BootCD v3.x Technical Documentation

Aaron Klingaman

Revision History

Revision 1.0 November 17, 2005 Revised by: AK
Initial draft.

Table of Contents

Overview

This document describes in detail how the PlanetLab boot CD is built and operates when running on a node. Older boot CDs, including 2.x cds, are not the focus of this document, and are no longer being deployed on production systems.

Background

Since the early days of PlanetLab, all production nodes are configured during setup to only start up off of the cdrom, with a PlanetLab boot cd always left in the drive. The intention is to allow a machine to be able to restart into a known environment, for debugging system problems, or as a way to still access the machine but not have any potentially compromised code to run if the system is believed to be compromised.

Soure Code

All 3.x boot cd source code is located in the repository 'bootcd_v3' on the PlanetLab CVS system. For information on how to access CVS, consult the PlanetLab website. Unless otherwise noted, all file references refer to this repository.

Basic Operation

The operation of the boot cd, when a machine is started off of one, is fairly straight forward. Essentially, it loads a linux kernel, configures the hardware and network, and fetches a signed script to execute. This generic operation allows for the boot cds to be used for any number of operations, whether they are installing machines or debug problems.

The full operation of a boot cd, from the moment it is booted, is described in the following diagram.

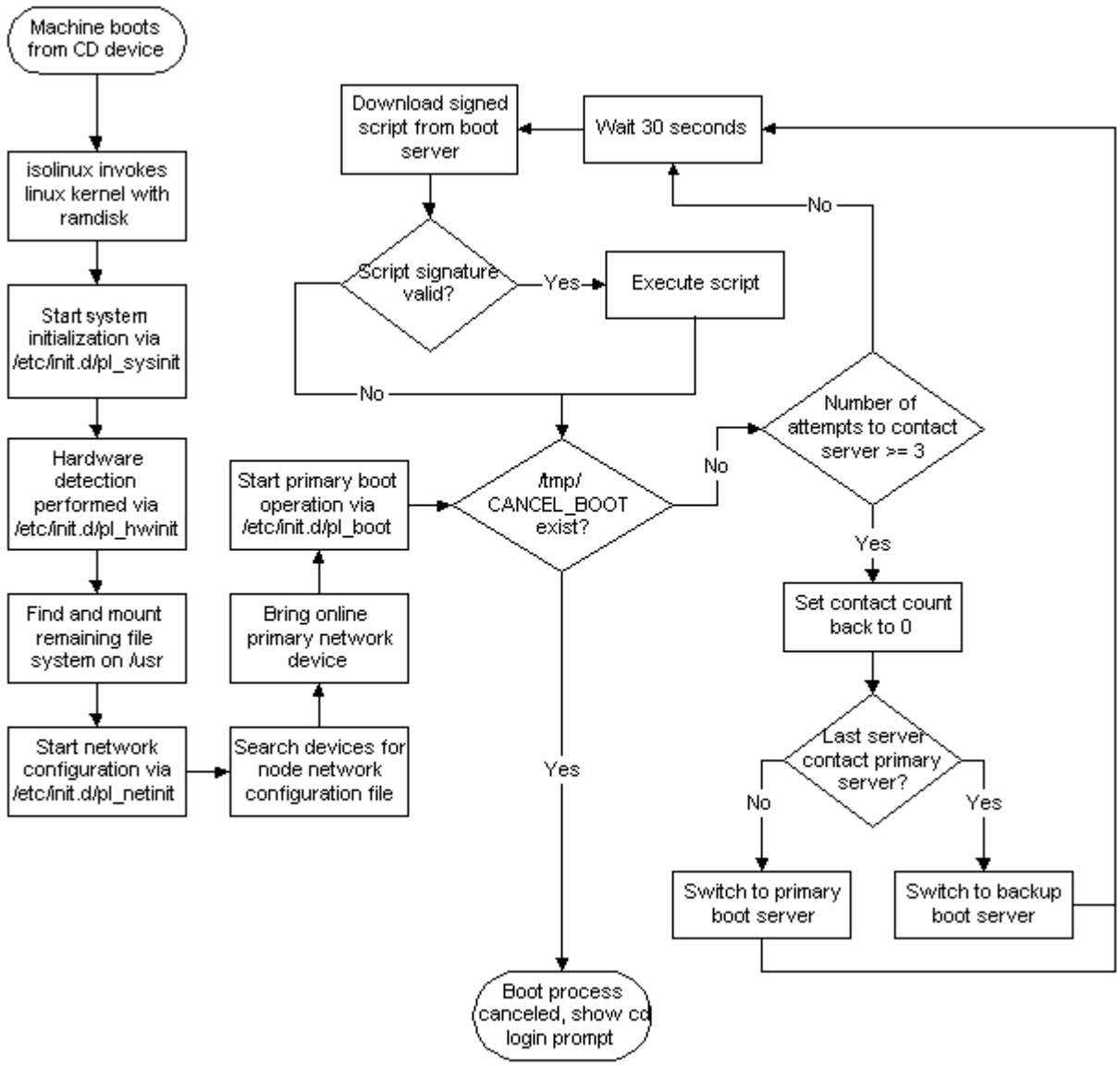

Figure 1. BootCD Operation Flowchart

Security

Ensuring that the boot cd provided a secure node boot mechanism was a primary concern during its development. The following requirements we used:

1. The boot cd should be immutable. At any point, a PlanetLab administator should be able to reboot a machine into a known safe environment to inspect or debug the node.
2. The cd should verify that the servers it contacts for executable scripts should be PlanetLab Central servers, and not someone posing as one.
3. The scripts executed are to be signed by PlanetLab Central.

Accomplishing 1. is fairly easy: simply require the cds to be burned onto a write once media. Once that is accomplished, it is up to local site administators to ensure physical security of the node so the cd is not switched out. Further work may be

done by executed scripts to validate a boot cd, if necessary (though not currently implemented).

Number two is accomplished through the use of SSL certificates. The PlanetLab Central boot server, running Apache at the time of this writing, uses a self signed SSL certificate. The boot cd, for each server it is to contact (a primary server, and a backup server), contains the CA certificates for those servers. Using the URL downloading tool curl, the scripts on the cd can ensure they are contacting a PlanetLab boot server, and not someone attempting to spoof one.

Number is is accomplished through the use of GPG public and private keys. There exists at PlanetLab Central a GPG private key that is used to sign the scripts downloaded and executed by the cd. The public key is located on the cd, and used to validate the signatures of the packages before execution.

Hardware Detection

After the Linux kernel is loaded, the first operation is to load the applicable hardware modules for devices on the system, including network drivers, disk drivers, and any others. This process is nearly identical to the process the BootManager uses. During the initial boot cd build process, the script sources/merge_hw_table.py from the bootmanager repository is invoked to create a lookup table to map PCI ids onto kernel modules. For more information about how this script operates, consult the BootMangaer technical documentation.

Building A BootCD

Previous PlanetLab boot cds were essentially boot cds from other projects, modified for use with PlanetLab. With the introduction of version 3.0 of the boot cd, they are now built from scratch. By doing this, we can ensure that the packages contain on the cd are fully up to date, only the packages we need for booting operations are installed (thus reducing the cd size), and the hardware detection mechanisms match that of the node installer (BootManager).

Though the cds are built from scratch, the process to build a cd is relatively simple, and are as follows:

1. The build process is currently only tested with and known to work with Fedora Core 2. You'll need root access on a FC2 machine.
2. Check out the boot cd repository from PlanetLab CVS:

   ```
   cvs -d :pserver:anon@cvs.planet-lab.org:/cvs co bootcd_v3
   ```

3. Initiate the build by running, from the bootcd_v3 directory:

   ```
   ./build.sh build default
   ```

4. When complete, the resultant iso image will be located in configurations/default/

The default configuration build above produces a boot cd that is configured to contact the primarily PlanetLab boot servers. To build a custom boot cd that contacts a different server, with a different SSL certificate and GPG key, you will need to create a custom configuration:

1. Change into the bootcd_v3/configurations directory:

   ```
   cd bootcd_v3/configurations
   ```

2. Copy the entire default directory, creating a new one with a short name for the custom configuration. The name is only used during the build process, and is not part of the actual cd.

   ```
   cp -r default mycustomcd
   ```

3. Edit the configuration file in the new directory. That file contains various fields that allow for the cd operation to be customized, see the section, Build Configuration Options for more information.
4. Once complete, the custom cd can be built with:

   ```
   ./build.sh build mycustomcd
   ```

Build Configuration Options

The configuration file for builds (the default being located at configurations/default/configuration, contains the following values that can be modified to result in a custom build boot cd:

- EXTRA_VERSION

  Set this to add extra version information to this cd. This will be added to the result ISO name, and on the cd. By doing so, you will be able to differentiate the cds from PlanetLab Boot cds (which have no extra version.

- DESCRIPTION

  A simple text description, one line, of the boot cd.

- ROOT_PASSWORD

  The crypted password to use for the root account on the boot cd. Only applies to the boot cd, not the root account on an installed and fully running node.

- PRIMARY_SERVER / BACKUP_SERVER

  The hostname of the server to attempt to contact first, and a backup server if that one fails.

- PRIMARY_SERVER_PORT / BACKUP_SERVER_PORT

  Which SSL port on the server we should contact (default SSL port is 443). This rarely will need to be changed.

- PRIMARY_SERVER_PATH / BACKUP_SERVER_PATH

  The path containing the script this cd should download and execute. Can either be a path to an exact file, like /boot/bootscript, or, can be a directory or dynamically executed file, like /boot/index.php or just /boot. In this case, the resultant output of that file/directory should be a signed and executable script.

- PRIMARY_SERVER_CERT / BACKUP_SERVER_CERT

  The SSL CA certificate(s) for the above server(s). This is used to validate that the server we are contacting has not been spoofed.

- PRIMARY_SERVER_GPG / BACKUP_SERVER_GPG

  The GPG public key(s) of the private key(s). that was used to sign the script that will be returned by PRIMARY_SERVER_PATH or BACKUP_SERVER_PATH

- NODE_CONFIGURATION_FILE

  If this cd is to be used exclusively by a single node, that node's network configuration file can be placed on the cd. This is the path on the local system to that configuration file, which will be copied to a known location on the cd and used during boot up.

With regard to file paths: for the locations of the keys, certificates, and optionally node configuration files, it is easiest to place these files inside the directory with the bootcd configuration file, and simply use the name of the file for the value. See the default configuration file for an example.

Build Package Sources

The packages used during the build process, by default, are downloaded from the current PlanetLab Central boot server. To change this, the file yum.conf, in the checked out bootcd_v3 directory, will need to be modified. The yumgroups.xml file, also located in the same directory, is used by the build process to identify which packages should be placed on the resultant cd image. This file should be located in one of the yum rpm repositories specified in yum.conf.