Welcome to

Systems Security - 1

Intro to IT-security 2024

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

Slides are available as PDF, kramse@Codeberg https://codeberg.org/kramse/
intro-to-it-security-system-security-1.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

- Prepare Virtual Machines
- Prepare tools for the exercises
- Learn to find resources, files and programs/libraries
- Concrete Expectations

Exercises

- Debian Linux exercises

Photo by Thomas Galler on Unsplash

Plan and Time Schedule

System security

- Day 1: Introduction, vulnerabilities, scanning with Nmap, get an overview
- Day 2: User accounts, authentication, athorization, Access Control Lists, Confinement and isolation
- Day 3: Benchmarking and Auditing
- Day 4: Work on mandatory
- Day 2: Present mandatory – as many as possible

- 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.

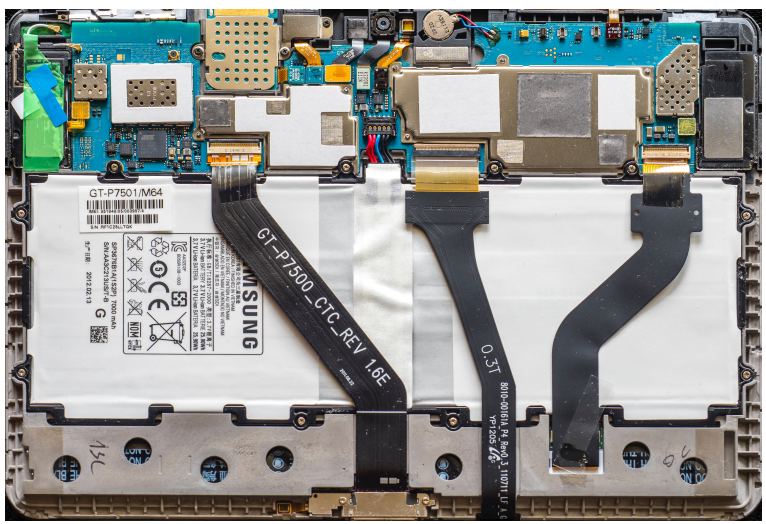
Course Materials

This material is in multiple parts:

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

Additional resources from the internet

Goals: System Security



System security - because we have some common security subjects across our environments

Plan for system security in this course

Picture from the OpenBSD project, software blobs

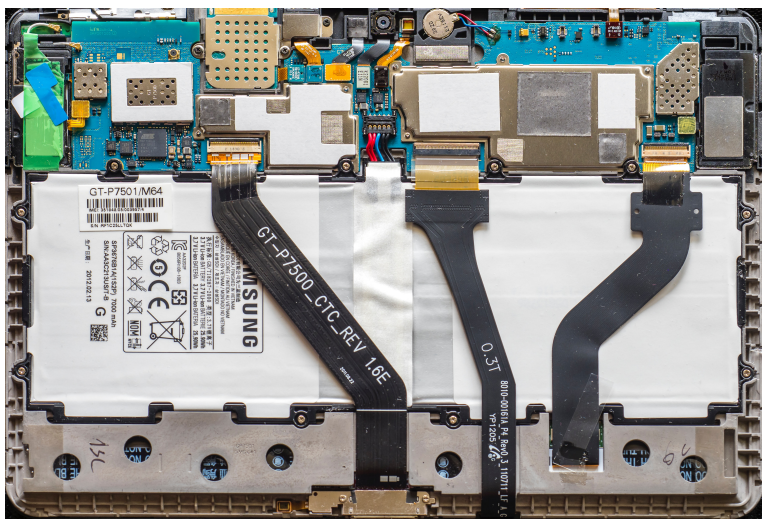
- Get started looking at this big subject
- 5 days with mix of teaching and doing exercises
- Last day dedicated to you presenting a mandatory

About the exercises

You will need in your group – 2-3 persons is recommended: Docker or similar container technology, Browser – researching stuff on the internet

I will use a virtual machine with Debian 12 Bookworm for this! Most exercises can be executed from this VM. You may be able to run exercises from your normal operating system, but it may take longer than just adding a Debian VM and doing it.

What is Infrastructure



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

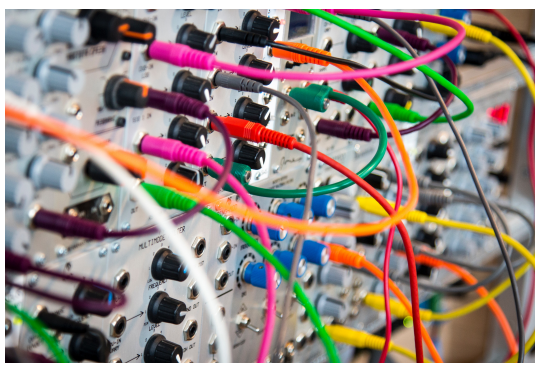
Photo by Alexander Schimmeck on Unsplash

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

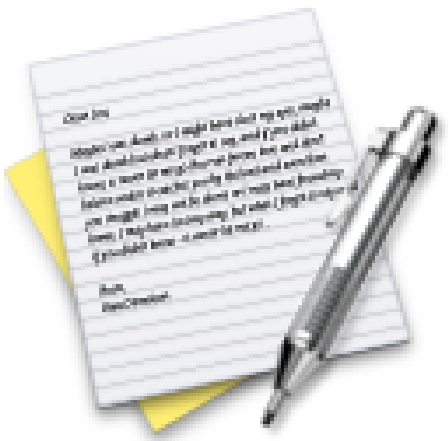
Photo by Kelly Sikkema on Unsplash

Integration Challenges

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

Photo by Thomas Drouault on Unsplash

Exercise



Now lets do the exercise

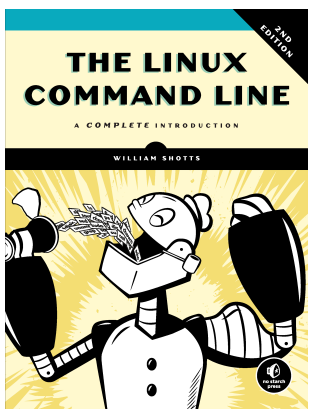
ℹ Mitre ATT&CK Framework 15min

which is number 1 in the exercise PDF.

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

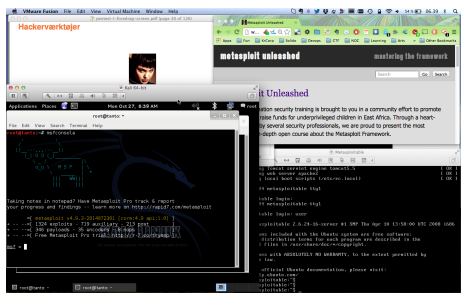
Not curriculum but explains how to use Linux

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

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 12 Bookworm 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 security-courses, find the software-security folder https://github.com/kramse/security-courses/tree/master/courses/system-and-software/software-security
- Look into the files named: `intro-to-it-security-week-7.pdf`, `intro-to-it-security-week-8.pdf` and `intro-to-it-security-exercises.pdf`
- Install Docker on a laptop in your team – you might already have it

Hint: If you have a Debian virtual machine you can install the docker software as a package using the Ansible tool, by checking out a repo `kramse-labs` and running Ansible:

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

Why introduce Git and Github?

We introduce Git here also because Github, one of the most popular places to store Git repositories has added security tools which you can use.

An example of a security feature at Github is the use of dependencies, when a project is stored on Github they can scan for outdated dependencies which have security issues.

You can read more about the features available, and some common problems in software in their article: *How GitHub secures open source software* Feb 23, 2021 // 10 min read
https://resources.github.com/security/open-source/how-github-secures-open-source-software/

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.

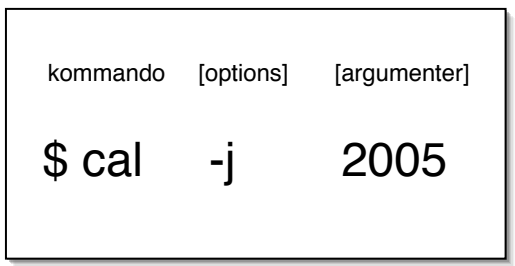
Command syntax

```
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.
```

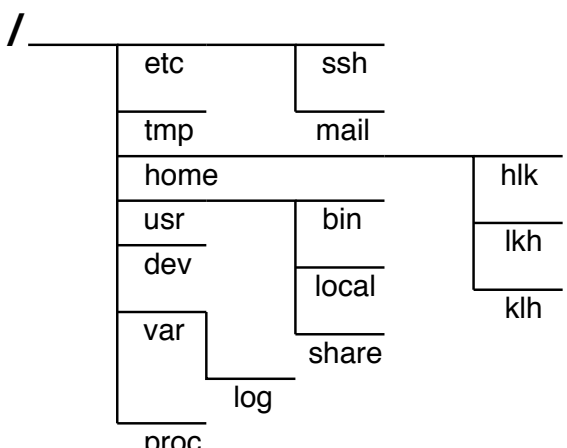
The year 1752

```
  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:

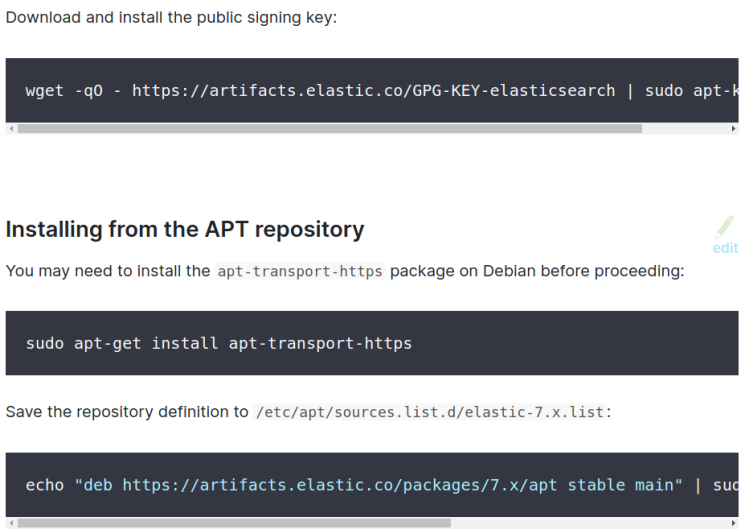
```
cd kramse-labs/suricatazeek
ansible-playbook -v 1-dependencies.yml 2-suricatazeek.yml 3-elasticstack.yml 4-configuration.yml
```

"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

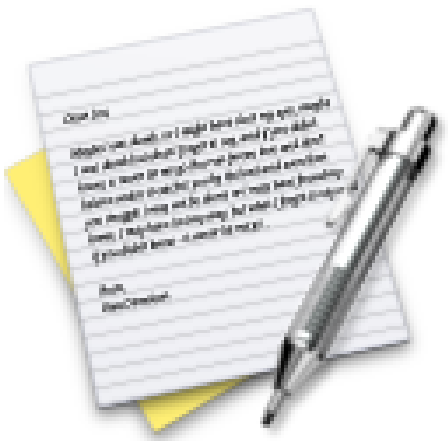
```
user@Projects:tt$  git clone https://github.com/kramse/kramse-labs.git
Cloning into 'kramse-labs'...
remote: Enumerating objects: 283, done.
remote: Total 283 (delta 0), reused 0 (delta 0), pack-reused 283
Receiving objects: 100% (283/283), 215.04 KiB | 898.00 KiB/s, done.
Resolving deltas: 100%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

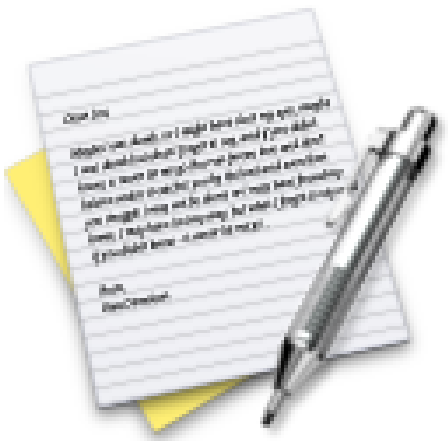


Now lets do the exercise

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

which is number 2 in the exercise PDF.

Exercise

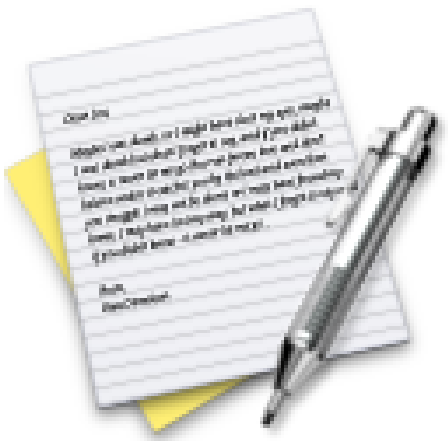


Now lets do the exercise

⚠ Check your Debian VM 10min

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

Exercise

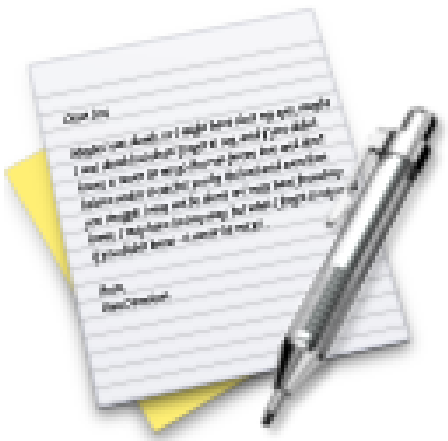


Now lets do the exercise

⚠ Investigate /etc 10min

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

Exercise

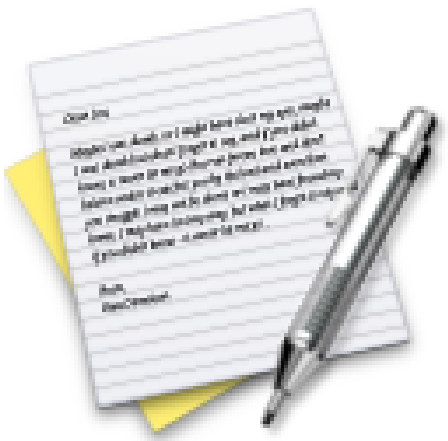


Now lets do the exercise

⚠ Enable UFW firewall - 10min

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

Exercise



Now lets do the exercise

Git tutorials - 15min

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

End of part I

Take a break!

Goals part II: Increase Security Awareness

Fact of life: Software has errors, hardware fails

Sometimes software can be made to fail in interesting ways

Humans can be social engineered

We are being attacked by criminals - including paranoid governments

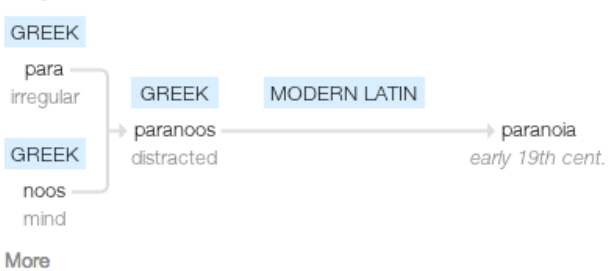
Paranoia defined

par·a·noi·a
/ˌparəˈnoiə/

noun
noun: **paranoia**

1. a mental condition characterized by delusions of persecution, unwarranted jealousy, or exaggerated self-importance, typically elaborated into an organized system. It may be an aspect of chronic personality disorder, of drug abuse, or of a serious condition such as schizophrenia in which the person loses touch with reality.
   synonyms: persecution complex, delusions, obsession, psychosis More
2. suspicion and mistrust of people or their actions without evidence or justification.
   "the global paranoia about hackers and viruses"

Origin



More

Source: google paranoia definition many years ago

Face reality

From the definition:

> suspicion and mistrust of people or their actions **without evidence or justification**. **the global paranoia about hackers and viruses**

It is not paranoia when:

- Criminals sell your credit card information and identity theft
- Trade infected computers like a commodity
- Governments write laws that allows them to introduce back-doors - and use these
- Governments do blanket surveillance of their population, implement censorship, threaten citizens and journalist

You are not paranoid when there are people actively attacking you!

I recommend we have appropriate paranoia (DK: passende paranoia)

Overlapping Security Incidents

New data breaches nearly every week, these from danish news site version2.dk

Problem, we need to receive data from others

Data from others may contain malware

Have a job posting, yes
- then HR will be expecting CVs sent as .doc files

XZ backdoor

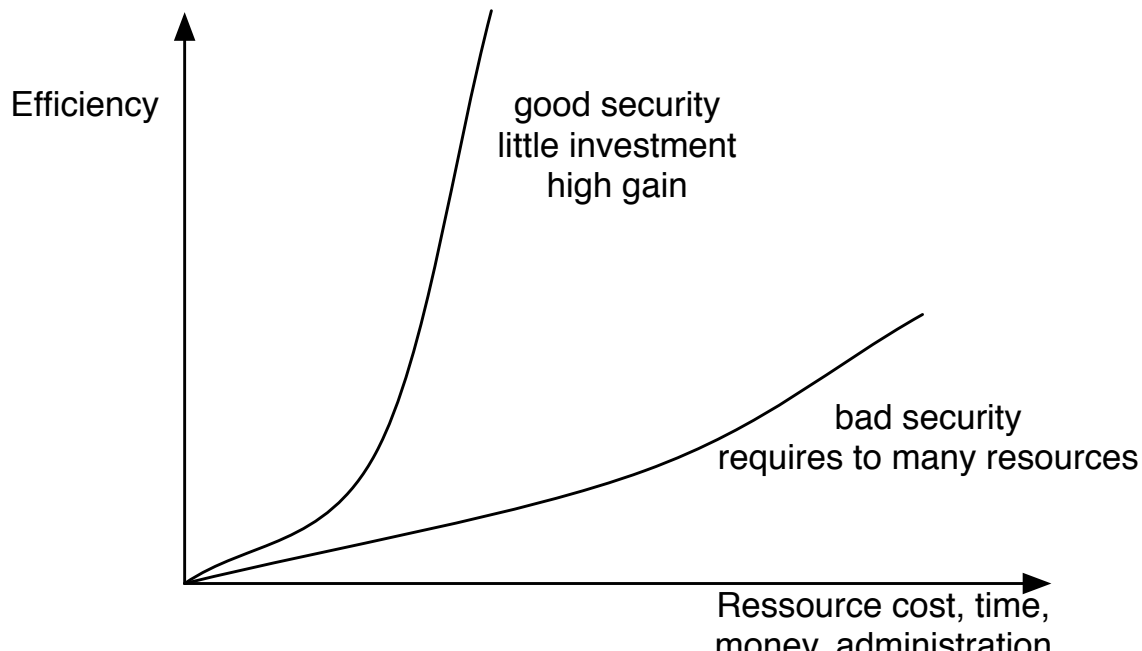
This month, only days ago it surfaced that someone injected backdoors into some software named XZ. *Inside the failed attempt to backdoor SSH globally — that got caught by chance*

> What happened here is now well documented elsewhere, so I shall not recap it much, but essentially somebody appears to have hijacked the open source XZ project by social engineering the volunteer developer into handing over maintainer access after they cited some mental health issues, used the package XZ Utils to piggy back into syst XZ, allowing sshd to be hooked to trojan it on Linux distributions that use systemd.
>
> The **trojan allows somebody a private key to hijack sshd to execute commands**, amongst other functions. It is highly advanced.

Source: https://doublepulsar.com/inside-the-failed-attempt-to-backdoor-ssh-globally-that-got-caught-by-chance-bbfe628

- Post by AndresFreundTec https://mastodon.social/@AndresFreundTec/112180083704606941

Good security



You always have limited resources for protection - use them as best as possible

Recommendations

Keep updated!

- read web sites, books, articles, mailing lists, Twitter, ...

Always have a chapter on security evaluation

- any process must have security, like RFC Request for Comments have

Incident Response

- you WILL have security incidents, be prepared

Write down security policy

- including software and e-mail policies

Advice

Use technology

Learn the technology - read the freaking manual

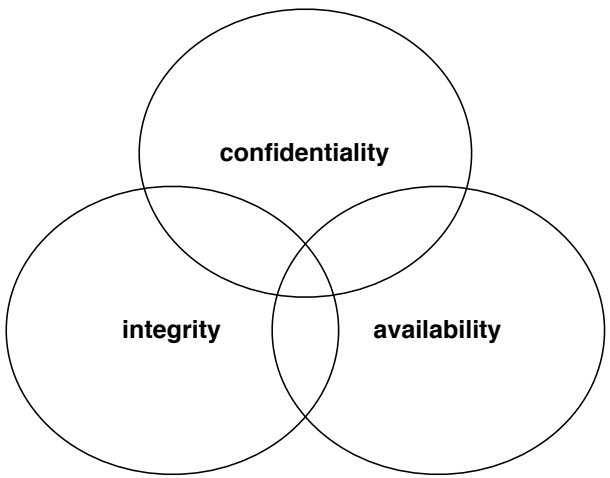
Think about the data you have, upload, facebook license?! WTF!

Think about the data you create

- Turn off features you don't use
- Turn off network connections when not in use
- Update software and applications
- Turn on encryption: IMAP**S**, POP3**S**, HTTP**S** also for data at rest, full disk encryption, tablet encryption
- Lock devices automatically when not used for 10 minutes
- Dont trust fancy logins like fingerprint scanner or face recognition on cheap devices

But which features to disable? Let the security principles guide you

Confidentiality, Integrity and Availability



We want to protect something

Confidentiality - data kept a secret

Integrity - data is not subjected to unauthorized changes

Availability - data and systems are available when needed

Work together

Team up!

We need to share security information freely

We often face the same threats, so we can work on solving these together

Goals of Security

Prevention - means that an attack will fail

Detection - determine if attack is underway, or has occured - report it

Recovery - stop attack, assess damage, repair damage

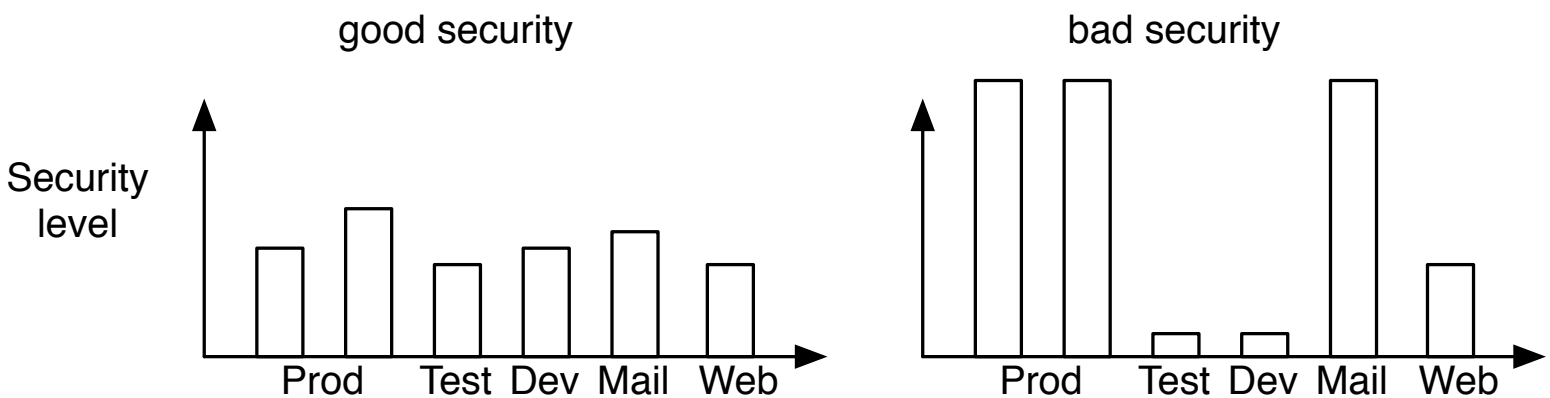
Policy and Mechanism

Definition 1-1. A *security policy* is a statement of what is, and what is not, allowed.

Definition 1-2. A *security mechanism* is a method, tool or procedure for enforcing a security policy.

Quote from Matt Bishop, Computer Security section 1.3

Balanced security



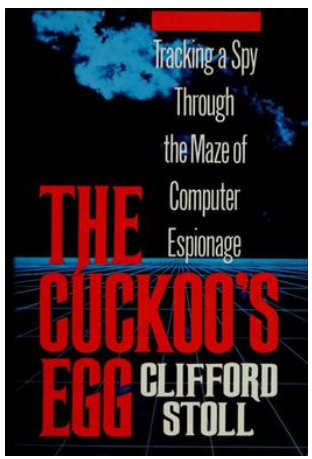
Better to have the same level of security

If you have bad security in some part - guess where attackers will end up

Hackers are not required to take the hardest path into the network

Realize there is no such thing as 100% security

Cuckoo's Egg 1986 A real spy story



Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage,
Clifford Stoll

During his time at working for KGB, Hess is estimated to have broken into 400 U.S. military computers

Source: https://en.wikipedia.org/wiki/Markus_Hess

Morris Internet Worm - 30 years ago

Used multiple vulnerabilities:

- Sendmail Debug functionality, we have similar things and Google Hacking
- Buffer overflow in fingerd, we still have those
- Weak passwords/password cracking, list of 432 words and /usr/dict/words, same problem today
- Trust between systems rsh, rexec, think Domain Admin today
- Found new systems using /etc/hosts.equiv, .rhosts, .forward, netstat ...

Also known as the Morris Internet Worm

The Internet Worm Program: An Analysis
Purdue Technical Report CSD-TR-823, Eugene H. Spafford

Resulted in creation of the CERT, http://www.cert.org

Internet Worms history repeats itself

Camouflage, tried to hide, malware today hides as well

- Program name set to 'sh', looks like a regular shell
- Used fork() to change process ID (PID)
- Worms in the 2000s spread quickly, like Code Red 2001 to approx 350.000 systems in a week
- SQL Slammer "It spread rapidly, infecting most of its 75,000 victims within ten minutes."

New malware today can use the same strategies

Except a lot of malware tries to stay hidden, less noisy

Using a small password list of 50 words it is possible to create your own botnet with 100.000s

Trusted Computing Base

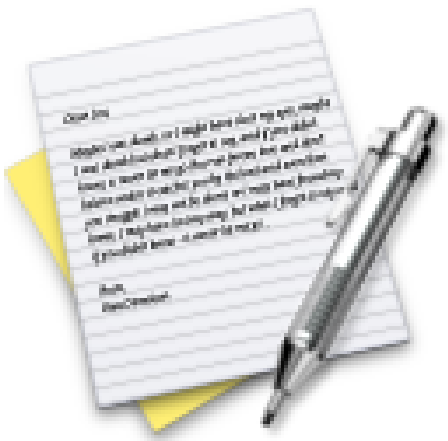
Definition 20-6. A *trusted computing base* (TCB) consists of all protection mechanisms within a computer system – including hardware, firmware, and software – that are responsible for enforcing a security policy

Quote from Matt Bishop, Computer Security

Keeping this small, simple and understandable help keeping systems more secure.

Example the Qubes OS depend on few security-critical components:
https://www.qubes-os.org/doc/security-critical-code/

Exercise

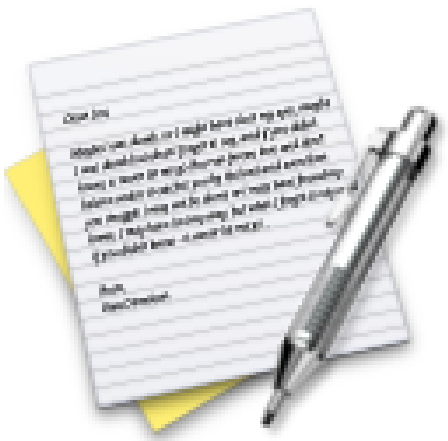


Now lets do the exercise

⚠ Discover active systems ping and port sweep 15min

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

Exercise

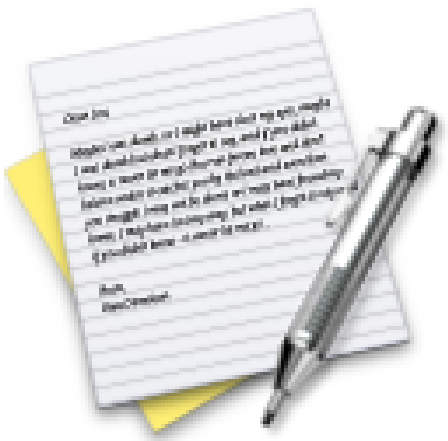


Now lets do the exercise

⚠ Execute nmap TCP and UDP port scan 15min

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

Exercise



Now lets do the exercise

⚠ Perform nmap OS detection 15min

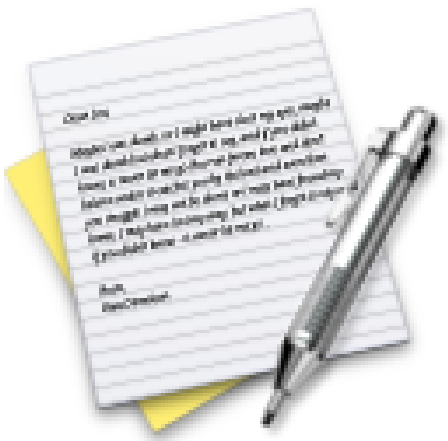
which is number **9** in the exercise PDF.

Exercise

Now lets do the exercise

Nmap full scan - 15min

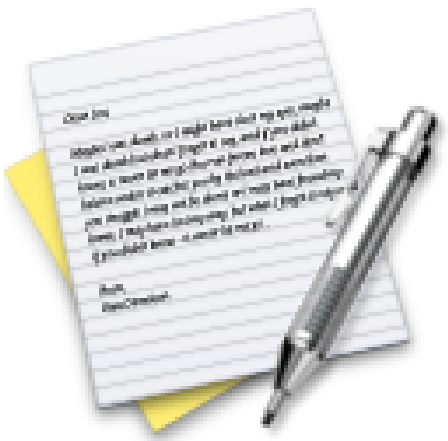
which is number **10** in the exercise PDF.



Now lets do the exercise

Reporting Nmap HTML 10min

which is number 11 in the exercise PDF.

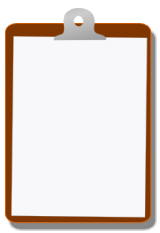


Now lets do the exercise

Nmap Scripting Engine NSE scripts 20min

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