McAfee® Command Live Scanner for Windows
Live Linux Boot CD Project

Quick-Start Guide

Version 1.0.0
February 8, 2010
Michael G. Spohn
mspohn@malware-hunters.net

Table of Contents

Introduction

This document is a condensed version of the McAfee Command Line Scanner Project (MCLSP) *User's Guide*. It was written for intermediate or advanced Linux administrators who require minimal guidance to build a custom Linux distro. It highlights the steps required to build a custom Ubuntu Linux distro that runs the McAfee A/V command line scanner for Windows.

If you prefer more detailed documentation or are not a strong Linux user, then I recommend you refer to the *User's Guide*. Also, be sure to read the *FAQ* document which provides good background information about this project.

Build Requirements

In order to build a custom live Ubuntu distro for the McAfee command line scanner you will need the following:

- A build platform running Ubuntu Linux.

  This is simply a workstation or laptop running Ubuntu Linux. You will use this system to build a custom live Ubuntu distro with the McAfee scan engine. I strongly recommend you use Ubuntu Desktop Version 10.10 (Maverick Meerkat) due to its extensive driver coverage for the latest hardware. **Note**: Do not try to build this project using a Live CD/USB stick. Ramdisk versions of Ubuntu do not have enough available memory to build this project.

- The latest copy of the McAfee Anti-Virus Command Line Scanner for Windows.

  Unfortunately, this product is no longer freely available on the Internet. The scan engine has been removed from the SuperDAT distribution. You will need to contact McAfee to obtain a licensed copy. Evaluation versions of the product are available here. You are responsible for complying with McAfee's licensing requirements.

- An Ubuntu Linux distribution ISO for the base re-master image.

  This ISO should be the same version of Ubuntu you use for your build environment. I have done extensive testing on Ubuntu 10.10 Desktop (Maverick) for both my build platform and base re-master image. You can download a base Ubuntu ISO here. If you are partial to Ubuntu 10.04 Desktop (Lucid) it will work just as well.

- The MCLSP build system archive.

  This tar archive contains the scripts and configuration files you need to build a custom live Linux ISO. You can download the archive from www.malware-hunters.net.

- A connection to the Internet

  Your build system must be connected to the Internet because the build scripts download packages required for the build, and DAT files from the McAfee ftp site.

Creating Build Environment

Setting up the build environment is pretty painless. It involves extracting the MCLSP build file archive to a folder on the build platform. The steps below describe this process:

- Download the MCLSP project file archive and unzip it.

  I recommend you place this file in the /opt folder and unzip it with the command:

  sudo tar –zxvf mcslsp.tar.gz

  This will extract the files into a folder named /opt/mclsp

- Copy the base re-master ISO to the build folder.

  Make sure there is a copy of the Ubuntu base re-master ISO in the root of the mclsp project folder. In my build environment, the file is:

  /opt/mclsp/Ubuntu-10.10-desktop-i386.iso

- Copy the McAfee command line scanner product zip file to /opt/mclsp/mcafee.

  Place the McAfee command line scanner product zip file in the folder /opt/mclsp/mcafee. You do not need to extract it since the build scripts will do this for you.

  NOTE: The /opt/mclsp/mcafee folder is where you should place any files that you want available on the custom distro ISO. This includes EXTRA.DAT files or other tools you need. All of the configuration and operational files of the custom distro can be found in the /home/mcafee folder.

  You now have a proper build environment you will use to create a custom live Ubuntu distro.

Configuration Options

You need to make some decisions about your configuration options because they get baked into the custom distro ISO you are about to build. There are three configuration points in the build process; 1) The desktop background image you want to use, 2) The A/V scan environment options in the av_config.ini file, and 3) The McAfee scan engine options in the av_options.txt file.

1. Configuring the desktop background image.

   The build configuration settings will use the file */home/mcafee/backgrounds/av_background.png* as the desktop background of your custom build. If you want to customize the background image, simply replace the existing file with your own. It is recommended you provide an image with a resolution of 1600x1200 because it renders well on most display devices.

2. Configuring the environment options.

   You control the behavior of the McAfee A/V scan environment by changing settings in the */opt/mclsp/mcafee/av_config.ini* file. If you look at this file you will see it has a typical ini file layout. The most important sections are [AUTORUN] and [MOUNT].

   The *autoscan* option in the [AUTORUN] section determines whether a McAfee scan is automatically run at boot time. The default setting is '1' which means a scan of all FAT/NTFS volumes on the target system will run at boot. Set this value to '0' if you do not want this to occur.

   The *mount* option in the [MOUNT] section determines if FAT/NTFS volumes are mounted in read-only or read-write mode. The default setting is to mount all devices as read-only (ro). This means the McAfee scan engine will *report* on viruses found, *but will not clean or delete compromised files*. If you want the scan engine to clean or delete infected files, change this setting to (rw).

3. Configuring the McAfee scan engine options.

   McAfee has provided excellent documentation with its command line scanner product. You can find it in the PDF document named *vcl6wpg.pdf* in the product distribution zip file. It is highly suggested you read this document to understand the power and capabilities of the scan engine.

   You control the behavior of the McAfee A/V scan engine by changing settings in the */opt/mclsp/mcafee/av_options.txt* file. This file contains the scan options that will be used by the command line scan.exe engine. There are a lot of options so it is critically important you understand them. The default options file provided will scan all files on a volume and report them in */tmp/mcafee/av_badlist.txt.* It will take no action against any found malware. You will certainly want to change the options in this file to suit your particular needs.

   Note: To enable the McAfee scan engine to clean malware from a host you need to do two things: 1) Change the *mount* parameter in the [MOUNT] INI file section to *rw* **and** 2) Add the /CLEAN option to the *av_options.txt* file.

Building Custom ISO

To build your custom live Ubuntu ISO with the McAfee scan engine you need to run three scripts: 1) *prereq.sh*, 2) *chroot.sh*, and 3) *build_custom_iso.sh*. These scripts are heavily commented and describe what each set of commands do. It is highly recommended you review each script to get an idea of how they work. These scripts must be run with root privileges so you *must* be logged in as root or use the *sudo* prefix. Also be sure you are connected to the Internet prior to running the scripts.

1. Run script *prereq.sh*.

```
sudo chdir /opt/mclsp/
sudo ./prereq.sh
```

The *prereq.sh* script sets up the build environment. It creates the required folder structure and extracts the contents of the base re-master ISO file, and mounts it on the loopback device.

2. Run script chroot.*sh*.

```
sudo chdir /opt/mclsp/
sudo .chroot edit /chroot.sh
```

The *chroot.sh* script does most of the heavy lifting for customizing our distribution. It removes lots of unwanted packages to shrink the size of the ISO file, downloads and installs the latest version of wine, creates our custom *initrd.gz* file, and sets some default configuration items.

It is important to understand what the command sudo .chroot edit /chroot.sh does.

sudo	runs the command as root.
chroot	changes the base of our build system root filesystem to /opt/mclsp/edit. This folder contains the root filesystem of our custom build distro.
/chroot.sh	runs the script *chroot.sh* from the base of the chroot file system (i.e. /opt/mclsp/edit.

This script will take a few minutes to run because it has a lot of work to do.

3. Run script *build_custom_iso.sh*.

```
sudo chdir /opt/mclsp/
sudo . /build_custom_iso.sh
```

The *build_custom_ISO.sh* script creates the custom ISO file used to clean hosts of malware. It takes the changes made by the *chroot.sh* script and bakes them into our custom distro. The output of this script is our custom ISO named *ubuntu-10.10-Desktop-i386_mcafee_av.iso.*

Deployment

Now that you have a custom ISO, you need to burn the image to bootable media such as a CD/DVD or USB thumb drive. This is easily done using open source tools. On Linux, my tools of choice are *Brasero* for burning CD/DVD's and *UNetbootin* for burning ISO's to thumb drives. You can install these apps on your build system using the Synaptic Package Manager or do it quickly from a terminal window using the below commands:

```
sudo apt-get install brasero
```

```
sudo apt-get install unetbootin
```

Once installed, you can access these applications from the Applications | System menu. You now can create bootable media to automatically boot and scan Windows hosts for malware.

Troubleshooting

I spent a lot of time tweaking the build environment to make it as easy as possible for you to create a custom Ubuntu distro to clean systems of malware. That said, the three build scripts do a lot of work and there are many moving parts. If you are having trouble with your builds, review the below items to assist in your troubleshooting efforts.

- *You must be root to install and execute the scripts*

  There is no way around this since we must mount file systems and make changes to the environment. Be sure to use the *sudo* command (preferred), or login as root prior to running the build scripts.

- *Make sure your build environment is the same version as the base re-master*

  In other words – your host build environment should be the same version of Ubuntu Linux as the base system you are going to re-master. If you use *Ubuntu-10.10-desktop-i386.iso* for your custom build then you should be building the custom ISO on *Ubuntu 10.10 Desktop.* I have not experienced it, but there is a lot of discussion on the Internet about complications of mixing hosting and build versions of custom distro's. Save yourself some headaches and make sure your host and build environments are the same.

- *Make sure you have connected to the Internet*

  The build scripts download packages and files from the Internet so be sure your build host has Internet access before you try to build a custom ISO.

- *Make sure the chroot environment has networking configured properly*

  The *prereq.sh* script copies your build system *resolv.conf* and *hosts* file to the chroot location (/opt/mclsp/edit). When running the *chroot.sh* script, if there are problems connecting to the Internet, you need to ensure the chroot environment network settings are configured properly. You may have to edit the *prereq.sh* script as required.

- *Make sure you run all three scripts in the proper order*

  Once you have successfully built a custom ISO, you only need to re-run the chroot.sh and build_custom_iso scripts to ensure your build has the latest McAfee DAT file. If you are having trouble successfully building a custom ISO, reboot your host system and run the three scripts again in the correct order: *prereq.sh, chroot.sh* and *build_custom_iso.sh.*

If you are still having trouble, send me an Email describing the issue and I will look into it when I can. mspohn@malware-hunters.net.