Universität Zürich[UZH]

# 0690 - Communication Systems Lab Course

*Exercise 6*

October 25, 2022

Assistants: Christian Killer, Jan von der Assen, Dr. Alberto Huertas

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

# 1 Introduction

In this lab session, you will solve a set of Capture The Flag (CTF) like interactive challenges against a Linux host. Although some CTF challenges are built to be extremely difficult and require highly sophisticated solutions and deep knowledge to solve, the fifteen levels you will encounter in this exercise allow you to start from zero, not requiring much more than the ability to read documentation. In general, the goal of this lab session is not only focusing on security vulnerabilities, but to learn and apply the standard Unix shell tool set to solve problems.

## 1.1 man pages

Unix systems normally come equipped with documentation for all software and commands in a standardized format, called manual pages or "manpages" in short. During the exercise, commands are listed which can be used for achieving the current objective. They are given in the traditional Unix notation command(section), where the section indicates on which manpage section the command is detailed, as there might be several manpages with the same name, or detailing different aspects (eg. configuration and invocation) of some commands. You can look up how to use a specific command by reading its manpage, using the `man` command:

```
man [section] command
```

For example:

```
man 1 ls
```

Note that the section part is optional if there is only one section containing a manpage under that name. Note that not all the given commands are always necessary to solve each level.

## 1.2 Command piping

As detailed in pipe(7) and bash(1), commands can be chained one after another in a way that allows to use the standard output of the first command as the standard input of the second command. This is done by concatenating commands using the pipe symbol (|). In the following example showing a console session, the output of the echo command (which prints text on the standard output) is used as input for the bc command (which evaluates mathematical expressions):

```
level00@wargames: $ echo 1+1 | bc
2
```

## 1.3 The awk command

Several levels mention `awk` as a command to use. It is a command that works as a small scripting language designed to quickly modify text files and run common tasks for which no simple command exists. The reference to `awk` is given as a hint that if you are somewhat familiar with its syntax, you might find it easier to solve the task.

However, all tasks in this exercise can be solved without using `awk`, so don't spend too much time trying to learn how it works.

## 2 Structure of this Exercise

The exercise consists of fifteen levels, each posing a challenge which can be solved using various standard Unix command line tools. The goal of each level is to find a password to enter the next level. The challenges are set up in a virtual machine on your PC, which you can access over SSH. **Each level has an SSH login with the password given from solving the previous level.** After connecting over SSH, you are given a restricted bash shell, on which only a basic subset of the usual GNU/Linux command set can be used. For example, you are not given any real text editor. Even so, the available tools are enough to solve each level.

### 2.1 Initial Setup (for Linux)

Install **VirtualBox** and import the virtual machine used in this exercise with the following commands. This can be done using the Virtualbox GUI or the `vboxmanage` command.

```
sudo apt-get update; sudo apt-get install virtualbox; sudo apt-get install curl
curl -O 192.168.3.1/wargames.ova
vboxmanage import /path/to/image.ova --options keepallmacs
```

The virtual machine needs its own corresponding subnet on your host PC for connecting to it. Add a virtual network with the following two commands:

```
vboxmanage hostonlyif create
vboxmanage hostonlyif ipconfig vboxnet0 --ip 10.10.0.1 --netmask 255.255.255.0
```

After that, `ip a` should list the new interface with correct IP and subnet settings. Find the name of the imported VM and start it:

```
vboxmanage list vms
vboxmanage startvm --type headless <machine>
```

## 3 Levels

For each level, write down the command (or, all commands) you used to get to the password and send an e-mail with the commands for all levels to vonderassens@ifi.uzh.ch, killer@ifi.uzh.ch. You don't need to include all the commands you used for trying to get to the solution, a one-liner which works after you've figured out how to solve a level is sufficient.

### 3.1 Level 00

`ssh level00@10.10.0.2`
**Password**: comsys

**Goal**: Find the secret key, which is the password to the next level.
**Useful commands**: `ls(1)`, `cat(1)`

## 3.2 Level 01

```
ssh level01@10.10.0.2
```

**Goal**: Find the secret key, which is the password to the next level.
**Useful commands**: `ls(1)`, `getopt(1)`, `cat(1)`, `awk(1)`

## 3.3 Level 02

```
ssh level02@10.10.0.2
```

**Goal**: The number of lines in data.txt is the password to the next level.
**Useful commands**: `ls(1)`, `cat(1)`, `wc(1)`, `awk(1)`

## 3.4 Level 03

```
ssh level03@10.10.0.2
```

**Goal**: The number of bytes in both data files minus total number of lines is the password to the next level.
**Useful commands**: `echo`, `cat(1)`, `wc(1)`, `bc(1)`, `awk(1)`

## 3.5 Level 04

```
ssh level04@10.10.0.2
```

**Goal**: The ”password” program in the home folder asks for the password to the next level. However, the program is badly written and contains the password in plaintext. You can run the program with `./password`
**Useful commands**: `ls(1)`, `file(1)`, `strings(1)`, `hd(1)`

## 3.6 Level 05

```
ssh level05@10.10.0.2
```

**Goal**: Find the true password, which is unique amongst the others.
**Useful commands**: `ls(1)`, `cat(1)`, `sort(1)`, `uniq(1)`, `awk(1)`

## 3.7 Level 06

```
ssh level06@10.10.0.2
```

**Goal**: josuah gave the key to his neighbour.
**Useful commands**: `ls(1)`, `cat(1)`, `cut(1)`, `grep(1)`, `wc(1)`, `awk(1`

## 3.8 Level 07

```
ssh level07@10.10.0.2
```

**Goal**: The password is in a single file somewhere in the data directory tree.
**Useful commands**: `cat(1)`, `find(1)`

## 3.9 Level 08

```
ssh level08@10.10.0.2
```

**Goal**: The password is in a file in the data directory tree which is owned by the root user.
**Useful commands**: `cat(1)`, `find(1)`

## 3.10 Level 09

```
ssh level09@10.10.0.2
```

**Goal**: The two files differ by a few lines. Somewhere there is a single line missing in one of the files, which is the password. **Useful commands**: `ls(1)`, `cat(1)`, `diff(1)`

## 3.11 Level 10

```
ssh level10@10.10.0.2
```

**Goal**: A running process called ”fileguard” has the file containing the password open.
**Useful commands**: `ls(1)`, `cat(1)`, `ps(1)`, `lsof(8)`, `grep(1)`

## 3.12 Level 11

```
ssh level11@10.10.0.2
```

**Goal**: The password has been encoded with the ROT18 cipher, which is a combination of ROT13 and ROT5. ROT13 means that every letter has been replaced by the one 13 places further in the alphabet (wrapping around), and ROT5 means 5 was added to every number (modulo 10).
**Useful commands**: `ls(1)`, `cat(1)`, `tr(1)`

## 3.13 Level 12

`ssh level12@10.10.0.2`

**Goal**: The password file has been encoded and compressed several times. Find out the encoding and compression commands to recover the password.
**Useful commands**: `ls(1), cat(1), file(1), ???`

## 3.14 Level 13

`ssh level13@10.10.0.2`

**Goal**: There is a daemon listening on localhost port 1234 (TCP). It will tell you the password to the next level if you send it the command ”password”
**Useful commands**: `ls(1), cat(1), nc(1), echo(1)`

## 3.15 Level 14

`ssh level14@10.10.0.2`

**Goal**: Same as in the previous level, but this time the daemon listens on some unknown port in range 10000-20000, which you need to figure out yourself. **Useful commands**: `ls(1), cat(1), nc(1), echo(1), netstat(8)`

## 3.16 Level 15

**Goal**: This level does not contain any challenge, it serves as the final level for which a password is obtainable. If you are able to login to level 15, you’re finished!

**Do not forget to send your answers to vonderassen and killer@ifi.uzh.ch.**