Linux Kernel Debugging

Stephen Dennison

April 28, 2013
Goal

Results
  Progress
  Process

Questions
Objective

Learn more about kernel debugging
Chapter 18: Debugging

- Does not contain the word breakpoint
Chapter 18: Debugging

- Does not contain the word breakpoint
- Not to sell it short, it contains some very nice hints and LOTS of other good information
Chapter 18: Debugging

- Does not contain the word breakpoint
- Not to sell it short, it contains some very nice hints and LOTS of other good information
- However, only in passing does it mention:
  - gdb
  - kgdb
What I hope to accomplish...

- Set breakpoints
- Single step
- Disassemble unknown code
- Resolve symbols
- Source level debugging
- All the typical debugger stuff
What I accomplished...

- Set breakpoints
- Single step
- Disassemble unknown code
- Resolve symbols
- Source level debugging
- All the typical debugger stuff (maybe)
Overview

- Launch VM
- Edit/build/edit...
- Deploy
- Build on the target
- Load the module
- Connect GDB
- Load Symbols
- Enjoy... but don’t Control+C! Use the Qemu monitor.
Use `-s` for remote debugging over a serial port, it’ll be TCP port 1234 on localhost, and use `-monitor stdio` so you can see the monitor and the contents of the VM at the same time.
The typical process of editing the source. Compiling is just to make sure that it’s ready for deploying to the VM. Avoid editing source on the VM unless you’ve taken steps to prevent losing it all when the VM locks up.
Deploy

Copy the entire project to the VM.
If the kernel is the same on the VM and host, you only need to deploy the compiled module.
Build on the target

Compile for debugging:

- `EXTRA_CFLAGS=-g`
Load the module

insmod
Connect GDB

Pretty much what you’d expect...

- `gdb vmlinux`
  - or just `gdb`
  - but if you want symbols...
- `target remote :1234`
Load Symbols

add-symbol-file module-path load-offset
To get that offset:

- cat /sys/modules/modulname/sections/.text
- or sudo grep modulename /proc/kallsyms
Questions

???