Welcome to

0. Introduction

KEA Kompetence Computer Systems Security 2024

Henrik Kramselund he/him han/ham xhek@kea.dk @kramse

Slides are available as PDF, kramse@Github

0-Introduction-system-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: xhek@kea.dk Mobile: +45 2026 6000

You are welcome to drop me an email

Goals for part I

- 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 part I

- 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

Time schedule

- 10:00 Welcome – Get started, begin part 1
- 10:45 break
- 11:00 Part 1 continued
- 12:30 Lunch
- 13:00 Part 2 Lecture and exercises
- 16:30 Summary of the day, key points, conclusions
- 17:00 End of day

- We aim at maximum of 45min of lecture without breaks.
- There may be times where 45min lecture is followed by exercises. You can get started immediately or take a break.

This will be the basic plan for each day

Course Materials

This material is in multiple parts:

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

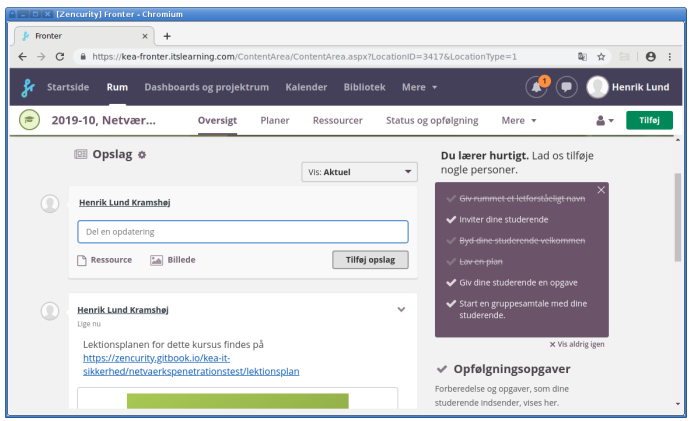
Books listed in the lecture plan Additional resources from the internet

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

A special thanks to William D. (Bill) Young Associate Professor of Instruction and Research Scientist, The University of Texas at Austin

When asked if I could borrow parts from his CS361 *Introduction to Computer Security* he graciously wrote: "You are welcome to use them freely. You can credit me at the beginning."

Fronter Platform



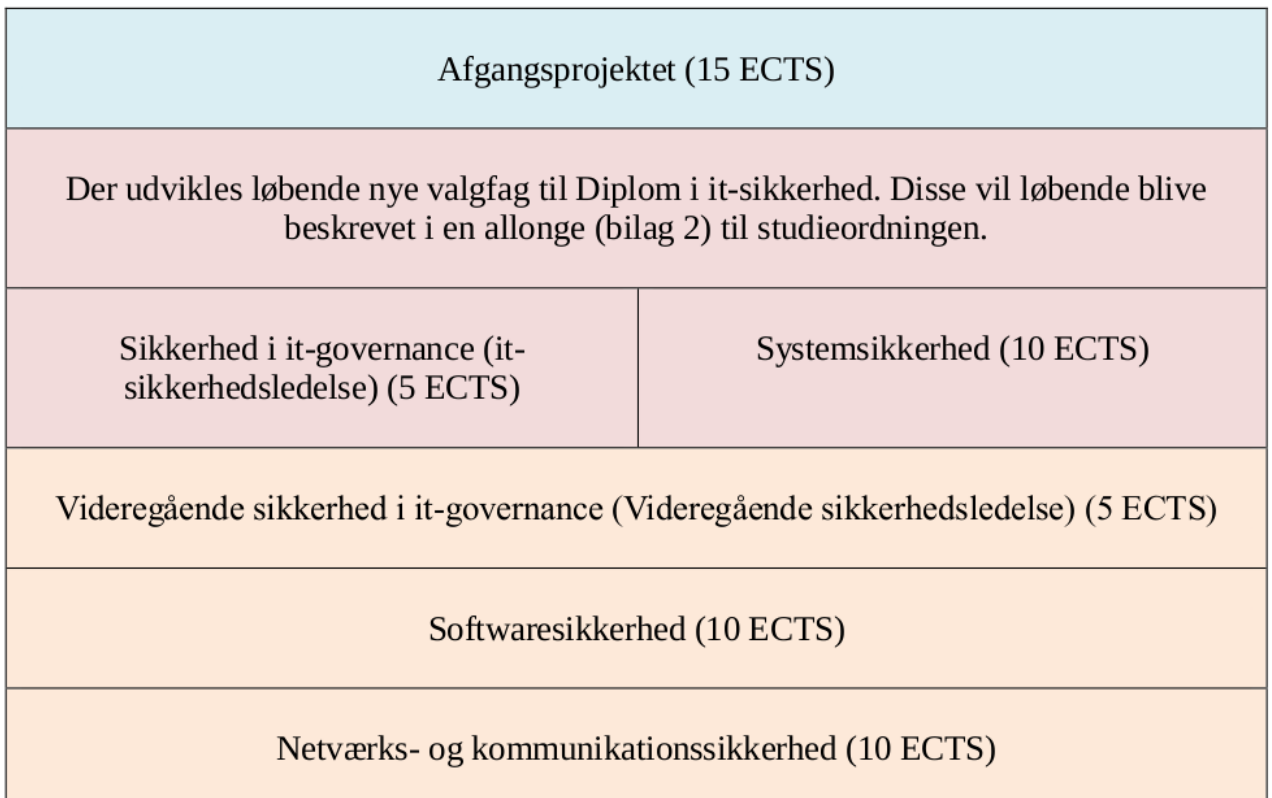
We will use fronter a lot, both for sharing educational materials and news during the course.

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



Course: VF 3 Computer Systems Security (10 ECTS)

Teaching dates: mondays 10:00 - 17:00
8/1, 15/1, 22/1, 29/1, 26/2, 4/3, 11/3 2024

Exam: 25/3 2024

Photo by Pawel 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:
- 2 Mandatory assignments
- Both mandatory assignments are required in order to be entitled to the exam.

Course Description

From: STUDIEORDNING Diplomuddannelse i it-sikkerhed Marts 2022
Indhold: Den studerende kan udføre, udvælge, anvende, og implementere praktiske tiltag til sikring af firmaets udstyr og har viden og færdigheder der supportere dette.

Viden

Den studerende har viden om:

- Generelle governance principper / sikkerhedsprocedurer
- Væsentlige forensic processer
- Relevante it-trusler
- Relevante sikkerhedsprincipper til systemsikkerhed
- OS roller ift. sikkerhedsovervejelser
- Sikkerhedsadministration i DBMS.

kea

Færdigheder

Den studerende kan:

- Udnytte modforanstaltninger til sikring af systemer
- Følge et benchmark til at sikre opsætning af enhederne
- Implementere systematisk logning og monitering af enheder
- Analysere logs for incidents og følge et revisionsspor
- Kan genoprette systemer efter en hændelse.

Kompetencer

Den studerende kan:

- håndtere enheder på command line-niveau
- håndtere værktøjer til at identificere og fjerne/afbøde forskellige typer af endpoint trusler
- håndtere udvælgelse, anvendelse og implementering af praktiske mekanismer til at forhindre, detektere og reagere over for specifikke it-sikkerhedsmæssige hændelser
- håndtere relevante 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

Form groups of 2-3 students

In groups of 2 students, brainstorm for 5 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

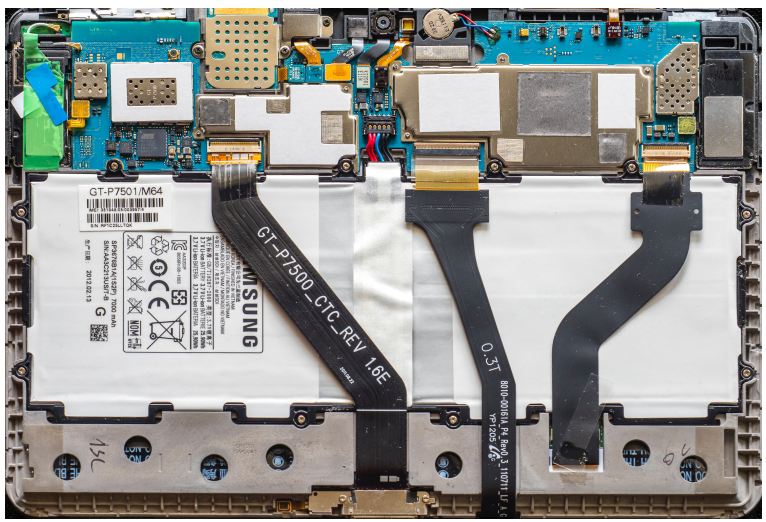
Goals and plans

"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 System Security
  Getting the best books and papers
- Present a lot of useful sources, tools
- Prepare you for production use of the knowledge

What is Infrastructure



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

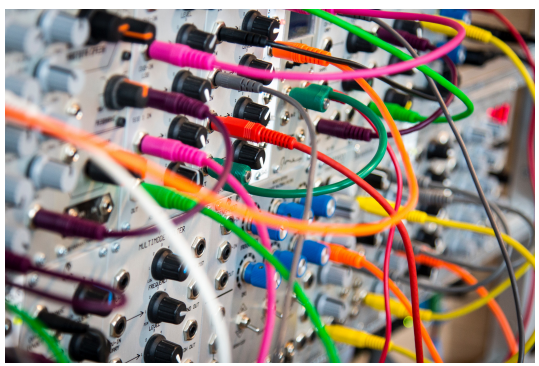
Photo by Alexander Schimmeck on Unsplash

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/systemsikkerhed/lektionsplan

Primary literature

Primary literature - not all chapters are part of the curriculum:

Free graphics by Lumen Design Studio

- *Mastering Linux Security and Hardening* (MLSH) - Third Edition, Donald A. Tevault, 2023 ISBN: 9781837630516
- *Defensive Security Handbook: Best Practices for Securing Infrastructure* (DSH), Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7
- *Forensics Discovery* (FD), Dan Farmer, Wietse Venema 2004, Addison-Wesley ISBN: 9780201634976
  This book is currently available for "free":
  http://fish2.com/security/ – also uploaded to Fronter

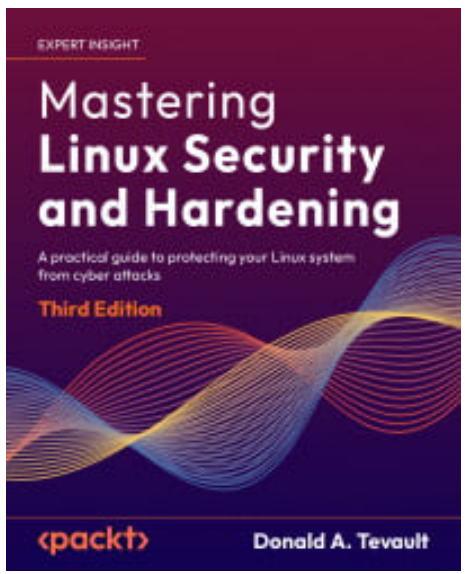
Other papers and resources will also be part of the curriculum!

Course overview

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

and the lecture plan in Fronter
(Source is also in Git https://github.com/kramse/kea-it-sikkerhed)

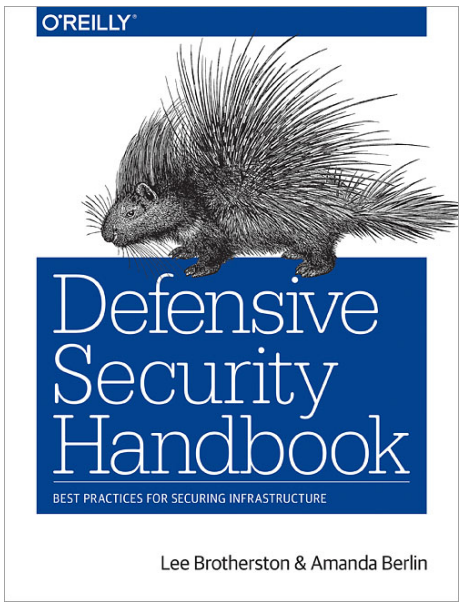
Book: Mastering Linux Security and Hardening (MLSH)



Mastering Linux Security and Hardening (MLSH), third edition, Donald A. Tevault, 2023 ISBN: 9781837630516

https://www.packtpub.com/product/mastering-linux-security-and-hardening-third-edition/9781837630516

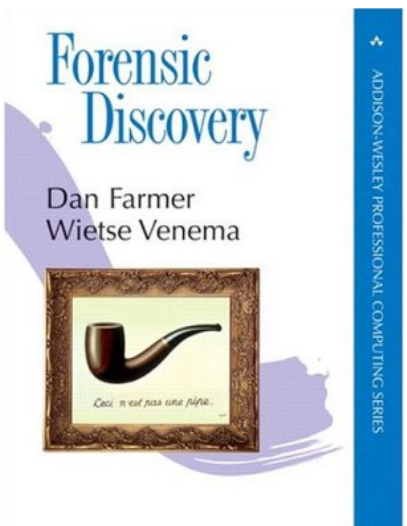
Book: Defensive Security Handbook (DSH)



Defensive Security Handbook: Best Practices for Securing Infrastructure, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7 – Note: 2nd edition scheduled for this year

https://www.oreilly.com/library/view/defensive-security-handbook/9781491960370/

Book: Forensics Discovery (FD)



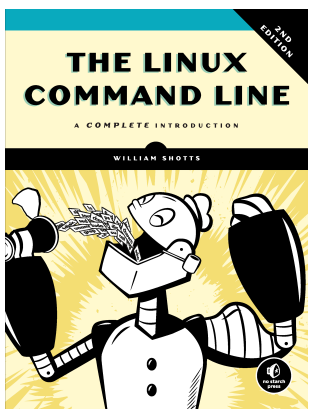
Forensics Discovery, Dan Farmer, Wietse Venema 2004, Addison-Wesley.

Can be found at http://www.porcupine.org/forensics/forensic-discovery/

Supporting literature books

- *The Linux Command Line: A Complete Introduction*, 2nd Edition
  by William Shotts
- *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
- *The Debian Administrator's Handbook*, Raphaël Hertzog and Roland Mas
  https://debian-handbook.info/ - shortened DEB
- *Kali Linux Revealed Mastering the Penetration Testing Distribution*
  Raphaël Hertzog, Jim O'Gorman - shortened KLR

Book: The Linux Command Line

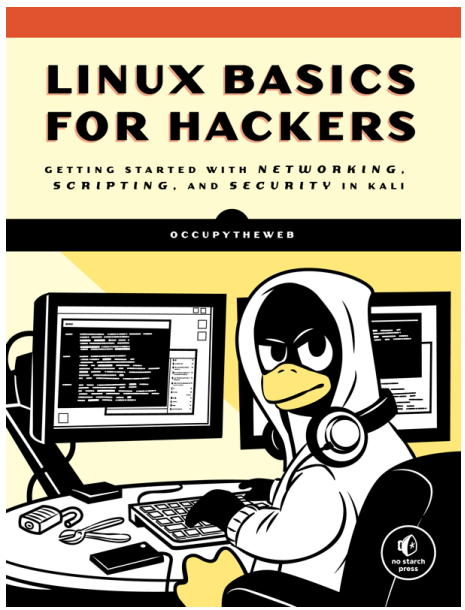


The Linux Command Line: A Complete Introduction , 2nd Edition by William Shotts

Print: https://nostarch.com/tlcl2
Download – internet edition https://sourceforge.net/projects/linuxcommand

Not curriculum but explains how to use Linux

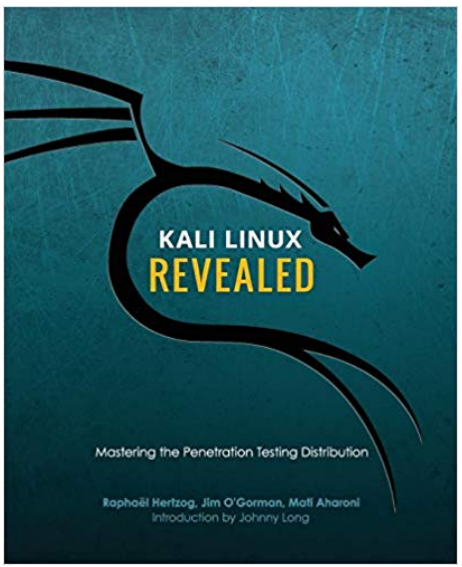
Book: Linux Basics for Hackers (LBfH)



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: Kali Linux Revealed (KLR)



Kali Linux Revealed Mastering the Penetration Testing Distribution

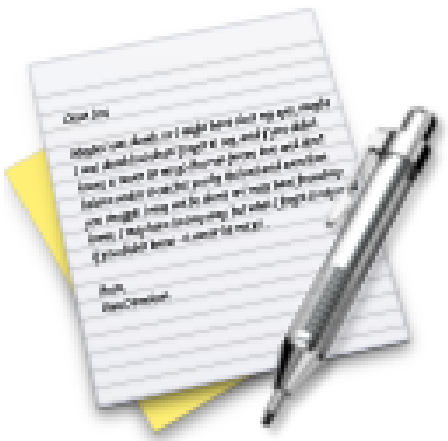
https://www.kali.org/download-kali-linux-revealed-book/
Not curriculum but explains how to install Kali Linux

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

Exercise

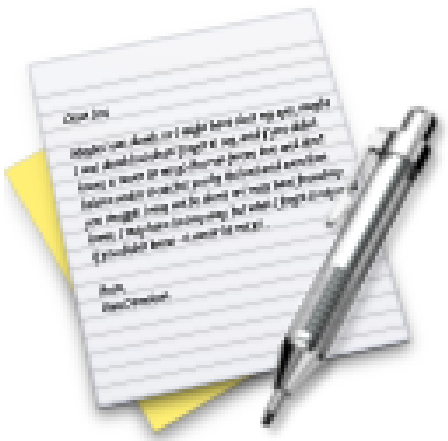


Now lets do the exercise

⚠ Download Kali Linux Revealed (KLR) Book 10min

which is number 1 in the exercise PDF.

Exercise



Now lets do the exercise

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

which is number 2 in the exercise PDF.

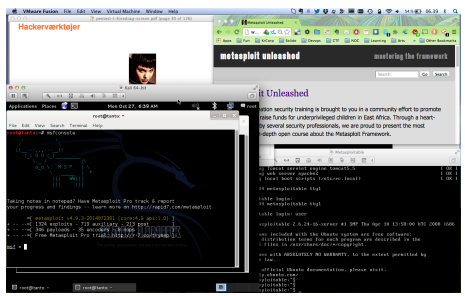
Technologies used in this course

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, Netflow
- Web technologies and services: REST, API, HTML5, CSS, JavaScript
- Tools like cURL, Zeek, Git and Github
- Aggregated example platforms: Elastic stack, logstash, elasticsearch, kibana, grafana, Filebeat

This list is not complete or a promise

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 10 Buster 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

Mixed exercises

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

- Linux as an operating system – user database in /etc/
- Linux command line
- Demo: Ansible
- Git, Python, scripting
- Demo: Elasticsearch – how to run a *service*

Note: today we will consider all these optional, we won't be able to do them all

Later we will return to them!

Aftale om test af netværk

Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 6 måneder straffes den, som uberettiget 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!

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 – Debian and Kali only
- Get the lab instructions, from

  https://github.com/kramse/kramse-labs/tree/master/suricatazeek

Yes, reusing instruction for the Suricata Zeek workshop - tested and working!

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.

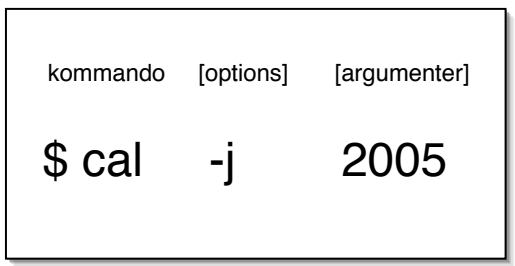
Command syntax

```
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 grouing 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



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 cur-
   rent 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 reforma-
tion (although a few did not recognize it until the early 1900's.)  Ten
days following that date were eliminated by the reformation, so the cal-
endar for that month is a bit unusual.
```

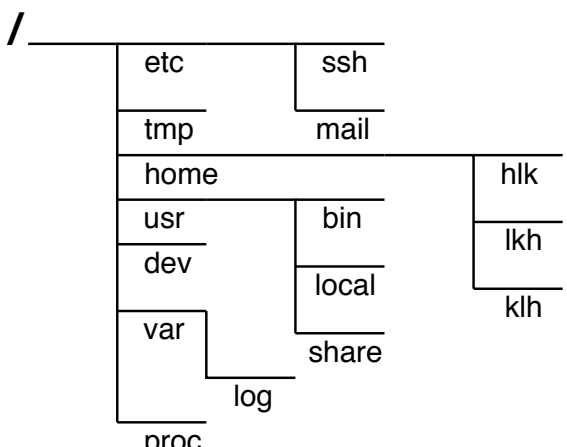
The year 1752

```
  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  3   4                 1  2      1  2  3  4  5  6
   5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
  12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
  19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
  26 27 28 29 30        24 25 26 27 28 29 30  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  2  3   4                 1        1  2 14 15 16
   5  6  7  8  9 10 11   2  3  4  5  6  7  8  17 18 19 20 21 22 23
  12 13 14 15 16 17 18   9 10 11 12 13 14 15  24 25 26 27 28 29 30
  19 20 21 22 23 24 25  16 17 18 19 20 21 22
  26 27 28 29 30 31     23 24 25 26 27 28 29
                        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
- 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`

Ansible

From my course materials:
Ansible is great for automating stuff, so by running the playbooks we can get a whole lot of programs installed, files modified - avoiding the Vi editor.

- Easy to read, even if you don't know much about YAML
- https://www.ansible.com/ and https://en.wikipedia.org/wiki/Ansible_(software)
- Great documentation
  https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html

Ansible Dependencies

- Ansible based on Python, only need Python installed
  https://www.python.org/
- Often you use Secure Shell for connecting to servers
  https://www.openssh.com/
- Easy to configure SSH keys, for secure connections

Ansible playbooks

Example playbook content, installing software using APT:

```
apt:
    name: "{{ packages }}"
    vars:
      packages:
        - nmap
        - curl
        - iperf
        ...
```

Running it:

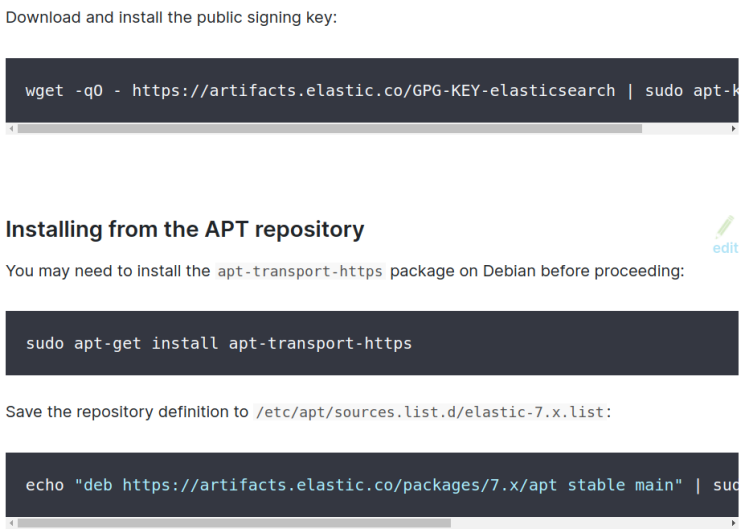
```
cd kramse-labs/suricatazeek
ansible-playbook -v 1-dependencies.yml 2-suricatazeek.yml 3-elasticstack.yml 4-configuration.yml
```

"YAML (a recursive acronym for "YAML Ain't Markup Language") is a human-readable data-serialization language." https://en.wikipedia.org/wiki/YAML

Python and YAML – Git

- We need to store configurations
- Run playbooks
- Problem: Remember what we did, when, how
- Solution: use git for the playbooks
- Not the only version control system, but my preferred one

Alternative



Download and install the public signing key:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-k
```

Installing from the APT repository

edit

You may need to install the `apt-transport-https` package on Debian before proceeding:

```
sudo apt-get install apt-transport-https
```

Save the repository definition to `/etc/apt/sources.list.d/elastic-7.x.list`:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | suc
```

My playbooks allow installation of a whole Elastic stack in less then 10 minutes,

compare to:

Getting started with the Elastic Stack

https://www.elastic.co/guide/en/elastic-stack-get-started/current/get-started-elastic-stack.html

Git getting started

Hints:
Browse the Git tutorials on https://git-scm.com/docs/gittutorial
and https://guides.github.com/activities/hello-world/

- What is git
- Terminology

Note: you don't need an account on Github to download/clone repositories, but having an acccount allows you to save repositories yourself and is recommended.

Demo: Ansible, Python, Git!

Running Git will allow you to clone repositories from others easily. This is a great way to get new software packages, and share your own.

Git is the name of the tool, and Github is a popular site for hosting git repositories.

- Go to https://github.com/kramse/kramse-labs
- Lets explore while we talk

Demo: output from running a git clone

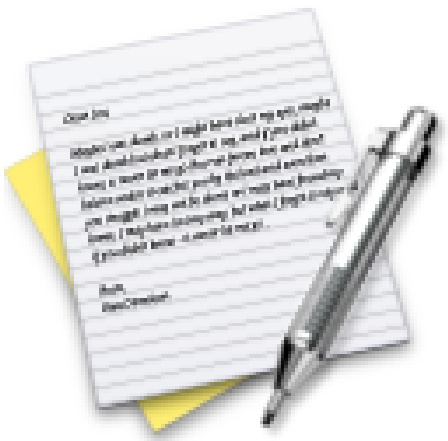
```
user@Projects:tt$ git clone https://github.com/kramse/kramse-labs.git
Cloning into 'kramse-labs'...
remote: Enumerating objects: 283, done.
remote: Total 283 (delta 0), reused 0 (delta 0), pack-reused 283
Receiving objects: 100% (283/283), 215.04 KiB | 898.00 KiB/s, done.
Resolving deltas: 100% (145/145), done.

user@Projects:tt$ cd kramse-labs/

user@Projects:kramse-labs$ ls
LICENSE  README.md  core-net-lab  lab-network  suricatazeek  work-station
user@Projects:kramse-labs$ git pull
Already up to date.
```

for reference at home later

Exercise



Now lets do the exercise

⚠ Check your Kali VM, run Kali Linux 30 min

which is number 3 in the exercise PDF.

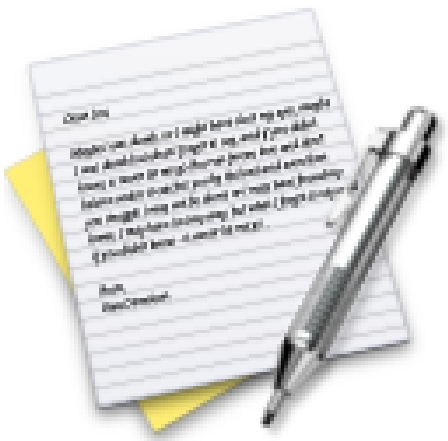
Exercise

Now lets do the exercise

⚠ Check your Debian VM 10min

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

Exercise

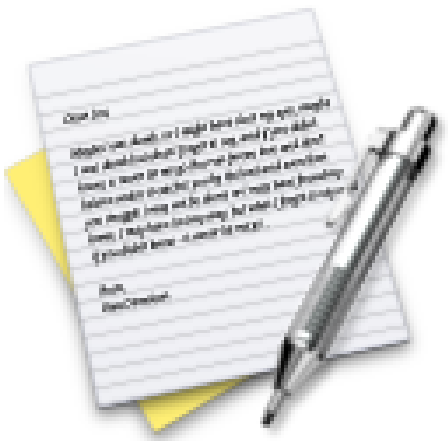


Now lets do the exercise

⚠ Investigate /etc 10min

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

Exercise



Now lets do the exercise

⚠ Enable UFW firewall - 10min

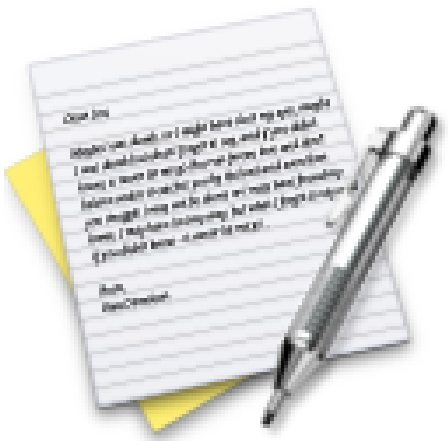
which is number **6** in the exercise PDF.

Exercise

Now lets do the exercise

Git tutorials - 15min

which is number **7** in the exercise PDF.

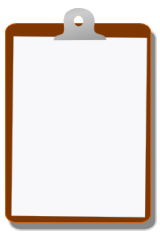


Now lets do the exercise

Use Ansible to install Elastic Stack

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!