NETWORK ADMINISTRATION
COURSE PROJECT REPORT

TITLE

Setting up Virtual LAN on a Cisco L3 Switch

PROJECT ID

Project ID (as per the shared spreadsheet) = 8

TEAM

Member 1: Gursimran Singh
2014041
gursimran14041@iiitd.ac.in

Member 2: Harish Fulara
2014143
harish14143@iiitd.ac.in

OBJECTIVE

The objective of this project is to learn Cisco's IOS (Internetwork Operating System) CLI and then to use this interface to configure a Cisco L3 Switch with the following configuration:

- atleast 3 Virtual LANs
- DHCP (Dynamic Host Configuration Protocol) Setup with MAC Address Binding and IPv6 Autoconfiguration

HARDWARE

We were provided with a Cisco L3 Switch connected to IIIT-Delhi's campus network. Specifications of the aforementioned switch are as follows:

Command
Switch# **show inventory**

```
NAME: "2", DESCR: "WS-C2960S-48TD-L"
PID: WS-C2960S-48TD-L  , VID: V04  , SN: FOC1550Z555
```

Command
Switch# **show ver**

```
Cisco IOS Software, C2960S Software (C2960S-UNIVERSALK9-M), Version
12.2(55)SE3, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Thu 05-May-11 16:56 by prod_rel_team
Image text-base: 0x00003000, data-base: 0x01B00000
```

```
ROM: Bootstrap program is Alpha board boot loader
BOOTLDR: C2960S Boot Loader (C2960S-HBOOT-M) Version 12.2(55r)SE,
RELEASE SOFTWARE (fc1)

Switch uptime is 3 weeks, 6 days, 6 hours, 11 minutes
System returned to ROM by power-on
System image file is "flash:/c2960s-universalk9-mz.122-55.SE3/c2960s-universalk9-
mz.122-55.SE3.bin"


This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco WS-C2960S-48TD-L (PowerPC) processor (revision F0) with 131072K bytes of
memory.
Processor board ID FOC1550Z555
Last reset from power-on
2 Virtual Ethernet interfaces
1 FastEthernet interface
50 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.

512K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : C4:0A:CB:70:AD:80
Motherboard assembly number     : 73-11905-08
Power supply part number        : 341-0327-04
Motherboard serial number       : FOC15502EZR
Power supply serial number      : LIT15420J0D
Model revision number           : F0
Motherboard revision number     : A0
Model number                    : WS-C2960S-48TD-L
Daughterboard assembly number   : 73-11933-04
Daughterboard serial number     : FOC15482KS7
System serial number            : FOC1550Z555
Top Assembly Part Number        : 800-30925-04
Top Assembly Revision Number    : B0
Version ID                      : V04
CLEI Code Number                : COMGD00ARD
```

```
Daughterboard revision number   : A0
Hardware Board Revision Number  : 0x01


Switch Ports Model              SW Version          SW Image
------ ----- -----              ----------          ----------
*    2 52    WS-C2960S-48TD-L   12.2(55)SE3         C2960S-UNIVERSALK9-M


Configuration register is 0xF
```

DETAILED REPORT

1. Accessing the Command Line Interface

Since the Switch was located in a remote network it was configured to use **Telnet** or **SSH** for remote connections. We were further required to use a VPN client to access the Switch located in IIIT-Delhi's private network.

To make a Telnet connection to the switch,

Command	**Purpose**
telnet {*hostname* | *ip_addr*}	Makes a Telnet connection from your host to the switch that you want to access.
Login: **admin** Password: ***password***	Initiates authentication. **Note**: If no password has been configured, press **Return**.
Switch# **exit**	Exits the session when finished.

Alternatively, to make an SSH connection to the switch,

Command	**Purpose**
ssh {*hostname* | *ip_addr*}	Makes an SSH connection from your host to the switch that you want to access.

2. Virtual LANs

We can use virtual LANs (VLANs) to divide the network into separate logical areas. VLANs can also be considered as broadcast domains. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

2.1. VLAN Ranges (as per Cisco which follows IEEE 802 IQ Standard)

VLANs Numbers	**Range**	**Usage**
1	Normal	Cisco default. You can use this VLAN, but you cannot modify or delete it.
2 - 1005	Normal	You can create, use, modify, and delete these VLANs.

1006 - 4094	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: • State is always active. • VLAN is always enabled. You cannot shut down these VLANs.
3968 – 4047 and 4094	Internally allocated	These 80 VLANs, plus VLAN 4094, are allocated for internal use. You cannot create, delete, or modify any VLANs within the block reserved for internal use.

2.2. Creating, Deleting and Modifying VLANs

VLANs are numbered from 1 to 4094. All configured ports belong to the default VLAN when the switch is first brought up. The default VLAN (VLAN1) uses only default values, and we cannot create, delete, or suspend activity in the default VLAN.

Depending on the range of the VLAN, we can configure the following parameters for VLANs (except the default VLAN):

- VLAN name
- Shutdown or not shutdown

When we delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. However, the system retains all the VLAN-to-port mapping for that VLAN, and when we reenable, or recreate, the specified VLAN, the system automatically reinstates all the original ports to that VLAN.

Once a VLAN is created, it is automatically in the active state.
To create and delete a VLAN,

Command	Purpose
Switch# **configure terminal**	Enters configuration mode.
Switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Creates a VLAN or a range or VLANs. If a number is entered that is already assigned to a VLAN, the switch puts us into the VLAN configuration submode for that VLAN. If we enter a number that is assigned to an internally allocated VLAN, the system returns an error message. VLAN1 is the default VLAN and cannot be created or deleted. We cannot create or delete those VLANs that are reserved for internal use.

Command	Purpose
Switch(config-vlan)# **no vlan** {*vlan-id* | *vlan-range*}	Deletes the specified VLAN or range of VLANs and removes us from the VLAN configuration submode. We cannot delete VLAN1 or the internally allocated VLANs.

2.3. Configuring the VLANs

We can modify the following parameters:

- Name
- Shut down

Command	**Purpose**
Switch# **configure terminal**	Enters configuration mode.
Switch(config)# **vlan** *{vlan-id* | *vlan-range}*	Enters VLAN configuration submode. If the VLAN does not exist, the system first creates the specified VLAN.
Switch(config-vlan)# **name** *vlan-name*	Names the VLAN. Can enter up to 32 alphanumeric characters to name the VLAN.
Switch(config-vlan)# **no shutdown**	Enables the VLAN.

We can now assign ports to a VLAN in the following way,

Command	**Purpose**
Switch# **configure terminal**	Enters configuration mode.
Switch(config)# **interface** *{type slot/port* | **port-channel** *number}*	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port or a port channel.
Switch(config-if)# **switchport access vlan** *vlan-id*	Sets the access mode of the interface to the specified VLAN.

To verify the VLAN configuration following commands may be used,

Command	**Purpose**
Switch# **show running-config vlan** *{vlan_id* | *vlan_range}*	Displays VLAN information.
Switch# **show vlan** [**brief** | **id** [vlan_id | vlan_range] | **name** name | **summary**]	Displays selected configuration information for the defined VLAN(s).

3. DHCP Setup

3.1. Enabling Cisco IOS DHCP Server and Relay Agent Features

By default, the Cisco IOS DHCP server and relay agent features are enabled on switch. To reenable these features if they are disabled, the following command may be used:

Command	**Purpose**
Switch(config)# **service dhcp**	Enables the Cisco IOS DHCP server and relay features on the switch. Use **no** form of this command to disable the Cisco IOS

	DHCP server and relay features.

3.2. Configuring a Database Agent OR Disabling Conflict Logging

To configure DHCP, we need to

- either specify a DHCP database agent (ex FTP, TFTP, RCP) that stores DHCP bindings (**cmd 1**) or
- if we don't specify a database agent we need to disable recording of DHCP address conflicts on DHCP Server (**cmd 2**).

Command	**Purpose**
Switch(config)# **ip dhcp database** url [**timeout** *seconds* **| write-delay** *seconds*]	Configures the database agent and the interval between database updates and database transfers.
Switch(config)# **no ip dhcp conflict logging**	Disables DHCP address conflict logging.

3.3. Excluding IP Addresses

The DHCP Server assumes that all IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients. We must specify the IP address that the DHCP Server should not assign to clients. To do so, the following command are used in global configuration mode:

Command	**Purpose**
Switch(config)# **ip dhcp excluded-address** *low-address* [*high-address*]	Specifies the IP addresses that the DHCP Server should not assign to DHCP clients.

3.4. Configuring DHCP pool name, Entering pool config mode

The following command allows us to give a name to ip address pool and simultaneously enters in DHCP pool configuration mode:

Command	**Purpose**
Switch(config)# **ip dhcp pool** *name*	Creates a name for the DHCP Server address pool and places you in DHCP pool configuration mode (identified by the dhcp-config# prompt).

3.5. Configuring DHCP Address Pool Subnet Mask

Giving a network number and a subnet mask to specify the pool of valid IP addreses.

Command	**Purpose**
Switch(dhcp-config)# **network** *network-number* [*mask* | /*prefix-length*]	Specifies the subnet network number and mask of the DHCP address pool.

3.6. Configuring a Domain name for the client

The domain name for a DHCP client places the client in the general grouping of networks that make up the domain.

Command	Purpose
Switch(dhcp-config)# **domain-name** *domain*	Specifies the domain name for the client.

3.7. Configuring IP DNS Server for Client

DHCP clients query DNS IP servers when they need to correlate host names to IP addresses. To configure the DNS IP servers that are available to a DHCP client, use the following command in DHCP pool configuration mode:

Command	Purpose
Switch(dhcp-config)# **dns-server** *address* [*address2 ... address8*]	Specifies the IP address of a DNS server that is available to a DHCP client.

3.8. Configuring the Default Router for Client

After a DHCP client has booted, the client begins sending packets to its default router. The IP address of the default router should be on the same subnet as the client.

Command	Purpose
Switch(dhcp-config)# **default-router** *address* [*address2 ... address8*]	Specifies the IP address of the default router for a DHCP client.

3.9. Configuring the Address Lease Time

By default, each IP address assigned by a DHCP Server comes with a one-day lease, which is the amount of time that the address is valid.

Command	Purpose
Switch(dhcp-config)# **lease** {*days* [*hours*] [*minutes*] | **infinite**}	Specifies the duration of the lease. The default is a one-day lease. **show ip dhcp binding to check**

3.10. Configuring Manual MAC Bindings

An address binding is a mapping between the IP address and MAC address of a client. Manual bindings are IP addresses that have been manually mapped to the MAC addresses of hosts that are found in the DHCP database. To configure a manual binding, first create a host pool, then specify the IP address of the client and client identifier or hardware address.

Following is a Step by Step procedure:

Command	Purpose
Switch(config)# **ip dhcp pool** *name*	Creates a name for the a DHCP Server address pool and places you in DHCP pool configuration mode
Switch(dhcp-config)# **host** *address* [*mask* | /*prefix-length*]	Specifies the IP address and subnet mask of the client.
Switch(dhcp-config)# **client-identifier** *unique-*	Specifies the unique identifier for DHCP clients.

identifier	This command is used for DHCP requests.
Switch(dhcp-config)# **hardware-address** *hardware-address type*	Specifies a hardware address for the client. This command is used for BOOTP requests. (optional)
Switch(dhcp-config)# **client-name** *name*	Specifies the name of the client using any standard ASCII character. (optional)

3.11. Enable DHCP on an interface

- Create a DHCP pool name (done above)
- Specify a network number and a mask (done above)

Command	**Purpose**
Switch(dhcp-config)# **import all**	Import DHCP option parameters into the DHCP server database.
Switch(dhcp-config)# **exit**	Exits DHCP pool configuration mode.
Switch(config)# **interface** *type number*	Configures an interface and enters interface configuration mode.
Switch(config-if)# **ip addres dhcp** [**client-id** *interface name*] [**hostname** *host-name*]	Specifies that the interface acquires an IP address through DHCP.

4. IPv6 Stateless AutoConfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) server. With IPv6, a device on the link advertises any global prefixes in Router Advertisement (RA) messages, as well as its willingness to function as a default device for the link. RA messages are sent periodically and in response to device solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Device solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

4.1. Setting Up a Link-Local Address

Step #	Command	Purpose
1	Switch# **interface vlan** *vlan-number*	Creates an interface and enters interface configuration mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **interface** *type number*	Configures an interface and enters interface configuration mode.
4	Switch(config-if)# **ip addres autoconfig**	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on an interface and enables IPv6 processing on the interface.

4.2. IPv6 Stateless Autoconfiguration

Step #	Command	Purpose
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **interface** *type number*	Configures an interface and enters interface configuration mode.
4	Switch(config-if)# **ip addres autoconfig**	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on an interface and enables IPv6 processing on the interface.

EXTRA CONFIGURATION

1. IP Access Control Lists

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one permit statement and possibly one or more deny statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into

the network.

- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same permit or deny statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a permit statement, packets are processed after they are received, and when you configure a deny statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you configure a permit statement, packets are sent to the output buffer, and when you configure a deny statement, packets are discarded.
- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

1.1. Sample IP Addresses, Wildcard Masks, and Match Results

Address	Wildcard Mask	Match Result
0.0.0.0	255.255.255.255	All addresses will match the access list conditions.
172.18.0.0/16	0.0.255.255	Network 172.18.0.0
172.18.5.2/16	0.0.0.0	Only host 172.18.5.2 matches
172.18.8.0	0.0.0.7	Only subnet 172.18.8.0/29 matches
172.18.8.8	0.0.0.7	Only subnet 172.18.8.8/29 matches
172.18.8.15	0.0.0.3	Only subnet 172.18.8.15/30 matches
10.1.2.0	0.0.252.255 (noncontiguous bits in mask)	Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0

1.2. Standard ACL

If you want to filter on source address only, a standard access list is simple and sufficient. There are two alternative types of standard access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features than numbered access lists.

1.2.1. Creating Named SACL

Summary Steps:

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
4. **remark** *remark*
5. **deny** {*source* [*source-wildcard*] | **any**} [**log**]
6. **remark** *remark*
7. **permit** {*source* [*source-wildcard*] | **any**} [**log**]
8. Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.
9. **end**
10. **show ip access-list**

Detailed Steps:

Step #	Command	Purpose
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **ip access-list standard** *name*	Defines a standard IP access list using a name and enters standard named access list configuration mode.
4	Switch(config-std-nacl)# **remark** *remark*	Adds a user-friendly comment about an access list entry.
5	Switch(config-std-nacl)# **deny** {*source* [*source-wildcard*] | **any**} [**log**]	Denies the specified source based on a source address and wildcard mask.
6	Switch(config-std-nacl)# **remark** *remark*	Adds a user-friendly comment about an access list entry.
7	Switch(config-std-nacl)# **permit** {*source* [*source-wildcard*] | **any**} [**log**]	Permits the specified source based on a source address and wildcard mask.
8	Repeat 4 - 7	Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list.
9	**end**	Exits standard named access list configuration mode and enters privileged EXEC mode.
10	**show ip access-list**	Displays the contents of all current IP access lists.

1.2.2. Creating numbered SACL

IP standard access lists are numbered 1 to 99 or 1300 to 1999.

Summary Steps:

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]
4. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
5. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
6. **end**
7. **show ip access-list**

Detailed Steps:

Step #	Command	Purpose
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]	Permits the specified source based on a source address and wildcard mask.
4	Switch(config-std-nacl)# **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]	Denies the specified source based on a source address and wildcard mask.
5	Repeat 3 - 4	Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list.
6	**end**	Exits standard named access list configuration mode and enters privileged EXEC mode.
7	**show ip access-list**	Displays the contents of all current IP access lists.

1.3. Extended ACL

If you want to filter on anything other than source address, you need to create an extended access list. There are two alternative types of extended access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features.

1.3.1. Creating Named EACL

Summary Steps:

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** tos] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **permit** protocol source [source-wildcard] destination [destination-wildcard] [**option** option-name] [**precedence** precedence] [**tos** tos] [**established**] [**log** | **log-input**] [**time-range** time-range-name] [**fragments**]
6. Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.
7. **end**
8. **show ip access-list**

Detailed Steps:

Step #	Command	Purpose
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
	Switch(config)# **ip access-list extended** *name*	Defines an extended IP access list using a name and enters extended named access list configuration mode.
3	Switch(config-ext-nacl)# **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Denies any packet that matches all of the conditions specified in the statement.
4	Switch(config-ext-nacl)# **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Permits any packet that matches all of the conditions specified in the statement.
5	Repeat 3 - 4	Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list.
6	Switch(config-ext-nacl)# **end**	Exits standard named access list configuration mode and enters privileged EXEC mode.
7	Switch# **show ip access-list**	Displays the contents of all current

		IP access lists.

1.3.2. Creating Numbered EACL

Summary Steps:

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.
8. **end**
9. **show ip access-list**

Detailed Steps:

Step #	**Command**	**Purpose**
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **access-list** *access-list-number* **remark** *remark*	Adds a user-friendly comment about an access list entry.
4	Switch(config)# **access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Permits any packet that matches all of the conditions specified in the statement.
5	Switch(config)# **access-list** *access-list-number* **remark** *remark*	Adds a user-friendly comment about an access list entry.
6	Switch(config)# **access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Denies any packet that matches all of the conditions specified in the statement.
7	Repeat 3-6	
8	**end**	Exits standard named access list configuration mode and enters privileged

		EXEC mode.
9	**show ip access-list**	Displays the contents of all current IP access lists.

1.4. Applying ACL to an interface

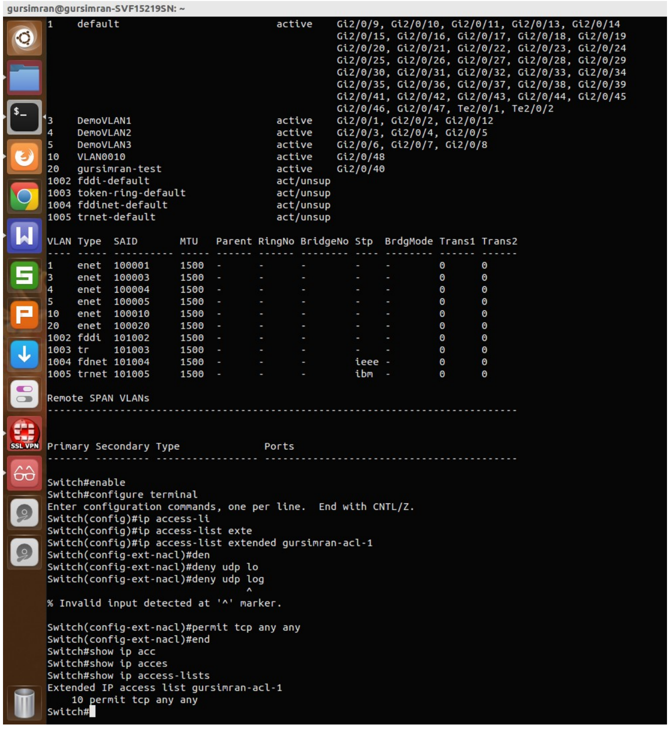
Summary Steps:

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* \| *access-list-name*} {**in** \| **out**}
5. **end**

Detailed Steps:

Step #	Command	Purpose
1	Switch# **enable**	Enables privileged EXEC mode.
2	Switch# **configure terminal**	Enters global configuration mode.
3	Switch(config)# **interface** *type number*	Specifies an interface and enters interface configuration mode.
4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}	Applies the specified access list to the inbound interface.
5	**end**	Exits standard named access list configuration mode and enters privileged EXEC mode.

Example:



gursimran@gursimran-SVF15219SN: ~

```
1    default                          active    Gi2/0/9, Gi2/0/10, Gi2/0/11, Gi2/0/13, Gi2/0/14
                                                Gi2/0/15, Gi2/0/16, Gi2/0/17, Gi2/0/18, Gi2/0/19
                                                Gi2/0/20, Gi2/0/21, Gi2/0/22, Gi2/0/23, Gi2/0/24
                                                Gi2/0/25, Gi2/0/26, Gi2/0/27, Gi2/0/28, Gi2/0/29
                                                Gi2/0/30, Gi2/0/31, Gi2/0/32, Gi2/0/33, Gi2/0/34
                                                Gi2/0/35, Gi2/0/36, Gi2/0/37, Gi2/0/38, Gi2/0/39
                                                Gi2/0/41, Gi2/0/42, Gi2/0/43, Gi2/0/44, Gi2/0/45
                                                Gi2/0/46, Gi2/0/47, Te2/0/1, Te2/0/2
3    DemoVLAN1                        active    Gi2/0/1, Gi2/0/2, Gi2/0/12
4    DemoVLAN2                        active    Gi2/0/3, Gi2/0/4, Gi2/0/5
5    DemoVLAN3                        active    Gi2/0/6, Gi2/0/7, Gi2/0/8
10   VLAN0010                         active    Gi2/0/48
20   gursimran-test                   active    Gi2/0/40
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
3    enet  100003     1500  -      -      -        -    -        0      0
4    enet  100004     1500  -      -      -        -    -        0      0
5    enet  100005     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

Switch#enable
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#ip access-li
Switch(config)#ip access-list exte
Switch(config)#ip access-list extended gursimran-acl-1
Switch(config-ext-nacl)#den
Switch(config-ext-nacl)#deny udp lo
Switch(config-ext-nacl)#deny udp log
                                 ^
% Invalid input detected at '^' marker.

Switch(config-ext-nacl)#permit tcp any any
Switch(config-ext-nacl)#end
Switch#show ip acc
Switch#show ip acces
Switch#show ip access-lists
Extended IP access list gursimran-acl-1
    10 permit tcp any any
Switch#
```

2. VLAN Trunk Protocol

VLAN Trunk Protocol (VTP) reduces administration in a switched network. When you configure a new VLAN on one VTP server, the VLAN is distributed through all switches in the domain. This reduces the need to configure the same VLAN everywhere. VTP is a Cisco-proprietary protocol that is available on most of the Cisco Catalyst series products.

2.1. VTP Modes

You can configure a switch to operate in any one of these VTP modes:

- **Server—**In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters, such as VTP version and VTP pruning, for the entire VTP domain. VTP servers advertise their VLAN configuration to other switches in the same VTP domain and synchronize their VLAN configuration with other switches based on advertisements received over trunk links. VTP server is the default mode.
- **Client—**VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- **Transparent—**VTP transparent switches do not participate in VTP. A VTP transparent switch does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements, but transparent switches do forward VTP advertisements that they receive out their trunk ports in VTP Version 2.
- **Off** (configurable only in CatOS switches)—In the three described modes, VTP advertisements are received and transmitted as soon as the switch enters the management domain state. In the VTP off mode, switches behave the same as in VTP transparent mode with the exception that VTP advertisements are not forwarded.

Configuration

Issue these commands from the VLAN database mode:

- **vtp [client | server | transparent]**
- **vtp domain** *name*

From enable mode, issue these commands in order to monitor VTP operation:

- **show vtp counters**
- **show vtp status**