CS451 – Software Analysis

Lecture 0

Introduction to Course and Logistics

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

What is this course all about?

- In many courses we learn how to develop software
  - Use programming systems/languages
  - Leverage software engineering techniques
  - Realize algorithms
  - Become familiar with APIs and frameworks
- Little attention is given to handling or processing existing software

Problem 1

- You have a large program with many different source files
- You want to see where most of your time is spent when running your program
- What do you do?

Problem 2

- You have a program, but you don’t have access to the source code
- The program has a bug in a function
- You know the bug and how to fix it, what can you do?

Problem 3

- You are given an executable
- You don’t know anything about it
- It may be malicious or not
- Can you safely run it?

Problem 4

- You are given a C program with 50K LoCs (lines of code) and around 2K different functions
- You want to log each function call in a log file
- Can you do this?

Problem 5

- You have a program that is processing data and then it crashes
- You need to see which code is executed before the process is crashing
- What do you do?

Problem 6

- You want to replace the memory allocator of a program
- What is a memory allocator?
- How can you plug a new memory allocator to an existing program?
- Do you need to recompile the program?

Software analysis

- This is just a fraction of problems that someone can solve with appropriate techniques of software analysis
- Some of the problems look tough
  - They might be, but good solutions exist
- Problems are from **different** domains
  - System administration, program development, malware analysis, performance optimization, forensics, etc.

Better understanding

- Developing the techniques gives us a better understanding of
  - How software works
  - How systems execute software
- Do you know how a program locates and calls the code of printf()?
- Do you know how a debugger implements a breakpoint?
- Have you ever modified a compiler?

Contents – bird’s eye view

- The structure of binaries
  - ELF format
  - Important binary technologies, such as the use of GOT/PLT
- Techniques for binary analysis
  - Light inspection and disassembly (linear/recursive)
  - Debugging (ptrace())
  - Static and dynamic
- Techniques for generating enhanced binaries
  - Binary re-writing
- Techniques for instrumenting source code
  - Extending software using modern compiler frameworks (LLVM)

Approach

- The course is heavily based on hands-on experience
- Several tools will be implemented from scratch
- Analysis techniques will be applied to actual software
- A specific Virtual Machine image is used through out the course (see later slides)

Class structure

- Two lectures
  - Monday/Thursday according to the timetable
- Lab lecture
  - Wednesday (will be announced)

Lectures

- Slides will be available in advance
- Incorporate a lot of hands-on experience
  - Most of the useful work is in demonstrations that I do
- Real-time examples
- Homework

Homework

- Each lecture has a set of tasks that I do in real-time
  - Sometimes, I may actually build a small tool as part of the lecture
- My part will be at around 90-95% finished
  - Your homework is to deliver the remaining 5-10%
- Your steps will be clearly stated in the last slide of the lecture
  - Your 5-10% is based on trivial steps
  - **However**, you need to understand my 90-95% in order to achieve these steps

Homework

- You do not need to submit your homework
  - **Do not send me an e-mail** with your work (see below)
- It is not graded
- You can also skip it, entirely
  - Bad idea, since you will have a hard time in doing the assignments, midterm and final
- In each lecture, there will be 5-10 minutes time discussing issues for the homework of the previous lecture

Lab lecture

- Scheduled on Wednesday
  - No labs during the first week
  - There will be announcements
- It takes place on a lecture room (not in the labs)
- You need to have a laptop with the VM installed in order to follow the steps
- Typical lectures are already lab-based; what’s new?
  - Lab lectures are based on concepts we are already familiar
  - More complicated demos

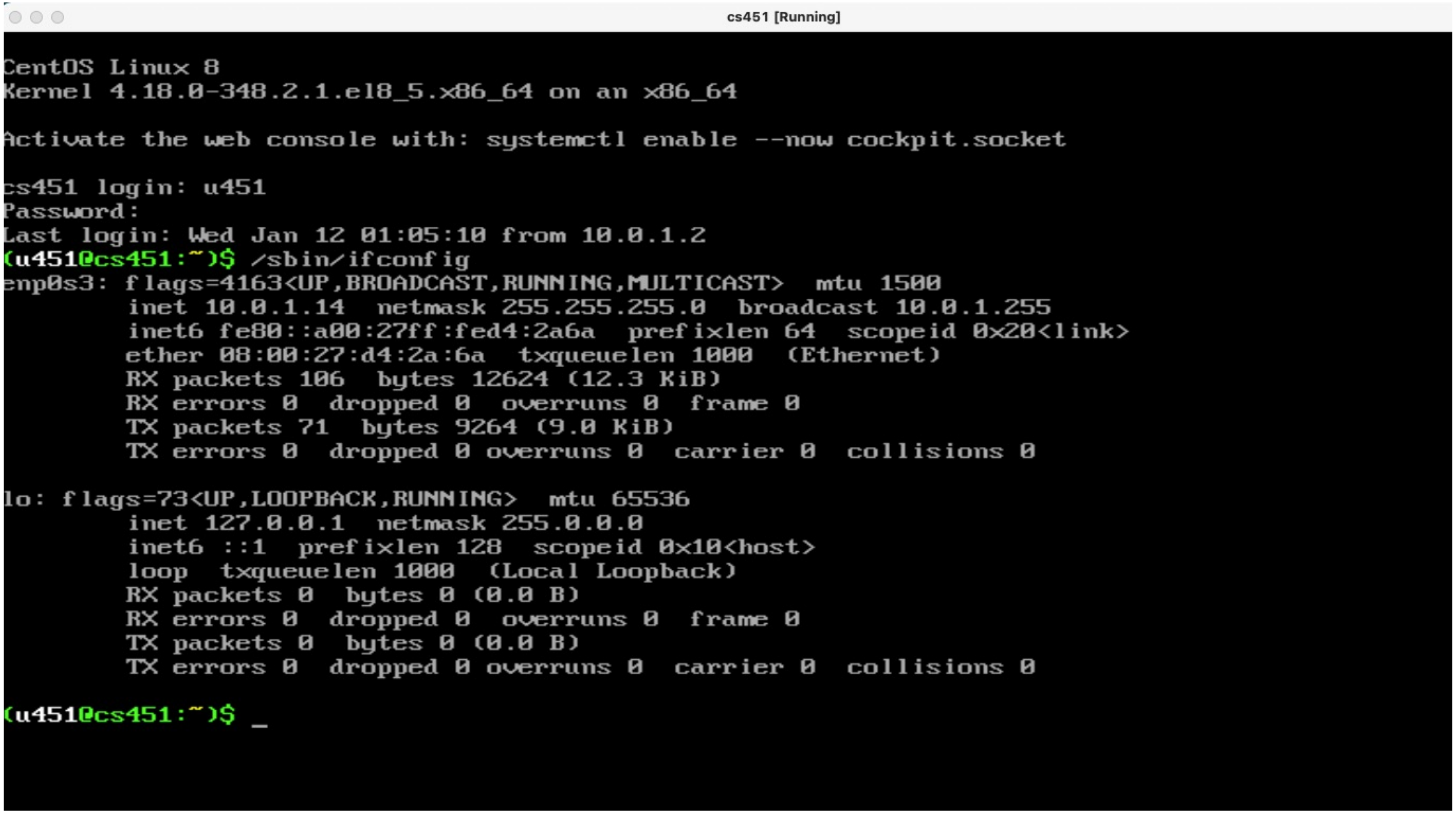
Virtual Machine

- You can get the VM image using the link below
  - https://drive.google.com/open?id=19LZjmbF63BTEqHwRSrG2fmHHYHpaQL5c
- It is a CentOS 8 Linux based system
  - With some required packages pre-installed
- You can build your own VM
  - Try to use CentOS 8,Ubuntu 16, or Debian, so that I can help you
  - My personal VM is Debian GNU/Linux 11 (bullseye)

How to use the VM

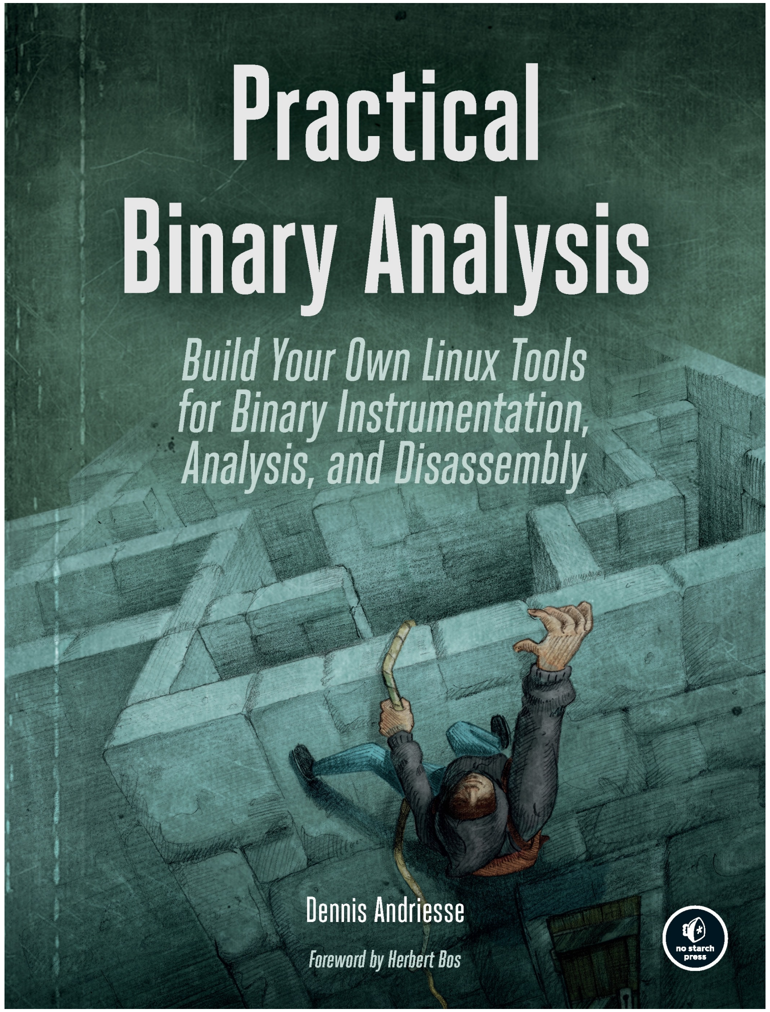
- There is a user ‘u451’ on the system
- The installation is very minimal (no GUI)
- A convenient way is to run the VM in your VirtualBox
  - Use a recent version (e.g., 6.1 or later)
- And connect to the VM using ssh from the host

Getting the IP address



```
CentOS Linux 8
Kernel 4.18.0-348.2.1.el8_5.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

cs451 login: u451
Password:
Last login: Wed Jan 12 01:05:10 from 10.0.1.2
(u451@cs451:~)$ /sbin/ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.14  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::a00:27ff:fed4:2a6a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d4:2a:6a  txqueuelen 1000  (Ethernet)
        RX packets 106  bytes 12624 (12.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 71  bytes 9264 (9.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

(u451@cs451:~)$ _
```

Book


Practical Binary Analysis
Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly
Dennis Andriesse
Foreword by Herbert Bos
no starch press

Logistics

- Your grade is based on the following distribution
  - Final exam – 50%
  - Midterm exam – 30%
  - Assignments – 20%
- You need to score at least 5 to pass the course

Communication

- Microsoft Teams
  - For communication and virtual lectures
  - Lecture material and timetable
- E-mail/Microsoft Teams
  - Assignment submission

Homework

- Download the VM and try to run it in your system with VirtualBox
  - https://drive.google.com/open?id=19LZjmbF63BTEqHwRSrG2fmHHYHpaQL5c
  - VirtualBox: https://www.virtualbox.org
- The VM is around 11GB
  - Have patient, use a good connection
- **Use chat in Microsoft Teams** to report problems
  - So that people with the same problem can receive help faster