

System Integration F2020

Exercises

Henrik Lund Kramshoej
hlk@zencurity.com

January 23, 2020



Contents

1	Download Kali Linux Revealed (KLR) Book 10 min	2
2	Check your Kali VM, run Kali Linux 30 min	3
3	Check your Debian VM 10 min	4
4	Investigate /etc 10 min	5
5	Centralized syslog 15min	7
6	Create Kibana Dashboard 15min	9
7	Enable firewall	11

Preface

This material is prepared for use in *System Integration F2020* and was prepared by Henrik Lund Kramshøj, Zencurity Aps. It describes the setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github
Look for system-integration-exercises in the repo security-courses.

These exercises are expected to be performed in a training setting with network connected systems. The exercises use a number of tools which can be copied and reused after training. A lot is described about setting up your workstation in the repo

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

Prerequisites

This material expect that participants have a working knowledge of internet from a user perspective. Basic concepts such as web site addresses, IP-addresses and email should be known as well.

Have fun and learn

Exercise content

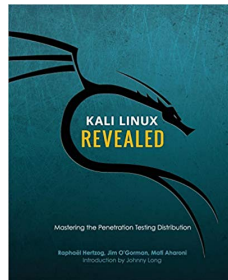
Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective
- **Purpose:** What is to be the expected outcome and goal of doing this exercise
- **Suggested method:** suggest a way to get started
- **Hints:** one or more hints and tips or even description how to do the actual exercises
- **Solution:** one possible solution is specified
- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

Exercise 1

Download Kali Linux Revealed (KLR) Book 10 min



Kali Linux Revealed Mastering the Penetration Testing Distribution

Objective:

We need a Kali Linux for running tools during the course. This is open source, and the developers have released a whole book about running Kali Linux.

This is named Kali Linux Revealed (KLR)

Purpose:

We need to install Kali Linux in a few moments, so better have the instructions ready.

Suggested method:

Create folders for educational materials. Go to <https://www.kali.org/download-kali-linux-revealed-book/> Read and follow the instructions for downloading the book.

Solution:

When you have a directory structure for download for this course, and the book KLR in PDF you are done.

Discussion:

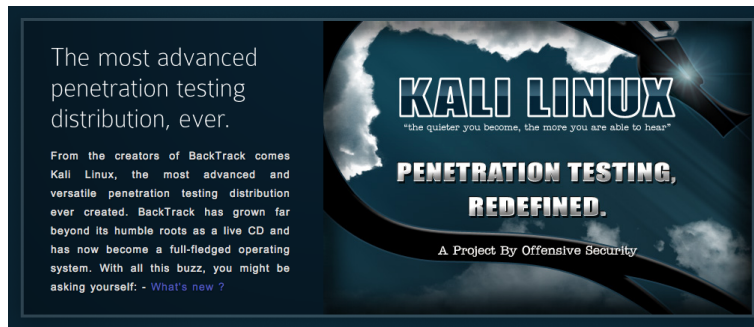
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux is a free pentesting platform, and probably worth more than \$10.000

The book KLR is free, but you can buy/donate, and I recommend it.

Exercise 2

Check your Kali VM, run Kali Linux 30 min



Objective:

Make sure your virtual machine is in working order.

We need a Kali Linux for running tools during the course.

Purpose:

If your VM is not installed and updated we will run into trouble later.

Suggested method:

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

Read the instructions for the setup of a Kali VM.

Hints:

If you allocate enough memory and disk you wont have problems.

Solution:

When you have a updated virtualisation software and Kali Linux, then we are good.

Discussion:

Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux includes many hacker tools and should be known by anyone working in infosec.

Exercise 3

Check your Debian VM 10 min



Objective:

Make sure your virtual Debian 9 machine is in working order.

We need a Debian 9 Linux for running a few extra tools during the course.

This is a bonus exercise - only one Debian is needed per team.

Purpose:

If your VM is not installed and updated we will run into trouble later.

Suggested method:

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

Read the instructions for the setup of a Kali VM.

Hints:

Solution:

When you have a updated virtualisation software and Kali Linux, then we are good.

Discussion:

Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Exercise 4

Investigate /etc 10 min

Objective:

We will investigate the /etc directory on Linux

We need a Debian 9 Linux and a Kali Linux, to compare

Purpose:

Start seeing example configuration files, including:

- User database /etc/passwd and /etc/group
- The password database /etc/shadow

Suggested method:

Boot your Linux VMs, log in

Investigate permissions for the user database files passwd and shadow

Hints:

Linux has many tools for viewing files, the most efficient would be less.

```
hlk@debian:~$ cd /etc
hlk@debian:/etc$ ls -l shadow passwd
-rw-r--r-- 1 root root  2203 Mar 26 17:27 passwd
-rw-r----- 1 root shadow 1250 Mar 26 17:27 shadow
hlk@debian:/etc$ ls
... all files and directories shown, investigate more if you like
```

Showing a single file: less /etc/passwd and press q to quit

Showing multiple files: less /etc/* then :n for next and q for quit

Trying reading the shadow file as your regular user:

```
user@debian-9-lab:/etc$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Why is that? Try switching to root, using su or sudo, and redo the command.

Solution:

When you have seen the most basic files you are done.

Discussion:

Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Exercise 5

Centralized syslog 15min

Objective:

See how server syslog is configured on regular Unix/Linux.

Centralized syslogging and example system can demonstrate how easy it is to get started

Purpose:

The main idea of this exercise is to understand how easy network connected systems can send log data.

This should be the common case, sending logs off system - to avoid an attacker being able to hide tracks and logs from exploits performing intrusion and escalation.

Suggested method:

Log into your local Linux systems or network devices, see how syslog is configured.

Hints:

Look in the config file, may be in /etc/syslog or /etc/syslog-ng/syslog-ng.conf

Sample output from old-skool syslogd

```
*.err;kern.debug;auth.notice;authpriv.none;mail.crit    /dev/console
*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none /var/log/messages
kern.debug;user.info;syslog.info                        /var/log/messages
auth.info                                                /var/log/authlog
authpriv.debug                                          /var/log/secure
...
# Uncomment to log to a central host named "loghost".
#*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none @loghost
#kern.debug,user.info,syslog.info                        @loghost
#auth.info,authpriv.debug,daemon.info                   @loghost
```

Solution:

When you understand how to configure syslog from a couple of devices and has looked up which protocol and port it uses. (default is 514/udp)

Discussion:

There are syslog senders for Windows too. Other systems define their own format for sending, example Beats - lightweight data shippers <https://www.elastic.co/products/beats>

I recommend using the elastic stack, previously the ELK stack, <https://www.elastic.co/products/>. The products can be used without license and can give a lot of experience with this kind of product. This will enable you to better describe your logging needs for evaluating other products.

This is done using Logstash as the server - can also receive SNMP traps!

Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite “stash.” - often Elasticsearch <https://www.elastic.co/products/logstash>

Other very popular systems are:

- Splunk <https://www.splunk.com>
- Graylog <https://www.graylog.org/>
- InfluxDB <https://www.influxdata.com/>
- Grafana The open platform for analytics and monitoring <https://grafana.com/>
- Prometheus Monitoring system & time series database <https://prometheus.io/>

Remember doing logging og performance metrics can also become a security characteristics. Availability is a critical metric for most commercial systems.

Exercise 6

Create Kibana Dashboard 15min



Objective:
See Kibana and understand how it is configured.

Purpose:
Kibana is a very popular system for creating dashboards from data in elasticsearch.
Learning how to create and import dashboards is a good exercise.

Suggested method:
Instructor will provide a running Elasticsearch and Kibana for this exercise.

Note: usually Kibana should be available on port 5601 on localhost (127.0.0.1) only!
It is recommended to keep this configuration and then add a web server like Nginx or Apache in front. This will further allow authentication and other features.

Using Firefox visit Kibana on the link provided by the instructor.

If this is the first time you need to select `logstash-*` as a default index. Note: Kibana is an advanced and powerful tool in itself.

Read how dashboards can be loaded using shell command, example the ones from:
<https://github.com/StamusNetworks/KTS6>

The commands are similar to

```
git clone https://github.com/StamusNetworks/KTS6.git
cd KTS6
bash load.sh
```

Hints:

Logstash and Elastic stack are a great way to get started with dashboarding.

However, running a big installation is harder than it looks. Make sure to have multiple servers and good monitoring.

Solution: When you have browsed Kibana, seen how you can add graphs and combine them into dashboards - using the GUI you are done. Previously creating dashboards was harder and often required programming knowledge.

Discussion:

Making dashboard are an art form. We will NOT start creating beautiful dashboards.

If you want, there is a SELKS LiveCD dedicated to suricata which also includes more tools for administration of rules and getting alerts:
<https://www.stamus-networks.com/open-source/>

Exercise 7

Enable firewall

Objective:

Turn on a firewall and configure a few simple rules.

Purpose:

See how easy it is to restrict incoming connections to a server.

Suggested method:

Install a utility for firewall configuration.

You should also perform Nmap port scan with the firewall enabled and disabled.

Hints:

Using the ufw package it is very easy to configure the firewall on Linux.

Install and configuration can be done using these commands.

```
root@debian01:~# apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ufw
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 164 kB of archives.
After this operation, 848 kB of additional disk space will be used.
Get:1 http://mirrors.dotsrc.org/debian stretch/main amd64 ufw all 0.35-4 [164 kB]
Fetched 164 kB in 2s (60.2 kB/s)
...
root@debian01:~# ufw allow 22/tcp
Rules updated
Rules updated (v6)
root@debian01:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@debian01:~# ufw status numbered
Status: active
```

	To	Action	From
	--	-----	----
[1]	22/tcp	ALLOW IN	Anywhere
[2]	22/tcp (v6)	ALLOW IN	Anywhere (v6)

Also allow port 80/tcp and port 443/tcp - and install a web server. Recommend Nginx `apt-get install nginx`

Solution:

When firewall is enabled and you can still connect to Secure Shell (SSH) and web service, you are done.

Discussion:

Further configuration would often require adding source prefixes which are allowed to connect to specific services. If this was a database server the database service should probably not be reachable from all of the Internet.

Web interfaces also exist, but are more suited for a centralized firewall.

Configuration of this firewall can be done using ansible, see the documentation and examples at https://docs.ansible.com/ansible/latest/modules/ufw_module.html

Should you have both a centralized firewall in front of servers, and local firewall on each server? Discuss within your team.