Prof. Dr. C. Eckert
Tamas Lengyel
Fatih Kilic
Thomas Kittel
Sebastian Vogl

Institut für Informatik
Lehrstuhl für
Sicherheit in der Informatik

Assignment 1: Setup & LKM Introduction

13.10.2014

1 Register for Mailing List

We will use a mailing list for all official communication. I also urge you to use this to ask questions and answer one another's questions if they come up. The mailing list address is:
`rootkit@sec.in.tum.de`

In order to subscribe for the mailing list send a blank email to:
`rootkit-subscribe@sec.in.tum.de`

Once you have been added to the mailing list send an email with your group number, your name, and your partner's name.

2 Install Debian 7.6 x86_64

Install Debian 7.6.0 "wheezy" in the virtual machine using the iso image provided at:
`http://cdimage.debian.org/debian-cd/7.6.0/amd64/iso-cd/debian-7.6.0-amd64-netinst.iso`
Make sure you install the 64-bit version. My advice would be to choose a disk size of 10GB and to install the whole system in a single partition, except for the swap partition. Installing a desktop environment is not necessary, keep this installation as simple as possible. When asked, I recommend choosing to install "SSH server" and "Standard system utilities".

2.1 Remove VirtualBox Guest Extensions

Debian will detect that the system is running in a VirtualBox VM and install guest extensions. These will not work with the vanilla kernel that we install in the next step, so it is best to simply remove them. These packages begin with "virtualbox-ose-guest".

3 Install Vanilla Kernel 3.16.4

The kernels released with distributions of Linux are often patched to suit the needs of the particular distribution. Since we want to make sure that our kernel behaves as specified, we will use a so called vanilla kernel. A vanilla kernel is the kernel as released by the Linux kernel developers. Use the kernel source provided at:
`https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.16.4.tar.xz`
and compile the kernel for your Debian system.

3.1 Prerequisites

If you followed my installation instructions, the only prerequisite packages that need to be installed are "xz-utils" and "build-essential".
`# aptitude install xz-utils build-essential`

3.2 Configuring the Kernel

Begin by downloading the kernel
`# wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.16.4.tar.xz`
and extracting it.
`# tar xvJf linux-3.16.4.tar.xz`

To configure the kernel, we will use the same configuration file that is currently in use for the new kernel. This means that you should copy `/boot/config-[kernel version]` to the top-level of your build directory and rename it to `.config`. Then run
`# make oldconfig`
You will be given several prompts, simply hit enter to choose the default choice for all prompts until you are back at the command line.

3.3 Compile and Install the Kernel

Begin by compiling the kernel
`# make`
Then, compile the modules
`# make modules`
Next, install the modules
`# make modules_install`
Finally, install the kernel
`# make install`

3.4 Create ramdisk and Update grub

Begin by creating a ramdisk
`# mkinitramfs -o /boot/initrd.img-3.16.4 3.16.4`
Finally, update the bootloader
`# update-grub2`

4 Create sysmap.h Script (submitted)

The system map file is a file that is often found in the boot directory of Linux distributions. This file stores the the addresses of kernel symbols and is of the form:
`address type symbol_name`
Symbols of type `D` (initialized data), `R` (read-only data), and `T` (code) are of interest to us.

Write a bash script that creates a header file (sysmap.h) that makes the symbols usable in your code (it might be easiest to do this with `#DEFINE`, but I will leave it up to you).

5 LKM Programming - mod.ko (submitted)

Create a Linux LKM (mod.ko) that does the following:

a) Use `printk` to perform any output. This output should use the `KERN_INFO` log level.

b) Your module should output welcome and goodbye messages when mod.ko is loaded and unloaded, respectively.

c) Your module should contain a function `print_nr_procs()`. This function should output the number of processes in the system.

d) After the welcome message, this module should call `print_nr_procs()` when loaded.

As a reference use the tutorial found at `http://tldp.org/LDP/lkmpg/2.6/html/index.html` (this is for 2.6 kernels, but the simple hello world examples are relevant for 3.x as well). Finally, include a Makefile and a README file in your submission. The Makefile should make sure that I can compile your module with `make` and the README file should give me any extra information I need in order to get your submission running.