



Welcome to

## 0. Introduction

# KEA Kompetence OB2 Software Security

Henrik Kramselund he/him han/ham hlk@zecurity.com @kramse  

Slides are available as PDF, kramse@Github  
0-Introduction-software-security.tex in the repo security-courses

# Contact information



- Henrik Kramselund, he/him internet samurai mostly networks and infosec
- Network and security consultant Zencurity, teach at KEA and activist
- Master from the Computer Science Department at the University of Copenhagen, DIKU
- Email: [hlk@zencurity.dk](mailto:hlk@zencurity.dk)      Mobile: +45 2026 6000

You are welcome to drop me an email

# Goals for today



Todays goals:

- Welcome, course goals and expectations, get to know eachother
- Create a good starting point for learning
- Learn to find resources, files and programs/libraries
- Concrete Expectations
- Prepare tools for the exercises, Prepare Virtual Machines

Photo by Thomas Galler on Unsplash

# Plan for today



- Introduce lecturer and students
- Expectations for this course
- Literature list walkthrough
- Prepare tools for the exercises
- Kali and Debian Linux introduction

## Exercises

- Kali Linux installation
- Debian Linux installation

Linux is a toolbox we will use and participants will use virtual machines

# Exercises



## Hardware

Since we are going to be doing exercises, each team will need two virtual machines.

The following are two recommended systems:

- One based on Debian, running software servers and web applications
- One based on Kali Linux, running attacks against software
- Setup instructions and help <https://github.com/kramse/kramse-labs>

Linux is a toolbox we will use and participants will use virtual machines

## Time schedule



17:00 - 18:15 Introduction and basics

18:15 - 18:45 – 30min break

18:45 - 19:30 – 45min Teaching

19:30 - 19:45 – 15min break

19:45 - 20:30 – 45min Teaching

This will be the basic plan for each evening

# Course Materials



This material is in multiple parts:

- Slide shows - presentation - this file
- Exercises - PDF which is updated along the way

Books listed in the lecture plan

Additional resources from the internet

Note: the presentation slides are not a substitute for reading the books, papers and doing exercises, many details are not shown

# Fronter Platform



The screenshot shows a web browser window for the Fronter platform. The URL is <https://kea-fronter.itslearning.com/ContentArea/ContentArea.aspx?LocationID=3417&LocationType=1>. The page displays a news feed with two items from 'Henrik Lund Kramshøj'. The top item is a 'Dei en opdatering' (Share update) and the bottom item is a 'Lige nu' (Now) post about a lesson plan. A tooltip box is overlaid on the screen, titled 'Du lærer hurtig'. It contains the text 'Lad os tilføje nogle personer.' and a list of actions with checkboxes:

- Giv rummet et lejlighedsigt navn.
- Inviter dine studerende
- Bød dine studerende velkommen
- Lav en plan
- Giv dine studerende en opgave
- Start en gruppesamtale med dine studerende.

At the bottom of the tooltip, there is a link 'Vis aldrig igen' (Never again).

We will use fronter a lot, both for sharing educational materials and news during the course.

You will also be asked to turn in deliverables through fronter

<https://kea-fronter.itslearning.com/>

If you haven't received login yet, let us know

# Overview Diploma in IT-security



Afgangsprojektet (15 ECTS)	
Der udvikles løbende nye valgfag til Diplom i it-sikkerhed. Disse vil løbende blive beskrevet i en allonge (bilag 2) til studieordningen.	
Sikkerhed i it-governance (it-sikkerhedsledelse) (5 ECTS)	Systemsikkerhed (10 ECTS)
Videregående sikkerhed i it-governance (Videregående sikkerhedsledelse) (5 ECTS)	
Softwaresikkerhed (10 ECTS)	
Netværks- og kommunikationssikkerhed (10 ECTS)	

# Course Data



## **Course: Software Security Ob 2 Softwaresikkerhed (10 ECTS)**

Teaching dates - fall 2021 tuesdays and thursdays 17:00 - 20:30

31/8, 2/9, 7/9, 9/9, 14/9, 16/9, 21/9, 23/9, 28/9, 30/10, 5/10, 7/10, 12/10, 14/10

Exam: 26/10 2021

Photo by Paweł Janiak on Unsplash

## Deliverables and Exam



- Exam
- Individual: Oral based on curriculum
- Graded (7 scale)
- Draw a question with no preparation. Question covers a topic
- Try to discuss the topic, and use practical examples
- Exam is 30 minutes in total, including pulling the question and grading
- Count on being able to present talk for about 10 minutes
- Prepare material (keywords, examples, exercises, wireshark captures) for different topics so that you can use it to help you at the exam
- Deliverables:
- 1 Mandatory assignments
- Both mandatory assignments are required in order to be entitled to the exam.

# Course Description



From: STUDIEORDNING Diplomuddannelse i it-sikkerhed August 2018

Indhold: Modulet fokuserer på sikkerhedsperspektivet i software, blandt andet programkvalitet og fejlhåndterings samt datahåndterings betydning for en software arkitekturs sårbarheder. Elementet introducerer også til forskellige designprincipper, herunder "security by design".

## Viden

Den studerende har viden om: Hvilken betydning programkvalitet har for it-sikkerhed ift.:

- Trusler mod software
- Kriterier for programkvalitet
- Fejlhåndtering i programmer
- Forståelse for security design principles, herunder:
  - Security by design
  - Privacy by design

# Færdigheder



## Færdigheder

Den studerende kan:

Tage højde for sikkerhedsaspekter ved at:

- Programmere håndtering af forventede og uventede fejl
- Definere lovlige og ikke-lovlige input data, bl.a. til test
- Bruge et API og/eller standard biblioteker
- Opdage og forhindre sårbarheder i programkoder
- Sikkerhedsvurdere et givet software arkitektur

# Kompetencer



## Kompetencer

Den studerende kan:

- Håndtere risikovurdering af programkode for sårbarheder.
- Håndtere udvalgte krypteringstiltag

Final word is the Studieordning which can be downloaded from

<https://kompetence.kea.dk/uddannelser/it-digitalt/diplom-i-it-sikkerhed>

Studieordning\_for\_Diplomuddannelsen\_i\_IT-sikkerhed\_Aug\_2018.pdf

# Expectations alignment



In groups of 2 students, brainstorm for 10 minutes on what topics you would like to have in this course

Use 5 minutes more on Agreeing on 5 topics and prioritize these 5 topics

PS We will from time to time have exercises, groups dont need to be the same each time.

## Prerequisites



This course includes exercises and getting the most of the course requires the participants to carry out these practical exercises

We will use Linux for some exercises but previous Linux and Unix knowledge is not needed

It is recommended to use virtual machines for the exercises

Security and most internet related security work has the following requirements:

- Network experience
- Server experience
- TCP/IP principles - often in more detail than a common user
- Programming is an advantage, for automating things
- Some Linux and Unix knowledge is in my opinion a **necessary skill**
  - too many new tools to ignore, and lots found at sites like Github and Open Source written for Linux

# Goals and plans

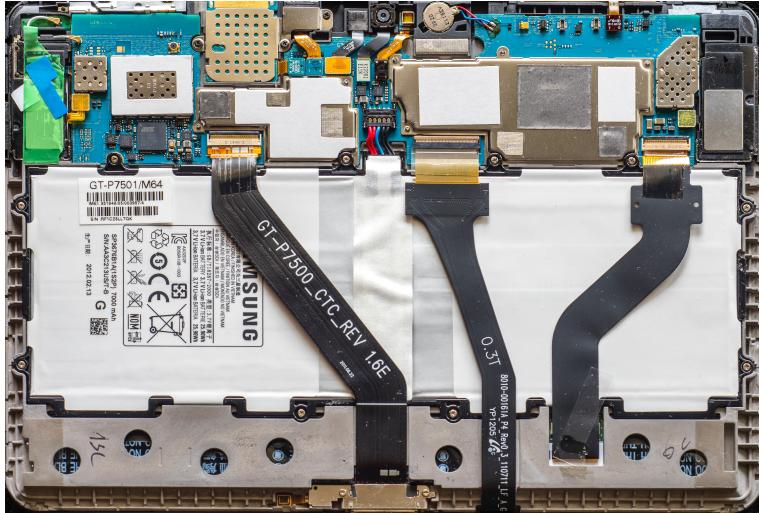


“A goal without a plan is just a wish.”  
Antoine de Saint-Exupéry

I want this course to

- Include everything required by studieordningen
- Be practical – you can do something useful
- Kickstart your journey into this subject  
Getting the best books and papers
- Present a lot of useful sources, tools
- Prepare you for production use of the knowledge

# What is Infrastructure – Software



- Enterprises today have a lot of computing systems supporting the business needs
- These are very diverse and often discrete systems

Photo by Alexander Schimmeck on Unsplash

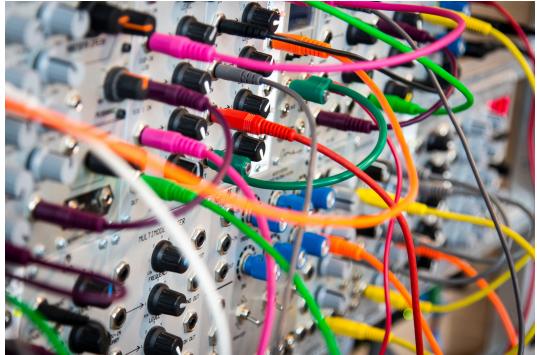
# Business Challenges



- Accumulation of software
- Legacy systems
- Partners
- Various types of data
- Employee churn, replacement

Photo by Adam Bignell on Unsplash

# Software Challenges



- Complexity
- Various languages
- Various programming paradigms, client server, monolith, Model View Controller
- Conflicting data types and available structures
- Steam train vs electric train

Photo by John Barkiple on Unsplash

# Developers Challenges



- Work in teams across organisation - and partners, vendors, sub-contractors
- Work with legacy systems, old technology
- Learn new Technologies

Photo by Kelly Sikkema on Unsplash

# Integration Challenges



- Enable communication between components
- Need mediator, interpreter, translator
- Recognize standard patterns

Photo by Thomas Drouault on Unsplash

## Course overview



We will now go through a little from the Table of Contents in the books.

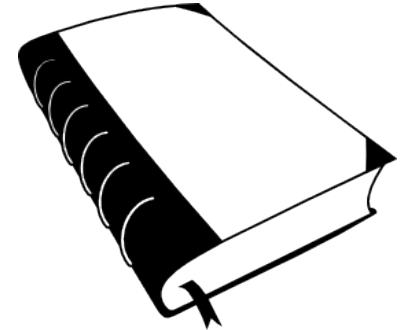
and the *Lektionsplan*

<https://zencurity.gitbook.io/kea-it-sikkerhed/softwaresikkerhed/lektionsplan>

# Primary literature



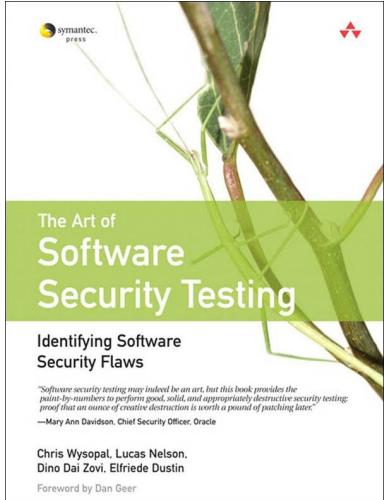
Primary literature:



Free graphics by Lumen Design Studio

- *The Art of Software Security Testing Identifying Software Security Flaws*, Chris Wysopal, ISBN: 9780321304865, AoST or the Green Book
- *Web Application Security*, Andrew Hoffman, 2020, ISBN: 9781492053118 called WAS Available in PDF if you give them an email address
  - <https://www.nginx.com/resources/library/web-application-security/>
- Pwning OWASP Juice Shop Official companion guide to the OWASP Juice Shop Can be found online for free, but recommend buying the PDF from <https://leanpub.com/juice-shop> - suggested price USD 5.99

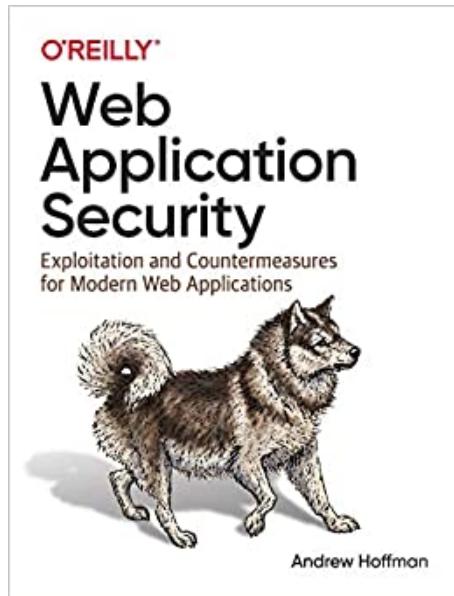
# Book: The Art of Software Security Testing



*The Art of Software Security Testing Identifying Software Security Flaws*

Chris Wysopal ISBN: 9780321304865, AoST or the Green Book

# Web Application Security



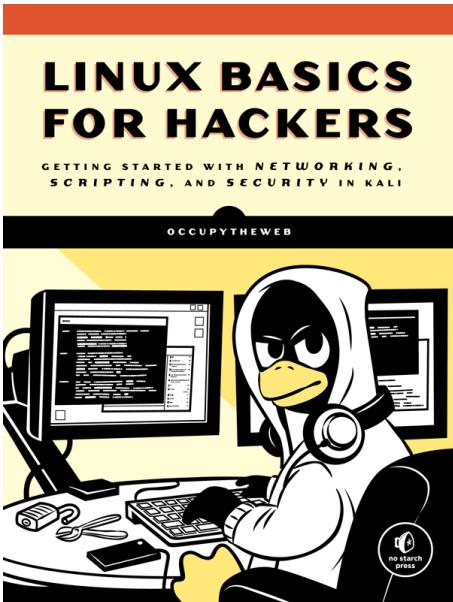
*Web Application Security*, Andrew Hoffmann, 2020, ISBN: 9781492053118 called WAS

## Supporting literature



- *Linux Basics for Hackers Getting Started with Networking, Scripting, and Security in Kali*. OccupyTheWeb, December 2018, 248 pp. ISBN-13: 978-1-59327-855-7 - shortened LBfH
- *Gray Hat Hacking: The Ethical Hacker's Handbook*, fifth edition Allen Harper ISBN: 9781260108415
- *Kali Linux Revealed Mastering the Penetration Testing Distribution* Raphael Hertzog, Jim O'Gorman - shortened KLR
- *The Debian Administrator's Handbook*, Raphaël Hertzog and Roland Mas  
<https://debian-handbook.info/> - shortened DEB
- Very optional and older but recommended *24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them*, Michael Howard, David LeBlanc, John Viega, ISBN: 9780071626750, 2010 The McGraw-Hill Companies, named 24-deadly below

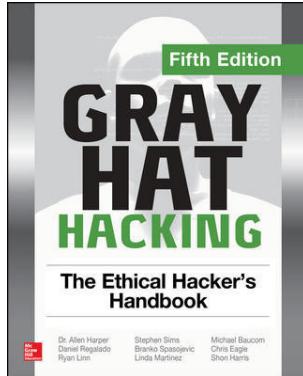
## Book: Linux Basics for Hackers (LBhf)



*Linux Basics for Hackers Getting Started with Networking, Scripting, and Security in Kali* by OccupyTheWeb December 2018, 248 pp. ISBN-13: 9781593278557

<https://nostarch.com/linuxbasicsforhackers> Not curriculum but explains how to use Linux

# Book: Gray Hat Hacking (Grayhat)



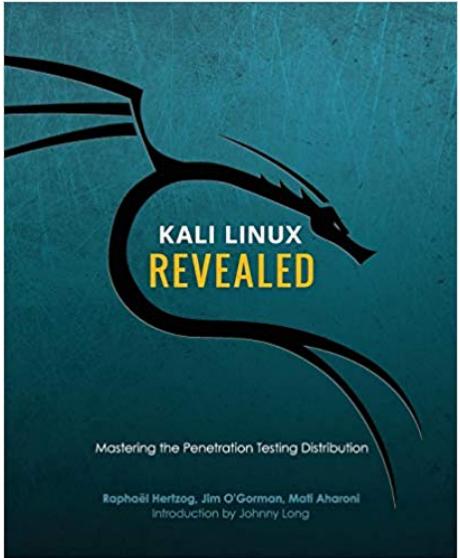
*Gray Hat Hacking: The Ethical Hacker's Handbook*, fifth edition

Authors: Allen Harper, Daniel Regalado, Ryan Linn, Stephen Sims, Branko Spasojevic, Linda Martinez, Michael Baucom, Chris Eagle, Shon Harris, Published: May 18th 2018, Paperback ISBN: 978-1-260-10841-5 640 pp.

<https://www.mhprofessional.com/9781260108415-usa-gray-hat-hacking-the-ethical-hackers-handbook-fifth-edition-group>

Not curriculum but has some programming introduction which are very useful. Also this book is used in the KEA Network Pentest course

# Book: Kali Linux Revealed (KLR)



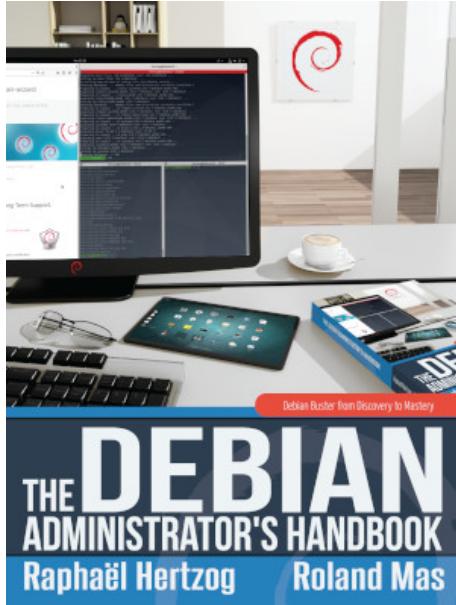
*Kali Linux Revealed Mastering the Penetration Testing Distribution*

Current link, may be updated:

<https://kali.training/courses/kali-linux-revealed/>

Not curriculum but explains how to install Kali Linux

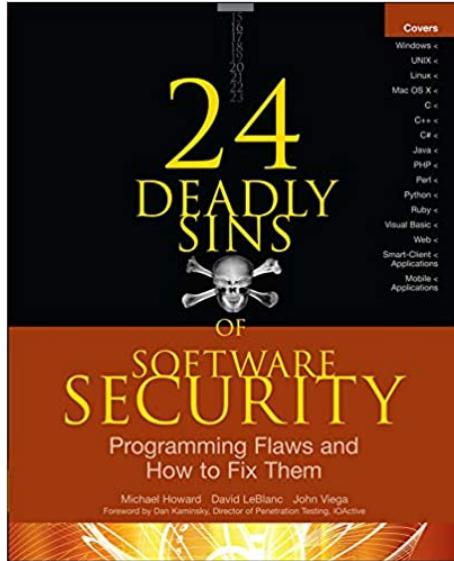
# Book: The Debian Administrator's Handbook (DEB)



*The Debian Administrator's Handbook*, Raphaël Hertzog and Roland Mas  
<https://debian-handbook.info/> - shortened DEB

Not curriculum but explains how to use Debian Linux

# 24 Deadly Sins of Software Security



*24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them*, Michael Howard, David LeBlanc, John Viega, ISBN: 9780071626750, 2010 The McGraw-Hill Companies, named 24-deadly below

Optional but recommended

# Exercise

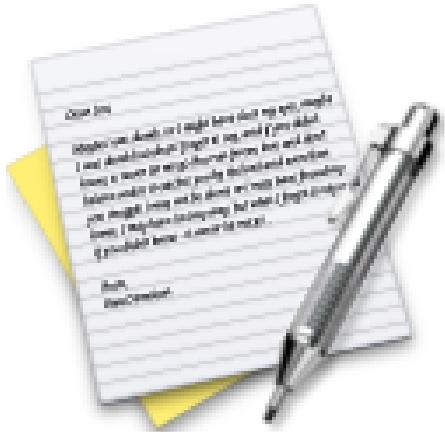


Now lets do the exercise

**Download Kali Linux Revealed (KLR) Book 10 min**

which is number **1** in the exercise PDF.

# Exercise

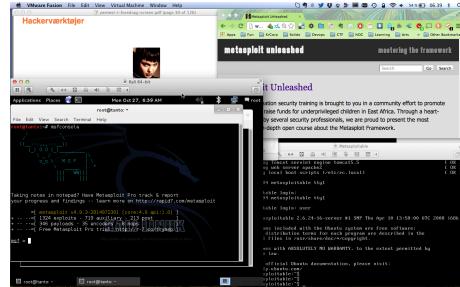


Now lets do the exercise

## Download Debian Administrator's Handbook (DEB) Book 10 min

which is number **2** in the exercise PDF.

# Hackerlab Setup



- Hardware: modern laptop CPU with virtualisation  
Dont forget to enable hardware virtualisation in the BIOS
- Virtualisation software: VMware, Virtual box, HyperV pick your poison
- Hackersoftware: Kali Virtual Machine amd64 64-bit <https://www.kali.org/>
- Linux server system: Debian amd64 64-bit <https://www.debian.org/>
- Setup instructions can be found at <https://github.com/kramse/kramse-labs>

It is enough if these VMs are pr team



## Technologies used in this course

The following tools and environments are examples that may be introduced in this course:

- Programming languages and frameworks Java, Python, regular expressions
- Development environments – choose your own IDE / Editor – I use **Atom**
- Networking and network protocols: TCP/IP, HTTP, DNS
- Web technologies and services: REST, API, HTML5, CSS, JavaScript
- Tools like cURL, Git and Github
- Optional - but demoed aggregated example platforms: Elastic stack, logstash, elasticsearch, kibana, grafana, Filebeat
- Cloud and virtualisation **Docker**

Kubernetes, Azure, AWS, microservices – are similar and can be added

This list is not complete or a promise

# OWASP Juice Shop Project



We will also use the OWASP Juice Shop Tool Project as a running example. This is an application which is modern AND designed to have security flaws.

Read more about this project at: [https://www.owasp.org/index.php/OWASP\\_Juice\\_Shop\\_Project](https://www.owasp.org/index.php/OWASP_Juice_Shop_Project)  
<https://github.com/bkimminich/juice-shop>

It is recommended to buy the Pwning OWASP Juice Shop Official companion guide to the OWASP Juice Shop from  
<https://leanpub.com/juice-shop> - suggested price USD 5.99

# Aftale om test af netværk



**Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 6 måneder straffes den, som ubrettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et anlæg til elektronisk databehandling.**

Hacking kan betyde:

- At man skal betale erstatning til personer eller virksomheder
- At man får konfiskeret sit udstyr af politiet
- At man, hvis man er over 15 år og bliver dømt for hacking, kan få en bøde - eller fængselsstraf i alvorlige tilfælde
- At man, hvis man er over 15 år og bliver dømt for hacking, får en plettet straffeattest. Det kan give problemer, hvis man skal finde et job eller hvis man skal rejse til visse lande, fx USA og Australien
- Frit efter: <http://www.stophacking.dk> lavet af Det Kriminalpræventive Råd
- Frygten for terror har forstærket ovenstående - så lad være!

## Mixed exercises



Then we will do a mixed bag of exercises to introduce technologies, find your current knowledge level with regards to:

- Linux
- Linux command line
- Git, Python and Ansible

Later we will return to them!

## Exercise CHAOS: Don't Panic – have fun learning



"It is said that despite its many glaring (and occasionally fatal) inaccuracies, the Hitchhiker's Guide to the Galaxy itself has outsold the Encyclopedia Galactica because it is slightly cheaper, and because it has the words 'DON'T PANIC' in large, friendly letters on the cover."

Hitchhiker's Guide to the Galaxy, Douglas Adams

# Your lab setup



- Go to GitHub, Find user Kramse, click through kramse-labs
- Look into the instructions for the Virtual Machine – Kali and Debian
- Get the lab instructions, from

<https://github.com/kramse/kramse-labs/>

Hint: you can install the Atom editor as a package using the Ansible tool, by checking out this repo and doing:

```
sudo apt install ansible git  
git clone https://github.com/kramse/kramse-labs.git  
cd work-station  
ansible-playbook -v 1-dependencies
```

# Exercise

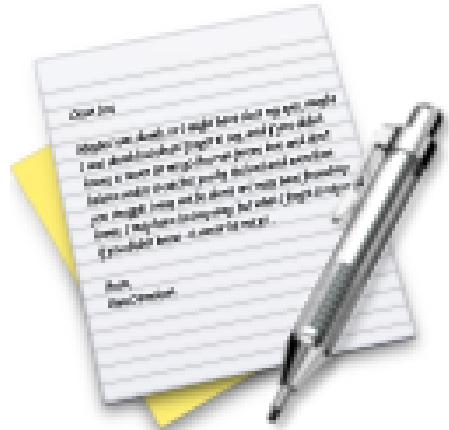


Now lets do the exercise

**Check your Kali VM, run Kali Linux 30 min**

which is number **3** in the exercise PDF.

# Exercise



Now lets do the exercise

## Check your Debian VM 10 min

which is number **4** in the exercise PDF.



# Command prompts in Unix

## Shells :

- sh - Bourne Shell
- bash - Bourne Again Shell, often the default in Linux
- ksh - Korn shell, original by David Korn, but often the public domain version used
- csh - C shell, syntax similar to C language
- Multiple others available, zsh is very popular

Windows have command.com, cmd.exe but PowerShell is more similar to the Unix shells

Used for scripting, automation and programs



## Command prompts

```
[hlk@fischer hlk]$ id  
uid=6000(hlk) gid=20(staff) groups=20(staff),  
0(wheel), 80(admin), 160(cvs)  
[hlk@fischer hlk]$ sudo -s  
[root@fischer hlk]#  
[root@fischer hlk]# id  
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon),  
20(staff), 80(admin)  
[root@fischer hlk]#
```

Note the difference between running as root and normal user. Usually books and instructions will use a prompt of hash mark # when the root user is assumed and dollar sign \$ when a normal user prompt.

# Command syntax



```
echo [-n] [string ...]
```

Commands are written like this:

- Always begin with the command to execute, like echo above
- Options typically short form with single dash -n
- or long options --version
- Some commands allow grouping options, tar -c -v -f becomes tar -cvf  
NOTE: some options require parameters, so tar -c -f filename.tar not equal to tar -fc filename.tar
- Optional options are in brackets [ ]
- Output can be saved using redirection, into new file/overwrite echo hello > file.txt or append echo hello >> file.txt
- Read from files wc -l file.txt or pipe output into input cat file.txt | wc -l  
wc is word count, and option l is count lines

# Unix Manual system



```
kommando [options] [argumenter]  
$ cal -j 2005
```

It is a book about a Spanish guy called Manual. You should read it. – Dilbert

Manual system in Unix is always there!

Key word search `man -k` see also `apropos`

Different sections, can be chosen

See `man crontab` the command vs the file format in section 5 `man 5 crontab`



# A manual page

## NAME

cal - displays a calendar

## SYNOPSIS

cal [-jy] [[month] year]

## DESCRIPTION

cal displays a simple calendar. If arguments are not specified, the current month is displayed. The options are as follows:

-j      Display julian dates (days one-based, numbered from January 1).

-y      Display a calendar for the current year.

The Gregorian Reformation is assumed to have occurred in 1752 on the 3rd of September. By this time, most countries had recognized the reformation (although a few did not recognize it until the early 1900's.) Ten days following that date were eliminated by the reformation, so the calendar for that month is a bit unusual.

# The year 1752



```
user@Projects:$ cal 1752
```

...

April							May							June								
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa		
														1	2		1	2	3	4	5	6
1	2	3	4				3	4	5	6	7	8	9	7	8	9	10	11	12	13		
5	6	7	8	9	10	11	10	11	12	13	14	15	16	14	15	16	17	18	19	20		
12	13	14	15	16	17	18	17	18	19	20	21	22	23	21	22	23	24	25	26	27		
19	20	21	22	23	24	25	24	25	26	27	28	29	30	28	29	30						
26	27	28	29	30			31															

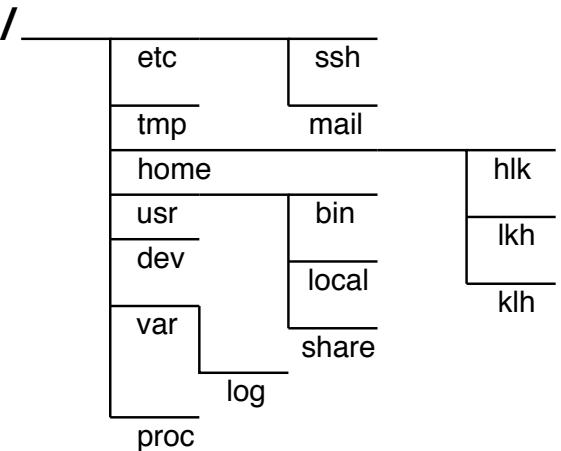
July							August							September							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
														1		1	2	14	15	16	
1	2	3	4				2	3	4	5	6	7	8	17	18	19	20	21	22	23	
5	6	7	8	9	10	11	9	10	11	12	13	14	15	24	25	26	27	28	29	30	
12	13	14	15	16	17	18	16	17	18	19	20	21	22								
19	20	21	22	23	24	25	23	24	25	26	27	28	29								
26	27	28	29	30	31		30	31													

...

# Linux configuration in /etc



- Command line is a requirement in the *studieordningen* ☺
- Linux and Unix uses a single virtual file system  
[https://en.wikipedia.org/wiki/Unix\\_filesystem](https://en.wikipedia.org/wiki/Unix_filesystem)
- No drive letters like the ones in MS-DOS and Microsoft Windows
- Everything starts at the root of the file system tree / - NOTE: *forward slash*
- One special directory is /etc/ and sub directories which usually contain a lot of configuration files



# Installing software in Debian – apt



## DESCRIPTION

apt provides a high-level commandline interface for the package management system. It is intended as an end user interface and enables some options better suited for interactive usage by default compared to more specialized APT tools like apt-get(8) and apt-cache(8).

### update (apt-get(8))

update is used to download package information from all configured sources. Other commands operate on this data to e.g. perform package upgrades or search in and display details about all packages available for installation.

### upgrade (apt-get(8))

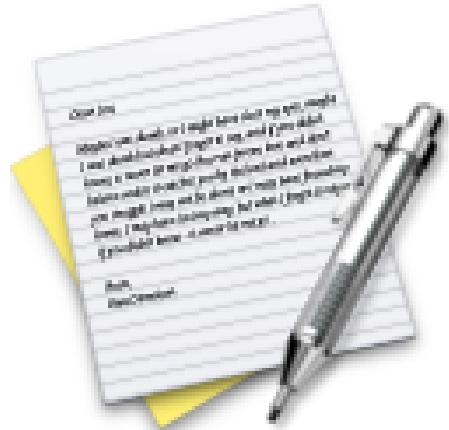
upgrade is used to install available upgrades of all packages currently installed on the system from the sources configured via sources.list(5). New packages will be installed if required to satisfy dependencies, but existing packages will never be removed. If an upgrade for a package requires the removal of an installed package the upgrade for this package isn't performed.

### full-upgrade (apt-get(8))

full-upgrade performs the function of upgrade but will remove currently installed packages if this is needed to upgrade the system as a whole.

- Install a program using apt, for example `apt install nmap`

# Exercise

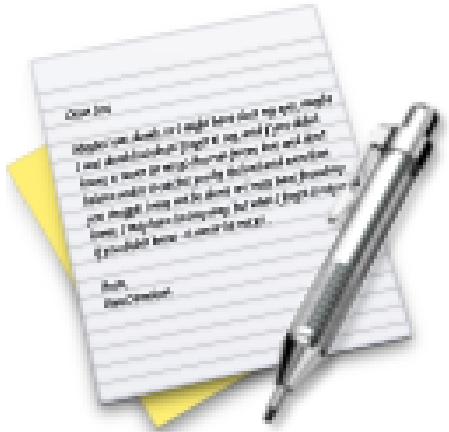


Now lets do the exercise

**Investigate /etc 10 min**

which is number **5** in the exercise PDF.

# Exercise

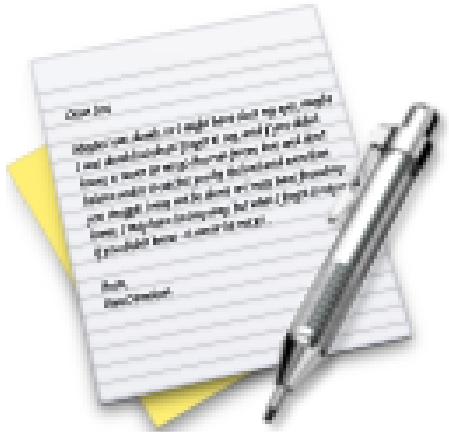


Now lets do the exercise

# Enable UFW firewall - 10 min

which is number **6** in the exercise PDF.

# Exercise



Now lets do the exercise

Git tutorials - 15min

which is number 7 in the exercise PDF.

## Exercise



Now lets do the exercise

# Run OWASP Juice Shop 45 min

which is number **8** in the exercise PDF.

## For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools

Buy the books!