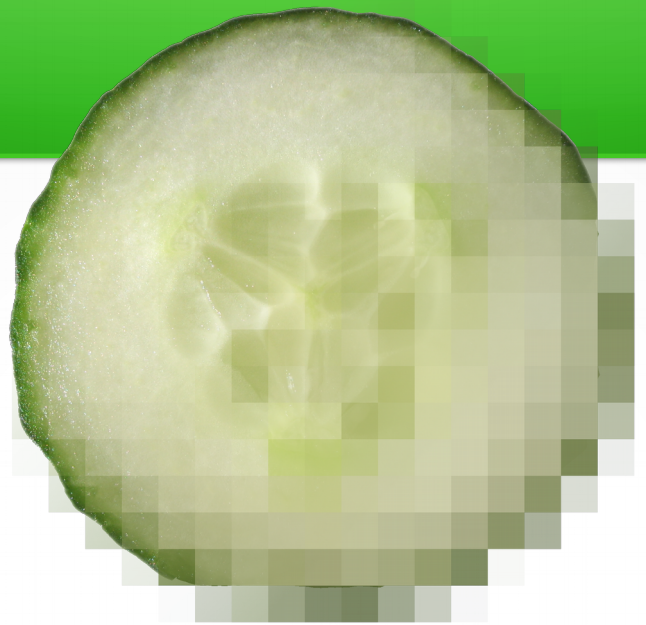


An Overview of How I Created the Cucumber Linux Distribution

Scott Court

February 16, 2018

Who am I?

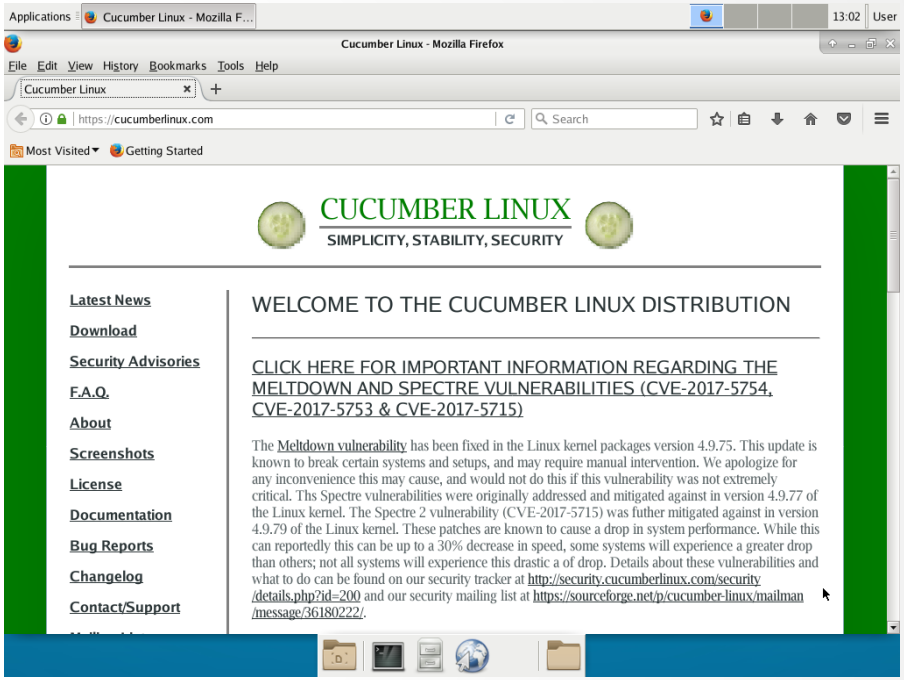
```
scott@Scott-Desktop:~$ ssh cucumberlinux.com

scott@cucumberlinux.com:~$ finger scott
Login: scott                            Name: Scott Court
Directory: /home/scott                  Shell: /bin/bash
On since Wed May 11 2016 10:00 (EST) on tty1

Major: CSEC
Year: Third year student
Preferred Scripting Language: Bash
Preferred Programming Language: C
Hobbies:
        Cucumber Linux
        No need for other hobbies, this takes up all my free time
```

What is Cucumber Linux?

- A Linux distribution built entirely from source, from scratch
- An independent distribution: it’s not based on any other Linux distribution.
- Inspired by: a mixture of Slackware and *Linux from Scratch*.
- Architectures: i686 and x86_64
- Package manager: pkgtools (borrowed from Slackware; uses tar archives as packages like Pacman).
- License: MIT
- Mission: provide a Linux distribution that follows the Unix Philosophy with a focus on simplicity, stability and security, and is usable as an every day, general purpose operating system.

What can it Do?

- It's functional as both a server and desktop operating system.
- Server Operating System:
  - Contains a fully functional LAMP stack as well as SSH, mail, DNS, FTP and Rsync servers.
  - Runs the web server and mail server for cucumberlinux.com.
  - Runs the primary mirror for Cucumber Linux (http and rsync).
- Desktop Operating System:
  - Comes with the XFCE desktop environment, a web browser, a mail client and an office suite.
  - It's running this presentation right now.

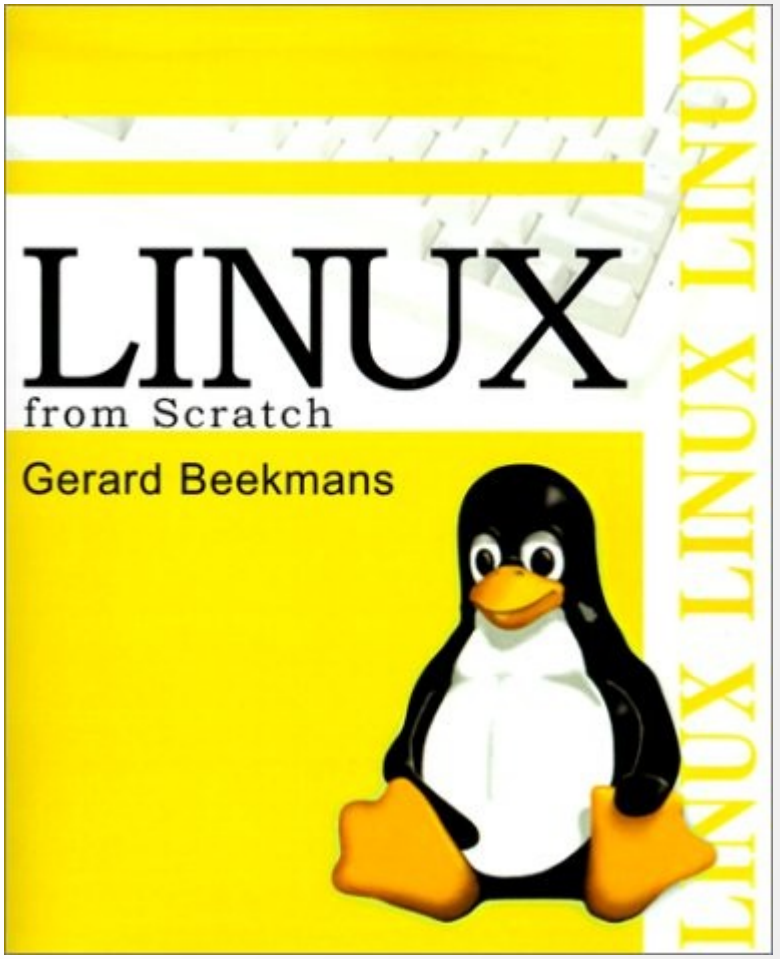


May 2016 – How it Began

- I had just completed my freshman year of college.
- I was looking for a project to work on that Summer.
- I had grown unhappy with the direction most every other Linux distribution was going in (namely the exodus from Sysvinit to Systemd).
- I had always had an interest in operating system development.
- So I set out to make my own Linux distribution, without Systemd.

May 2016 – July 2016

- May, June and July of 2016 were spent doing research and trying out different techniques for building a Linux system from scratch.
- This was marked by several failed attempts to build even a basic system.
- Eventually, I settled on using the book *Linux from Scratch 7.9* as a guide.



August 2016 – Real Work Began

- Work began on building the base system that would become Cucumber Linux 1.0.
- Before I could begin work on building the actual system, it was necessary to build a new toolchain (compiler, binutils and c library) that could be used to build the system.
  - This involved compiling GCC and binutils 6 separate times to resolve a circular dependency with glibc.
  - 24 other packages also had to be built to support the toolchain.
  - This took a long time.

August 2016 – Real Work Began

- Once the toolchain was complete, the real fun began: building all of the packages.
- This entailed:
  - Downloading the package's source archive.
  - Creating a Bash script to extract the source archive, build the package from source, and package it up into a Cucumber Linux compatible .txz (.tar.xz) binary package.
    - These scripts were dubbed "buildscripts."
    - Usually building the package from source involved something like running ./configure, make and make install.
    - Unfortunately, every package is structured slightly differently, so a unique buildscript had to be written for each package.
  - Running the buildscript twice: once for i686 and once for x86_64.
    - This created the binary packages that could actually be installed.
    - This had to be done as root (yikes!).
  - Uploading the buildscript, source tarball and binary packages to my rsync server.
- This process was repeated for each package.

The Anatomy of a Buildscript

- Breakdown of how this works:
  - Lines 1 – 24 (not shown): the license.
  - Lines 25 – 55: set some variables that will be used later in the build process.
  - Lines 56 – 63: create a temporary directory to build the package in.
  - Lines 65 – 66: extract the package source tarball.
  - Lines 69 – 76: build the package and install it to a staging directory (a “fake root”).
  - Lines 79 – 90: strip the resulting binaries (to reduce the package size) and copy over a couple of files:
    - doinst.sh: a shell script that runs post installation to perform additional commands needed to set up the package correctly (optional, not every package has one).
    - slack-desc: a text file containing a brief description of the package.
  - Lines 93 – 96: create the package tarball by tarring up the staging directory.
- Slackware and Linux from Scratch both provide similar scripts for building packages, which served as a starting point for most of the buildscripts.
- Every buildscript was open sourced under the terms of the MIT license.
- All of the buildscripts are available at http://mirror.cucumberlinux.com/cucumber/

```
less-481.tar.gz  less.buildscript  slack-desc
```

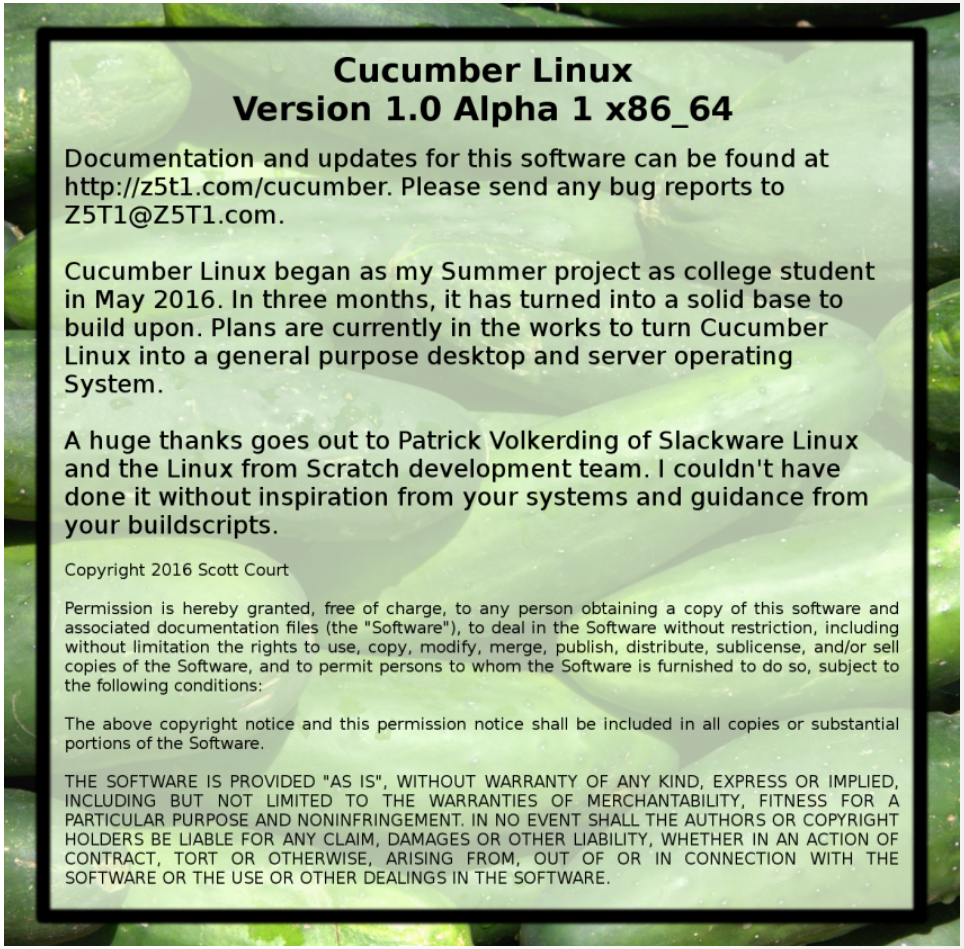
The directory listing for the less buildscript directory.

```
25 NAME=less
26 VERSION=${VERSION:-$(echo $NAME-*.tar.gz | rev | cut -f 3- -d . | cut -f 1 -d - | rev)}
27 BUILD=${BUILD:-1}
28
29 # Determine the architecture we are building on if it is not explicitly specified.
30 if [ -z "$CUCARCH" ]; then
31         case "$(uname -m)" in
32         # For the x86 series, force i686
33         i?86)
34                 export CUCARCH="i686"
35                 LIBDIRSUFFIX=""
36                 ;;
37         # For x86_64, set the libdir suffix to 64 (so we use /lib64 instead of /lib)
38         x86_64)
39                 export CUCARCH="x86_64"
40                 LIBDIRSUFFIX="64"
41                 ;;
42         # Use "uname -m" for all other architectures
43         *)
44                 export CUCARCH=$(uname -m)
45                 LIBDIRSUFFIX=""
46         esac
47 fi
48
49 PACKAGE=$NAME-$VERSION-$CUCARCH-$BUILD
50 BUILDDIR=${BUILDDIR:-/tmp/$PACKAGE}
51 DESTDIR=$BUILDDIR/bin
52
53 # Save the Original Working Directory
54 OWD=$(pwd)
55
56 # If the build directory already exists, remove it so we have a fresh start
57 if [ -e $BUILDDIR ]; then
58         rm $BUILDDIR -rf
59 fi
60
61 # Make a temporary directory to build/stage in
62 mkdir $BUILDDIR/{bin,src} -pv
63 cd $BUILDDIR/src
64
65 tar -xf "$OWD/$NAME-$VERSION.tar.gz" || exit 1
66 cd $NAME-$VERSION || exit 1
67
68 # Do the actual build
69 ./configure \
70         --prefix=/usr \
71         --libdir=/usr/lib${LIBDIRSUFFIX} \
72         --sysconfdir=/etc \
73         --localstatedir=/var \
74         --build=$CUCARCH-cucumber-linux || exit 1
75 make || exit 1
76 make install DESTDIR=$DESTDIR || exit 1
77
78 # Strip
79 find $DESTDIR/{,usr/}{bin,lib${LIBDIRSUFFIX},sbin} -type f \
80         -exec strip --strip-debug '{}' ';'
81
82 # Copy any additional package scripts
83 mkdir -pv $DESTDIR/install
84 if [ -e $OWD/slack-desc ]; then
85         cat $OWD/slack-desc > $DESTDIR/install/slack-desc
86 fi
87 if [ -e $OWD/doinst.sh ]; then
88         cat $OWD/doinst.sh > $DESTDIR/install/doinst.sh
89         chmod 755 $DESTDIR/install/doinst.sh
90 fi
91
92 # Compress the package
93 cd $DESTDIR
94 makepkg -l y -c n /tmp/$PACKAGE.txz || exit 1
95 echo "Package $PACKAGE was successfully built to:"
96 echo "/tmp/$PACKAGE.txz"
97
```

The buildscript for the less package.

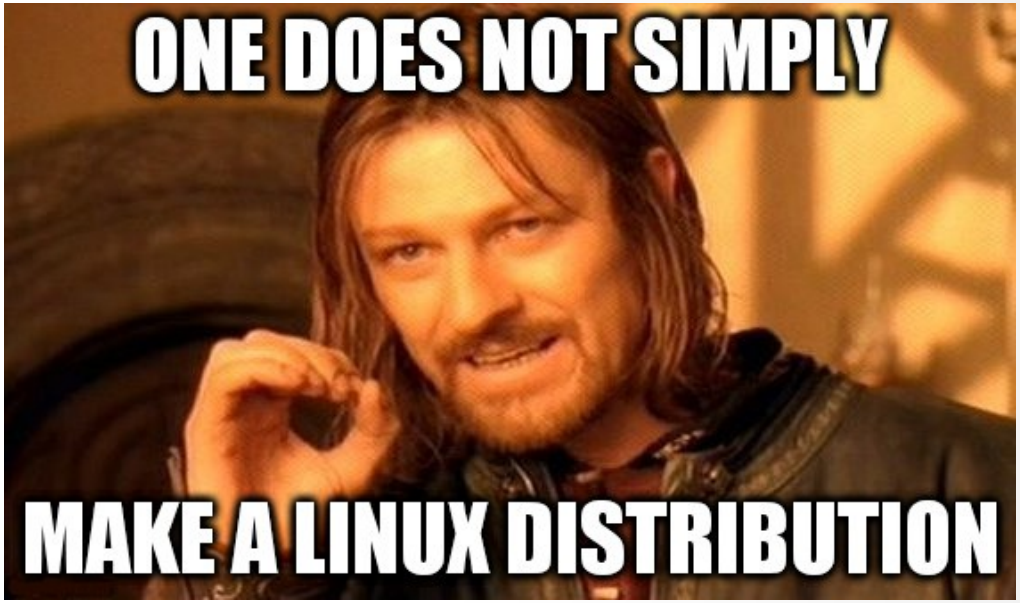
August 2016 – Alpha 1 Released

- At the end of August, the first alpha was released.
- What it had:
  - A shell
  - A functional compiler and toolchain
- What it didn’t have:
  - A desktop environment
  - Xorg
  - Any daemons
  - Proper networking support
  - Several essential programs
  - Systemd :)
- Total package count: 69



September 2016

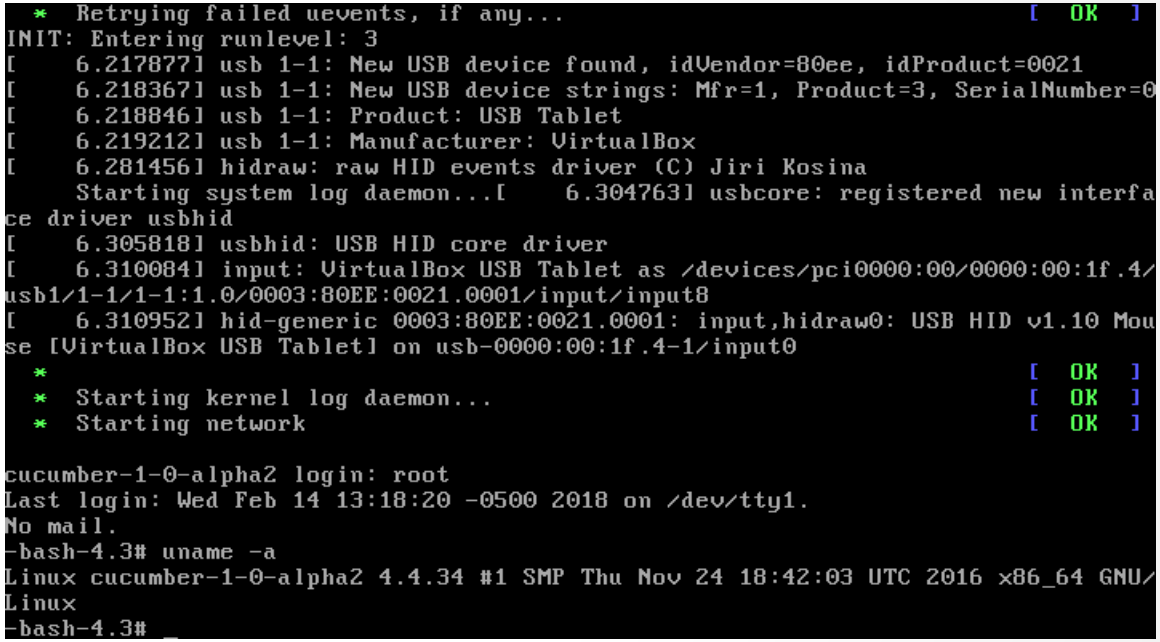
- This was a Summer project and now Summer was over, so I was done right?
- Not exactly. I had put too much effort into it to stop at creating a product that couldn't actually do anything useful, so I decided to continue working on it until I was satisfied it was reasonably complete (famous last words).



- It was at this point I decided to keep developing Cucumber Linux into a general purpose operating system, suited for desktop and server usage.

October 2016 – April 2017

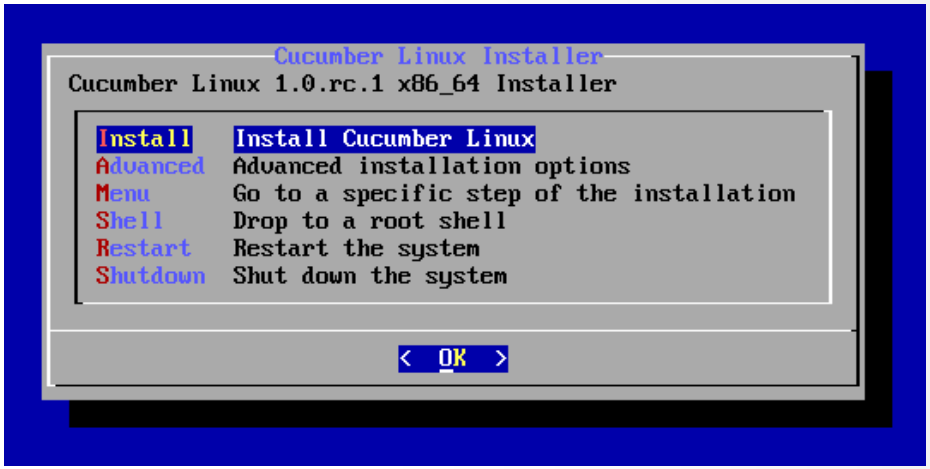
- Development continued:
  - Networking was fixed
  - Some daemons were built
  - I began cryptographically signing all the packages with GPG.
- In December 2016, the second alpha was released.
  - Total package count: 120
  - What it still didn't have:
    - A desktop environment
    - Xorg



- In early 2017, more daemons and the Xorg window system were added.
- Shortly thereafter, the XFCE desktop environment was added along with Firefox, Thunderbird, LibreOffice and some other common graphical applications.

May 2017 – Beta Released

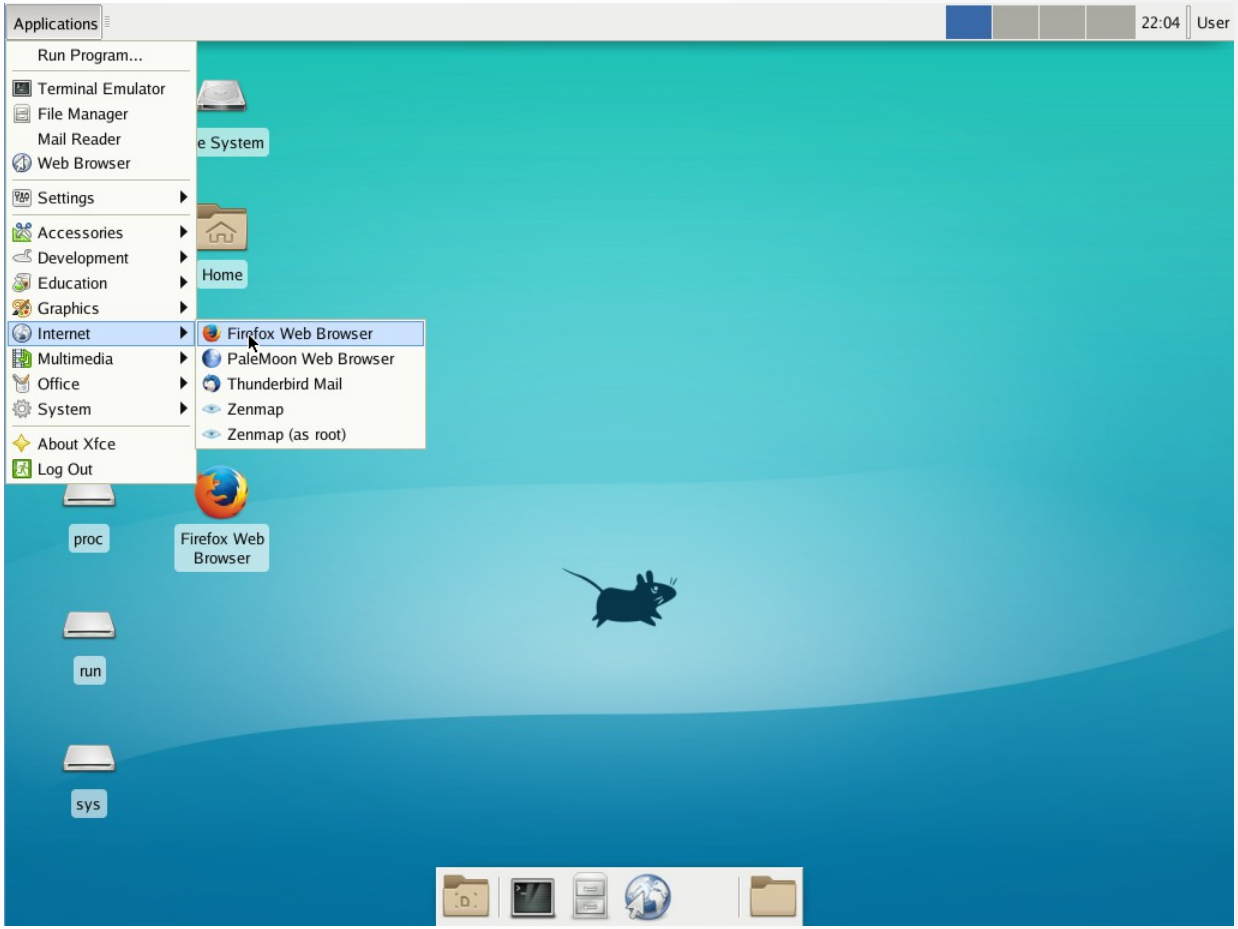
- In May 2017, the beta was released.
- What it had:
  - Xorg 7.7
  - XFCE desktop environment 4.12
  - Most commonly used graphical and command line applications.



- Total package count: 288
- At the release of the beta there was a feature freeze and the primary focus of development shifted to debugging in preparation for a stable release.
- This was followed by the first release candidate in June and the second release candidate in early July.

July 2017 – v. 1.0 Stable Release

- Finally, on July 10, 2017 Cucumber Linux 1.0 was released.
- What it had:
  - Linux 4.9 LTS
  - GNU Userspace Utilities
  - System V Init 2.88dsf
  - Apache 2.4
  - MariaDB 10.1
  - PHP 5.6
  - X.org 7.7
  - XFCE 4.12
  - Firefox Web Browser 52
  - Thunderbird Email Client 52
  - LibreOffice 5.3
- Final package count: 287



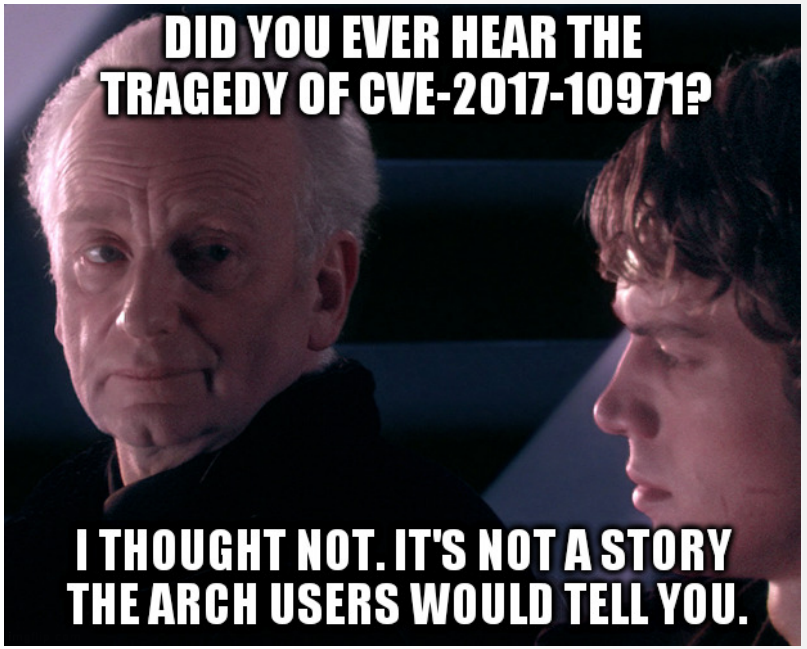
But I wasn't Done Yet

- After Cucumber Linux 1.0 was released, my primary focus shifted from developing the distribution to maintaining it (bug and security fixes).
- This was when I began to seriously get into Linux security.
  - I subscribed to the National Vulnerability Database RSS Feed (this is where new CVEs get announced).

  - I began patching all vulnerabilities (no matter how trivial they may seem) as soon as they were announced.
    - Sometimes packages would release official updates, so I could just download the new source tarball and rerun the buildscript for that package.
    - Not all packages were this nice though; other times I would have to manually backport patches.
  - I published and announced my patched packages, and was surprised to find that other distributions weren't releasing patches very quickly, if at all.
    - Usually, I would fix a vulnerability, and then it would be fixed in other distributions a few days to a few months after the fact.
    - One such early example of this was CVE-2017-10971.

The Tragedy of CVE-2017-10971

- A timeline of CVE-2017-10971 (a privilege escalation/arbitrary code execution vulnerability in Xorg-server).
  - Vulnerability disclosed on 7/6/2017.
  - Fixed in Cucumber Linux that same day.
  - Fixed in Debian 7/9/2017.
  - Fixed in SUSE on 7/14/2017.
  - Fixed in Arch Linux on 8/14/2017.
  - Fixed in Slackware on 8/15/2017.
  - Red Hat says they will not fix it.
- At this point I realized just how bad the security situation is in the Linux world.
  - So I decided from there on out to also make security a focus of Cucumber Linux.
  - It was shortly thereafter that “Simplicity, Stability, Security” was adopted as the project’s new motto.



Some Other Milestones

- August 2017 – Cucumber Linux was listed on DistroWatch.com.

DistroWatch.com
Put the fun back into computing. Use Linux, BSD.

Type Distribution Name | Go | Cucumber | Go
DuckDuckGo Site Search | Go | Random Distribution

• Home Page, Headlines
• DW Weekly, Comments
• Packages, Package Management
• Glossary, FAQ, Mobile Site

• Search, Sitemap
• Major Distributions
• Submit Distribution
• Upcoming Releases

News/Opinions/Reviews ▼ | Packages ▼ | Find/Submit Distro ▼ | Tutorials & Learning ▼ | Related Resources ▼ | DistroWatch.com ▼

DistroWatch.com: Cucumber Linux

Cucumber Linux

Last Update: 2017-08-26 01:22 UTC

Cucumber

- **OS Type:** Linux
- **Based on:** Independent
- **Origin:** USA
- **Architecture:** i686, x86_64
- **Desktop:** Xfce
- **Category:** Desktop, Server
- **Status:** Active
- **Popularity:** 193 (41 hits per day)

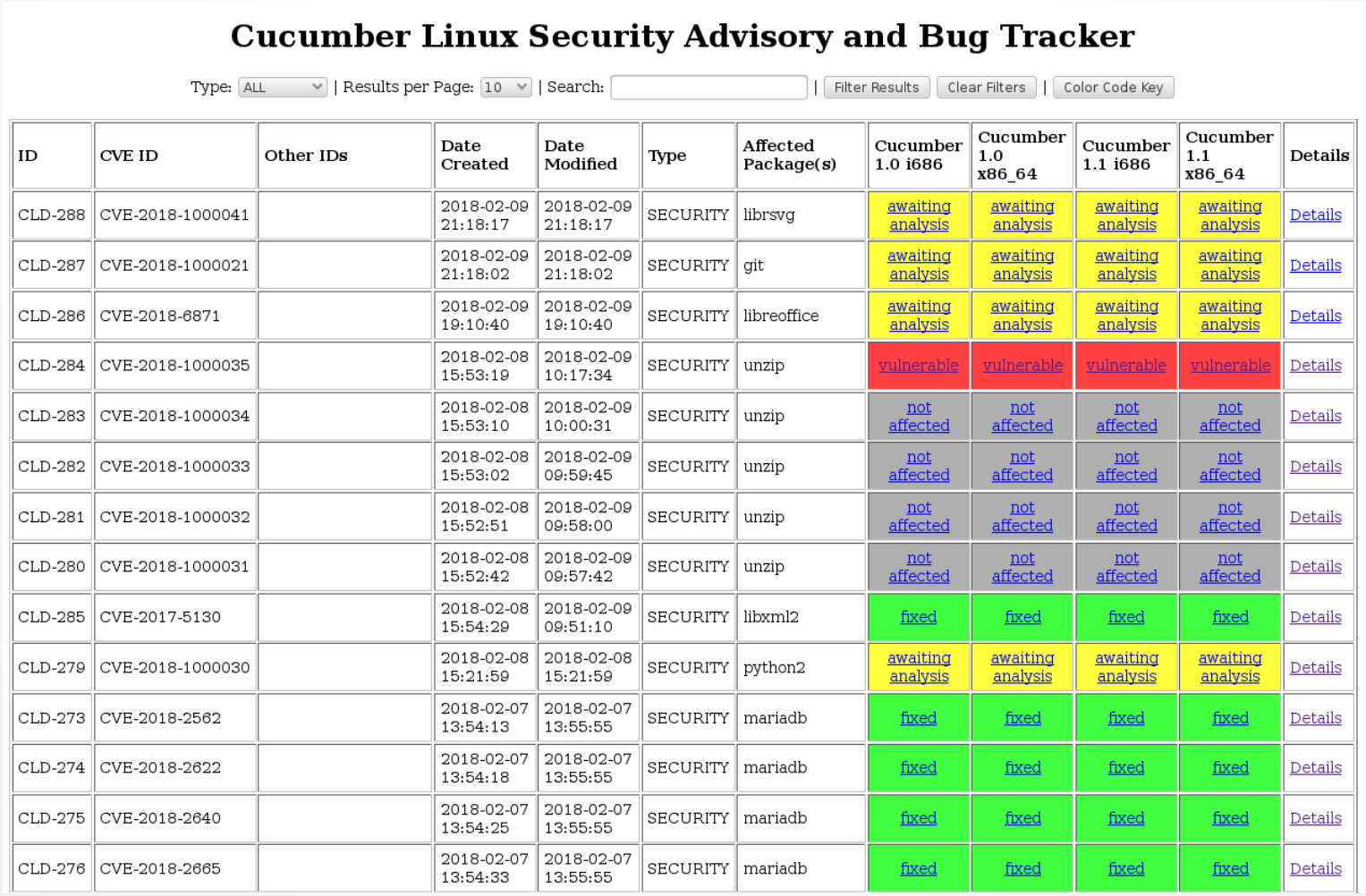
Cucumber Linux aims to provide a Linux distribution that is usable as an every day, general purpose operating system. It aims to this in as minimalistic a way as possible and in a way that follows the Unix Philosophy. Cucumber Linux favors simplicity and modularity of design over simplicity of use. While developed independently, Cucumber's design is heavily influenced by Slackware Linux.

Popularity (hits per day): 12 months: **279** (24), 6 months: **193** (41), 3 months: **223** (34), 4 weeks: **204** (37), 1 week: **226** (34)

Average visitor rating: **6.25**/10 from **4** review(s).

Some Other Milestones

- September 2017 – The Cucumber Linux security tracker debuted



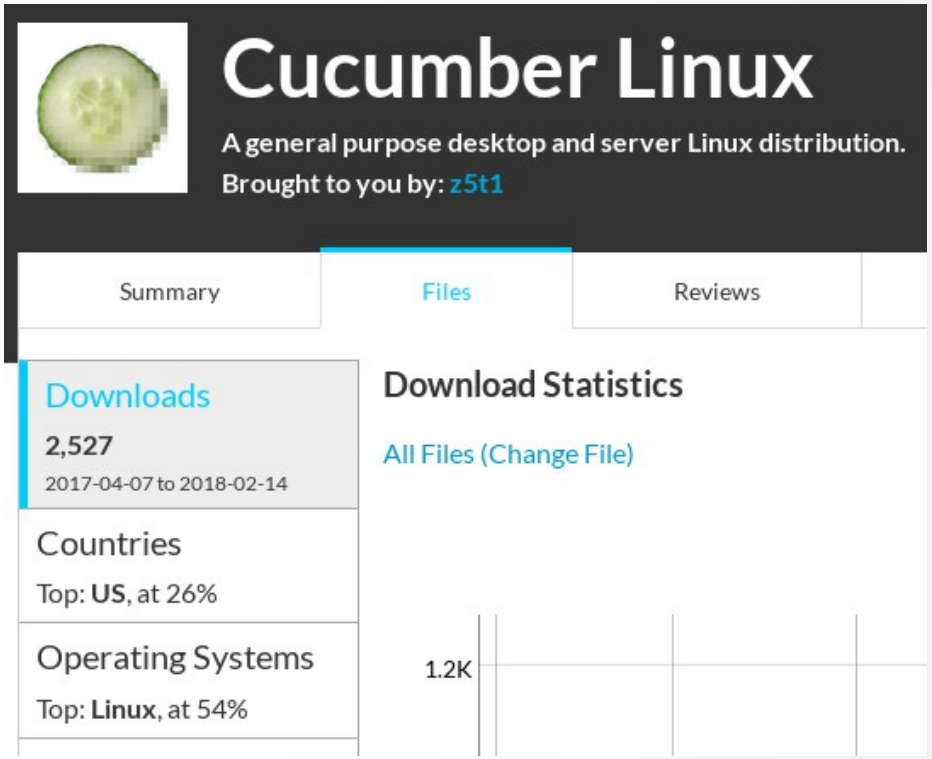
Cucumber Linux Security Advisory and Bug Tracker

Type: ALL | Results per Page: 10 | Search: | Filter Results | Clear Filters | Color Code Key

ID	CVE ID	Other IDs	Date Created	Date Modified	Type	Affected Package(s)	Cucumber 1.0 i686	Cucumber 1.0 x86_64	Cucumber 1.1 i686	Cucumber 1.1 x86_64	Details
CLD-288	CVE-2018-1000041		2018-02-09 21:18:17	2018-02-09 21:18:17	SECURITY	librsvg	awaiting analysis	awaiting analysis	awaiting analysis	awaiting analysis	Details
CLD-287	CVE-2018-1000021		2018-02-09 21:18:02	2018-02-09 21:18:02	SECURITY	git	awaiting analysis	awaiting analysis	awaiting analysis	awaiting analysis	Details
CLD-286	CVE-2018-6871		2018-02-09 19:10:40	2018-02-09 19:10:40	SECURITY	libreoffice	awaiting analysis	awaiting analysis	awaiting analysis	awaiting analysis	Details
CLD-284	CVE-2018-1000035		2018-02-08 15:53:19	2018-02-09 10:17:34	SECURITY	unzip	vulnerable	vulnerable	vulnerable	vulnerable	Details
CLD-283	CVE-2018-1000034		2018-02-08 15:53:10	2018-02-09 10:00:31	SECURITY	unzip	not affected	not affected	not affected	not affected	Details
CLD-282	CVE-2018-1000033		2018-02-08 15:53:02	2018-02-09 09:59:45	SECURITY	unzip	not affected	not affected	not affected	not affected	Details
CLD-281	CVE-2018-1000032		2018-02-08 15:52:51	2018-02-09 09:58:00	SECURITY	unzip	not affected	not affected	not affected	not affected	Details
CLD-280	CVE-2018-1000031		2018-02-08 15:52:42	2018-02-09 09:57:42	SECURITY	unzip	not affected	not affected	not affected	not affected	Details
CLD-285	CVE-2017-5130		2018-02-08 15:54:29	2018-02-09 09:51:10	SECURITY	libxml2	fixed	fixed	fixed	fixed	Details
CLD-279	CVE-2018-1000030		2018-02-08 15:21:59	2018-02-08 15:21:59	SECURITY	python2	awaiting analysis	awaiting analysis	awaiting analysis	awaiting analysis	Details
CLD-273	CVE-2018-2562		2018-02-07 13:54:13	2018-02-07 13:55:55	SECURITY	mariadb	fixed	fixed	fixed	fixed	Details
CLD-274	CVE-2018-2622		2018-02-07 13:54:18	2018-02-07 13:55:55	SECURITY	mariadb	fixed	fixed	fixed	fixed	Details
CLD-275	CVE-2018-2640		2018-02-07 13:54:25	2018-02-07 13:55:55	SECURITY	mariadb	fixed	fixed	fixed	fixed	Details
CLD-276	CVE-2018-2665		2018-02-07 13:54:33	2018-02-07 13:55:55	SECURITY	mariadb	fixed	fixed	fixed	fixed	Details

And so it Continues...

- October 2017 – development began on Cucumber Linux 1.1.
- November 2017 – Cucumber Linux 1.1 Beta was released.
  - This added mail server (OpenSMTPD + Dovecot) and DNS server (Bind) capabilities.
  - Total package count: 297
- February 2018 – Cucumber Linux reached 2500 all time downloads.
- Cucumber Linux 1.1 is currently slated for a release in the next month.



Questions?

Website: cucumberlinux.com

Email: scott@cucumberlinux.com

This presentation can be found online at
https://cucumberlinux.com/~scott/presentations/