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. morover, becomes foundamental 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 peeks 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 identiy 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.

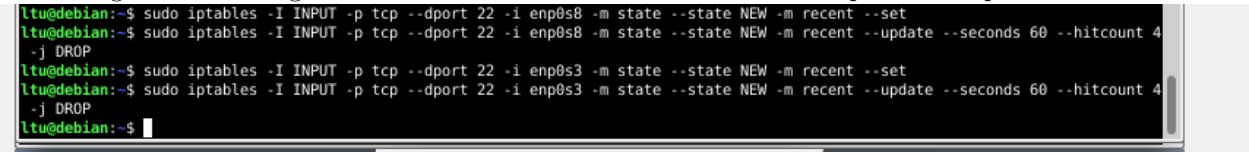


Figure 2: before applying the rules the port 22 results open.

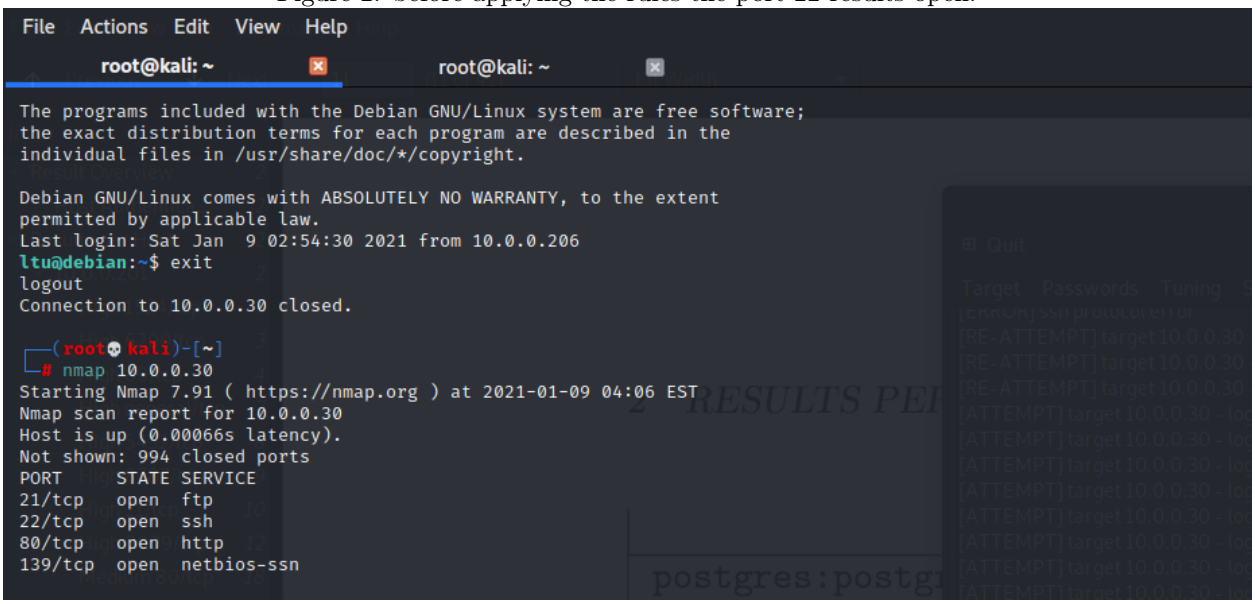


Figure 3: when we apply the rules the port results filtered and while we try to attempt multiple logins, xhydra presents some error messages due to the drop of our packets after more than 4 failed attempt.

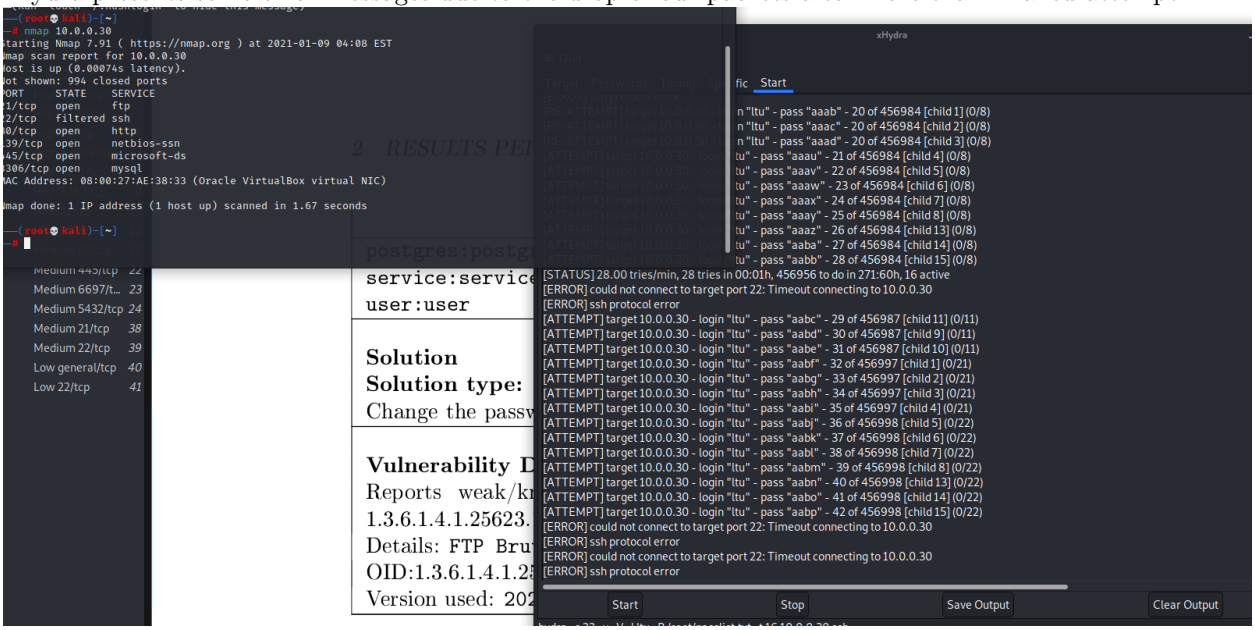


Figure 4: we can see that if we try to connect to SSH, the connection doesn't occurs due to the multiple failed connections.

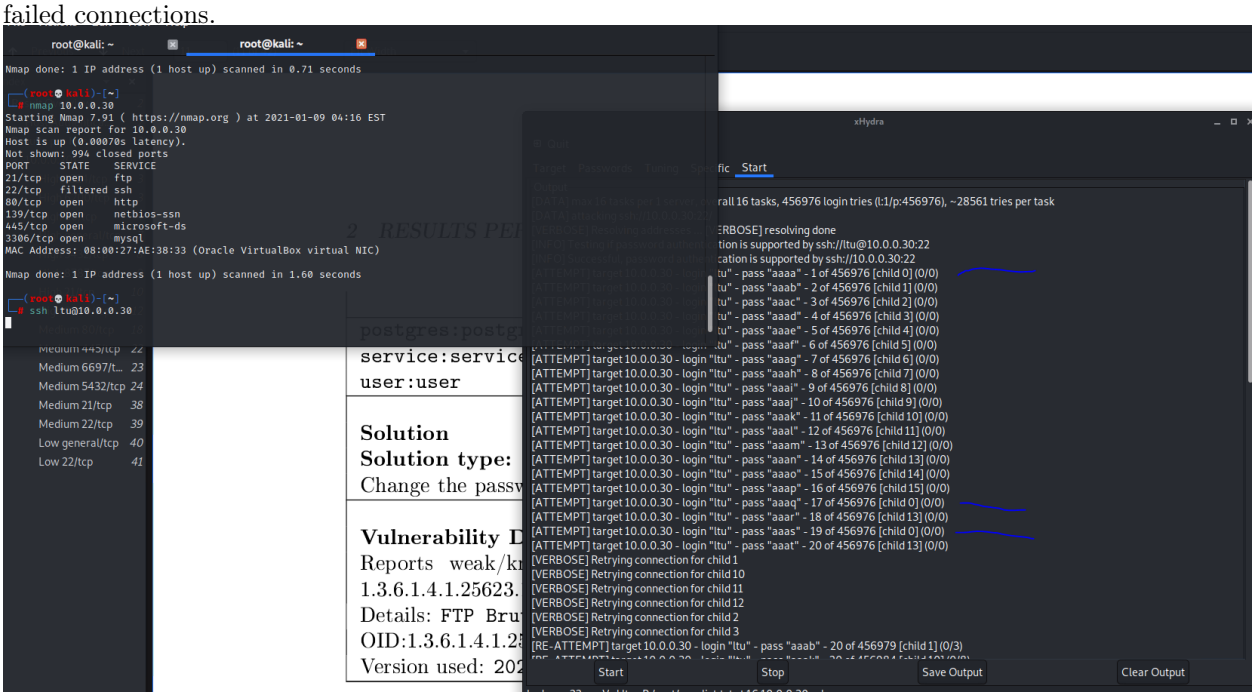


Figure 5: if the rules are removed (by changing the parameter *-I* to *-D*), the port will be open again and will be allowed to try multiple attempts.

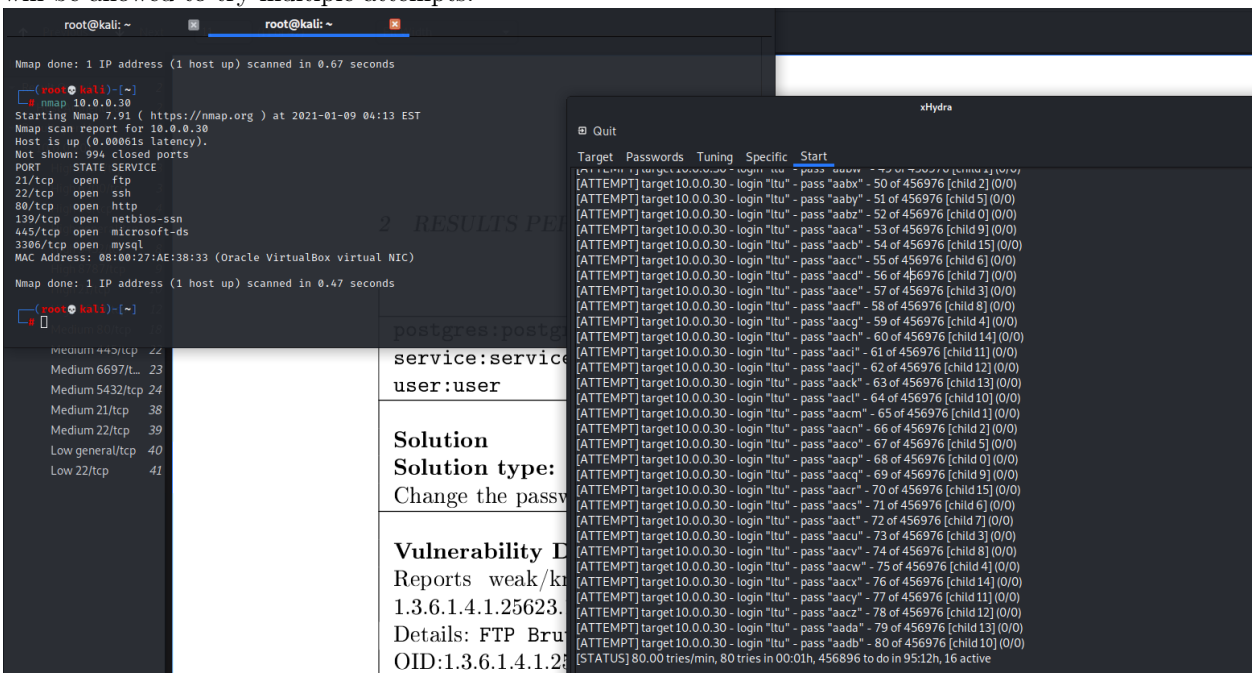


Figure 6: a RSA private key is generated and in the command , the parameter *-des3* states the cipher used to encrypt private key before outputting it in the file specified after the *-out* parameter and *2048* is size of the private key to generate in bits.

```
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
```

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*).

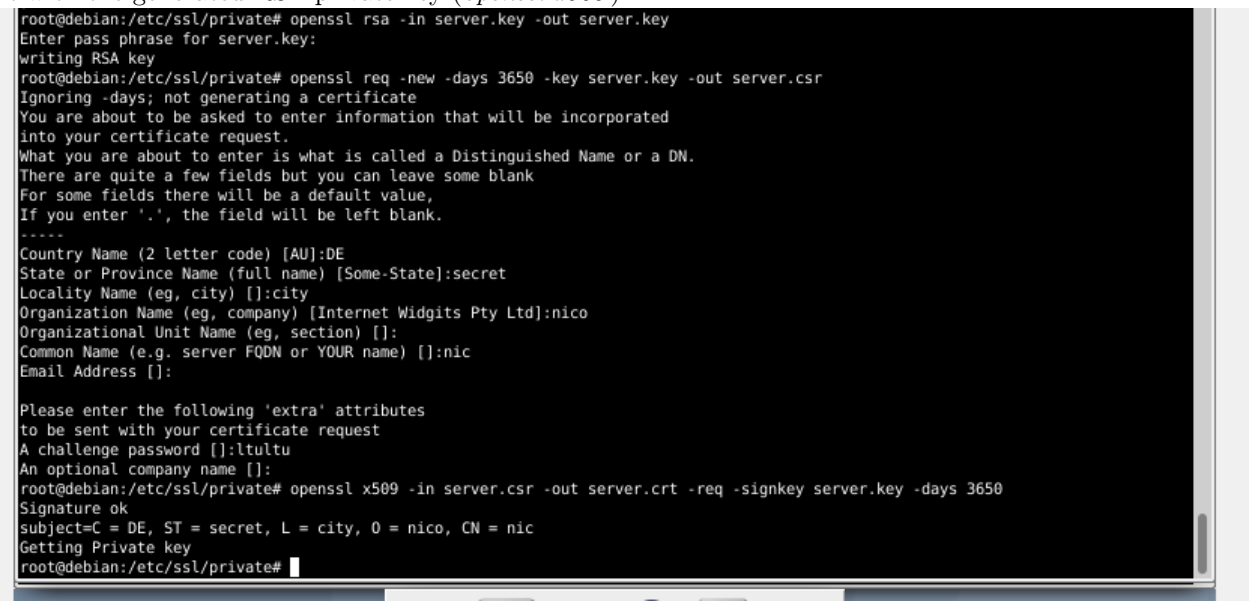


Figure 8:

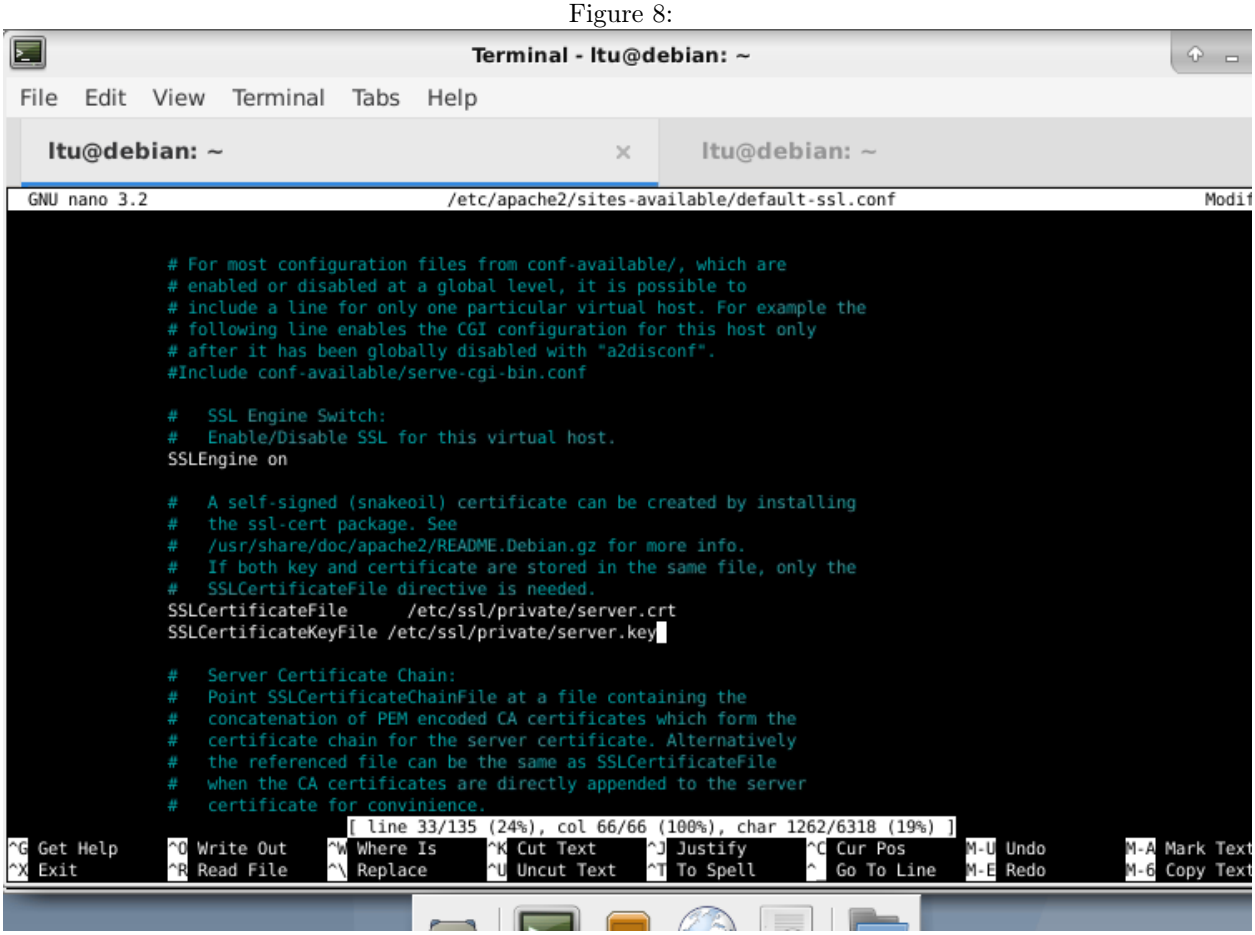

Terminal - ltu@debian: ~
File Edit View Terminal Tabs Help
ltu@debian: ~
ltu@debian: ~
GNU nano 3.2 /etc/apache2/sites-available/default-ssl.conf
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular 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

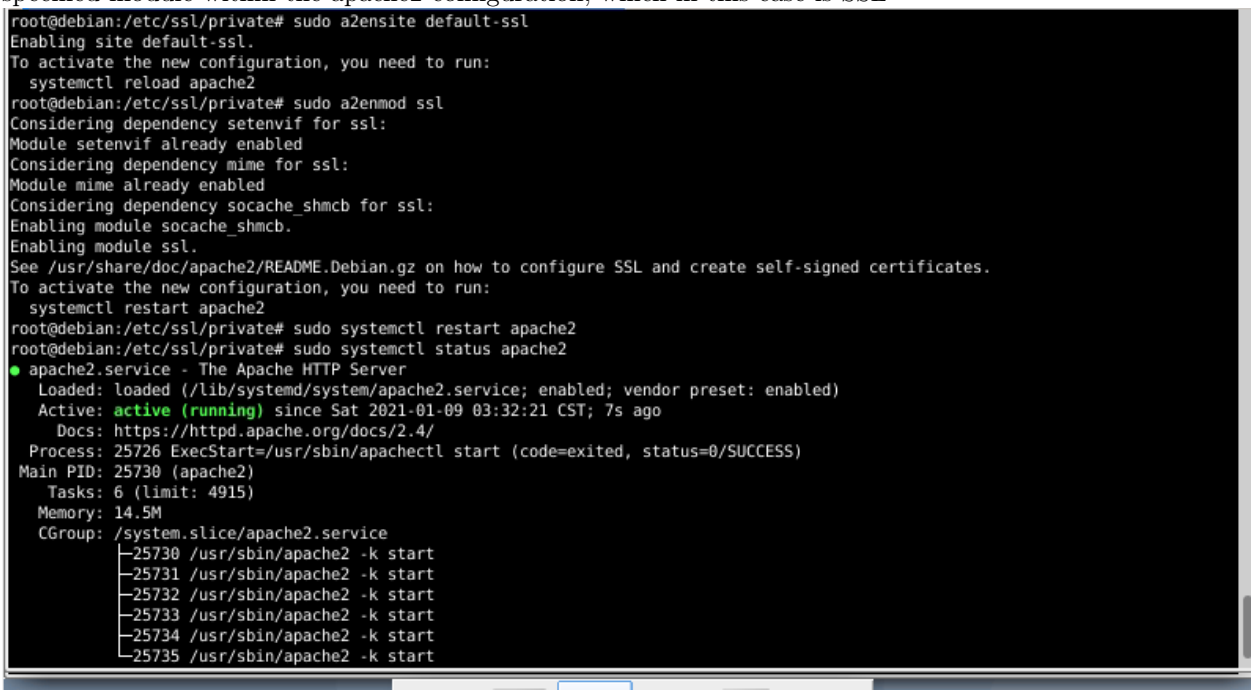


Figure 10: as we can see an unknown certificate is used

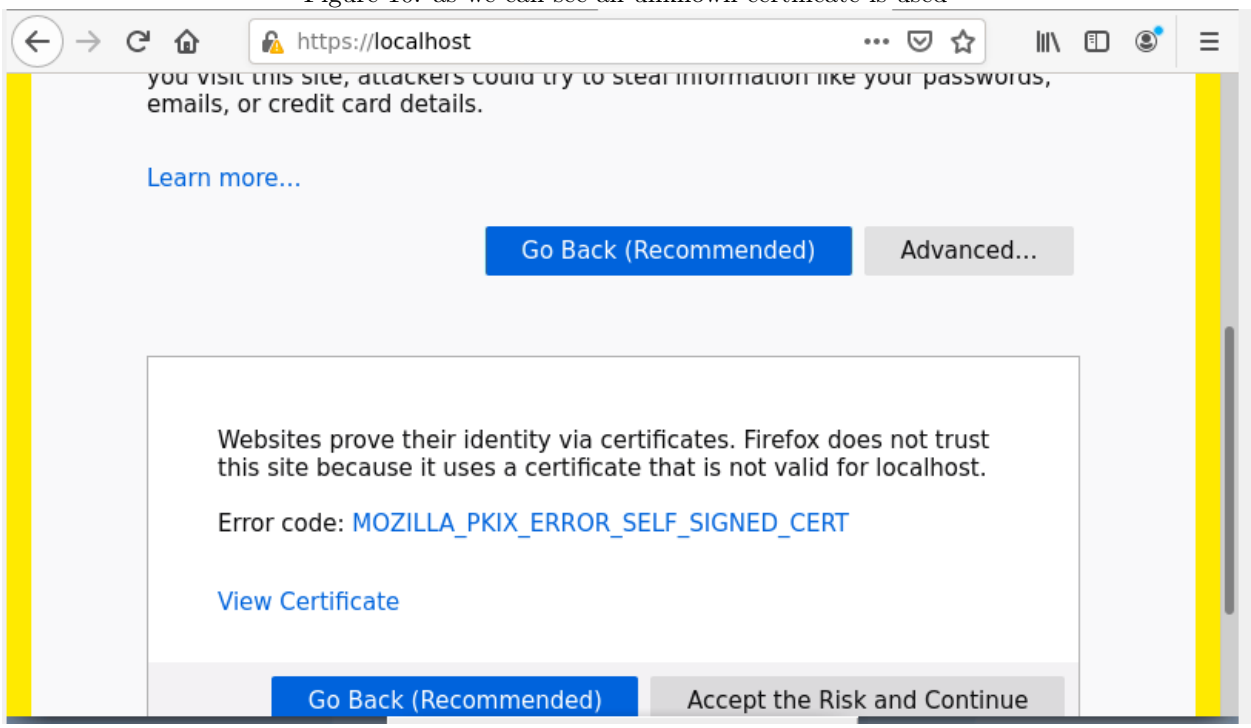


Figure 11: through Firefox we can see the information of the trusted certificates, and in this case we can analyze the certificate we have just signed as trusted and we see the information we entered during its creation, as shown in Figure 7

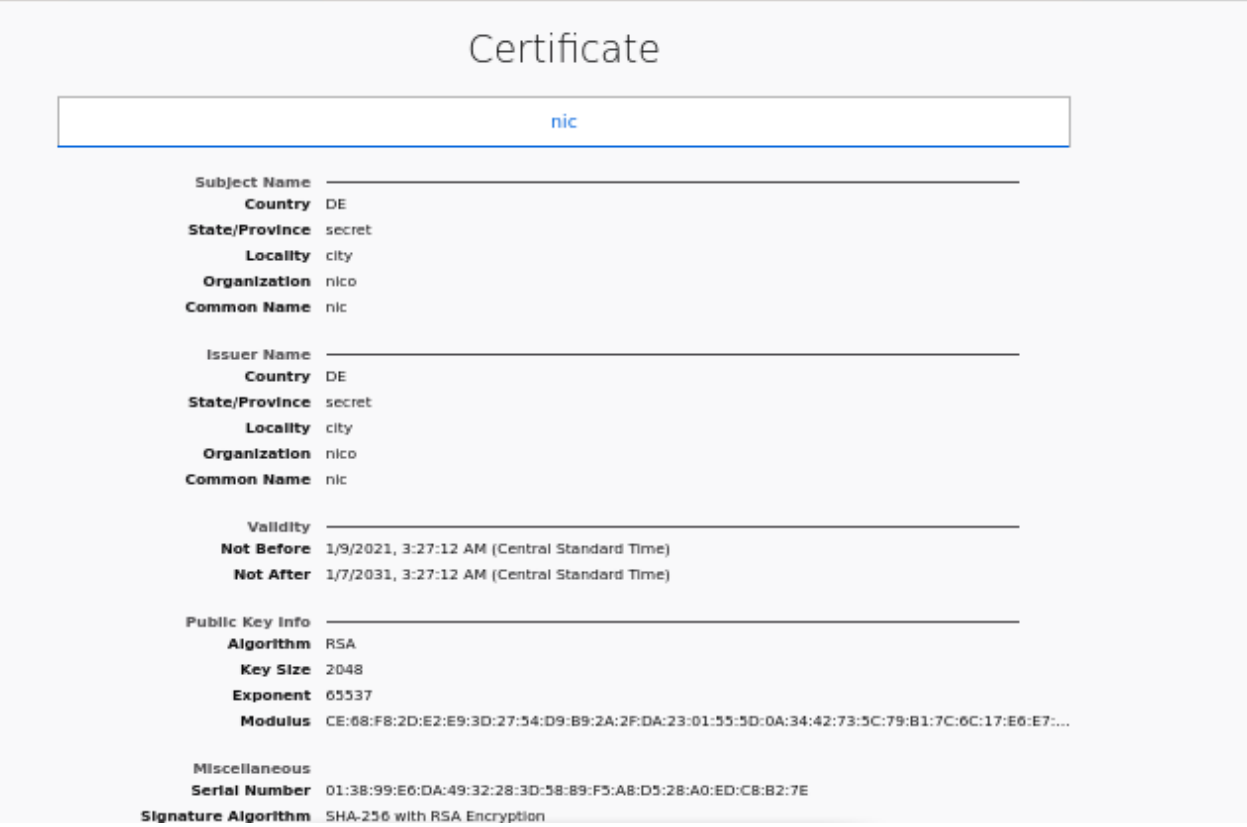


Figure 12:

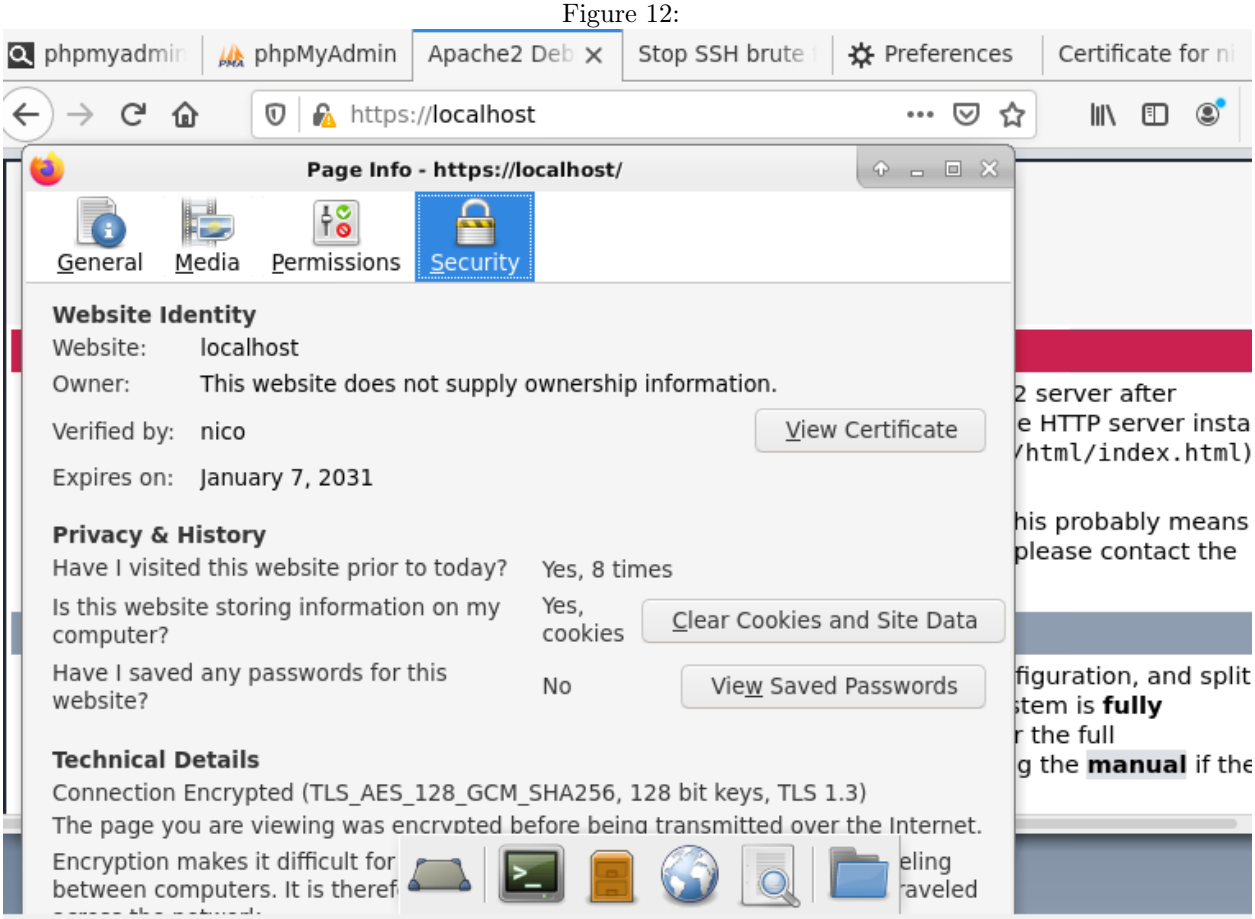

phpmyadmin
phpMyAdmin
Apache2 Deb
Stop SSH brute
Preferences
Certificate for ni
https://localhost
Page Info - https://localhost/
General
Media
Permissions
Security
Website Identity
Website: localhost
Owner: This website does not supply ownership information.
Verified by: nico
View Certificate
Expires on: January 7, 2031
Privacy & History
Have I visited this website prior to today? Yes, 8 times
Is this website storing information on my computer? Yes, cookies
Clear Cookies and Site Data
Have I saved any passwords for this website? No
View Saved Passwords
Technical Details
Connection Encrypted (TLS_AES_128_GCM_SHA256, 128 bit keys, TLS 1.3)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for
between computers. It is theref
2 server after
e HTTP server insta
/html/index.html)
his probably means
please contact the
figuration, and split
stem is fully
r the full
g the manual if the

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.

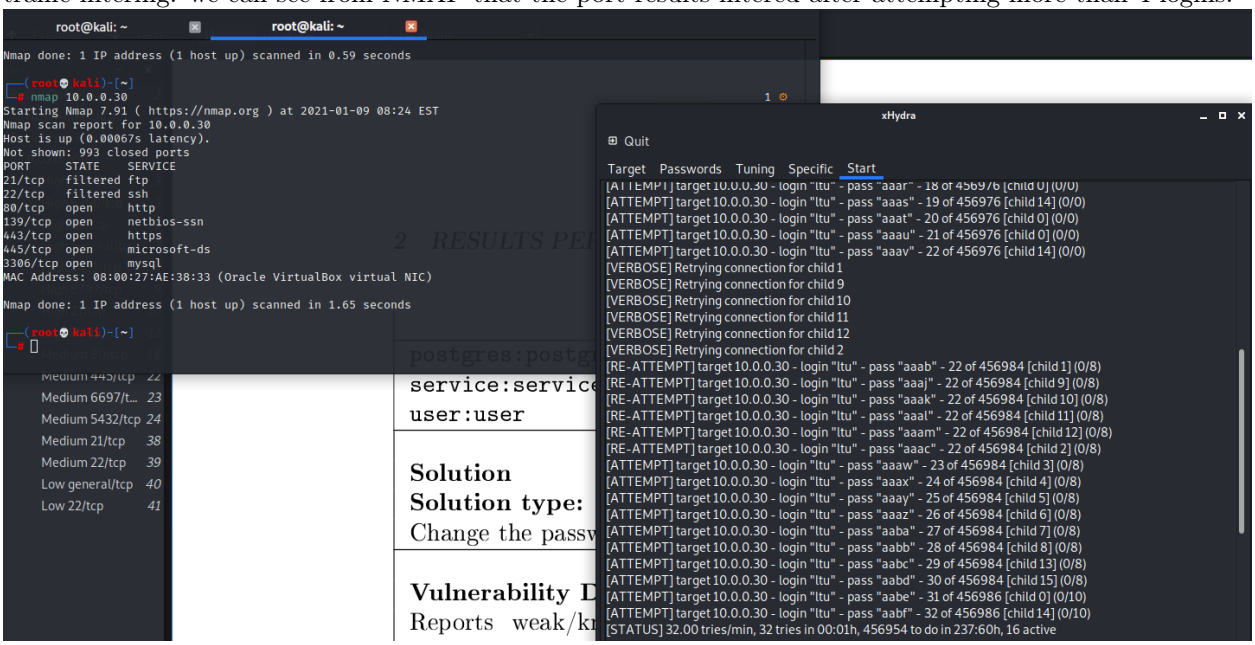


Figure 14: now we set some password policies

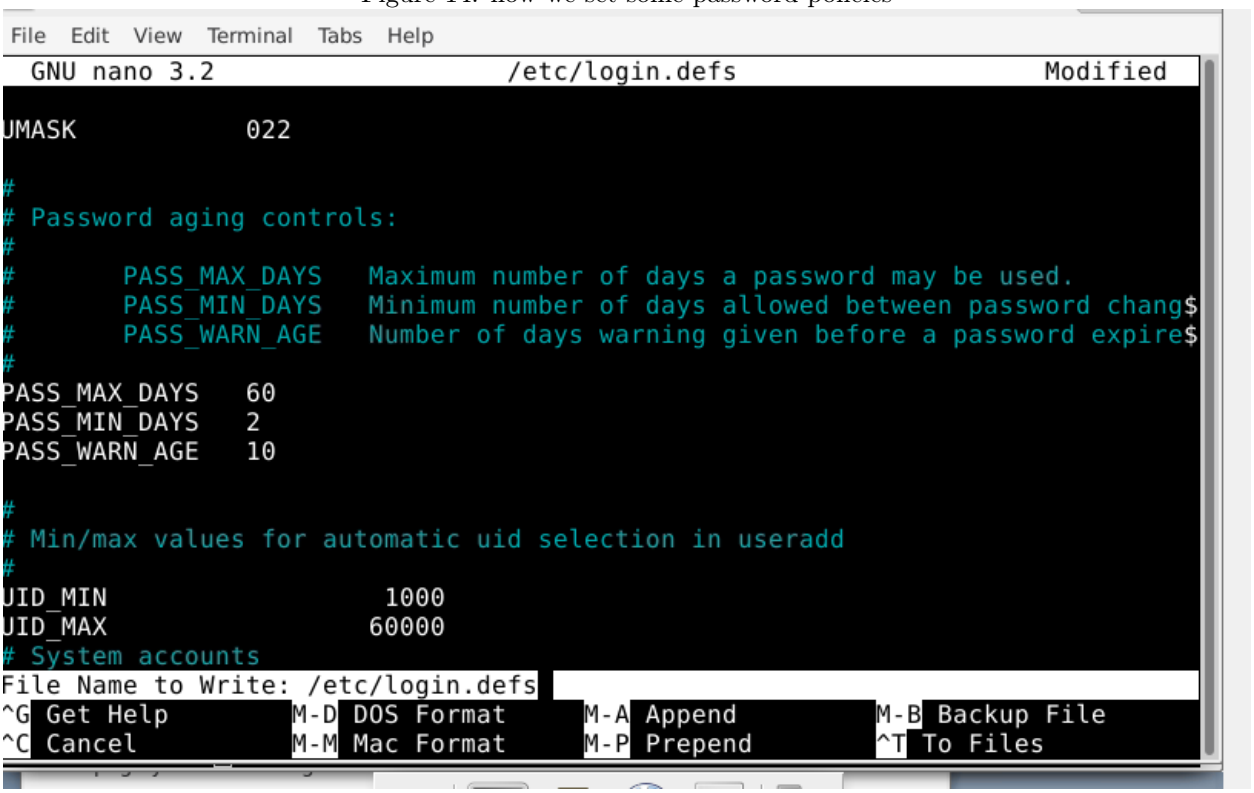


Figure 15: Thanks to the module *libpam-cracklib* we can set other policies.

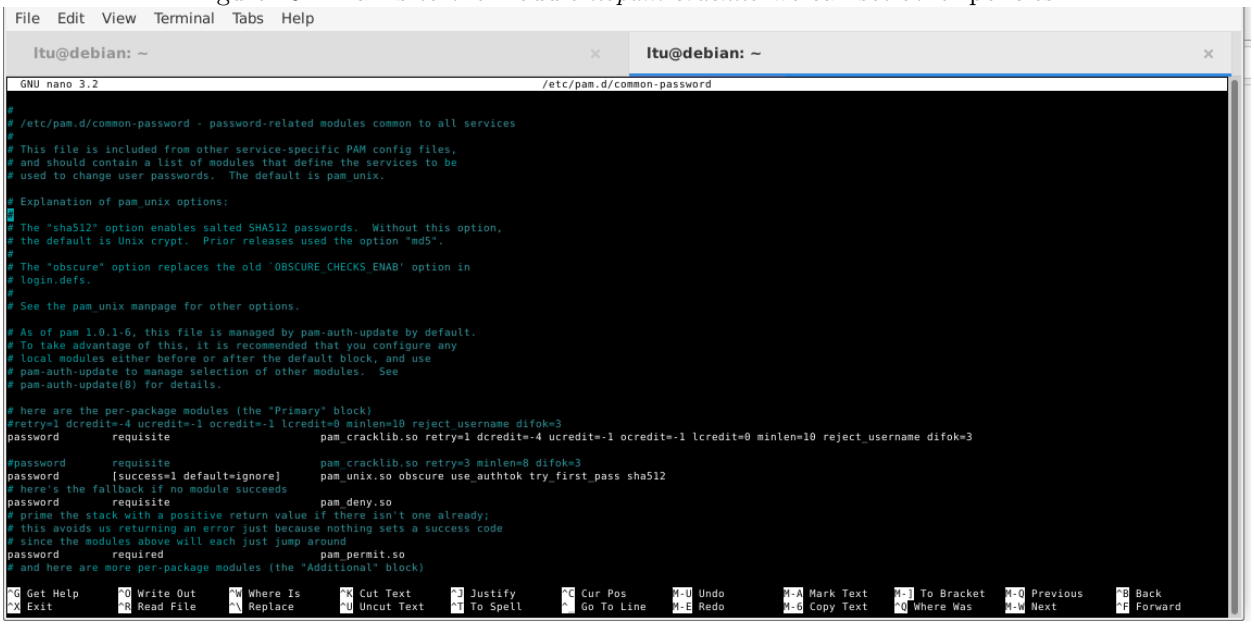


Figure 16: if we try to change the password to an user using password which are not accepted by the policies, an error is prompted.

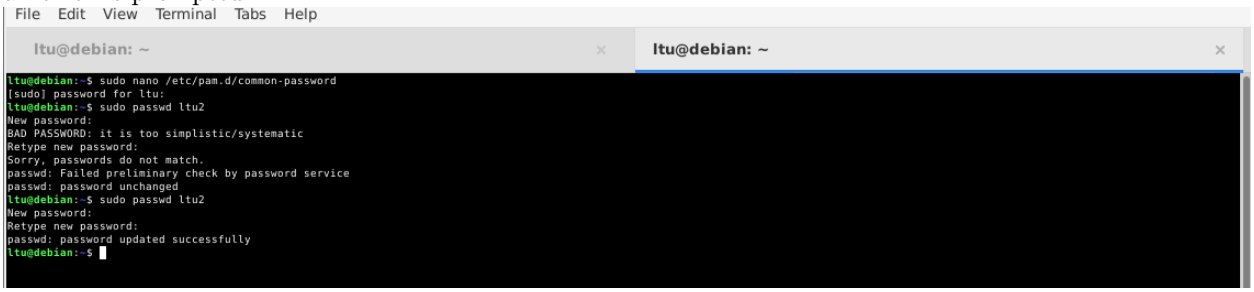


Figure 17:

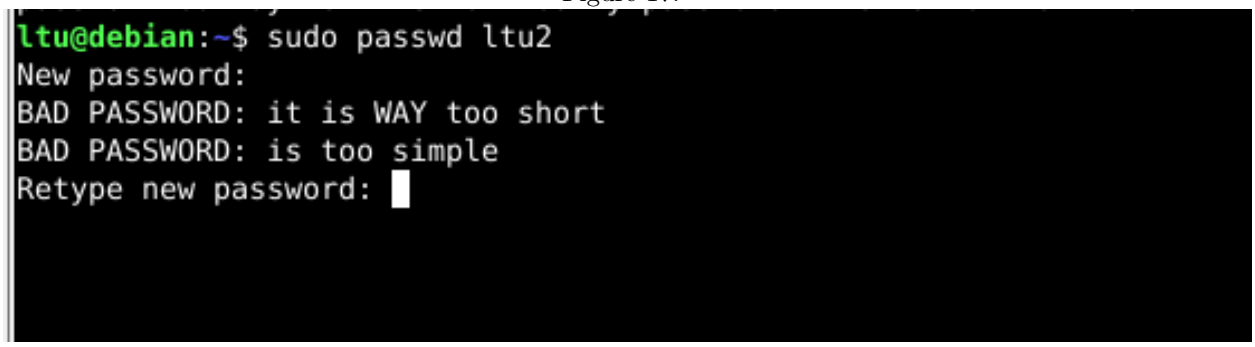


Figure 18:

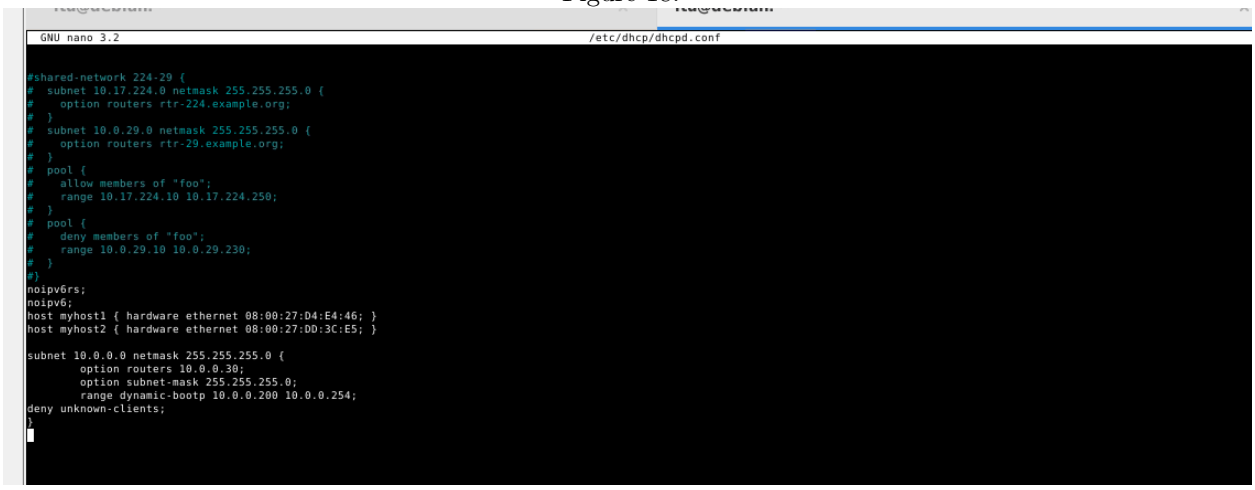


Figure 19:

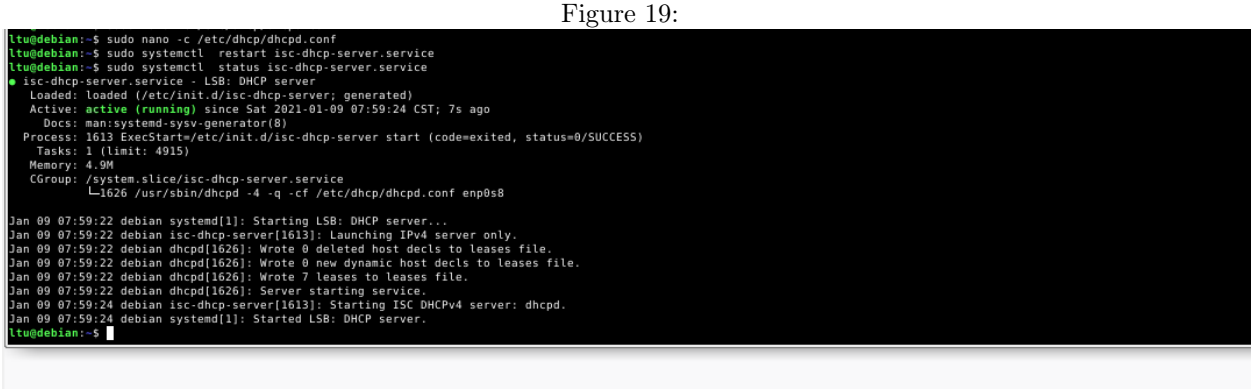


Figure 20: After setting a filter in the DHCP server allowing only known MAC addresses, i try to change the MAC address of the Kali VM

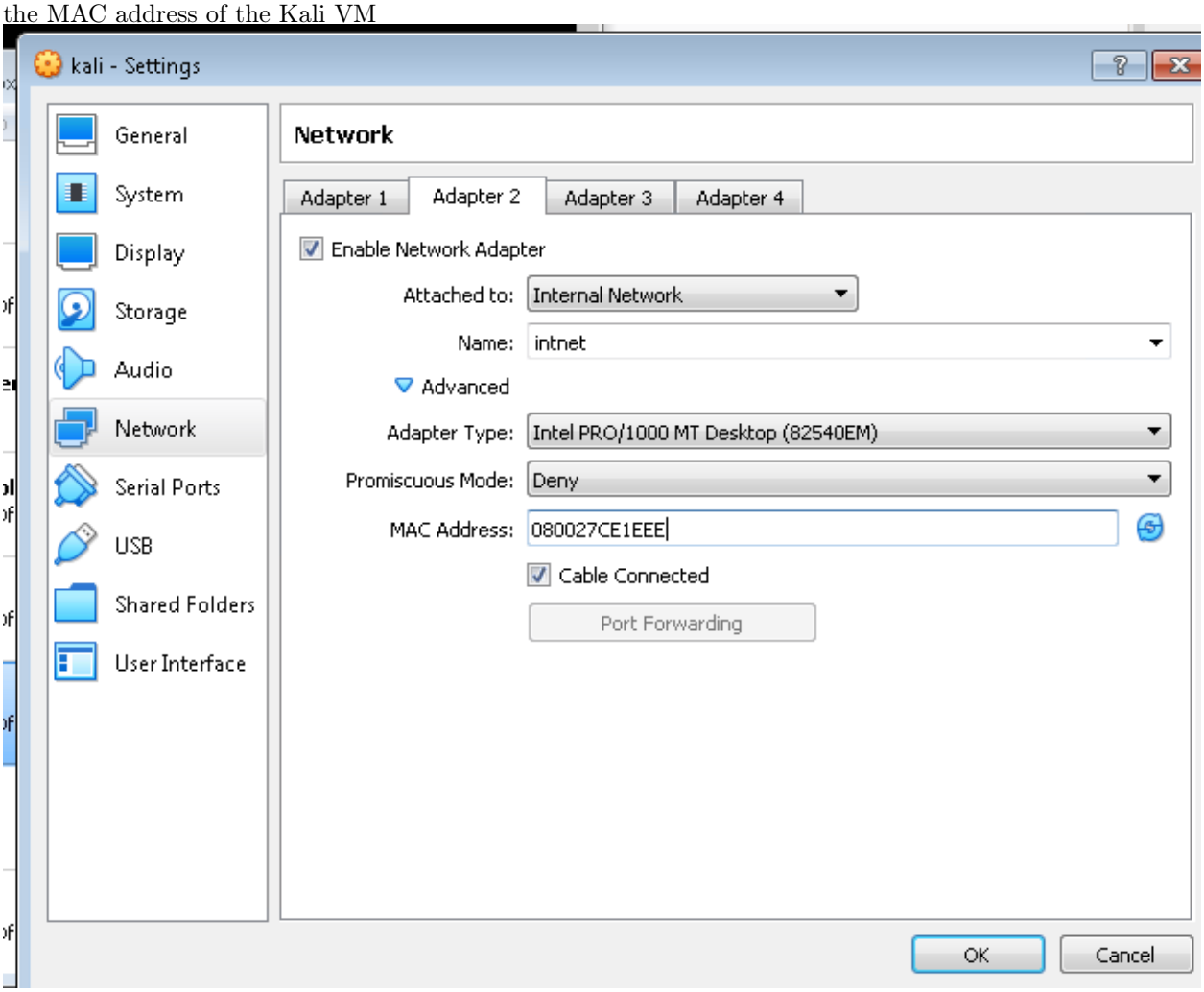


Figure 21: as we can see the VM doesn't receive any IP address from the DHCP server, due to its unknown MAC address to the server.

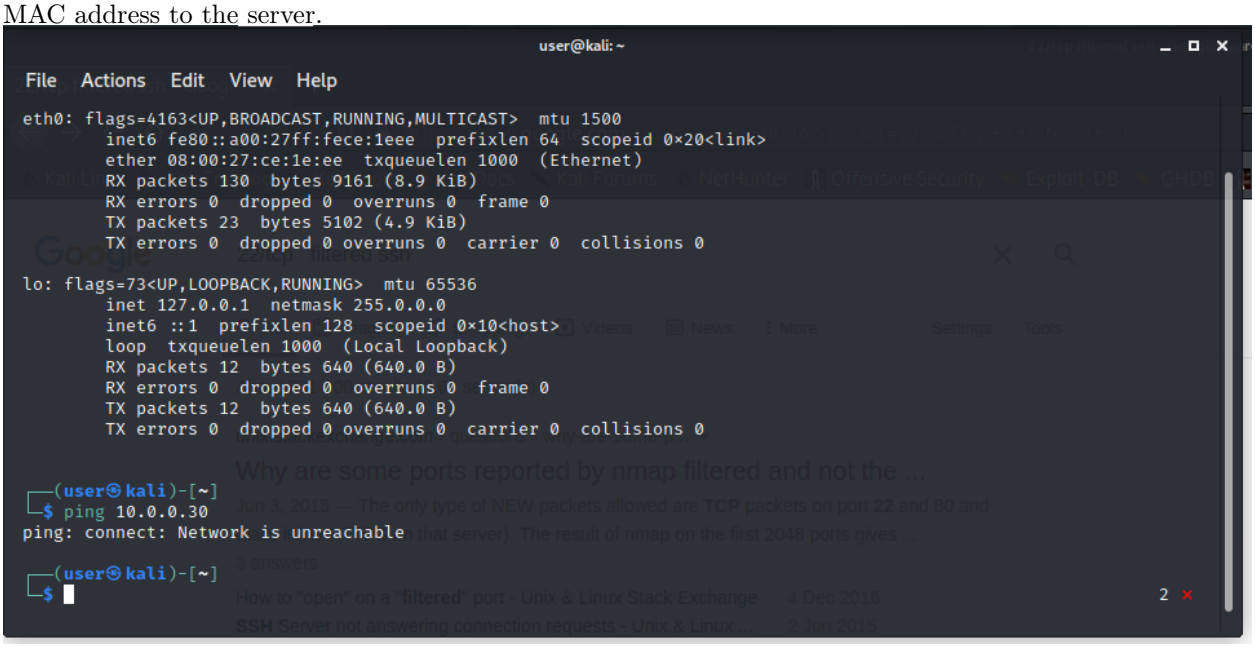


Figure 22: If the MAC is changed back to the one specified in the known hosts in the server, it will connect to the network.

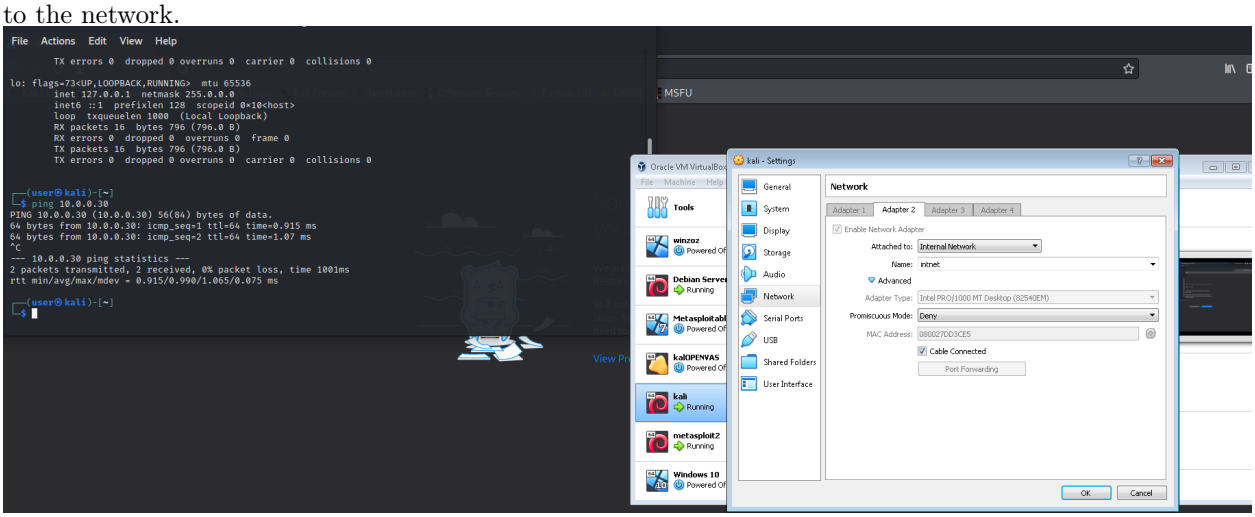


Figure 23:

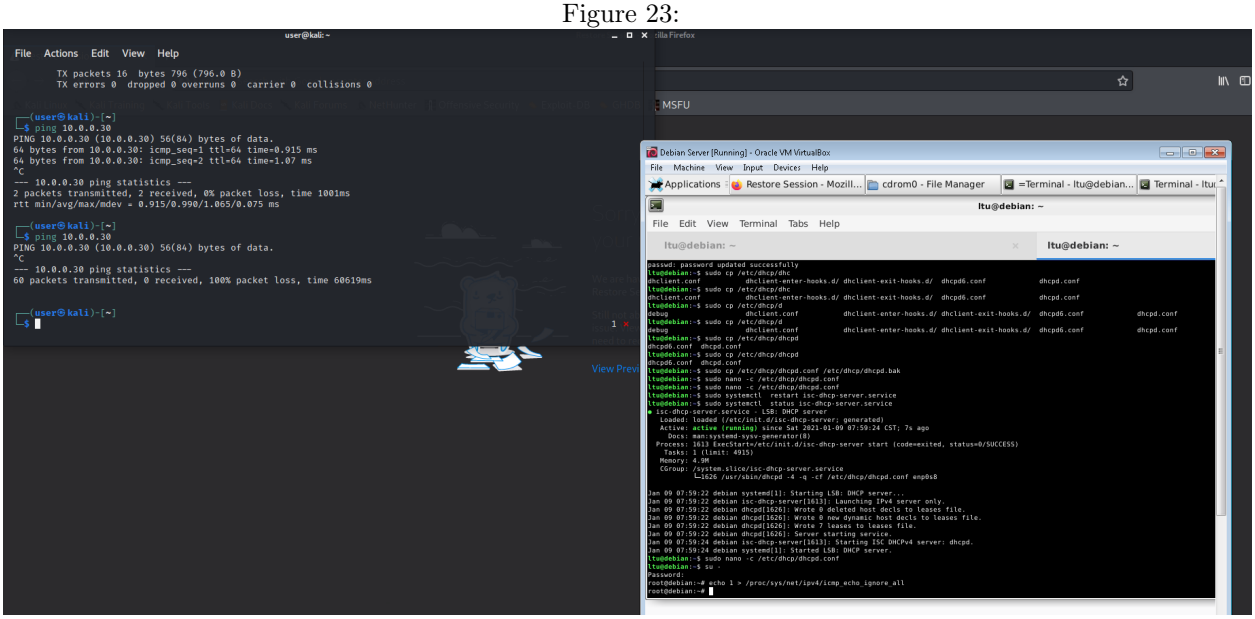


Figure 24: now the ICCMP traffic is ignored by the server, and we can see that the ping command will not receive any reply.

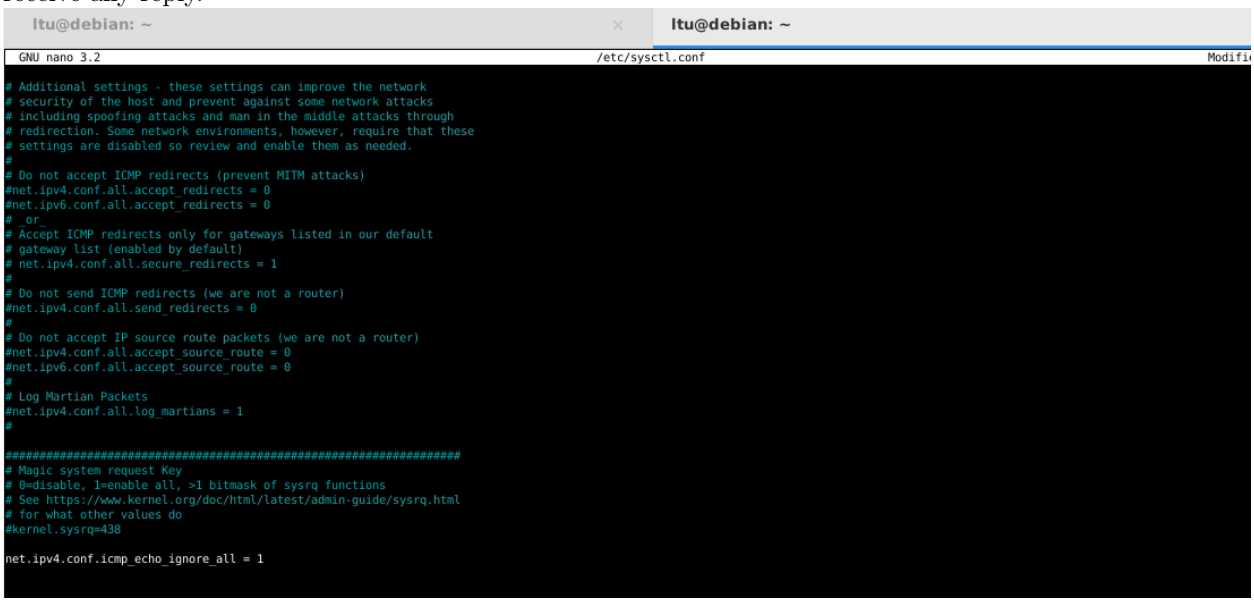


Figure 25:

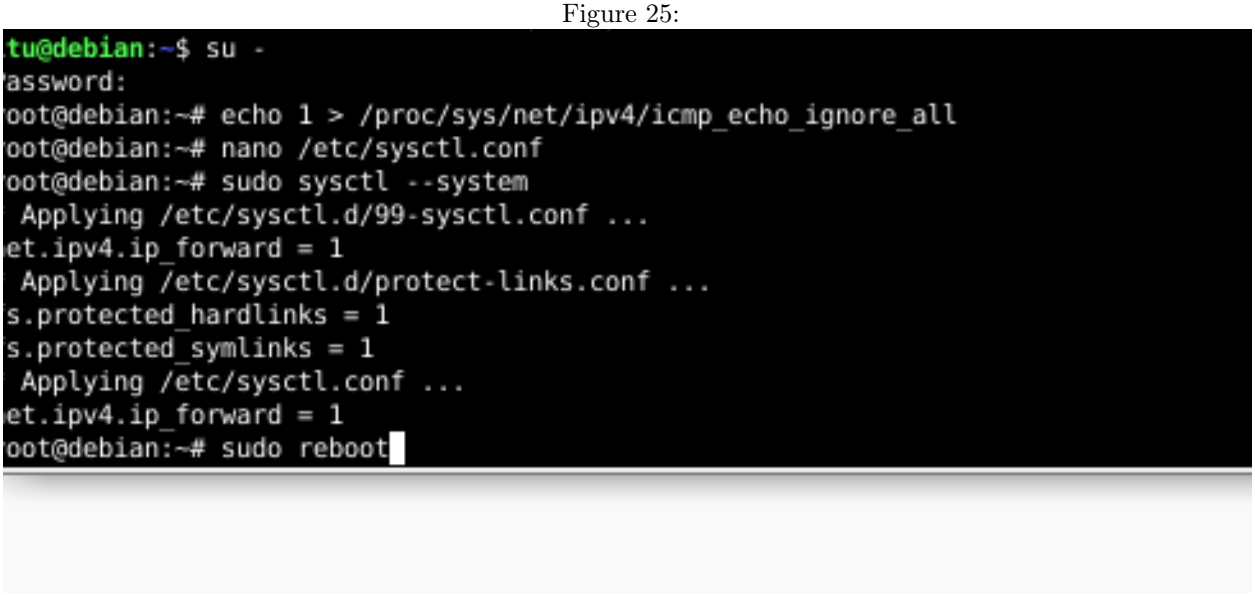


Figure 26:

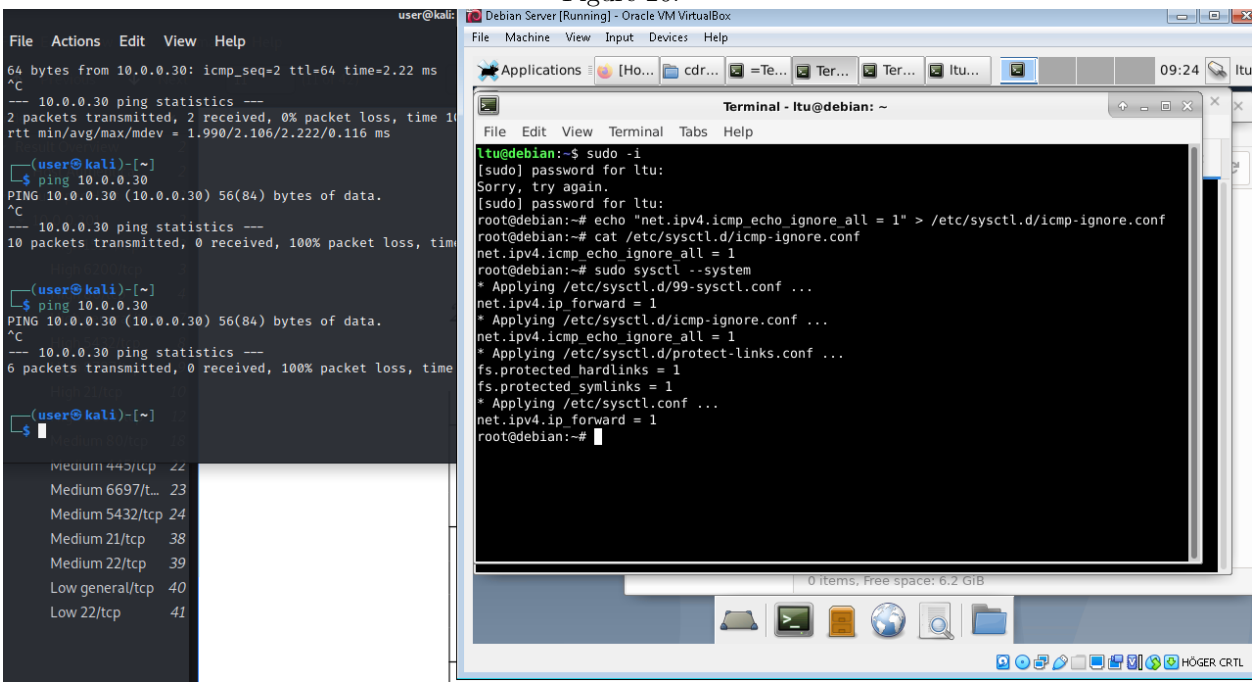


Figure 27:

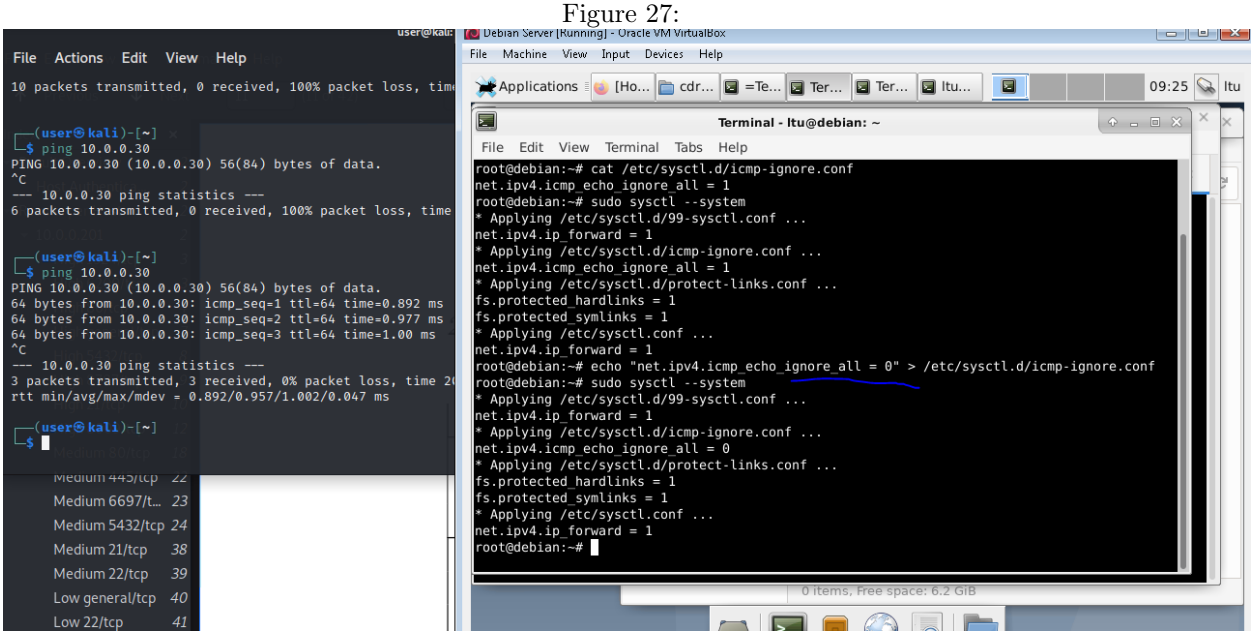

File Actions Edit View Help
10 packets transmitted, 0 received, 100% packet loss, time
(user㉿kali)-[~]
$ ping 10.0.0.30
PING 10.0.0.30 (10.0.0.30) 56(84) bytes of data.
^C
--- 10.0.0.30 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time
(user㉿kali)-[~]
$ ping 10.0.0.30
PING 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=0.892 ms
64 bytes from 10.0.0.30: icmp_seq=2 ttl=64 time=0.977 ms
64 bytes from 10.0.0.30: icmp_seq=3 ttl=64 time=1.00 ms
^C
--- 10.0.0.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2
rtt min/avg/max/mdev = 0.892/0.957/1.002/0.047 ms
(user㉿kali)-[~]
$
Medium 6697/t... 23
Medium 5432/tcp 24
Medium 21/tcp 38
Medium 22/tcp 39
Low general/tcp 40
Low 22/tcp 41
Debian Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
09:25
Terminal - ltu@debian: ~
File Edit View Terminal Tabs Help
root@debian:~# cat /etc/sysctl.d/icmp-ignore.conf
net.ipv4.icmp_echo_ignore_all = 1
root@debian:~# sudo sysctl --system
* Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.ip_forward = 1
* Applying /etc/sysctl.d/icmp-ignore.conf ...
net.ipv4.icmp_echo_ignore_all = 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:~# echo "net.ipv4.icmp_echo_ignore_all = 0" > /etc/sysctl.d/icmp-ignore.conf
root@debian:~# sudo sysctl --system
* Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.ip_forward = 1
* Applying /etc/sysctl.d/icmp-ignore.conf ...
net.ipv4.icmp_echo_ignore_all = 0
* 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:~#
0 items, Free space: 6.2 GiB

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

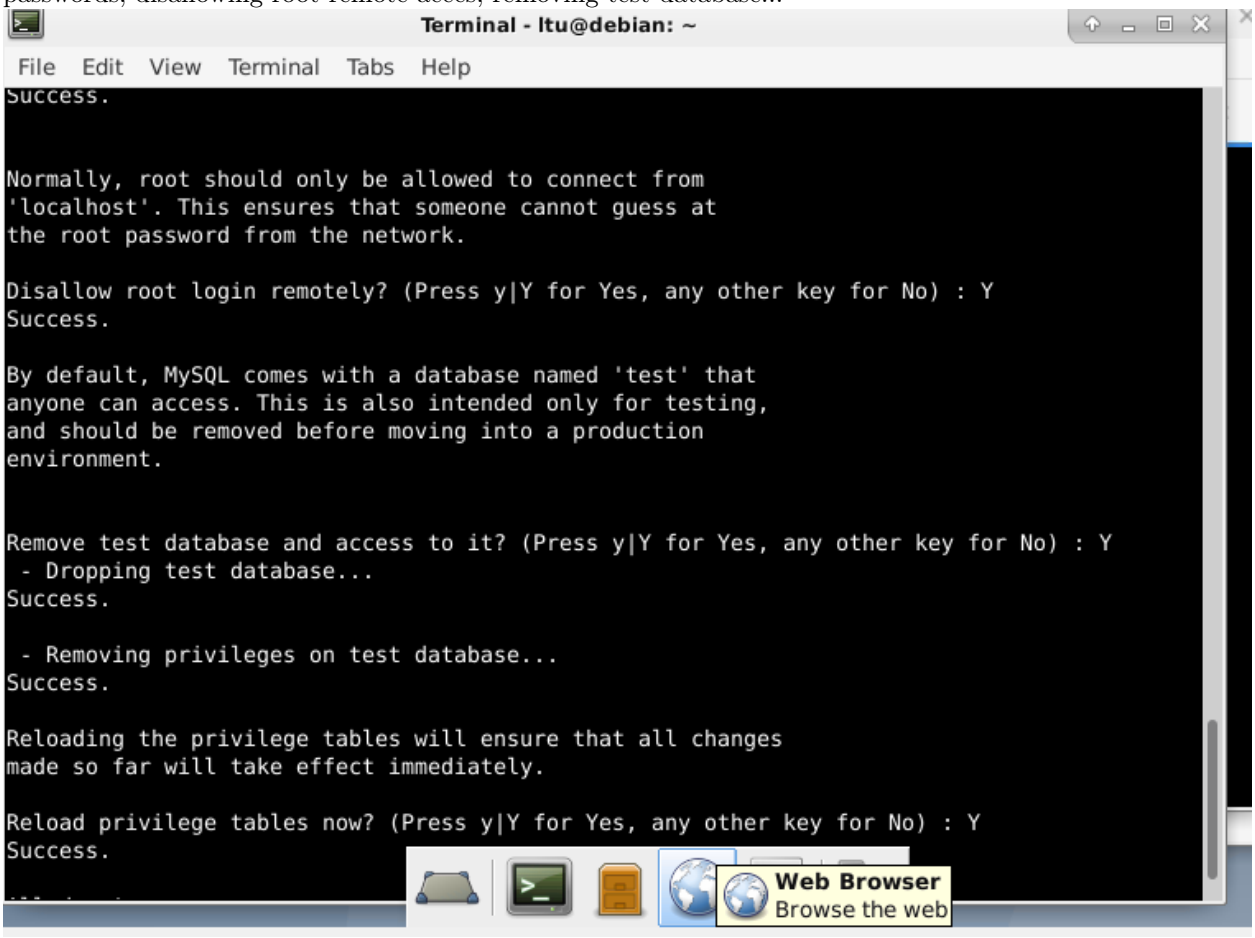


Figure 29: we allow access to phpmyadmin only from the server IP and after reloading the service we can test it.

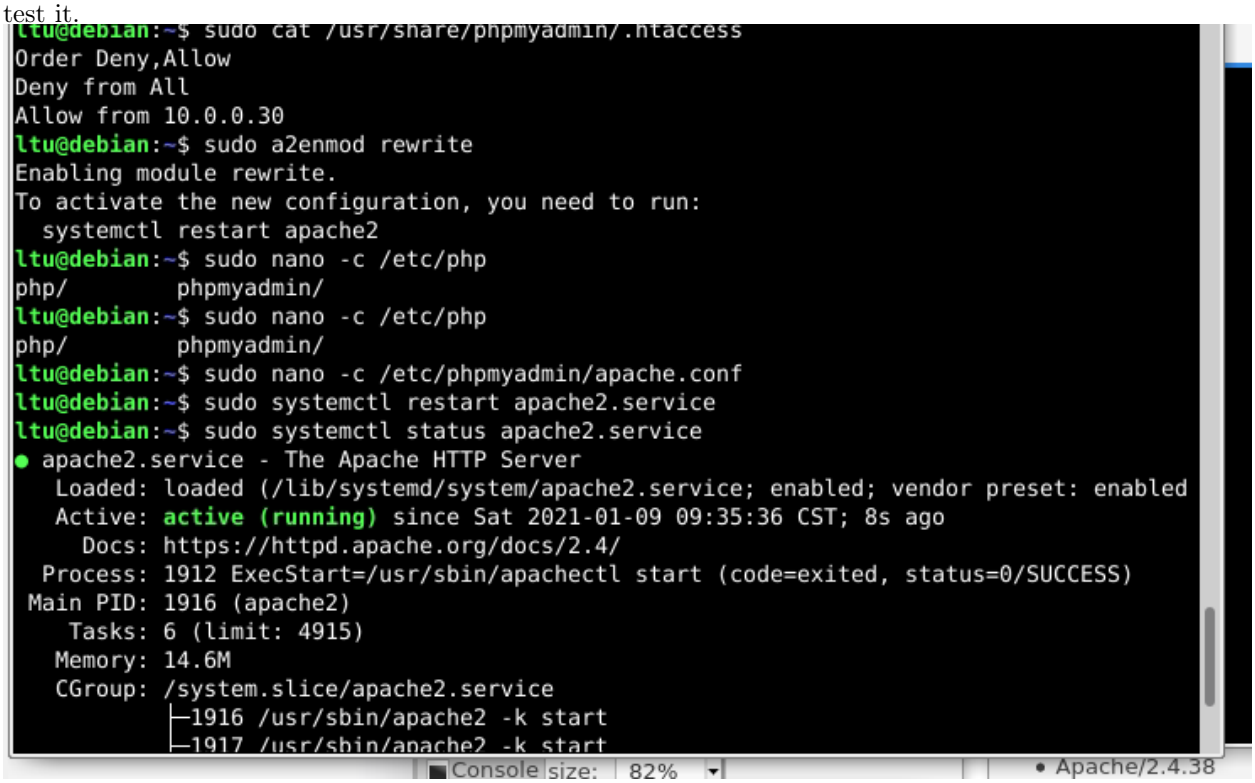


Figure 30: from the Debian machine it result possible to access to phpmyadmin, as expected.

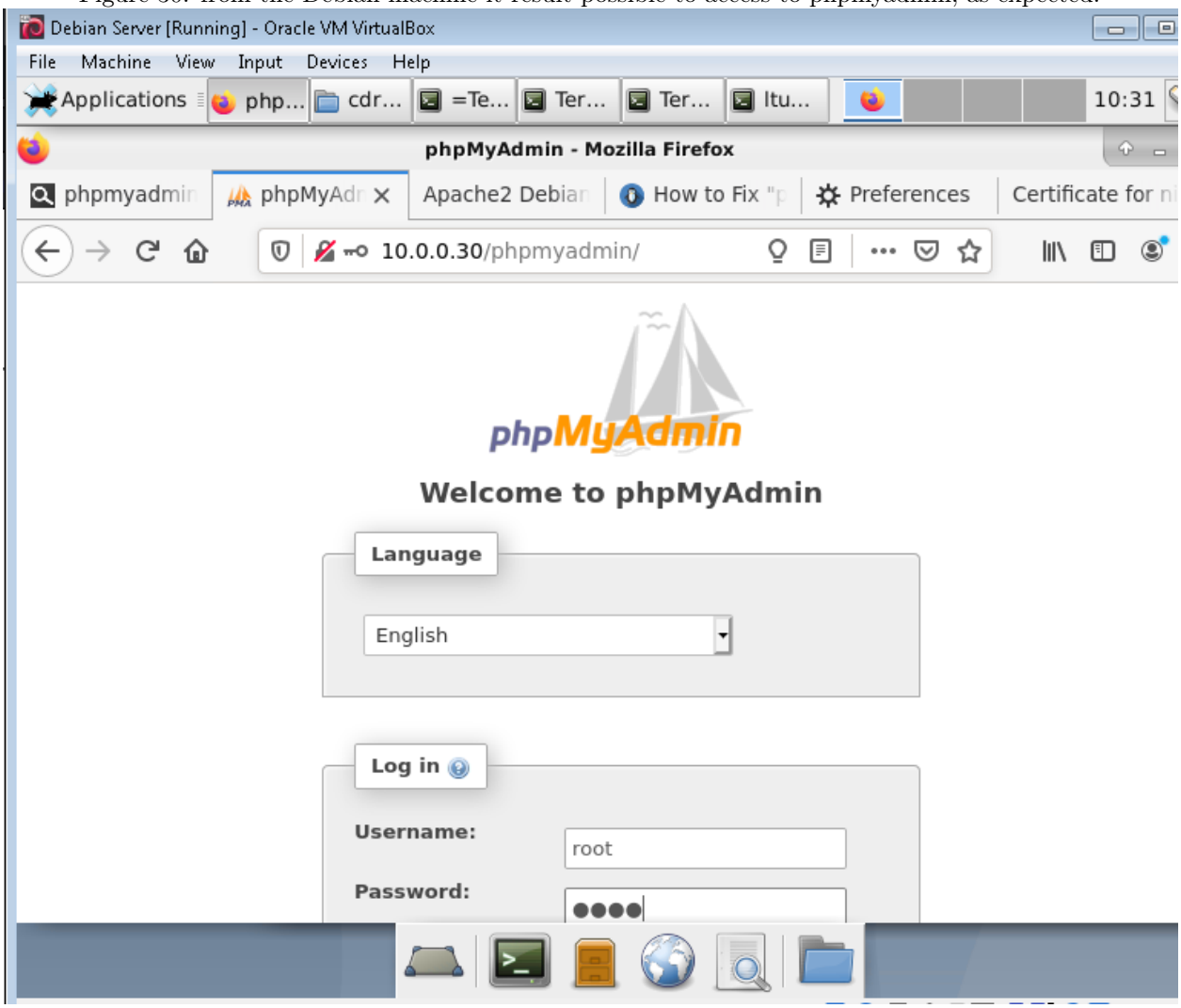
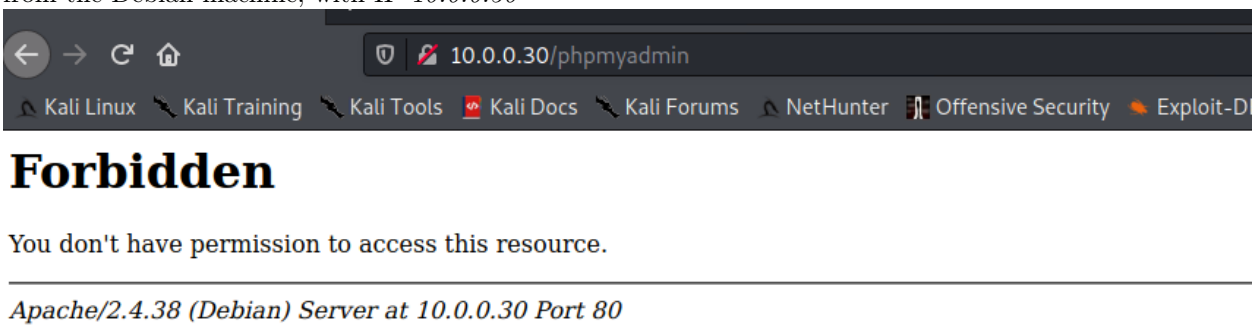


Figure 31: from the kali VM it is not possible to access to phpmyadmin, since the acccess is allowed only from the Debian machine, with IP *10.0.0.30*



4. thoughts about this week

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