Samba Server

Samba is an open source/free software suite and print services to Server Message Block (SMB) Clients.

It allows for interoperability between Linux/Unix Servers and Windows based clients.

It is a type of mechanism which is used to share the resources and object data across different operating system clients such as Linux & Windows.

The samba server mainly used in the network not only to share data but also to share domain.

Samba Server

The function of sharing data between samba server & samba client will be controlled by cifs (common internet file system) and the services such as smb (server message block) and nmb (netbios message block).

smb services use in file sharing cases and nmb services can be used in domain sharing cases.

Samba Server

Features of Samba

1) It maintains all clients data centralized in one system.

2) It supports anonymous & domain user policies.

3) It generates its log file at default location /var/log/samba.

4) It by default comes with tools to check the configuration, For example testparm at server system and smbclient at client systems.

5) It is used to share data and also domain using the configurations such as samba pdc & samba winbind.

Samba Server

Packages
- samba*.rpm

Port Numbers
- 137 NetBios Name Service
- 138 NetBios Datagram Service
- 139 NetBios Session Service

Configuration file
- /etc/samba/smb.conf

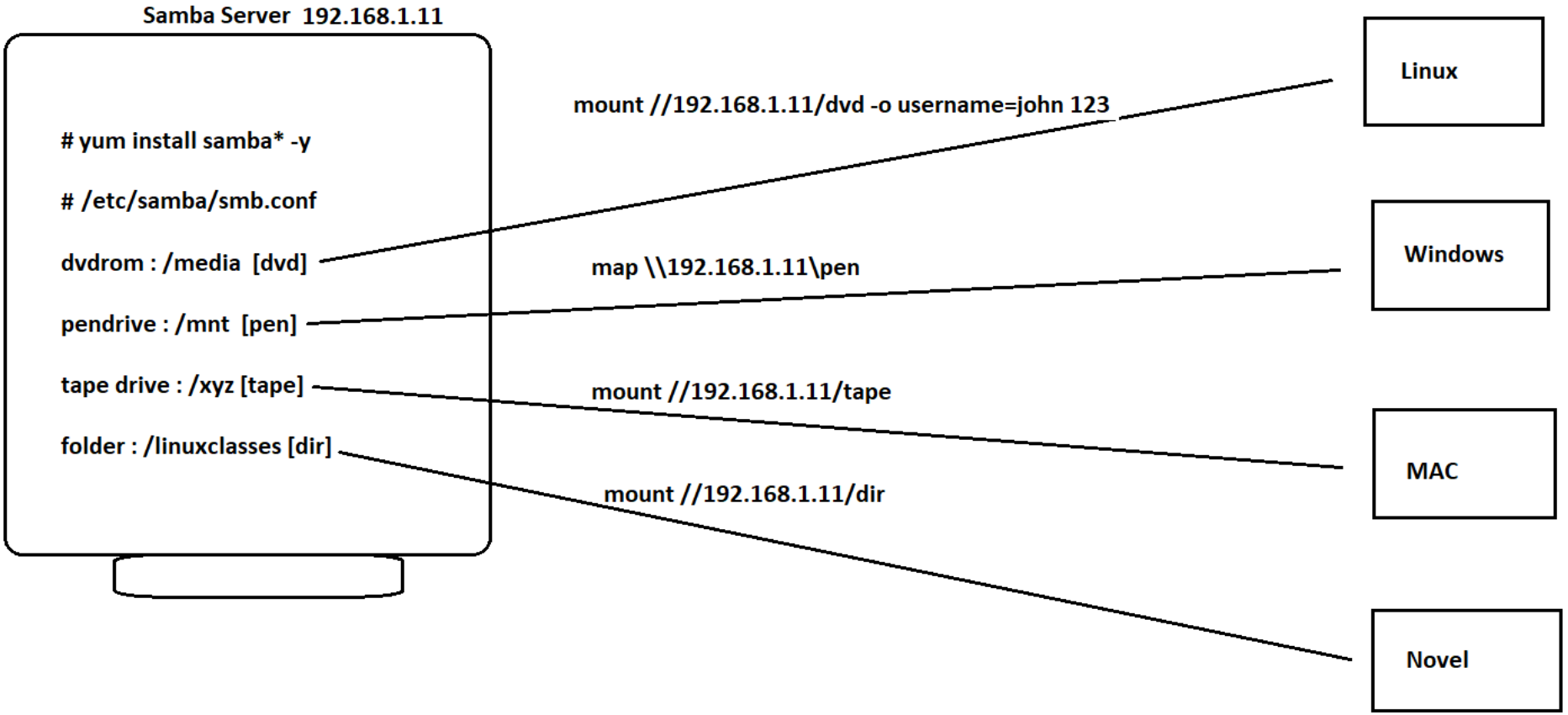
Service /Daemon
- smb

Samba Server

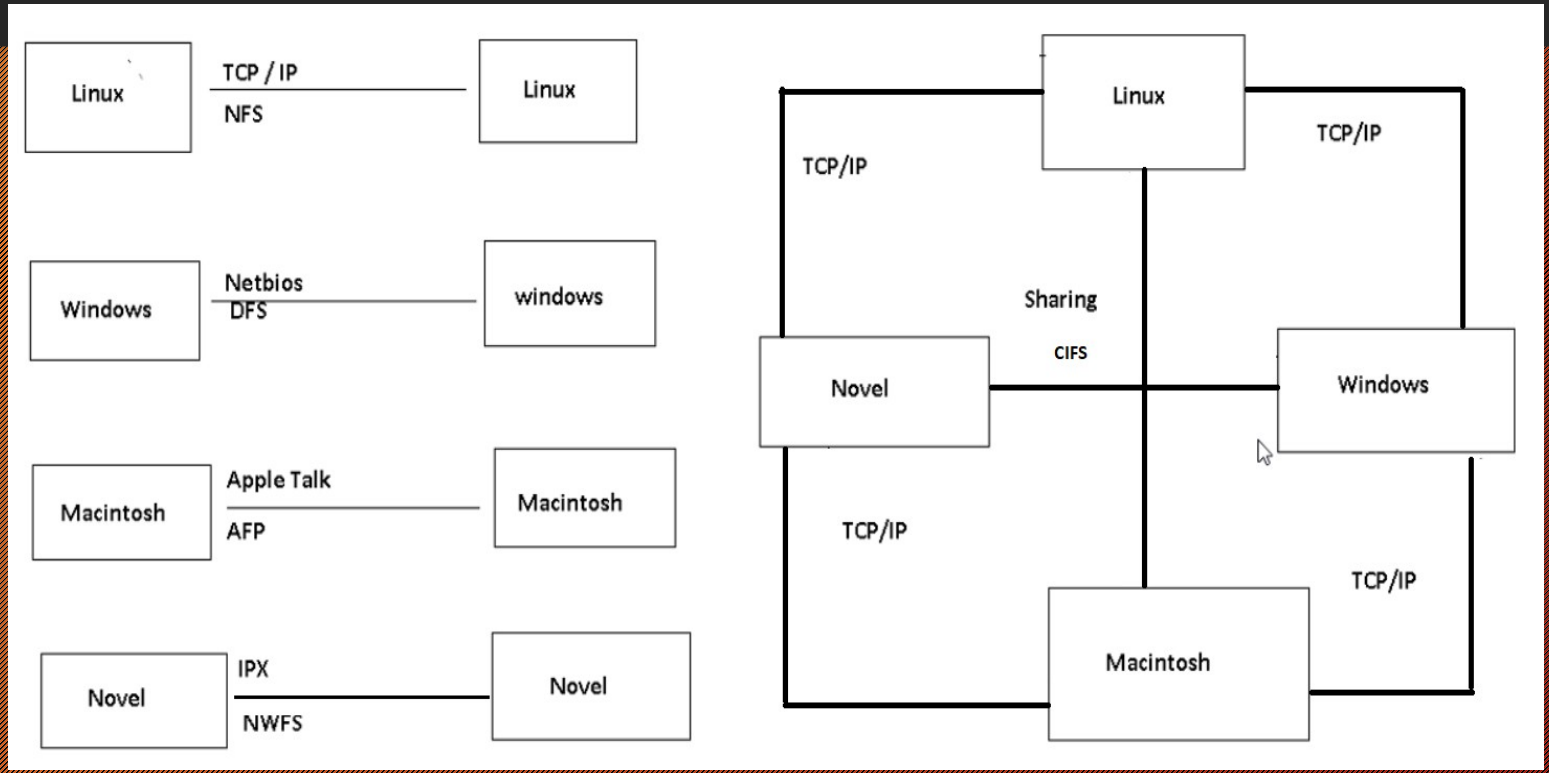
Samba File Sharing

Prerequisite

1) Share Name (Just a flat name)
2) Samba tools
3) Samba port numbers
4) Samba user policies
5) Samba services & daemons


Samba Server 192.168.1.11
yum install samba* -y
/etc/samba/smb.conf
dvdrom : /media [dvd]
pendrive : /mnt [pen]
tape drive : /xyz [tape]
folder : /linuxclasses [dir]
mount //192.168.1.11/dvd -o username=john 123
map \\192.168.1.11\pen
mount //192.168.1.11/tape
mount //192.168.1.11/dir
Linux
Windows
MAC
Novel

Samba Server



Samba Server

LAB SESSION

Linux Administration

❑ SELINUX

➢ **Introduction** - SELinux

- Security Enhanced Linux or SELinux is an advanced access control mechanism built into most modern Linux distributions. It was initially developed by the US National Security Agency to protect computer systems from malicious intrusion and tampering.
- SELinux is a set of predefined policies, it mainly concerns about the processes & Services.
- SELinux is a way to fine-tune such access control requirements. With SELinux, you can define what a user or process can do. It confines every process to its own domain so the process can interact with only certain types of files and other processes from allowed domains. This prevents a hacker from hijacking any process to gain system-wide access.

Linux Administration

Why SELinux

- SELinux implements what's known as **MAC** (Mandatory Access Control). This is implemented on top of what's already present in every Linux distribution, the **DAC** (Discretionary Access Control).
- To understand DAC, let's first consider how traditional Linux file security works.

- In a traditional security model, we have three entities: User, Group, and Other (u,g,o) who can have a combination of Read, Write, and Execute (r,w,x) permissions on a file or directory. If a user **jo** creates a file in their home directory, that user will have read/write access to it, and so will the **jo** group. The "other" entity will possibly have no access to it. In the following code block, we can consider the hypothetical contents of jo's home directory.

Linux Administration

•SELinux gives that extra layer of security to the resources in the system. It provides the MAC (mandatory access control) as contrary to the DAC (Discretionary access control). Before we dive into setting the SELinux modes, let us see what are the different SELinux modes of operation and how do they work. SELinux can operate in any of the 3 modes :

•**Enforced** : Actions contrary to the policy are blocked and a corresponding event is logged in the audit log.

•**Permissive** : Actions contrary to the policy are only logged in the audit log.

•**Disabled** : The SELinux is disabled entirely.

Linux Administration

- **Toggling the SELinux modes temporarily**

- To switch between the SELinux modes temporarily we can use the setenforce command as shown below :

- # setenforce [Enforcing | Permissive | 1 | 0]

0 -> Permissive
1 -> Enforcing

Or you can simply echo the values into the pseudo file - **/sys/fs/selinux/enforce** or **/selinux/enforce.**
echo [0|1] > /sys/fs/selinux/enforce
To check the current mode of SELinux :
getenforce
Enforcing

Linux Administration

- Changing SELinux modes Permanently

- One way of changing the SELinux mode permanently to either of Enforcing or Permissive is - to edit the /etc/sysconfig/selinux file and set SELINUX parameters value to either enforcing or permissive.

```
# ls -l /etc/sysconfig/selinux
lrwxrwxrwx. 1 root root 17 Mar  2 13:03 /etc/sysconfig/selinux -> ../selinux/config
```

Linux Administration

- Changing SELinux modes Permanently

```
# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- Edit this file and take a reboot of the system for the changes to take effect.

Linux Administration

- Changing SELinux modes Permanently

- We can also use the Kernel boot parameter at boot to set the SELinux mode. For this edit the /etc/grub.conf file and add the option "selinux=1 enforcing=[0|1]" to the boot parameters.

```
# cat /etc/grub.conf
.........
    root (hd0,0)
    kernel /vmlinuz-2.6.32-279.el6.x86_64 root=/dev/md3 selinux=1 enforcing=0
    initrd /initramfs-2.6.32-279.el6.x86_64.img
.........
```

- selinux=1 -> Enable the SELinux
- enforcing=0 -> Permissive mode
- enforcing=1 -> Enforcing mode

Linux Administration

➢Changing SELinux modes Permanently

- Disabling SELinux

- Sometimes when you are not well acquainted with SELinux functionalities, it is better to disable it. We can not disable the SELinux without a reboot. An alternative option would be – to set SELinux in Permissive mode. To completely disable SELinux edit the configuration file /etc/sysconfig/selinux or the /etc/selinux/config which is a soft link to /etc/sysconfig/selinux file.

Linux Administration

- Changing SELinux modes Permanently

```
# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- Edit this file and take a reboot of the system for the changes to take effect.

Linux Administration

➢SELinux Contexts

A SELinux context consists of an access control environment where decisions are made based on SELinux user, role, and type (and optionally a level):

- A SELinux user complements a regular Linux user account by mapping it to a SELinux user account, which in turn is used in the SELinux context for processes in that session, in order to explicitly define their allowed roles and levels.
- The concept of role acts as an intermediary between domains and SELinux users in that it defines which process domains and file types can be accessed. This will shield your system against vulnerability to privilege escalation attacks.
- A type defines an SELinux file type or an SELinux process domain. Under normal circumstances, processes are prevented from accessing files that other processes use, and and from accessing other processes, thus access is only allowed if a specific SELinux policy rule exists that allows it.

Linux Administration

SELinux Policy

- At the heart of SELinux' security engine is its policy. A policy is what the name implies: a set of rules that define the security and access rights for everything in the system. And when we say everything, we mean users, roles, processes, and files. The policy defines how each of these entities are related to one another.

- Some Basic Terminology

- To understand policy, we have to learn some basic terminology. We will go into the details later, but here is a brief introduction. An SELinux policy defines user access to roles, role access to domains, and domain access to types.

Linux Administration

SELinux Policy

- Users

- SELinux has a set of pre-built users. Every regular Linux user account is mapped to one or more SELinux users.

- In Linux, a user runs a process. This can be as simple as the user sverma opening a document in the vi editor (it will be sverma's account running the vi process) or a service account running the httpd daemon. In the SELinux world, a process (a daemon or a running program) is called a subject.

Linux Administration

SELinux Policy

- Roles

- A role is like a gateway that sits between a user and a process. A role defines which users can access that process. Roles are not like groups, but more like filters: a user may enter or assume a role at any time provided the role grants it. The definition of a role in SELinux policy defines which users have access to that role. It also defines what process domains the role itself has access to. Roles come into play because part of SELinux implements what's known as Role Based Access Control (RBAC).

Linux Administration

SELinux Policy

- Subjects and Objects

- A subject is a process and can potentially affect an object.

- An object in SELinux is anything that can be acted upon. This can be a file, a directory, a port, a tcp socket, the cursor, or perhaps an X server. The actions that a subject can perform on an object are the subject's permissions.

Linux Administration

SELinux Policy

- Domains are for Subjects

- A domain is the context within which an SELinux subject (process) can run. That context is like a wrapper around the subject. It tells the process what it can and can't do. For example, the domain will define what files, directories, links, devices, or ports are accessible to the subject.

Firewalld

- A packet filtering firewall reads incoming network packets and filters (allows or denies) each data packet based on the header information in the packet. The Linux kernel has built-in packet filtering functionality called Netfilter.

- FirewallD is frontend controller for iptables used to implement persistent network traffic rules. It provides command line and graphical interfaces.

- Two services are available in RHEL 7 to create, maintain, and display the rules stored by Netfilter:

1. firewalld
2. iptables

Firewalld

- The default firewall service in RHEL 7 is firewalld.
- It is a dynamic firewall manager which supports firewall (network) zones.
- The firewalld service has support for IPv4, IPv6, and for Ethernet bridges.

❑Firewalld

➢ Advantages over iptables

- Firewalld has the following advantages over iptables :

- Unlike the iptables command, the firewall-cmd command does not restart the firewall and disrupt established TCP connections.
- firewalld supports dynamic zones.
- firewalld supports D-Bus for better integration with services that depend on firewall configuration.

❑Firewalld

➢ Configuration options

➢ The firewalld service has two types of configuration options:

- **Runtime**: Changes to firewall settings take effect immediately but are not permanent. Changes made in runtime configuration mode are lost when the firewalld service is restarted.
- **Permanent**: Changes to firewall settings are written to configuration files. These changes are applied when the firewalld service restarts.

Firewalld

- Configuration files

- Configuration files for firewalld exist in two directories:

- /usr/lib/firewalld: Contains default configuration files. Do not make changes to these files. An upgrade of the firewalld package overwrites this directory.
- /etc/firewalld: holds system configuration files. These files will overwrite a default configuration.

Firewalld Components

- **Drop Zone: Any incoming packets are dropped, if we use this drop zone. This is same as we use to add iptables -j drop. If we use the drop rule, means there is no reply, only outgoing network connections will be available.**

- **Block Zone: Block zone will deny the incoming network connections are rejected with an icmp-host-prohibited. Only established connections within the server will be allowed.**

- **Public Zone: To accept the selected connections we can define rules in public zone. This will only allow the specific port to open in our server other connections will be dropped.**

Firewalld Components

- External Zone: This zone will act as router options with masquerading is enabled other connections will be dropped and will not accept, only specified connection will be allowed.
- DMZ Zone: If we need to allow access to some of the services to public, you can define in DMZ zone. This too have the feature of only selected incoming connections are accepted.
- Work Zone: In this zone, we can define only internal networks i.e. private networks traffic are allowed.

Firewalld Components

- **Home Zone:** This zone is specially used in home areas, we can use this zone to trust the other computers on networks to not harm your computer as every zone. This too allow only the selected incoming connections.
- **Internal Zone:** This one is similar to work zone with selected allowed connections.
- **Trusted Zone:** If we set the trusted zone all the traffic are accepted.

- Firewalld Components

  - For each zone you can define the following features:

    - Services: Predefined or custom services to trust. Trusted services are a combination of ports and protocols that are accessible from other systems and networks.

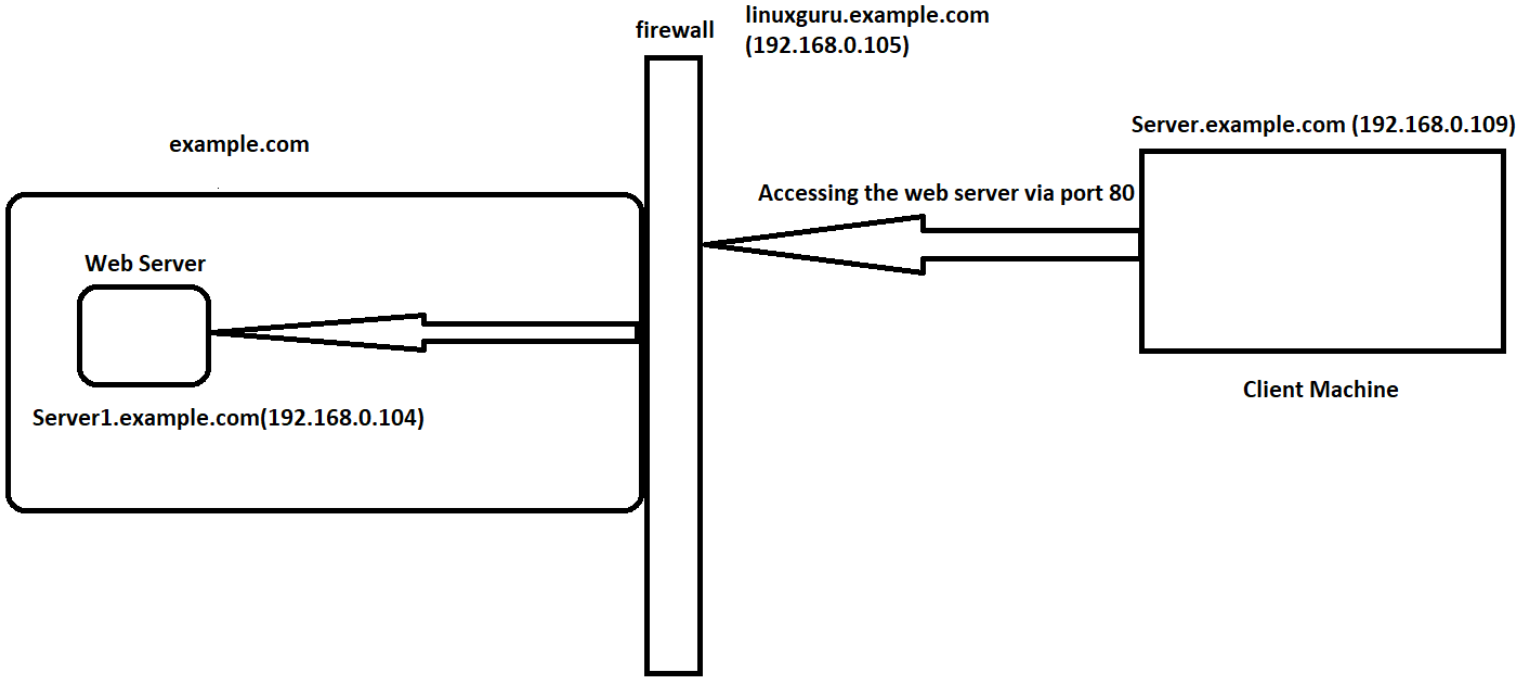
    - Ports: Additional ports or port ranges and associated protocols that are accessible from other systems and networks.

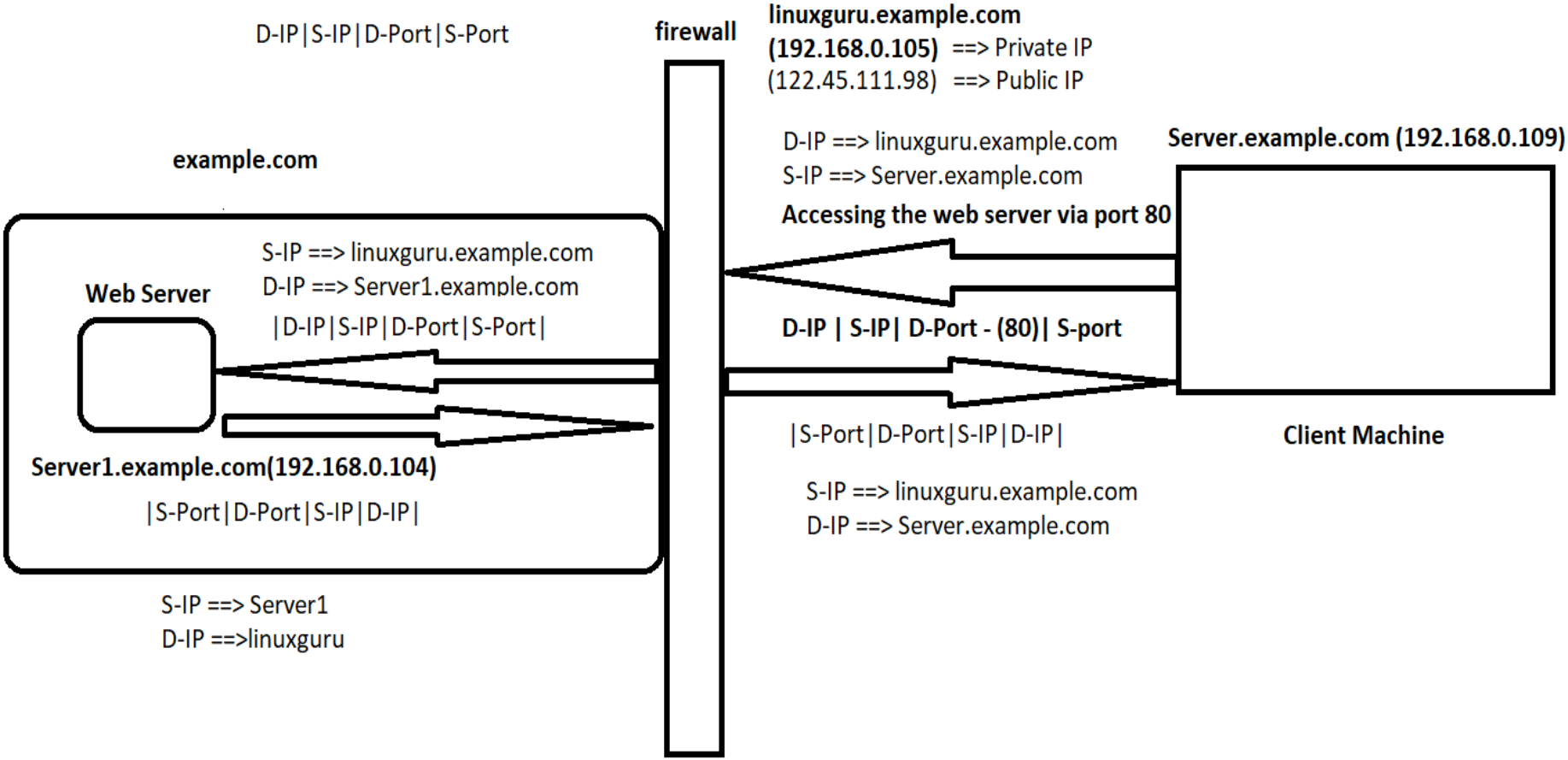
    - Masquerading: Translate IPv4 addresses to a single external address. With masquerading enabled, addresses of a private network are mapped to and hidden behind a public address.

Firewalld Components

- **Port Forwarding:** Forward inbound network traffic from a specific port or port range to an alternative port on the local system, or to a port on another IPv4 address.
- **ICMP Filter:** Block selected Internet Control Message Protocol messages.
- **Rich Rules:** Extend existing firewalld rules to include additional source and destination addresses and logging and auditing actions.
- **Interfaces:** Network interfaces bound to the zone. The zone for an interface is specified with the ZONE=option in the /etc/sysconfig/network-scripts/ifcfg file. If the option is missing, the interface is bound to the default zone.

- Lab Sessions
- Firewall Server : linuxguru.example.com (192.168.0.115)
- Client Server : Server.example.com (192.168.0.109)


firewall
linuxguru.example.com
(192.168.0.105)
example.com
Server.example.com (192.168.0.109)
Accessing the web server via port 80
Web Server
Server1.example.com(192.168.0.104)
Client Machine


D-IP|S-IP|D-Port|S-Port
firewall
linuxguru.example.com
(192.168.0.105) ==> Private IP
(122.45.111.98) ==> Public IP
D-IP ==> linuxguru.example.com
S-IP ==> Server.example.com
Server.example.com (192.168.0.109)
example.com
Accessing the web server via port 80
S-IP ==> linuxguru.example.com
D-IP ==> Server1.example.com
|D-IP|S-IP|D-Port|S-Port|
Web Server
D-IP | S-IP| D-Port - (80)| S-port
|S-Port|D-Port|S-IP|D-IP|
Client Machine
Server1.example.com(192.168.0.104)
|S-Port|D-Port|S-IP|D-IP|
S-IP ==> linuxguru.example.com
D-IP ==> Server.example.com
S-IP ==> Server1
D-IP ==>linuxguru

Rich Rules

Firewalld Rich Rules Command

Following command is used to manage the rich rules

```
firewall-cmd  [Option] [Rule]
```

Here ***firewall-cmd*** is the main command.

The ***Option*** is the type of operation which we want to perform on rule. Following table lists some common operations.

Option	Description
--add-rich-rule='[RichRule]'	Add specified [*RichRule*] rule to default zone. To add rule in other zone, provide its name as argument with ***--zone*** option. To add rule permanently use ***--permanent*** option.
--query-rich-rule='[RichRule]'	Figure out whether the specified rule is added in default zone or not. To query in other zone, provide its name with ***--zone*** option.
--remove-rich-rule='[RichRule]'	Remove specified [*RichRule*] rule from default zone. To remove rule from other zone, provide its name as argument with ***--zone*** option. To remove rule permanently use ***--permanent*** option.
--list-rich-rules	List all rules from default zone. To list rules from other zone, provide its name as argument with ***--zone*** option.

Rich Rules

The ***Rule*** is rich rule. Rich rule uses following syntax:-

```
Rule
[source] [destination]
{service|port|protocol|icmp-block|masquerade|forward-port}
        [log] [audit]
[accept|reject|drop]
```