

Welcome to

## 0. Introduction

# KEA Kompetence SIEM and Log Analysis

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

Slides are available as PDF, kramse@Github 

0-Introduction-siem-log-analysis.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](mailto:xhek@kea.dk)      Mobile: +45 2026 6000

You are welcome to drop me an email

# Goals for today



- Welcome, course goals and expectations
- Prepare Virtual Machines - hope you brought a laptop
- Create a good starting point for learning
- Concrete Expectations
- Prepare tools for the exercises

Photo by Thomas Galler on Unsplash

## Plan for today

- Create a good starting point for learning
- Introduce lecturer and students
- Expectations for this course
- Literature list walkthrough
- Prepare tools for the exercises
- Kali and Debian Linux introduction

Exercise theme: Chaos Computing

- Debian Linux installation
- Git tutorials, Python, Ansible
- Elastick Stack installation, Postman

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

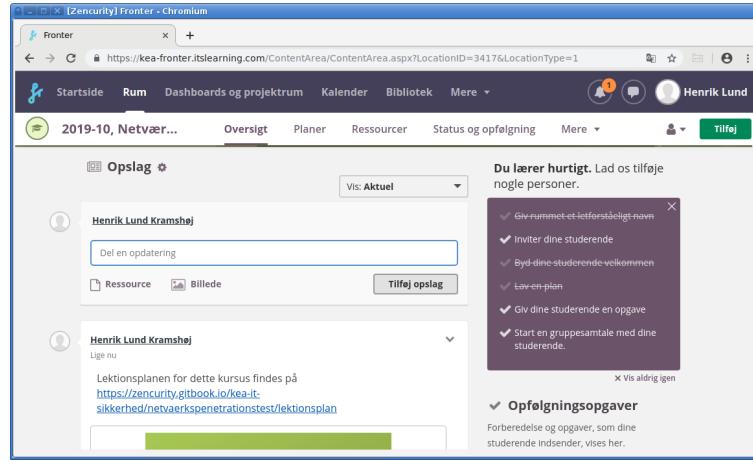
This will be the basic plan for each evening

## 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 and here
- 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



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: SIEM and Log Analysis (5 ECTS)

Teaching dates: mostly tuesdays and thursdays 17:00 - 20:30

24/10 2023, 31/10 2023, 7/11 2023, 14/11 2023, 21/11 2023, 28/11 2023, 5/12 2023

Exam: 19/12 2023

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
- Mandatory assignments are required in order to be entitled to the exam.

# Course Description

From: STUDIEORDNING Diplomuddannelse i it-sikkerhed August 2018  
VF4 SIEM og log analyse (5 ECTS)

## Indhold

Den studerende lærer om Security information and event management (SIEM), herunder hvordan man kan indsamle, administrere, og søge i sikkerhedshændelsesdata i et større IT system (komplekse systemer, IOT deployments, corporate IT).

## Læringsmål

Viden – Den studerende har viden om og forståelse for:

- Typiske SIEM arkitekturen
- Standard logformater og logtyper for standard systemer og komponenter
- Typiske SIEM produkter
- Juridiske krav til logning og bevarelse af data ifb. forensic analyse

## Færdigheder – Den studerende kan:

- Lave en baseline-analyse af en infrastruktur
- Bruge log-data til at identificere infrastrukturkomponenter
- Bruge et værktøj til at analysere system log-data og netværkstrafik til at finde sikkerhedshændelser
- Udvikle "dashboards" og alarmer der viser tegn på hændelser

## Kompetencer – Den studerende kan:

- Designe og implementere en SIEM løsning på tværs af diverse produkter
- Træffe beslutninger om hvilke data der skal indsamles i en givne situation
- Identificerer fejl i logopsamlingen
- Deltage i drøftelser på et praktisk og strategisk niveau i forhold til implementering af logmanagement/SIEM

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

**Security information and event management (SIEM)** is a subsection within the field of computer security, where software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware.

Vendors sell SIEM as software, as appliances, or as managed services; these products are also used to log security data and generate reports for compliance purposes.[1]

The term and the initialism SIEM was coined by Mark Nicolett and Amrit Williams of Gartner in 2005.[2]

Source: [https://en.wikipedia.org/wiki/Security\\_information\\_and\\_event\\_management](https://en.wikipedia.org/wiki/Security_information_and_event_management)

- Note: there are alerting examples towards the bottom of the page, with sources
- Closely related to log management, incident response

# Exercises

Exercise theme: Virtual Machines allows us play with tech

## Hardware

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

The following are recommended systems:

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

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

# Expectations alignment



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

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

I look forward to hearing your wishes, and hopefully we can accomodate some

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

## 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 SIEM and Logging  
Getting the best books with pointers about the closely related subject incident response
- Present a lot of useful sources, data types, tools
- Prepare you for production use of the knowledge  
Example you can take Linux, Ansible and Elasticsearch almost directly into production

We only have 5 ECTS, but a lot of flexibility.

## Some keywords relating to this course

Analysis Visualization Dashboards Data-driven Security  
SIEM architectures frameworks acquire process Zeek  
log formats data types databases JSON XML Security Operations Center  
(Incident Response) Intelligence R and Python fundamentals  
Practical application Building Infosec Ansible Playbooks  
Collect, mine, organize, and analyze relevant data sources  
Sort data create reporting and monitoring ports  
IP-address Netflow nfdump Elasticsearch real-world knowledge

- Lots of new terms, technologies and tools
- Its okay if too much, we will sort it out during next weeks

## 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

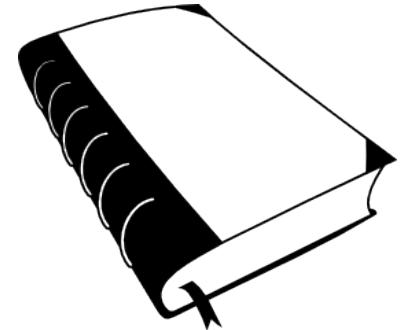
Security and most internet related security work has the following requirements:

- 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** for infosec work
  - too many new tools to ignore, and lots found at sites like Github and Open Source written for Linux

# Primary literature

Primary literature:

- *Data-Driven Security: Analysis, Visualization and Dashboards* Jay Jacobs, Bob Rudis ISBN: 978-1-118-79372-5 February 2014 <https://datadrivensecurity.info/> - short DDS
- *Crafting the InfoSec Playbook: Security Monitoring and Incident Response Master Plan* by Jeff Bollinger, Brandon Enright, and Matthew Valites ISBN: 9781491949405 - short CIP
- *Intelligence-Driven Incident Response*  
Scott Roberts. Rebekah Brown, ISBN: 9781098120689 **2nd edition**- short IDIR
- *Security Operations Center: Building, Operating, and Maintaining your SOC*  
ISBN: 9780134052014 Joseph Muniz - short SOC



Free graphics by Lumen Design Studio

Problem: You probably dont have the books yet ...

## 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/siem-and-log-analysis/lektionsplan>

### Why so many books?!

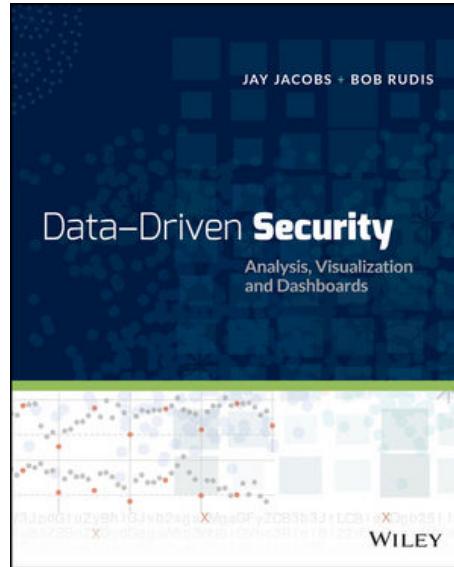
Because we not only want you to learn during the course, but be prepared for what comes after. These books are a library of relevant information that you will use when handling your job functions afterwards.

You will probably also use this course for Incident Response.

Some of you will be tasked with building a capability in-house, a SOC, virtual SOC or similar security group.

# Data-Driven Security: Analysis, Visualization and Dashboards

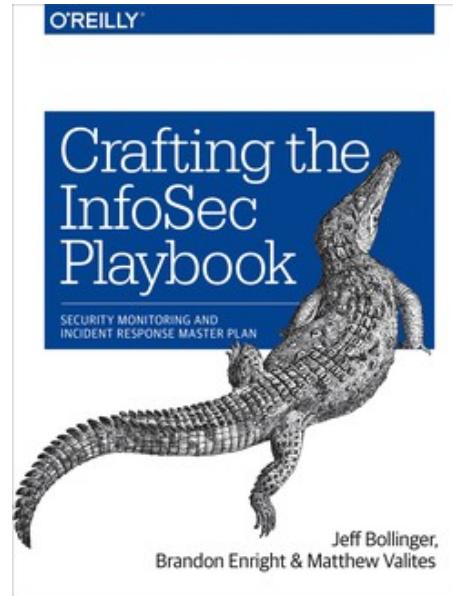
kea



*Data-Driven Security: Analysis, Visualization and Dashboards* Jay Jacobs, Bob Rudis  
ISBN: 978-1-118-79372-5 February 2014 <https://datadrivensecurity.info/> - short DDS

Our main book for this course. We will read a lot from this one. From basic data processing to dashboards

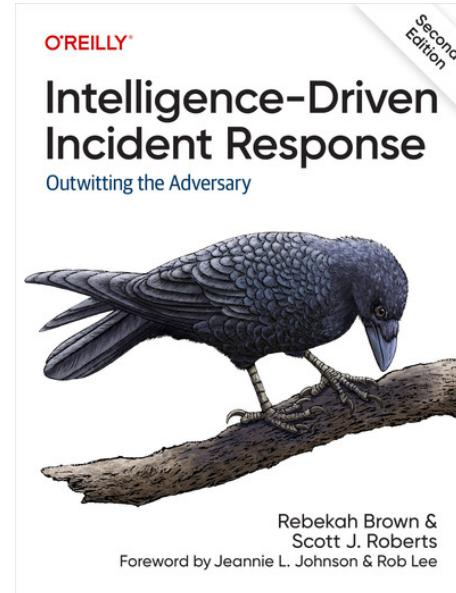
# Crafting the InfoSec Playbook



*Crafting the InfoSec Playbook: Security Monitoring and Incident Response Master Plan*  
by Jeff Bollinger, Brandon Enright, and Matthew Valites ISBN: 9781491949405 - short CIP

*Develop your own threat intelligence and incident detection strategy*

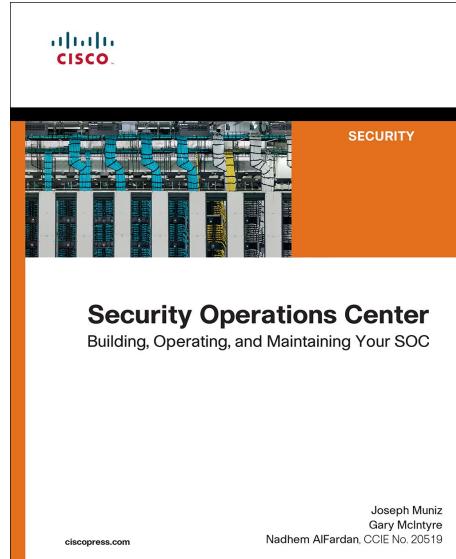
# Intelligence-Driven Incident Response



*Intelligence-Driven Incident Response*

Scott Roberts. Rebekah Brown, ISBN: 9781098120689 **2nd edition**- short IDIR

# Security Operations Center

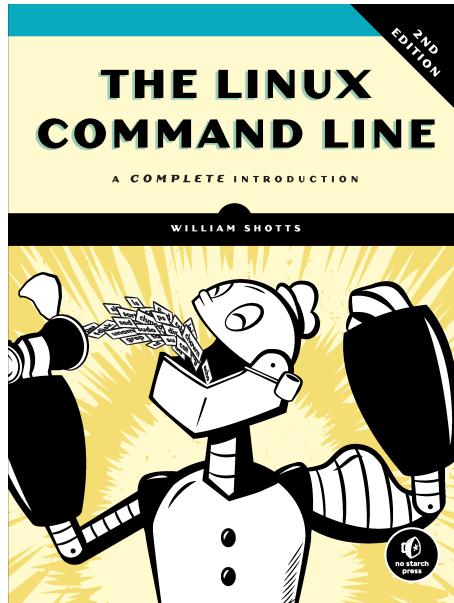


*Security Operations Center: Building, Operating, and Maintaining your SOC*  
ISBN: 9780134052014 Joseph Muniz - short SOC

## 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

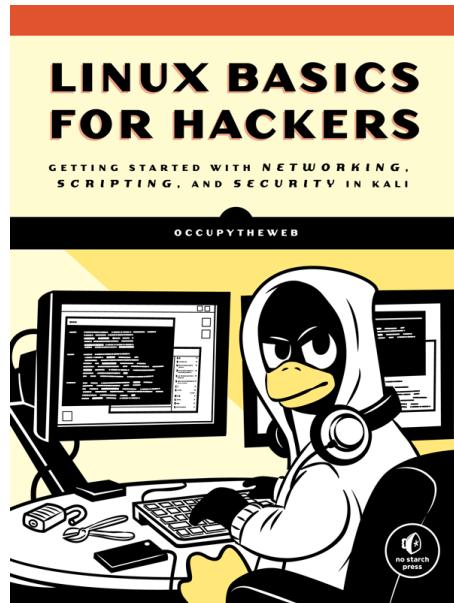
## Book: The Linux Command Line



*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>



*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

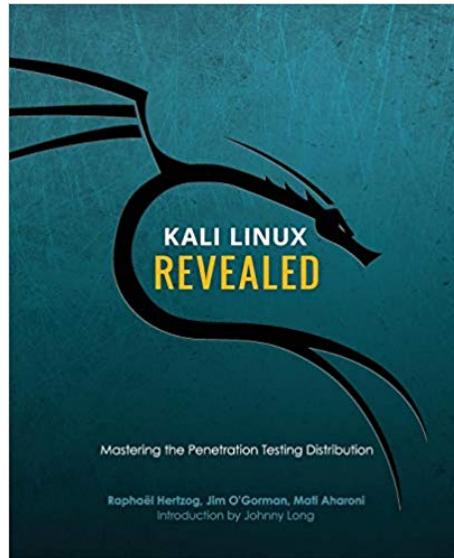
# The Debian Administrator's Handbook (DEB)



*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

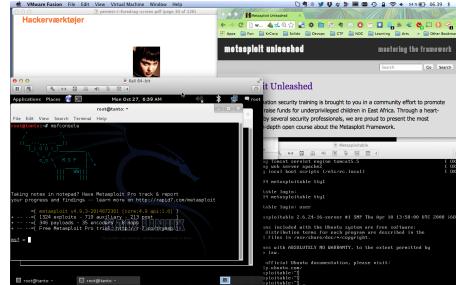
# Kali Linux Revealed (KLR)



*Kali Linux Revealed Mastering the Penetration Testing Distribution*

Not curriculum but explains how to install Kali Linux

# 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
- 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

## 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
- Elasticsearch – how to run a *service*
- Running Java on Linux – environment variables?!
- Ansible provisioning – installing and configuring software for production

**Note: today we will consider all these optional, we wont be able to do them all**

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

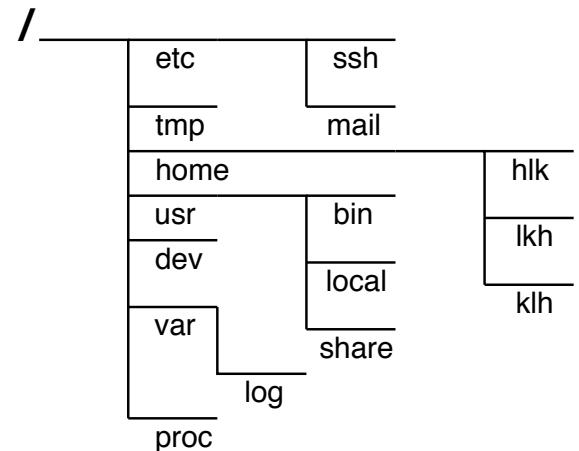
September

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
														1						
1	2	3	4											1	<b>2</b>	<b>14</b>	<b>15</b>	<b>16</b>		
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](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



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\\_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))
- Great documentation  
[https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt\\_module.html](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>



- 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



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" | su
```

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 account 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 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.”

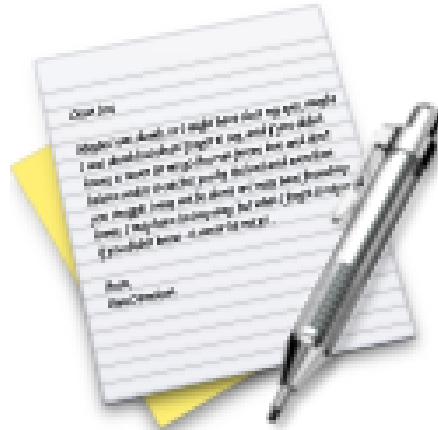
Hitchhiker’s Guide to the Galaxy, Douglas Adams

## Your lab setup

- Go to GitHub, Find user Kramse, click through kramse-labs
- Look into the instructions for the Virtual Machine – Debian 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!



Now lets do the exercise

## ⚠ Download Debian Administrator's Handbook – 10 min

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

# Exercise



Now lets do the exercise

## ⚠ Check your Debian VM – 10min

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

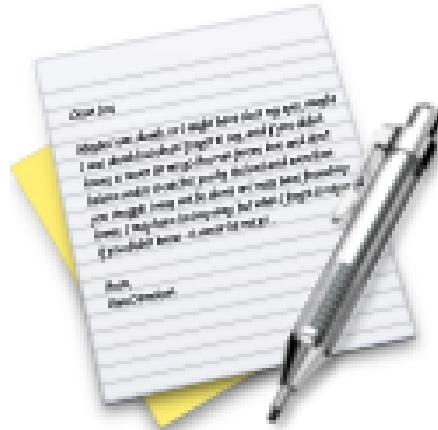


Now lets do the exercise

## ❶ Investigate /etc – 10min

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

# Exercise



Now lets do the exercise

## ⚠ Enable UFW firewall – 10min

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



Now lets do the exercise

## ⚠ Postman API Client – 20min

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



Now lets do the exercise

## ⚠ Git tutorials – 15min

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



Now lets do the exercise

## ⚠ Getting started with the Elastic Stack – 60min

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

# Exercise



Now lets do the exercise

## ⚠ Use Ansible to install programs – 10-60min

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

# Exercise



Now lets do the exercise

## ⚠ Configure Elasticsearch passwords – 15 min

which is number **9** in the exercise PDF.



Now lets do the exercise

## ⚠ Configure Kibana – 15 min

which is number **10** in the exercise PDF.

# Exercise



Now lets do the exercise

## ❶ Install JupyterLab – up to 30min

which is number **11** in the exercise PDF.

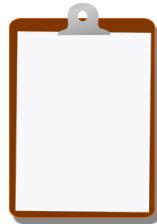


Now lets do the exercise

## i Making requests to Elasticsearch – 15-75min

which is number **12** 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! Create your VMs