

A7011E Homework 6

Nico Ferrari (nicfer-0@student.ltu.se)

January 13, 2021

1. Data Center security vulnerabilities and countermeasures

Data centers are the facilities housing the enterprise computer systems, the networking equipment and associated hardware needed to ensure connectivity other business networks and/or to the Internet and the supplies that protect the data center hardware and keep it up and running (such as cooling systems, generators,...). During the last years, organizations started to move more often to cloud or hybrid cloud solutions, where the providers of these services assume the responsibility of providing available and fault-tolerant computing resources as a service [5].

According to the report published by Verizon [13], DoS attacks are one of the most common attacks. The volume of the traffic's attack is drastically increased during the last decade, together with the complexity of these attacks. Organizations usually deploy security appliances such as firewall in their datacenters, however, due to the trend in D/DoS, the traffic of the attack results beyond their capabilities. New mitigation methods include the redirection of the traffic to cloud based services, resulting in a usually cheaper and scalable alternative. However this solution can bring privacy related problems since the traffic is monitored by third parties and it brings latency due to the redirection of the traffic. Other solutions, such the one proposed in [1], deploy security functions as software instead of hardware by using VMs, leveraging the automation and scalability features of NFV and SDN.

Virtualization, due to its benefits (such as isolation, fast recovery etc...) is a frequent solution in datacenters. However it brings new surfaces of attack. This is challenging problem specially in multi tenant datacenters. The attacks, in fact, can come from the guest OS to the hypervisor or vice versa. Security solutions are described in [12] involving the implementation of traditional security mechanisms such as intrusion detection software and firewall on the different layers of the virtualization such as the hypervisor and the guest OS. moreover, becomes fundamental the security related to the transportation of VMs images, how they are stored and how they are managed, due to their mobility [16].

During the last years we assisted to the always more frequent deployment of Software-Defined-Networks (SDN), implementing a logically centralized controller able to analyze traffic and configure new instructions to be forwarded to switches' tables, reacting to changes or abnormalities in the network. However, new attacks vectors are introduced, especially against the controller, where threats could bring down the entire network [11].

Power over-subscription is a trend in order to support more servers with the existing power infrastructures. Due to the improbability to have peaks simultaneously on the servers, more of them are placed on the power infrastructure of a data center than it can support if the maximum power consumption is reached by those servers at the same time [14]. This removes the need of upgrading the power infrastructure, which is usually extremely expensive, however, power consumption of servers might exceed power capacity, incurring in the risk of power outages. This makes it difficult to observe DoS attacks before the unexpected traffics can violate power budget. In [4] this problem has been named DOPE (Denial of Power and Energy). In [9] is proposed a solution which uses machine learning to monitor and learn the acceptable characteristics of the life cycle of the data center in order to identify abnormal load requests. When these request are identified, the framework preempts the resources to prevent the power attack. Some other techniques are proposed in [7], which suggest a CNN framework to detect patterns of known attacks and once the attack is identified, it takes defence measures. This approach works good with know attacks but has disadvantage is that the unknown attack cannot be detected. The deployment a mini-UPS on each server results in an alternative

way to defend against power attacks, since a short period of power outage will not bring down the server [15].

Another big threat shown by Verizon in its report is due to miss-configurations and human error. Last decades, in fact, have been affected by a large amount of data breach [6] [10], and many times the reason was human mistake. For example, as reported by MacKeeper [8], a leak of sensitive medical records of thousands patients has been discovered due to the unprotected Database where they were stored. The same company noticed a leak of voters' data due to a miss configured database stored on Amazon S3. server. As we can see, human factor plays an important role in the security of a datacenter. In order to mitigate this threat, awareness becomes necessary, as described in [2]. Especially with the always increasing computing power and resources available, some security measures become obsolete, such as passwords. Enforcing MFA could mitigate the problems related to passwords but new challenges are presents for alternatives such as biometrics. In fact, they represent a non cancellable feature which, if stolen, can compromise the digital identity of a user [3]. Awareness can mitigate also the common Social Engineering Attacks and Phishing.

2. References

- [1] Talal Alharbi, Ahamed Aljuhani, and Hang Liu. "Holistic DDoS mitigation using NFV". In: *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*. Institute of Electrical and Electronics Engineers Inc., Mar. 2017. ISBN: 9781509042289. DOI: 10.1109/CCWC.2017.7868480.
- [2] Carl Colwill. "Human factors in information security: The insider threat - Who can you trust these days?" In: *Information Security Technical Report* 14.4 (Nov. 2009), pp. 186–196. ISSN: 13634127. DOI: 10.1016/j.istr.2010.04.004.
- [3] Tran Khanh Dang, Van Quoc Phuong Huynh, and Hai Truong. "A Hybrid Template Protection Approach using Secure Sketch and ANN for Strong Biometric Key Generation with Revocability Guarantee". In: March (2018).
- [4] Xiaofeng Hou et al. "When power oversubscription meets traffic flood attack: Re-thinking data center peak load management". In: *ACM International Conference Proceeding Series*. Association for Computing Machinery, Aug. 2019. ISBN: 9781450362955. DOI: 10.1145/3337821.3337856.
- [5] Kashif Bilal et al. "Trends and Challenges in Cloud Datacenters". In: *IEEE Cloud Computing* (2014).
- [6] Kenneth J. Knapp, Gary D. Denney, and Mark E. Barner. "Key issues in data center security: An investigation of government audit reports". In: *Government Information Quarterly* 28.4 (Oct. 2011), pp. 533–541. ISSN: 0740624X. DOI: 10.1016/j.giq.2010.10.008.
- [7] Li et al. "Real-time DDoS attack detection based on deep learning." In: *Telecommun. Sci.* (2017).
- [8] *MacKeeper Report*. <https://mackeeper.com/blog/post/220-digital-medical-records-are-the-future-but-how-safe-is-your-private-data/>. Accessed: 2020-12-03.
- [9] Rajesh JS et al. "Securing Data Center Against Power Attacks". In: *Journal of Hardware and Systems Security volume* (2019).
- [10] *Risk Based Security Report*. <https://pages.riskbasedsecurity.com/2019-q1-breach-quickview-report>. Accessed: 2020-12-03.
- [11] A Samson. *Software Defined Networking: Identification of Pathways for Security Threats*. Tech. rep. 2010. URL: <http://dx.doi.org/10.1145/2980258.2980294>.
- [12] Karen Scarfone and Peter Mell. *Guide to Intrusion Detection and Prevention Systems (IDPS) (Draft) Recommendations of the National Institute of Standards and Technology*. Tech. rep.
- [13] *Verizon Report*. <https://enterprise.verizon.com/resources/reports/dbir/2019/results-and-analysis/>. Accessed: 2020-12-03.

- [14] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz André Barroso. “Power Provisioning for a Warehouse-sized Computer”. In: (2007), p. 530.
- [15] Zhang Xu et al. *Power Attack: An Increasing Threat to Data Centers*. Tech. rep. 2014.
- [16] Minjie Zhang and Raj Jain. *Virtualization Security in Data Centers and Clouds*. 2011.

3. Have you successfully completed Lab assignment (5)

Figure 1: inserting the new rules to filter the inbound traffic in esp0s3 and esp0s8 interfaces.

```
ltu@debian:~$ sudo iptables -I INPUT -p tcp --dport 22 -i enp0s8 -m state --state NEW -m recent --set
ltu@debian:~$ sudo iptables -I INPUT -p tcp --dport 22 -i enp0s8 -m state --state NEW -m recent --update --seconds 60 --hitcount 4
-j DROP
ltu@debian:~$ sudo iptables -I INPUT -p tcp --dport 22 -i enp0s3 -m state --state NEW -m recent --set
ltu@debian:~$ sudo iptables -I INPUT -p tcp --dport 22 -i enp0s3 -m state --state NEW -m recent --update --seconds 60 --hitcount 4
-j DROP
ltu@debian:~$
```

Figure 2: before applying the rules the port 22 results open.

```
File Actions Edit View Help
root@kali: ~ root@kali: ~
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan  9 02:54:30 2021 from 10.0.0.206
ltu@debian:~$ exit
logout
Connection to 10.0.0.30 closed.

(root@kali)-[~]
# nmap 10.0.0.30
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-09 04:06 EST
Nmap scan report for 10.0.0.30
Host is up (0.00066s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
```

Figure 3: when we apply the rules the port results filtered and while we try to attempt multiple logins, xhydra presents some error messages due to the drop of our packets after more than 4 failed attempt.

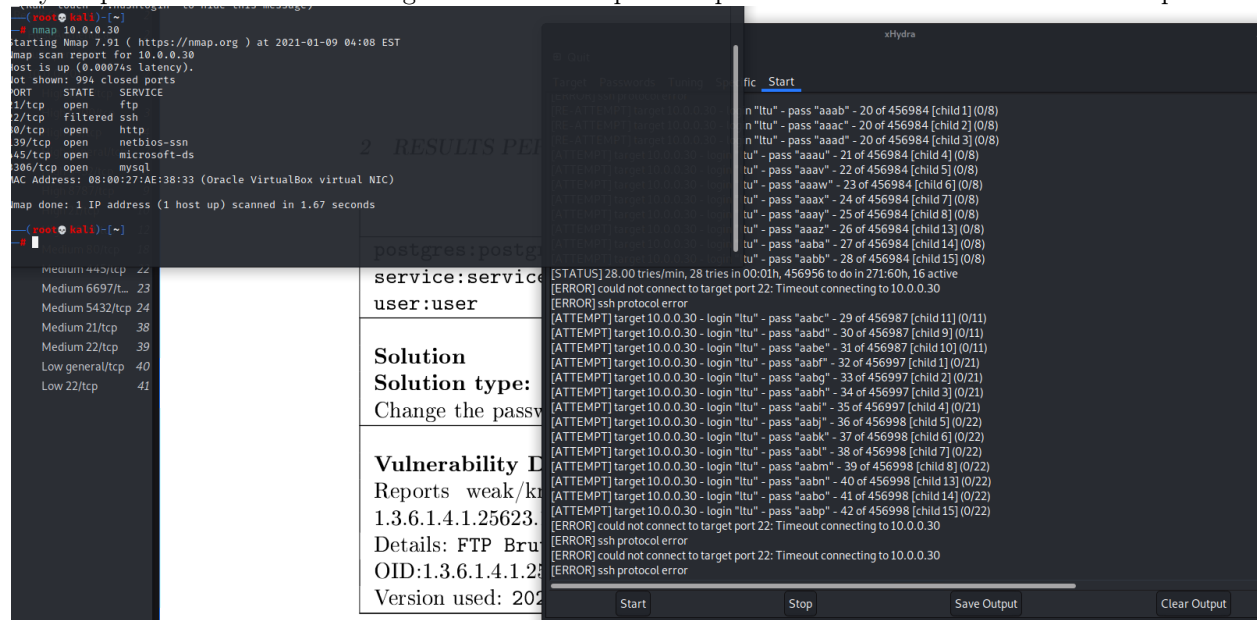


Figure 4: we can see that if we try to connect to SSH, the connection doesn't occurs due to the multiple failed connections.

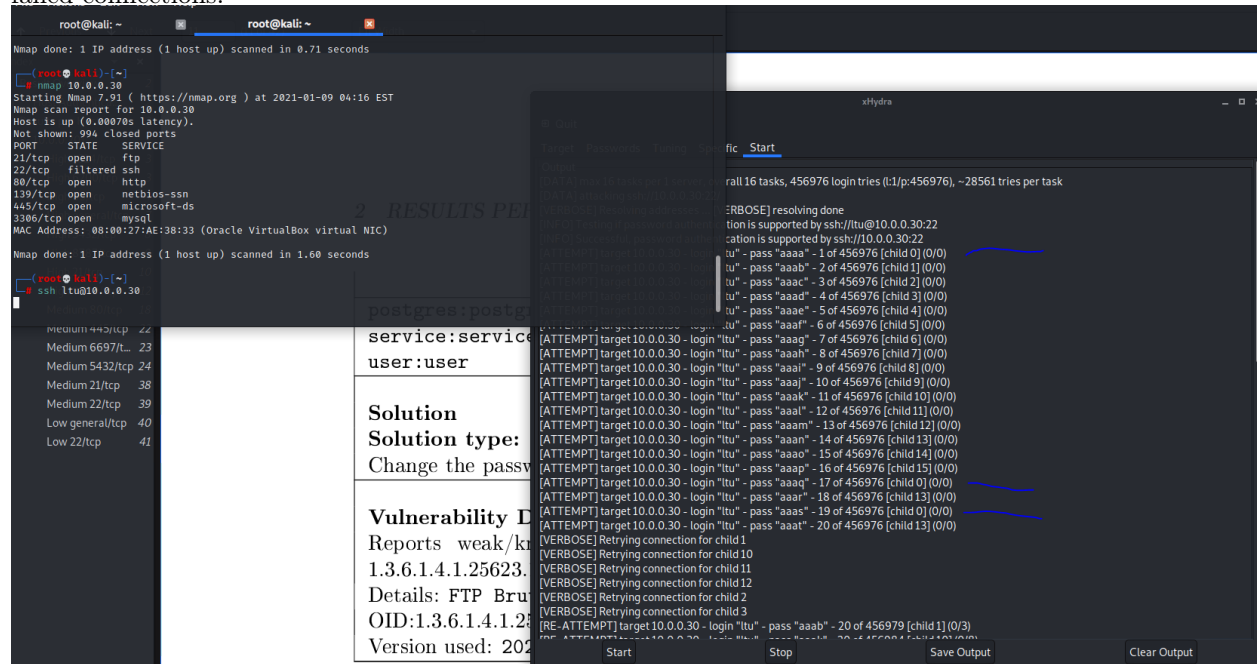


Figure 5: if the rules are removed (by changing the parameter *-I* to *-D*), the port will be open again and will be allowed to try multiple attempts.

The screenshot shows a Kali Linux terminal with two windows. The left window displays the output of an Nmap scan on 10.0.0.30, identifying open ports 21/tcp (ftp), 22/tcp (ssh), 80/tcp (http), 139/tcp (netbios-ssn), 445/tcp (microsoft-ds), and 3306/tcp (mysql). The right window shows a Hydra password cracking session targeting the SSH service on 10.0.0.30. The session lists various password attempts, mostly failing (0/0), and shows a status of 80.00 tries/min.

```

root@kali: ~
Nmap done: 1 IP address (1 host up) scanned in 0.67 seconds

root@kali: ~
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

root@kali: ~
Nmap scan report for 10.0.0.30
Host is up (0.00061s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
MAC Address: 08:00:27:AE:38:33 (Oracle VirtualBox virtual NIC)

root@kali: ~
Medium 6697/tcp 22
Medium 5432/tcp 24
Medium 21/tcp 38
Medium 22/tcp 39
Low general/tcp 40
Low 22/tcp 41

postgres:postgres
service:service
user:user

Solution
Solution type:
Change the password

Vulnerability D
Reports weak/known
1.3.6.1.4.1.25623.
Details: FTP Brute
OID:1.3.6.1.4.1.25623.

Hydra
Quit
Target Passwords Tuning Specific Start
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aaba" - 50 of 456976 [child 2] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aaby" - 51 of 456976 [child 5] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aabc" - 52 of 456976 [child 0] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aaca" - 53 of 456976 [child 9] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aach" - 54 of 456976 [child 15] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacb" - 55 of 456976 [child 6] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacd" - 56 of 456976 [child 7] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aace" - 57 of 456976 [child 3] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacf" - 58 of 456976 [child 8] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacg" - 59 of 456976 [child 4] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aach" - 60 of 456976 [child 14] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacj" - 61 of 456976 [child 11] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aack" - 62 of 456976 [child 12] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aack" - 63 of 456976 [child 13] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aack" - 64 of 456976 [child 10] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacm" - 65 of 456976 [child 1] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacn" - 66 of 456976 [child 2] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aaco" - 67 of 456976 [child 5] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacp" - 68 of 456976 [child 0] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacq" - 69 of 456976 [child 9] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacr" - 70 of 456976 [child 15] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacs" - 71 of 456976 [child 6] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aact" - 72 of 456976 [child 7] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacu" - 73 of 456976 [child 3] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacv" - 74 of 456976 [child 8] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacw" - 75 of 456976 [child 4] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacx" - 76 of 456976 [child 14] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacy" - 77 of 456976 [child 11] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aacz" - 78 of 456976 [child 12] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aada" - 79 of 456976 [child 13] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ltu" - pass "aadb" - 80 of 456976 [child 10] (0/0)
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 456896 to do in 95:12h, 16 active

```

Figure 6: a RSA private key is generated and in the command, the parameter *-des3* states the cipher used to encrypt private key before outputting it in the file specified after the *-out* parameter and *2048* is size of the private key to generate in bits.

The screenshot shows a Debian terminal where the user installs openssl and generates an RSA private key. The key is 2048 bits long and encrypted with des3. The user is prompted to enter a pass phrase for the key.

```

ltu@debian:~$ sudo apt-get install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1d-0+deb10u3).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
ltu@debian:~$ su
Password:
root@debian:/home/ltu# cd /etc/ssl/private/
root@debian:/etc/ssl/private# openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
root@debian:/etc/ssl/private# openssl -h
Invalid command '-h'; type "help" for a list.
root@debian:/etc/ssl/private# openssl help
Standard commands

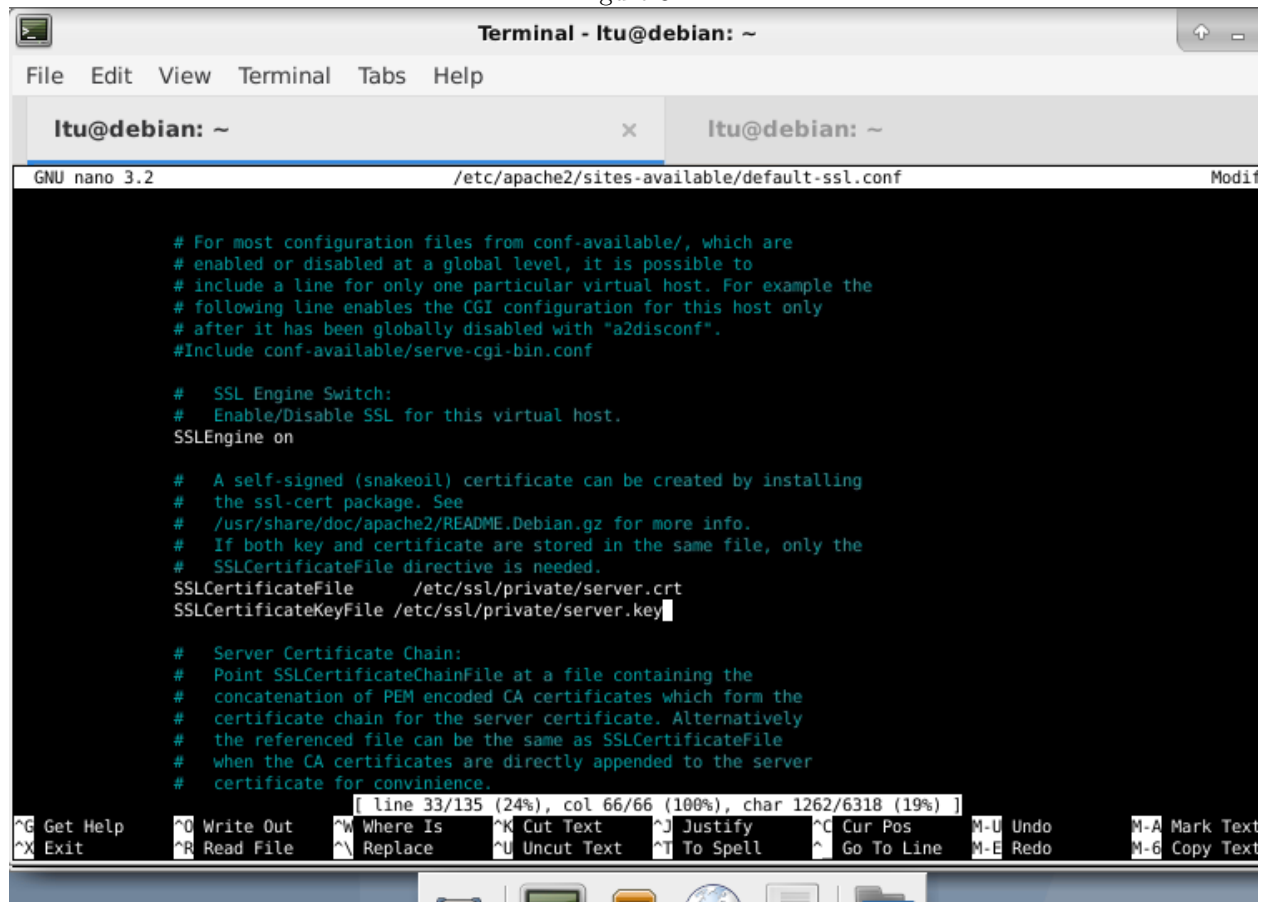
```

Figure 7: removing the passphrase from the private key and requesting and generating a certificate, signing it with the generated RSE private key (*openssl x509*).

```
root@debian:/etc/ssl/private# openssl rsa -in server.key -out server.key
Enter pass phrase for server.key:
writing RSA key
root@debian:/etc/ssl/private# openssl req -new -days 3650 -key server.key -out server.csr
Ignoring -days; not generating a certificate
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:secret
Locality Name (eg, city) []:city
Organization Name (eg, company) [Internet Widgits Pty Ltd]:nico
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:nic
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:ltultu
An optional company name []:
root@debian:/etc/ssl/private# openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650
Signature ok
subject=C = DE, ST = secret, L = city, O = nico, CN = nic
Getting Private key
root@debian:/etc/ssl/private#
```

Figure 8:



```
Terminal - ltu@debian: ~
File Edit View Terminal Tabs Help

ltu@debian: ~ x ltu@debian: ~
GNU nano 3.2 /etc/apache2/sites-available/default-ssl.conf Modif

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/private/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
[ line 33/135 (24%), col 66/66 (100%), char 1262/6318 (19%) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo M-6 Copy Text
```

Figure 9: *a2ensite* allows us to enable or disable an apache2 site / virtual host and *a2enmod* enables the specified module within the apache2 configuration, which in this case is SSL

```
root@debian:/etc/ssl/private# sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian:/etc/ssl/private# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:/etc/ssl/private# sudo systemctl restart apache2
root@debian:/etc/ssl/private# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-01-09 03:32:21 CST; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 25726 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 25730 (apache2)
    Tasks: 6 (limit: 4915)
   Memory: 14.5M
   CGroup: /system.slice/apache2.service
           └─25730 /usr/sbin/apache2 -k start
             └─25731 /usr/sbin/apache2 -k start
               └─25732 /usr/sbin/apache2 -k start
                 └─25733 /usr/sbin/apache2 -k start
                   └─25734 /usr/sbin/apache2 -k start
                     └─25735 /usr/sbin/apache2 -k start
```


Figure 10: as we can see an unknown certificate is used

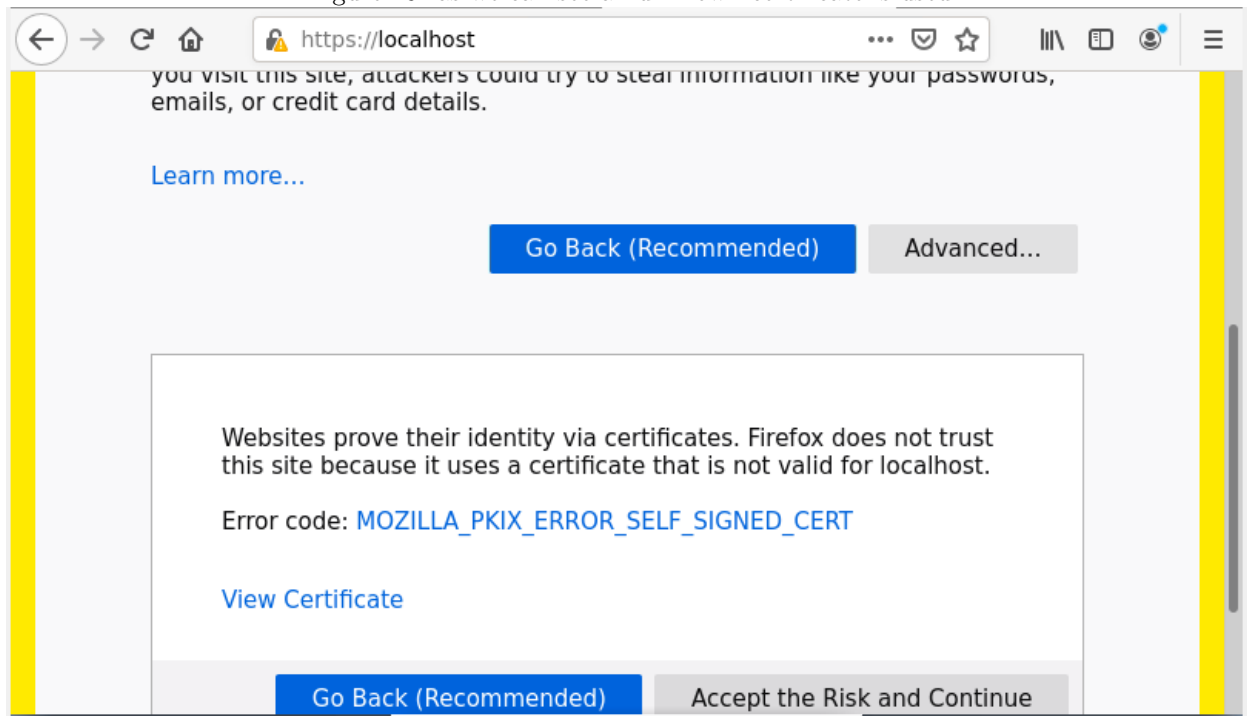


Figure 11: through Firefox we can see the information of the trusted certificates, and in this case we can analyze the certificate we have just signed as trusted and we see the information we entered during its creation, as shown in Figure 7

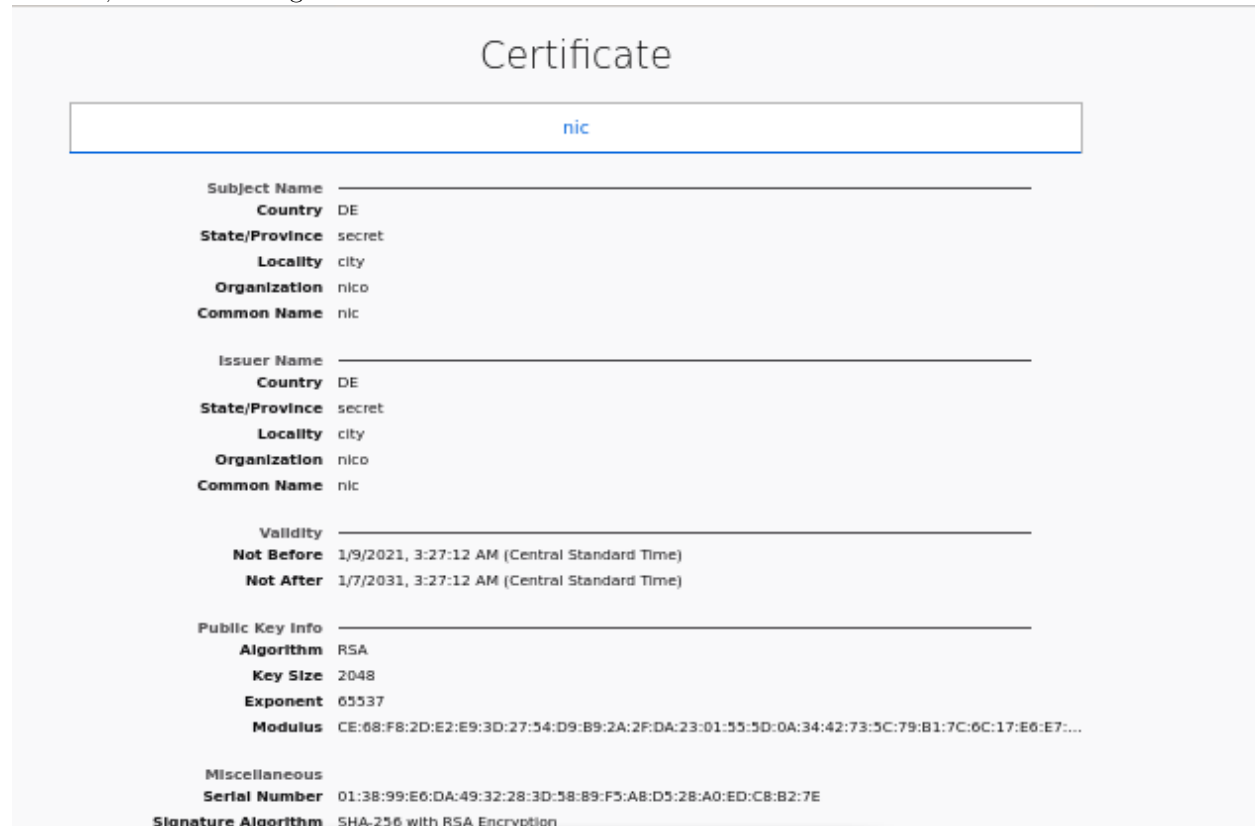


Figure 12:

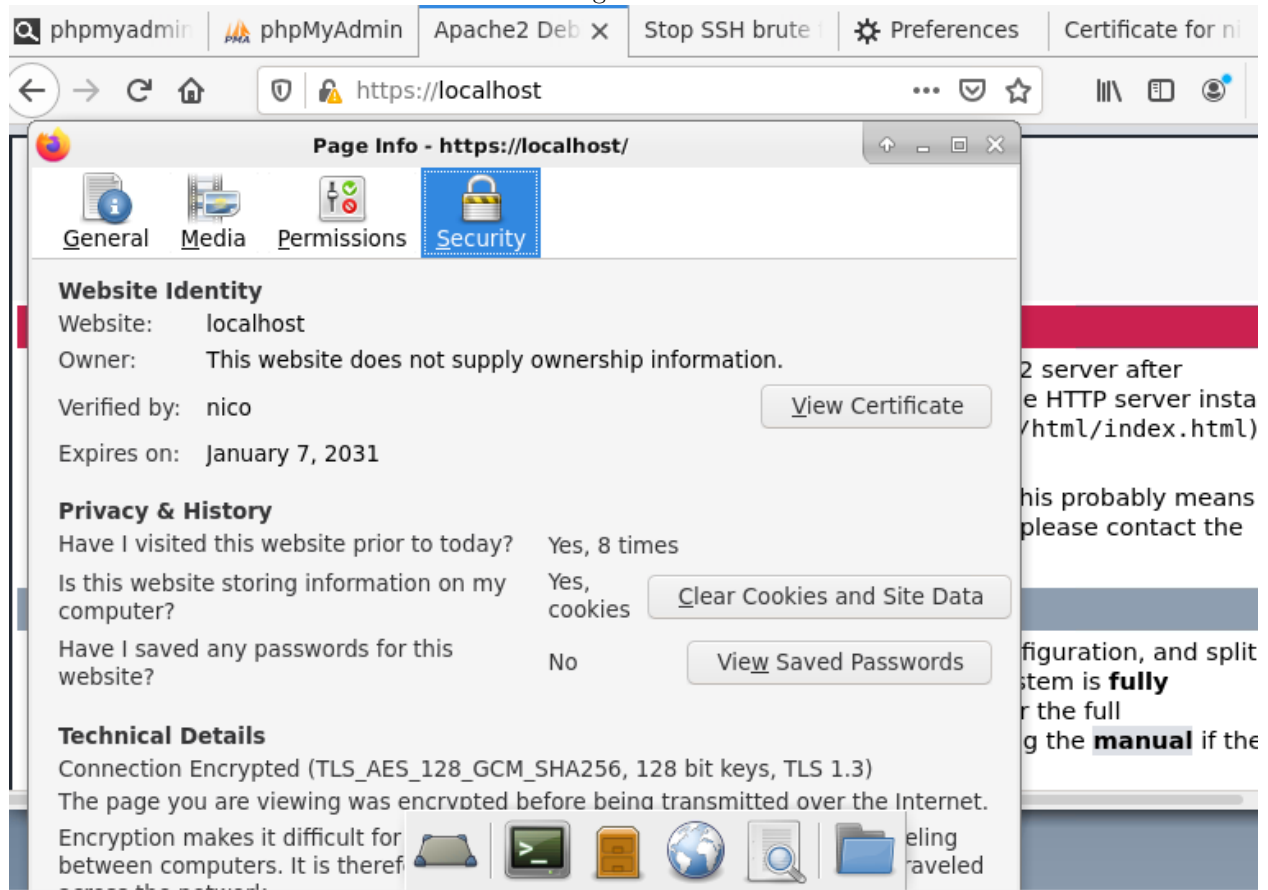


Figure 13: applying rules which filters FTP traffic gives us similar results to the ones presented for SSH traffic filtering. we can see from NMAP that the port results filtered after attempting more than 4 logins.

```

root@kali: ~
Nmap done: 1 IP address (1 host up) scanned in 0.59 seconds

root@kali: ~
# nmap 10.0.0.30
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-09 08:24 EST
Nmap scan report for 10.0.0.30
Host is up (0.00067s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
21/tcp    filtered ftp
22/tcp    filtered ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
MAC Address: 08:00:27:AE:38:33 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds

root@kali: ~
# nmap 10.0.0.30
Medium 443/tcp 22
Medium 6697/tcp 23
Medium 5432/tcp 24
Medium 21/tcp 38
Medium 22/tcp 39
Low general/tcp 40
Low 22/tcp 41

2 RESULTS PER
postgres:postgres
service:service
user:user

Solution
Solution type:
Change the passw

Vulnerability D
Reports weak/k
  
```

```

xHydra
Quit
Target Passwords Tuning Specific Start
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaar" - 18 of 456976 [child 0] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaas" - 19 of 456976 [child 14] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaat" - 20 of 456976 [child 0] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaau" - 21 of 456976 [child 0] (0/0)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaav" - 22 of 456976 [child 14] (0/0)
[VERBOSE] Retrying connection for child 1
[VERBOSE] Retrying connection for child 9
[VERBOSE] Retrying connection for child 10
[VERBOSE] Retrying connection for child 11
[VERBOSE] Retrying connection for child 12
[VERBOSE] Retrying connection for child 2
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaab" - 22 of 456984 [child 1] (0/8)
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaaj" - 22 of 456984 [child 9] (0/8)
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaak" - 22 of 456984 [child 10] (0/8)
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaal" - 22 of 456984 [child 11] (0/8)
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaam" - 22 of 456984 [child 12] (0/8)
[RE-ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaac" - 22 of 456984 [child 2] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaaw" - 23 of 456984 [child 3] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaax" - 24 of 456984 [child 4] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaay" - 25 of 456984 [child 5] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaz" - 26 of 456984 [child 6] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aaba" - 27 of 456984 [child 7] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aabb" - 28 of 456984 [child 8] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aabc" - 29 of 456984 [child 13] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aabd" - 30 of 456984 [child 15] (0/8)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aabe" - 31 of 456986 [child 0] (0/10)
[ATTEMPT] target 10.0.0.30 - login "ituu" - pass "aabl" - 32 of 456986 [child 14] (0/10)
[STATUS] 32.00 tries/min, 32 tries in 00:01h, 456954 to do in 237:60h, 16 active
  
```

Figure 14: now we set some password policies

```

GNU nano 3.2 /etc/login.defs Modified

UMASK          022

#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password chang$
#      PASS_WARN_AGE   Number of days warning given before a password expires$
#
PASS_MAX_DAYS   60
PASS_MIN_DAYS   2
PASS_WARN_AGE   10

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN         1000
UID_MAX         60000
# System accounts
File Name to Write: /etc/login.defs
^G Get Help      M-D DOS Format   M-A Append      M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend     ^T To Files

```

Figure 15: Thanks to the module *libpam-cracklib* we can set other policies.

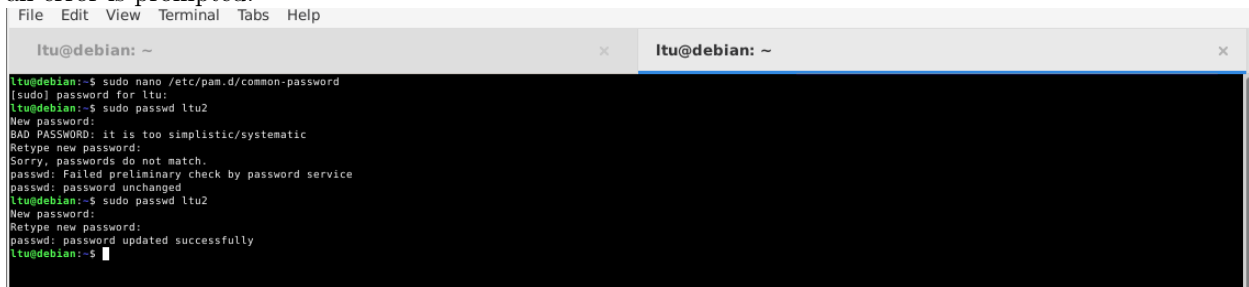
```

GNU nano 3.2 /etc/pam.d/common-password

# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.
#
# Explanation of pam_unix options:
#
# The "sha512" option enables salted SHA512 passwords. Without this option,
# the default is Unix crypt. Prior releases used the option "md5".
#
# The "obscure" option replaces the old 'OBSCURE_CHECKS_ENAB' option in
# login.defs.
#
# See the pam_unix manpage for other options.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
#
# here are the per-package modules (the "Primary" block)
password      requisite      pam_unix.so minlen=10 reject_username difok=3
password      requisite      pam_cracklib.so retry=1 dcredit=-1 lcredit=-1 minlen=10 reject_username difok=3
password      requisite      pam_cracklib.so retry=3 minlen=8 difok=3
password      [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass sha512
# here's the fallback if no module succeeds
password      requisite      pam_permit.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password      required       pam_permit.so
# and here are more per-package modules (the "Additional" block)

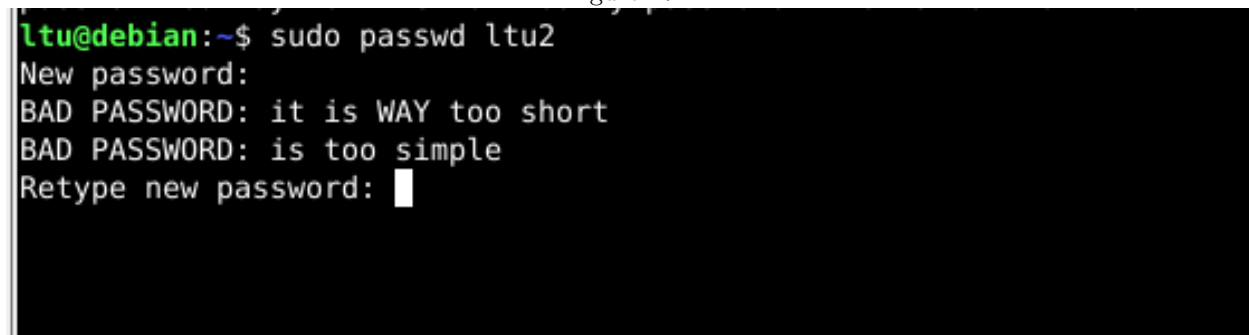
```

Figure 16: if we try to change the password to an user using password which are not accepted by the policies, an error is prompted.



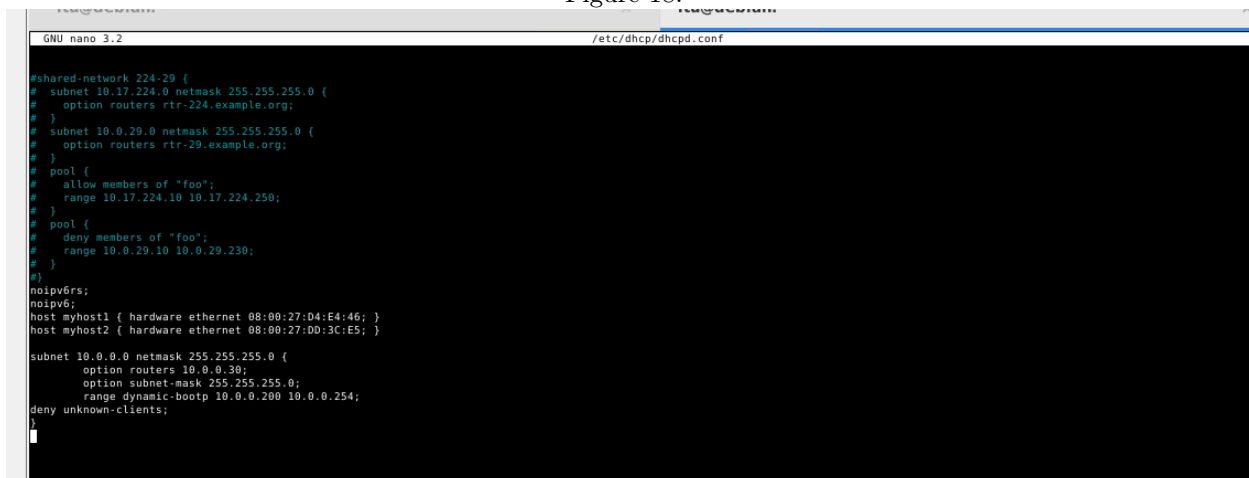
```
ltu@debian: ~  
ltu@debian:~$ sudo nano /etc/pam.d/common-password  
[sudo] password for ltu:  
ltu@debian:~$ sudo passwd ltu2  
New password:  
BAD PASSWORD: it is too simplistic/systematic  
Retype new password:  
Sorry, passwords do not match.  
passwd: Failed preliminary check by password service  
passwd: password unchanged  
ltu@debian:~$ sudo passwd ltu2  
New password:  
Retype new password:  
passwd: password updated successfully  
ltu@debian:~$
```

Figure 17:



```
ltu@debian:~$ sudo passwd ltu2  
New password:  
BAD PASSWORD: it is WAY too short  
BAD PASSWORD: is too simple  
Retype new password: █
```

Figure 18:



```
GNU nano 3.2 /etc/dhcp/dhcpd.conf  
#shared-network 224.29 {  
# subnet 10.17.224.0 netmask 255.255.255.0 {  
# option routers rtr-224.example.org;  
# }  
# subnet 10.0.29.0 netmask 255.255.255.0 {  
# option routers rtr-29.example.org;  
# }  
# pool {  
# allow members of "foo";  
# range 10.17.224.10 10.17.224.250;  
# }  
# pool {  
# deny members of "foo";  
# range 10.0.29.10 10.0.29.230;  
# }  
#}  
noipw6rs;  
noipv6;  
host myhost1 { hardware ethernet 08:00:27:04:E4:46; }  
host myhost2 { hardware ethernet 08:00:27:00:3C:ES; }  
  
subnet 10.0.0.0 netmask 255.255.255.0 {  
option routers 10.0.0.30;  
option subnet-mask 255.255.255.0;  
range dynamic-bootp 10.0.0.200 10.0.0.254;  
deny unknown-clients;  
}
```

Figure 19:

```
ltug@debian:~$ sudo nano -c /etc/dhcp/dhcpd.conf
ltug@debian:~$ sudo systemctl restart isc-dhcp-server.service
ltug@debian:~$ sudo systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Sat 2021-01-09 07:59:24 CST; 7s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 1613 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 4915)
   Memory: 4.9M
   CGroup: /system.slice/isc-dhcp-server.service
           └─1626 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp0s8

Jan 09 07:59:22 debian systemd[1]: Starting LSB: DHCP server...
Jan 09 07:59:22 debian isc-dhcp-server[1613]: Launching IPv4 server only.
Jan 09 07:59:22 debian dhcpd[1626]: Wrote 0 deleted host decls to leases file.
Jan 09 07:59:22 debian dhcpd[1626]: Wrote 0 new dynamic host decls to leases file.
Jan 09 07:59:22 debian dhcpd[1626]: Wrote 7 leases to leases file.
Jan 09 07:59:22 debian dhcpd[1626]: Server starting service.
Jan 09 07:59:24 debian isc-dhcp-server[1613]: Starting ISC DHCPv4 server: dhcpd.
Jan 09 07:59:24 debian systemd[1]: Started LSB: DHCP server.
ltug@debian:~$
```

Figure 20: After setting a filter in the DHCP server allowing only known MAC addresses, i try to change the MAC address of the Kali VM

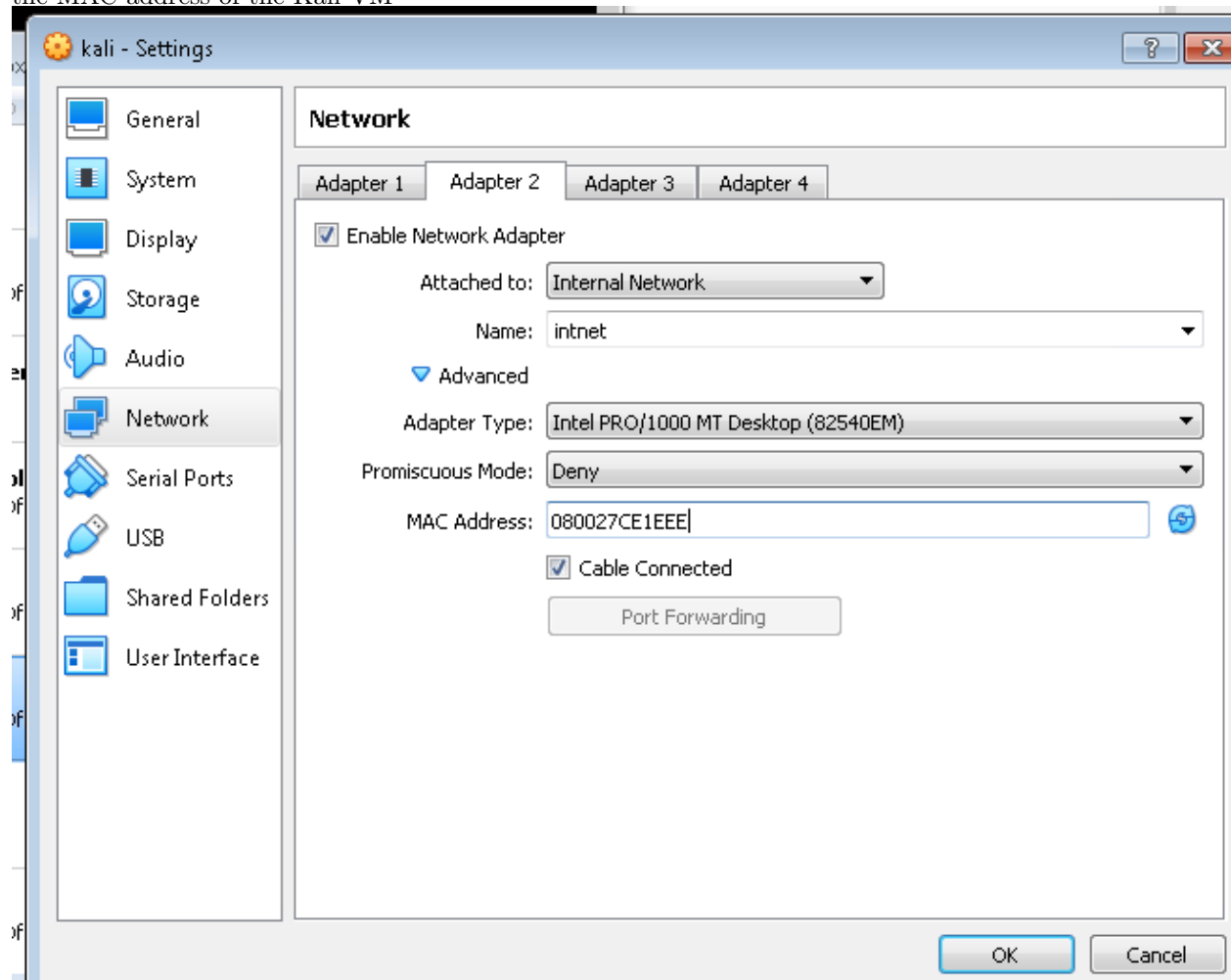


Figure 21: as we can see the VM doesn't receive any IP address from the DHCP server, due to its unknown MAC address to the server.

```

user@kali: ~
File Actions Edit View Help

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::a00:27ff:face:1eee prefixlen 64 scopeid 0<link>
    ether 08:00:27:ce:1e:ee txqueuelen 1000 (Ethernet)
    RX packets 130 bytes 9161 (8.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23 bytes 5102 (4.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 640 (640.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 640 (640.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(user@kali)-[~]
$ ping 10.0.0.30
ping: connect: Network is unreachable

```

Figure 22: If the MAC is changed back to the one specified in the known hosts in the server, it will connect to the network.

```

File Actions Edit View Help

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 16 bytes 796 (796.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 796 (796.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(user@kali)-[~]
$ ping 10.0.0.30
PING 10.0.0.30 (10.0.0.30) 56(84) bytes of data:
64 bytes from 10.0.0.30: icmp_seq=1 ttl=64 time=0.915 ms
64 bytes from 10.0.0.30: icmp_seq=2 ttl=64 time=1.07 ms
^C
-- 10.0.0.30 ping statistics --
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.915/0.990/1.065/0.075 ms

(user@kali)-[~]
$

```

Oracle VM VirtualBox - Settings

Network

Adapter 1: Adapter 2: Adapter 3: Adapter 4:

☒ Enable Network Adapter

Attached to: Internal Network

Name: Internet

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08:00:27:CE:1E:EE

☒ Cable Connected

☐ Port Forwarding

Figure 23:

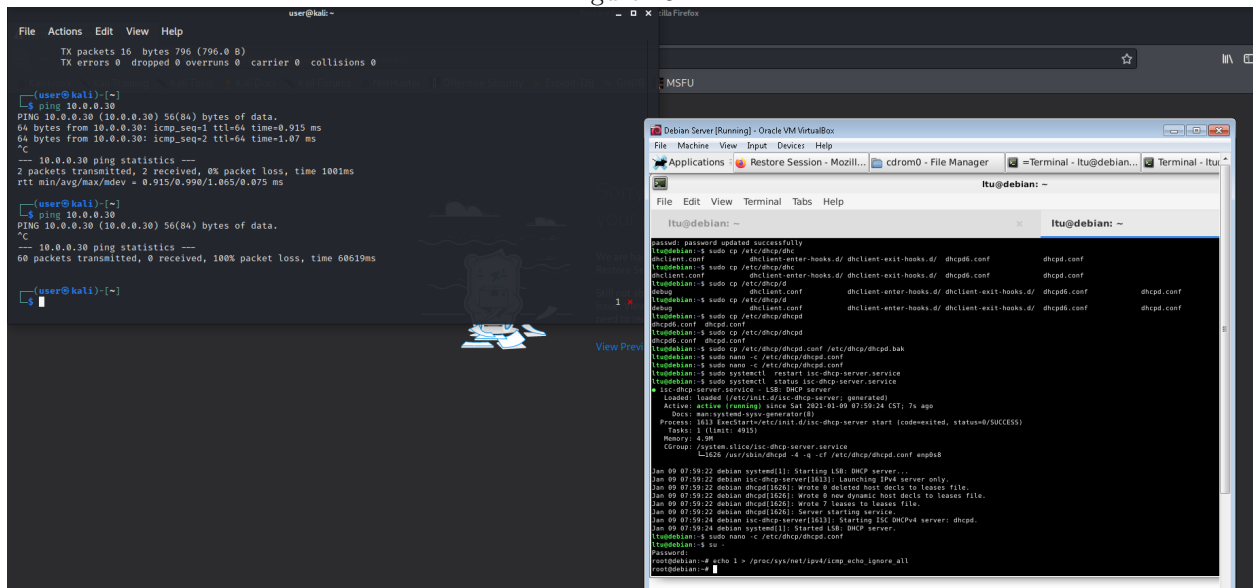


Figure 24: now the ICCMP traffic is ignored by the server, and we can see that the ping command will not receive any reply.

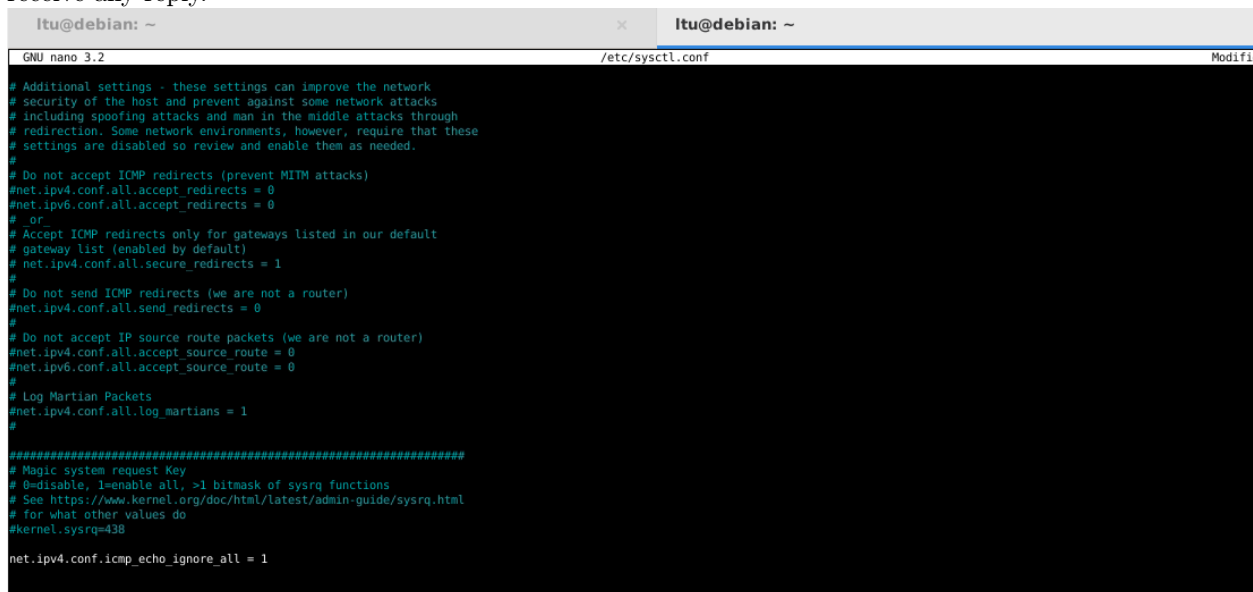


Figure 25:

```

tu@debian:~$ su -
password:
root@debian:~# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
root@debian:~# nano /etc/sysctl.conf
root@debian:~# sudo sysctl --system
Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.ip_forward = 1
Applying /etc/sysctl.d/protect-links.conf ...
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
Applying /etc/sysctl.conf ...
net.ipv4.ip_forward = 1
root@debian:~# sudo reboot

```

Figure 26:

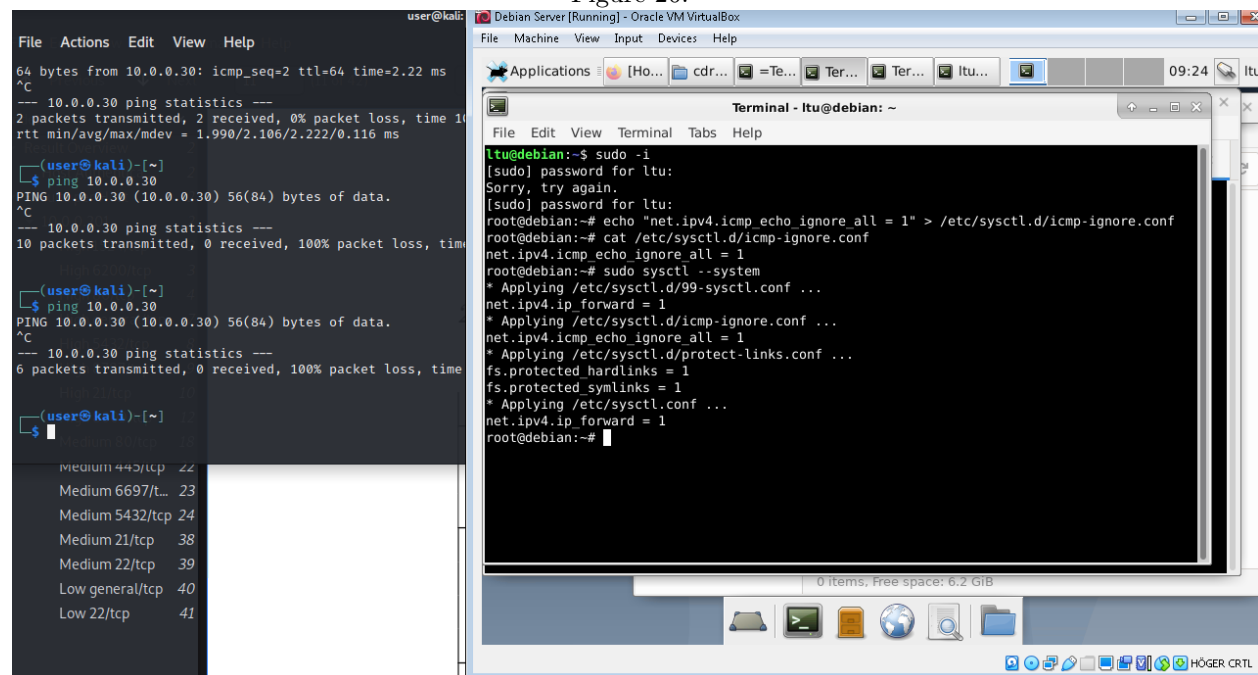


Figure 27:

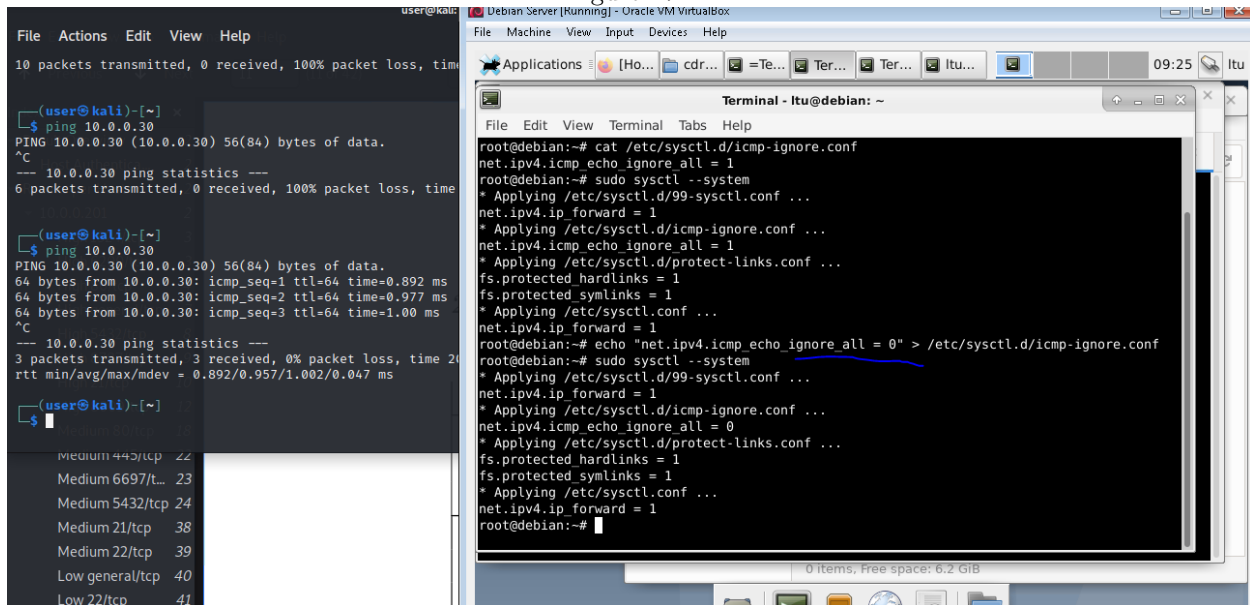


Figure 28: we harden the Mysql PHPmyadmin installation by setting policies such as allowing only "strong" passwords, disallowing root remote acces, removing test database...

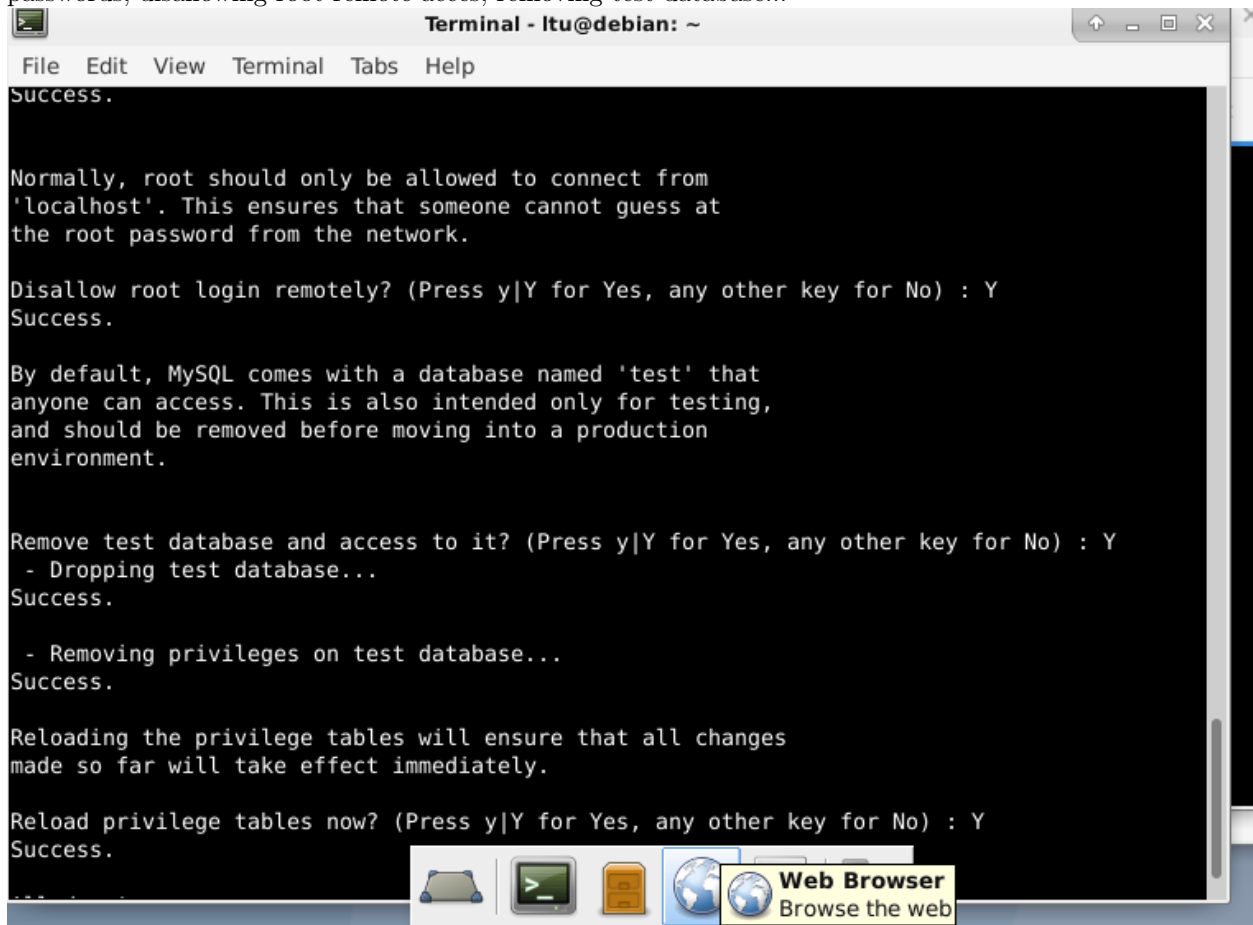
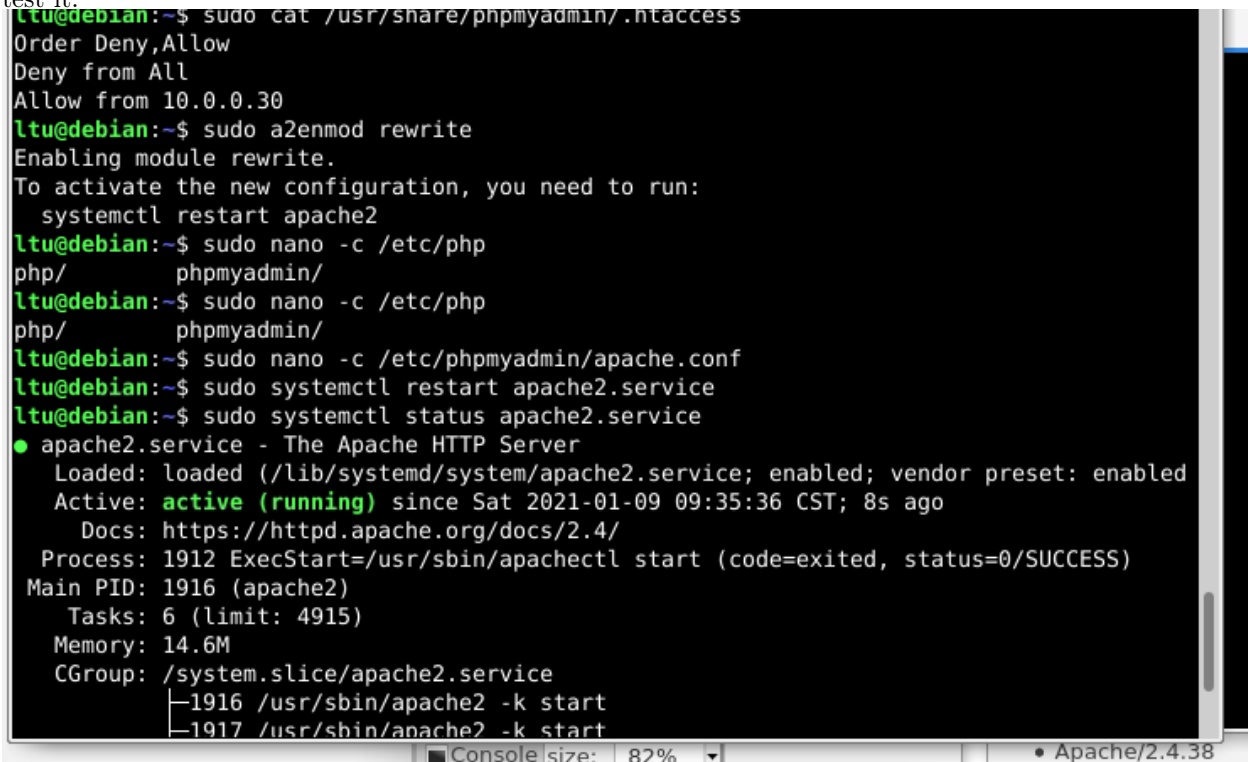


Figure 29: we allow access to phpmyadmin only from the server IP and after reloading the service we can test it.



```
ltu@debian:~$ sudo cat /usr/share/phpmyadmin/.htaccess
Order Deny,Allow
Deny from All
Allow from 10.0.0.30
ltu@debian:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
ltu@debian:~$ sudo nano -c /etc/php
php/      phpmyadmin/
ltu@debian:~$ sudo nano -c /etc/php
php/      phpmyadmin/
ltu@debian:~$ sudo nano -c /etc/phpmyadmin/apache.conf
ltu@debian:~$ sudo systemctl restart apache2.service
ltu@debian:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-01-09 09:35:36 CST; 8s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 1912 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 1916 (apache2)
    Tasks: 6 (limit: 4915)
   Memory: 14.6M
    CGroup: /system.slice/apache2.service
            └─1916 /usr/sbin/apache2 -k start
              └─1917 /usr/sbin/apache2 -k start
```

Console size: 82% | Apache/2.4.38

Figure 30: from the Debian machine it result possible to access to phpmyadmin, as expected.

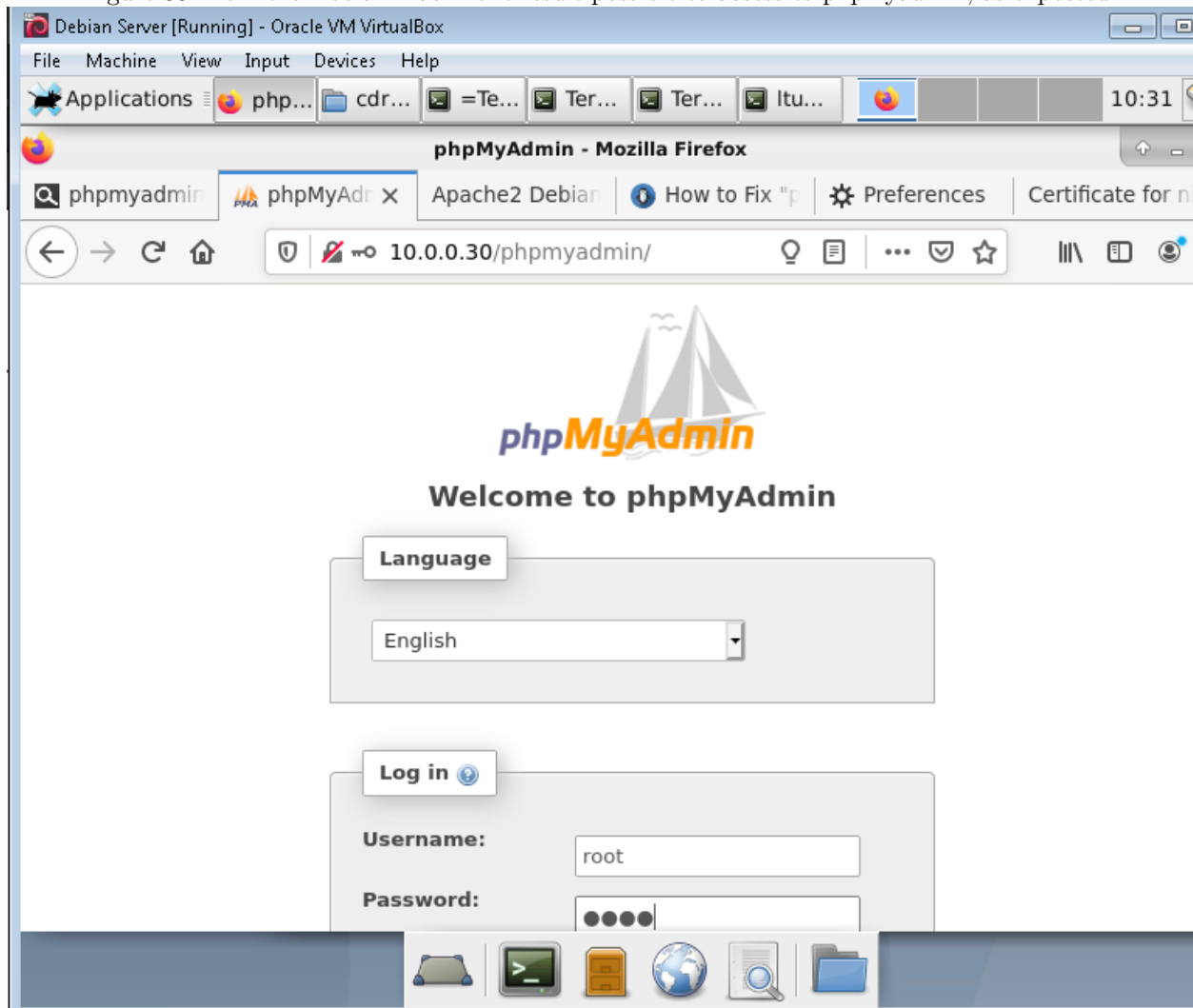
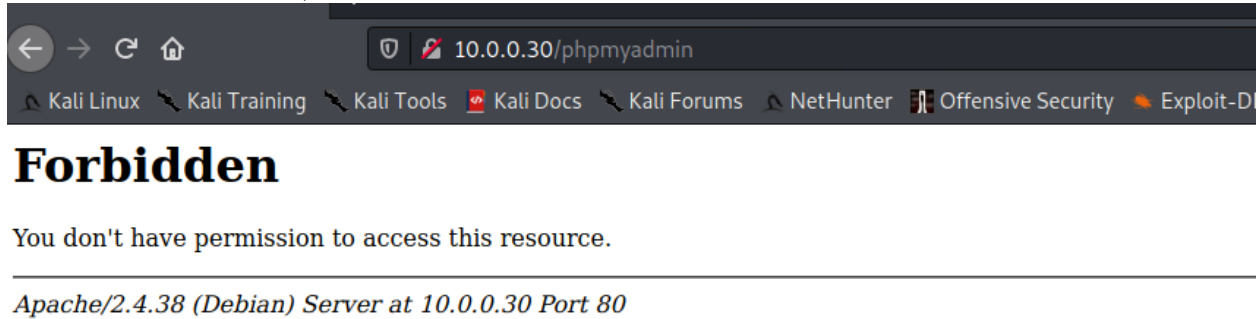


Figure 31: from the kali VM it is not possible to access to phpmyadmin, since the access is allowed only from the Debian machine, with IP *10.0.0.30*



4. thoughts about this week

This topic introduced me the SDN, which I didn't know before. Really interesting also the lab.