself
    - `sudo apt-get install build-essential devscripts fakeroot`
    - `sudo apt-get build-dep` **`PACKAGE`**
    - `dget` **`URL/to/PACKAGE.dsc`**
    - `dpkg-source -x` **`PACKAGE`**`.dsc`
    - `(cd` **`PACKAGE-*`** `&& debuild -uc -us)`
    - `sudo dpkg -i` **`PACKAGE_`**`*.deb`

Why *to* run the devel branch

- stable versions don't work on your hardware
- you want to help out with testing to ensure that the next version of Ubuntu is high-quality
- you want to help develop Ubuntu
- you are employed to develop Ubuntu :-)
- <zomg>NOT FOR THE FAINT OF HEART</zomg>
  - but, if you're here, you likely have the appropriate level of intestinal fortitude

Release process

- https://wiki.ubuntu.com/GutsyReleaseSchedule
- lots of churn in the first couple of months as we resynchronise with Debian
- regular installable milestone releases
- feature freeze two months before release
- focus on bug fixing for last two months, becoming increasingly more conservative
- last-minute bugs probably cannot be fixed in time for release unless they are world-shatteringly bad

Helping with testing

- upgrade regularly (at least weekly if not daily) and report bugs when things go wrong
  - `sudo apt-get update`
  - `sudo apt-get -dy dist-upgrade`
  - `sudo apt-get dist-upgrade`
- test the live CD
- try out fresh installs of milestone releases
- always read the release notes
- https://wiki.ubuntu.com/Testing
- https://wiki.ubuntu.com/Bugs
- https://wiki.ubuntu.com/BugSquad

Live CD testing

- can safely boot the live CD without touching your existing system
  - beware mounting filesystems from hibernated OSes
- relatively slow to run day-to-day, but an important and safe test
  - can install packages on the running live CD (subject to free memory)
  - much faster in a virtual machine (though requires even more memory)
- Live CD Persistence (when it works)
  - https://help.ubuntu.com/community/LiveCDPersistence

Installation testing options

- may choose desktop, alternate install CD, netboot, USB stick ...
  - desktop CD more user-friendly, though can be harder to fix if it fails
  - alternate install CD text-only, but very flexible and you can retry individual steps
  - https://help.ubuntu.com/7.04/installation-guide/
- install alongside existing system
  - requires unpartitioned space, or resizable volumes
  - installer bugs may eat data (make backups!)
  - important test case (especially alongside Windows)

Installation testing options (cont.)

- use a spare machine
  - revision control, rsync, unison, scp, NFS, Samba for keeping your files handy
  - safe and fast; uses more desk space and electricity
- use a virtual machine
  - several free/proprietary VM options (qemu, kvm, virtualbox, VMware, etc.)
  - frequent choice for heavy testers on the development team
  - safe and saves desk space and electricity; some slowdown; requires spare disk space

Upgrading

- very important to test
- lots of combinations with lots of possible failure modes
- some auto-testing done by development team
- benefits greatly from community testing
- filing bugs on problems you encounter helps us improve the upgrade process

Upgrading (cont.)

- automatic (may not always work)
  - `update-manager -c -d`
- manual
  - edit /etc/apt/sources.list, replace all codename references (e.g. “feisty” to “gutsy”)
    - `sudo perl -pi -e 's/feisty/gutsy/' /etc/apt/sources.list`
  - `sudo apt-get update`
  - `sudo apt-get dist-upgrade`
  - hold on to your hats (and see the rest of this talk!)

What can go wrong

- mirror problems
- uninstallable packages
- removals due to difficult upgrade logic
- administrative error
- crashes during upgrades
- hardware-specific breakage
- unpredictable bugs

Mirror problems

- checksum failures
  - development branch changes hourly; sometimes mirrors get skewed
- unauthenticated package warnings
  - don't install unauthenticated packages; this warning is for your protection (it could be a real attack)
- next mirror push may resolve the problems, so wait an hour and try again
- try temporarily using archive.ubuntu.com
  - remember to switch back! it's slow and expensive

Uninstallable packages

- incorrect dependencies
- file-level conflicts ("trying to overwrite '/bar/baz', which is also in package “foo")
- broken maintainer scripts
- note that apt will fail if the system is too inconsistent, and you may have to fall back to using dpkg directly

Unpack failures

- typically haven't done much to your system
- if due to file-level conflict, may be fixable using:
    - `sudo dpkg --force-overwrite -i /var/cache/apt/archives/foo.deb`
    - almost always indicates a missing Replaces field
- bugs in pre-installation scripts generally unfixable without repacking .deb
- sometimes pre-removal script of old package may be broken; may require editing by hand
    - see /var/lib/dpkg/info/**foo**.prerm

Configure failures

- package's files are unpacked, but post-installation script failed
- sometimes a retry is enough
  - `sudo dpkg --configure -a`
- sometimes post-installation script is broken; may require editing by hand
  - see /var/lib/dpkg/info/**foo**.postinst
- update-alternatives and dpkg-divert are often used and sometimes used wrongly; see the documentation if need be

Package system debugging

- dpkg itself only gives you an exit code, which isn't very useful
- read back through the output for the earliest error that caused dpkg to fail, and fix that
- when reporting a bug, quote the full output, not just the end
- most maintainer scripts are shell; to get a full execution trace, edit them in /var/lib/dpkg/info/, put 'set -x' on the second line, and retry

Package system debug (cont.)

- if debconf goes wrong (often indicated by exit codes that are multiples of 10), get a debconf trace:
  - `export DEBCONF_DEBUG=developer`
- exit codes 10 and 20 are usually bad arguments to db_*something*, while 30 is often just a missing “|| true” after db_input or db_go
- add verbose flags to commands

Package system debug (cont.)

- strace (sledgehammer tracing tool, output can be hard to read)
  - `sudo strace -f` **`dpkg-command...`** `2>/tmp/foo.out`
- debug apt-get dependency problems; output can be hard to read
  - `sudo apt-get -o Debug::pkgProblemResolver=true ...`
- occasionally dpkg itself goes wrong; see help
  - `dpkg —debug=help`
- dpkg debug options do not help with maintainer script problems!

Incorrect removals

- always check removal list carefully when performing a dist-upgrade
- update-manager has a list of known-good removals, but this may not be up to date
- "Following essential packages will be removed"
  - never say yes unless you are well beyond needing this talk
- packages that have been removed from the archive are usually fair game to remove
- check package states with `apt-cache policy` **`PACKAGE`**

Administrative error

- forcibly installed package that causes problems
  - `sudo dpkg --remove` **`foo`**`; sudo apt-get -f install`
- packaging system files were modified by hand
  - reinstall packages and next time use dpkg-divert or dpkg-statoverride as necessary
- packages from third-party archives
  - we don't deliberately break them, but sometimes it's unavoidable or unnoticed
  - consider removing them to make the upgrade finish, and reinstall later
  - report bugs to third party in question

Crashes during upgrades

- try to resist the urge to pull the power cord during upgrades, but if you must...
  - `sudo dpkg --configure -a`
  - `sudo apt-get -f install`
- "Package is in a very bad inconsistent state"
  - `dpkg --unpack /var/cache/apt/archives/foo.deb`
  - `dpkg --configure -a`
- /var/lib/dpkg/status is critically important
  - copies in /var/lib/dpkg/status-old and /var/backups/
- /var/lib/dpkg/available corrupt
  - `sudo dselect update`

Crashes during upgrades (cont.)

- files filled with zero bytes
  - XFS does this if you're unlucky; get a UPS :-)
- corrupt files belonging to packages
  - `dpkg -S` ***`/path/to/file`***
  - `sudo apt-get install --reinstall` **`PACKAGE`**
- /var/cache/apt files can be removed if corrupt
- debconf database files can sort of be removed if corrupt, but you may have to reinstall packages and re-answer questions afterwards
  - `sudo dpkg-reconfigure -plow` **`PACKAGE`**

Hardware-specific breakage

- most important area to report bugs
  - Not everyone has your hardware, so it can be a challenge to test all combinations.
- common failures
  - Kernel breakage
  - X.org breakage
  - Network breakage

Kernel breakage

- new kernel (or initramfs) breaks on boot
  - always keep at least one old known-good kernel
  - kernel ABI changes frequently in development branches so you will generally be able to boot an old one
    - linux-image-2.6.22-8-generic 2.6.22-8.18
- kernel seems to work, but oopses later
  - be cautious (is the filesystem oopsing?)
  - report a bug and consider reverting to an older kernel
    - include full oops log in /var/log/kern.log and/or dmesg

X.org breakage

- log in at console (Ctrl-Alt-F1)
- can generally fall back to vesa or vga driver
  - edit /etc/X11/xorg.conf, e.g.:
    - Section “Device”
    -     Driver “vesa”
    - EndSection
- restart X.org
  - `sudo /etc/init.d/gdm restart`
- should be less of a problem in future (bullet-proof-x), with automatic vesa/vga fallback

Network breakage

- network-manager takes down the network interface you were using
  - interaction with /etc/network/interfaces is hard
  - may require manual prodding (ifconfig, ifup/ifdown)
  - if want you, you can stop network-manager
    - `sudo /etc/dbus-1/event.d/26NetworkManagerDispatcher stop`
    - `sudo /etc/dbus-1/event.d/25NetworkManager stop`

Unpredictable bugs

- almost anything can happen in theory
- keep a login session open in case authentication breaks
  - check /var/log/auth.log for authentication problems
- may need to reboot in recovery mode to fix
  - in a pinch, (e)dit the grub menu item and use this for the kernel command-line arguments instead of “ro”:
    - rw init=/bin/sh
- report any library crashes or compilation breakage

Remote upgrades

- <broken record>not a good idea with development branches</broken record>
- keep an ssh session open
  - make sure you can still log in before closing it
  - may still lose the battle if network interfaces go away
- if you need to reboot remotely, invest in a remote console server :-)

Where to go for help

- report bugs: https://bugs.launchpad.net/ubuntu
- mailing lists:
  - ubuntu-users@lists.ubuntu.com
  - ubuntu-devel-discuss@lists.ubuntu.com
- web forums: http://ubuntuforums.org/
- IRC: #ubuntu or #ubuntu+1 on irc.freenode.net
  - please use #ubuntu-devel only if helping with development
  - please use #ubuntu-bugs if helping report/triage bugs

Questions?

Kees Cook
kees@ubuntu.com