Mudra's Digital Command Book

Table of Contents

Title	Page
Command Line Tips	3
Syslog Commands	4
Command History	5
Basic Configuration	5-6
Router IP Configuration	6-7
DHCP Configuration	8-9
Configure a Router as a DHCP Client	9
DHCP Debug Commands	9
SLAAC Configuration	10
Stateless DHCPv6 Configuration	10
Stateful DHCPv6 Configuration	10-11
NAT Configuration	11-12
PAT Configuration - Single Global Address	12
PAT Configuration - Global Address Pool	13
Dynamic NAT	14
Port Forwarding	14
Booting/Restoration Commands	15-16
Switchport Security	16-18
Configuring SVI and VLANs on a Switch	18-21
Legacy InterVLAN Routing	21
Modifying and Verifying VLANs	22
Layer 3 Switch Configuration	23

Configuring the Physical Layer of the Switch	**24**
Useful Switch Verification Commands	**24**
Maintaining Router and Switch Files	**25**
CDP	**26**
LLDP	**26**
RIPv2 Configuration	**27**
Single Area OSPFv2	**28-29**
Configure Standard Numbered ACLs	**29-30**
Modify ACLs	**30-31**
Configure Standard Named ACLs	**31**
Configure SSH	**31-32**
Configure Telnet	**32**
Secure VTY Lines with Standard ACLs	**33**
Network Management	**33**
Link Aggregation (EtherChannel) Commands	**34**
Mitigate VLAN Hopping Attacks	**35**
Mitigate DHCP Attacks	**35-36**
Mitigate ARP Attacks	**36**
Mitigate STP Attacks	**36-37**
Configure Extended ACLs	**37-38**
Command Prompt Commands	**39**
Wireless Controller Navigation	**39-40**
FTP Server Setup	**41**
Fixing Corrupted Disk Partition	**41**
HSRP Configuration	**42-43**

COMMAND LINE TIPS

Hot Keys and Shortcuts

Down arrow *– scroll through former commands*
Up arrow *– scroll backwards through former commands*
Tab *– completes the remainder of a partially typed command*
Ctrl+A *– moves to the beginning of the line*
Ctrl+E *– moves to the end of the line*
Ctrl+R *– redisplays a line*
Ctrl-Z *– Exits the config mode and returns to privileged EXEC mode*
Ctrl+C *– exits config mode or aborts current command*
Ctrl+Shift+6 *– Abort mission! Interrupts executing commands like a ping or traceroute*

Show Command Output Filters

| Section = Shows the entire section that starts with the filtering expression
| Include = Includes all output lines that match the filtering expression
| Exclude = Excludes all output lines that match the filtering expression
| Begin = Shows all the output lines from a certain point, starting with the line that matches the filtering

Syslog Protocols

Severity Level	Meaning
Level 0 - Emergency	System Down
Level 1 - Alert	Immediate Action Needed
Level 2 - Critical	Critical Condition
Level 3 - Error	Error Condition
Level 4 - Warning	Warning Condition
Level 5 - Notification	Normal but Significant Condition
Level 6 - Informational	Informational Message
Level 7 - Debugging	Debugging Message

Syslog Commands

`Router(config)# logging server-ip`	Specifies the IP address of the centralized syslog server.
`Router(config)# logging trap severity-level`	Indicates the maximum severity level to be logged to the syslog server. In this example, messages from levels 0 to 5 will be sent to the syslog server.
`Router(config)# logging source-interface int-id`	The router will use the IP address of the interface specified as the source for all syslog messages even if sent out a different exit interface.
`Router(config)# logging console`	Sends all syslog messages to the console (CLI).
`Router(config)# logging buffered`	Gives all syslog messages a timestamp. System clock must be set using NTP or manual setting.
`Router(config)# service timestamps log datetime`	Gives all syslog messages a timestamp. System clock must be set using NTP or manual setting.
`Router(config-line)# logging synchronous`	Line config command. Apply to line console 0. Prevents syslog output from interrupting your console session as you are typing commands.
`Router# show logging`	Displays information about current logging configuration, such as where messages are being sent. Also displays the log buffer.

Command History

`Router#show history`	Displays the commands currently stored in the history buffer. By default, the system captures the last ten command lines in its history buffer.
`Router#terminal history` `Router>terminal history`	Enables terminal history. This command can be run from either user or privileged EXEC mode.
`Router#terminal history size 50`	Configures the terminal history size. The terminal history can maintain 0-256 command lines.
`Router#terminal no history size`	Resets the terminal history size to its default value of 20 command lines in Cisco IOS 15.
`Router#terminal no history`	Disables terminal history.

Basic Configuration

`> enable`
`Switch# configure terminal`
`Switch(config)# hostname name`
	Hostname

`Switch(config)# enable secret password`	Securing Device Access for Privileged Exec mode

`Switch(config)# line console 0`
`Switch(config-line)# password password`
`Switch(config-line)# login`
	Console password secures console access

Switch(config)# **line vty 0 15** *The 0 15 are the range of vty lines the specific switch has* Switch(config-line)# **password** *password* Switch(config-line)# **login**	**VTY password secures telnet and SSH access.**
Switch(config)# **banner motd #***message***#** *The # are the delimiters used in the beginning and end of a message. Can be any character not included in the message.*	**Banner Message (MOTD)**

Router IP Configuration

Router(config)# interface *interface* **Router(config-if)# ip address** *specific ip address and subnet mask* **Router(config-if)# no shutdown**	**Applied Addressing-- Configuring Interfaces**
Router(config-if)# ipv6 address *GUA address/prefix-length*	**Entered in interface configuration mode, this command sets the IPv6 Global Unicast Address.**
Router(config-if)# ipv6 address *FE80::1 link-local*	**Sets a link local address of the interface. Must fall within the range of link-local IPv6 addresses: FE80 to FEBF**
Router(config)# ip route *network address-subnet mask-next [nexthop | exit-int]*	**Configuring Static Routes IPv4**
Router(config)# ipv6 unicast-routing	**Enables routing of IPv6 packets.**
Router(config)# ipv6 route *ipv6-prefix/prefix-length [nexthop | exit-int]* *example:* **Router(config)# ipv6 route** *2001:ABCD:0:1::/64 200:0:0:1::1*	**Configures a static route for the next hop ip address**

Router(config)# ip route 0.0.0.0 0.0.0.0 *next hop ip address or exit-interface*	**Configuring Default Routes IPv4**
Router(config)# ip route *network address-subnet mask-next [nexthop | exit-int] administrative distance* *Notice the difference? You add a higher administrative distance that the primary route*	**Configure a floating static route**
Router(config)# ipv6 route ::/0 *[nexthop | exit-int]*	**Configuring Default Routes IPv6**
Router(config)# interface *loopback number*	**Enabling and assigning a loopback address.**
Router(config)# interface *interface.decimal* **Router(config-if)# encapsulation dot1Q** *vlan-id* **Router(config-if)# ip address** *ip-address subnet-mask* **Router(config-if)# interface** *interface* **Router(config)# no shutdown**	**Router-on-a-Stick Subinterface Configuration**
R1# show ip interface brief **R1# show ipv6 interface brief**	**Verify Configuration**
R1# show ipv6 interface gigabitethernet 0/0/0	**Displays the interface status and all of the IPv6 addresses belonging to the interface. Along with the link local address and global unicast address, the output includes the multicast addresses assigned to the interface.**
R1# show interfaces *g0/0/1.10*	**Subinterfaces can be verified using the show interfaces subinterface-id**
R1# show running-config interface *int*	**Displays the current commands applied to the specified interface**
R1# show ip route **R1# show ipv6 route**	**Verify Routes**
R1# ping *ip address*	**Ping to make sure network connectivity**

DHCP Configuration

`Router(config)# ip dhcp excluded-address ip address`
 	
or
 	
`Router(config)# ip dhcp excluded-address low address - high address`
	Step 1: Exclude IP Addresses from the DHCP Pool
	
	Command marks specified addresses as reserved. Can specify just 1 address, or an optional range of reserved addresses.

`Router(config)# ip dhcp pool POOLNAME`	**Step 2: Create a Named DHCP Pool Creates a pool using the name specified and navigates to the DHCPv4 configuration mode for that pool.**

`Router(dhcp-config)# network network address subnet mask`
`Router(dhcp-config)# default-router gateway-ip`
`Router(dhcp-config)# dns-server ip address`
`Router(dhcp-config)# domain-name www.domain.com`
`Router(dhcp-config)# lease # of days`
	Step 3: Configure IP Parameters for Each Pool As Desired

`Router(config-if)# ip helper-address ip address`	**Use in interfaces of other networks needing access to the DHCP server to which it can forward DHCPDISCOVER broadcast messages.**
`Router(config)# no service dhcp`	**Disables DHCPv4 service on a router. DHCPv4 Service is by default on Cisco IOS.**
`Router(config)# service dhcp`	**Enables DHCPv4 service on a router (DHCP is enabled on router by default)**
`Router# show ip dhcp binding`	**Shows the list of IP addresses leased, binded to the MAC addresses of the host to which they have been leased.**

`Router# show running-config | section dhcp`	**Filters running config output to DHCPv4 commands.**
`Router# show ip dhcp server statistics`	**Displays counters related to the number of DHCP messages that have been sent or received; handy for troubleshooting.**
`Router# show ip interface int`	**Verify an interface on a router has been configured with a helper address to make the router a DHCP relay agent.**

Configure a Router as a DHCP Client

`Router(config-if)# ip address dhcp`	**Interface config command. Command tells the router to use DHCP to retrieve an IP address for this interface.**

DHCP Debug Commands

`Router(config)# ip access-list 99 permit udp any any eq 67`
 	
`Router(config)# ip access-list 99 permit udp any any eq 68`
	Creates an extended ACL number 99 to only permit traffic on UDP ports 67 and 68, which are the ports DHCP messages use. Only DHCP messages will be permitted using this command.

`Router# debug ip packet 99`	**Turns on debugging for IP packets that are permitted through the ACL created. The router will generate a syslog message describing the details of all IP packets processed.**

`Router# debug ip dhcp server events`	Generates syslog messages for DHCP server events such as address assignment and lease expiration.

SLAAC Configuration

`Router(config-if)# no ipv6 nd other-config-flag`	Sets the other-config-flag to 0
`Router(config-if)# no ipv6 nd managed-config-flag`	Sets the managed-config-flag to 0

Stateless DHCPv6 Configuration

`Router(config)# ipv6 unicast-routing`	Step 1: Enable IPv6 routing
`Router(config)# ipv6 dhcp pool` *POOLNAME*	Step 2: Create a DHCPv6 pool
`Router(config-dhcpv6)# dns-server` *ipv6 address* `Router(config-dhcpv6)# domain-name` *domain.com*	Step 3: Configure pool parameters
`Router(config-if)# ipv6 dhcp server` *POOLNAME*	Binds the pool specified to the interface
`Router(config-if)# ipv6 nd other-config-flag`	Sets the other-config-flag to 1
`Router(config-if)# no ipv6 nd managed-config-flag`	Sets the managed-config-flag to 0

`Router(config-if)# interface` *int*
`Router(config-if)# ipv6 enable`
`Router(config-if)# ipv6 address autoconfig`
	Steps to Configure the Client.

Stateful DHCPv6 Configuration

`Router(config)# ipv6 unicast-routing`	Step 1: Enable IPv6 routing

Router(config)# ipv6 dhcp pool *POOLNAME*	**Step 2: Create a DHCPv6 pool**
Router(config-dhcpv6)# address prefix *prefix/prefix-length* *Specifies the prefix and prefix length. Can use an optional* ***lifetime*** *keyword to specify a lifetime in seconds.* **Router(config-dhcpv6)# dns-server** *ipv6 address* **Router(config-dhcpv6)# domain-name** *domain.com*	**Step 3: Configure pool parameters**
Router(config-if)# ipv6 dhcp server *POOLNAME* **Router(config-if)# ipv6 nd managed-config-flag** **Router(config-if)# ipv6 nd prefix default no-autoconfig** ****The A flag is manually changed from 1 to 0 using the interface command* **ipv6 nd prefix default no-autoconfig**. *The A flag can be left at 1, but some client operating systems such as Windows will create a GUA using SLAAC and get a GUA from the stateful DHCPv6 server. Setting the A flag to 0 tells the client not to use SLAAC to create a GUA.*	**Step 4: Configure DHCPv6 on applicable interfaces with the managed-config-flag set to 1.**
Router(config-if)# ipv6 enable **Router(config-if)# ipv6 address dhcp**	**Configure the Client for Stateful DHCP**
Router(config-if)# ipv6 dhcp relay destination *ipv6-address [interface]*	**A DHCPv6 Relay Agent Configuration**

NAT Configuration

Router(config)# ip nat inside source static *local-ip address global ip-address*	**Step 1: Create the static NAT table entry**
Router(config-if)# ip nat inside **Router(config-if)# ip nat outside**	**Step 2: Mark all inside and outside interfaces**
Router# show ip nat translations	**Displays the NAT table on a NAT enabled router.**

`Router# show ip nat statistics`	**Displays counters related to NAT translations, such as how many packets have been translated.**
`Router# clear ip nat statistics`	**Resets counters related to NAT translations. Clear NAT statistics after troubleshooting NAT problems.**
`Router# debug ip nat`	**Displays basic information about each IP address translated**
`Router# debug ip nat detailed`	**Similar to debug ip nat command, but displays more information, including errors.**

PAT Configuration - Single Global Address

	Configure Port Address Translation
`Router(config)# access-list ACL# [permit | deny] source-ip source-wildcard`	**Step 1: Configure an ACL Used to Define Which Local Addresses Are Eligible for Translation**
`Router(config)# ip nat inside source list ACL# interface int overload` *Ties the ACL to the NAT process and specifies the exit interface whose IP address will be used for translation. Also allows the address to be overloaded (used for more than one inside device).*	**Step 2: Tie the ACL to the NAT Process and Specify the Global Address Used for Translation.**
`Router(config-if)# ip nat inside` `Router(config-if)# ip nat outside`	**Step 3. Mark Inside and Outside Interfaces:**

PAT Configuration - Global Address Pool

	Configure Port Address Translation
Router(config)# ip nat pool *POOLNAME first-ip last-ip* **netmask** *subnet-mask* *Creates a pool of global addresses to be used for NAT.*	**Step 1: Define a Pool of Global Addresses**
Router(config)# access-list *ACL#* **[permit | deny]** *source-ip source-wildcard*	**Step 2: Configure an ACL Used to Define Which Local Addresses Are Eligible for Translation**
Router(config)# ip nat inside source list *ACL#* **pool** *POOLNAME* **overload** *Ties the ACL specified to the NAT process and specifies the NAT pool to be used for translation. Also specifies the address will be overloaded.*	**Step 3: Tie the ACL to the NAT Process and Specify the NAT Pool to be Used for Translation**
Router(config-if)# ip nat inside **Router(config-if)# ip nat outside**	**Step 4. Mark Inside and Outside Interfaces:**
Router# show ip nat translations	**Displays the NAT table on a NAT enabled router.**
Router# show ip nat statistics	**Displays counters related to NAT translations, such as how many packets have been translated.**

Dynamic NAT

Router(config)# ip nat pool *POOLNAME first-ip last-ip* **netmask** *subnet-mask* *Creates a pool of global addresses to be used for NAT.*	**Step 1: Define a Pool of Global Addresses**
Router(config)# access-list *ACL#* **[permit | deny]** *source-ip source-wildcard*	**Step 2: Configure an ACL Used to Define Which Local Addresses Are Eligible for Translation**
Router(config)# ip nat inside source list *ACL#* **pool** *POOLNAME* *Ties the ACL specified to the NAT process and specifies the NAT pool to be used for translation.*	**Step 3: Tie the ACL to the NAT Process and Specify the NAT Pool to be Used for Translation. For Dynamic NAT, there is no overload.**
Router(config-if)# ip nat inside **Router(config-if)# ip nat outside**	**Step 4. Mark Inside and Outside Interfaces.**
Router# show ip nat translations	**Displays the NAT table on a NAT enabled router.**
Router# show ip nat statistics	**Displays counters related to NAT translations, such as how many packets have been translated.**

Port Forwarding

Router# ip nat inside source static [tcp | udp] *inside-IP inside-port outside-ip outside-port*	**Port Forwarding is just static NAT configured with an added port parameter.**

Important Booting/Restoring Commands

`Router# copy running-config startup-config`	**To Save Changes**

`Router# erase startup-config`
`Router# delete flash:vlan.dat`
`Router# reload`
	To Restore Factory defaults on a Router

`Switch(config)# boot system` *`flash:/c2960-lanbasek9-mz.150-2.SE/c2960-lanbasek9-mz.150-2.SE.bin`* *`*** Boot System command dissection:`* *`boot system`* **`The main command`** *`Flash:`* **`The storage device`** *`c2960-lanbasek9-mz.150-2.SE/`* **`The path to the file system`** *`c2960-lanbasek9-mz.150-2.SE.bin`* **`The IOS file name`**	**BOOT environment variable is set using the boot system global configuration mode command on a switch.**
`switch: set` **`switch: flash_init`** **`switch: dir flash:`** **`switch: BOOT=`***`flash:c2960-lanbasek9-mz.150-2.SE8.bin`*** **`switch: set`** **`switch: boot`** *`***`* *`Step 1. Connect a PC by console cable to the switch console port. Configure terminal emulation software to connect to the switch.`* *`Step 2. Unplug the switch power cord.`* *`Step 3. Reconnect the power cord to the switch and, within 15 seconds, press and hold down the Mode button while the System LED is still flashing green.`* *`Step 4. Continue pressing the Mode button until the System LED turns briefly amber and then solid green; then release the Mode button.`* *`Step 5. The boot loader switch: prompt appears in the terminal emulation software on the PC.`* *`After that, type in the following commands. Make the boot variable the IOS you found through the dir flash: command`*	**To recover from a system crash on a switch, do the following:**

`Switch# erase startup-config` `Switch# delete vlan.dat`	**Restores Switch to factory default.**

Switchport Security

`Switch(config)# interface range int#-#` `Switch(config-if-range)# shutdown`	**Shutdown unused ports.**
`Switch(config)# interface int` `Switch(config)#  switchport mode access`	**Set the interface you want to configure to *access* mode.**
`Switch(config-if)# switchport port-security`	**With this command, port security will be enabled with default settings.**
`Switch(config-if)# switchport port-security maximum #`	**Sets the maximum number of unique MAC addresses that can be used on a port**

`Switch(config-if)# switchport port-security violation shutdown`
`Switch(config-if)# switchport port-security violation restrict`
`Switch(config-if)# switchport port-security violation protect`
	The violations do the following:
	
	Shutdown - Stops forwarding traffic, shuts port down, and increases violation counter
	Restrict - Stops forwarding traffic, increases violation counter, and sends syslog message
	Protect - Stops forwarding traffic

`Switch(config-if)# switchport port-security mac-address mac-address`	**Configures a static secure MAC address on a switchport.**

`Switch(config-if)# switchport port-security aging { static | time time | type {absolute | inactivity}}`	***static*** **- Enable aging for statically configured secure addresses on this port.** ***time*** *time* **- Specify the aging time for this port. The range is 0 to 1440 minutes. If the time is 0, aging is disabled for this port.** ***type*** *absolute* **- Set the absolute aging time. All the secure addresses on this port age out exactly after the time (in minutes) specified and are removed from the secure address list.** ***type*** *inactivity* **- Set the inactivity aging type. The secure addresses on this port age out only if there is no data traffic from the secure source address for the specified time period.**
`Switch(config-if)# switchport port-security mac-address sticky`	**Configures the use of sticky learning on a switchport.**
`Switch# clear mac-address-table`	**Clears the CAM table**
`Switch# show port-security interface int`	**Displays port security settings for the interface, violation count, and last used source address. Output is specific to the interface specified.**
`Switch# show port-security address`	**Shows all secure MAC addresses known by the switch, what type they are, and what port they are on.**
`Switch# show running-config | begin int`	**Used to find the specific interface.**

`Switch# show mac-address-table`	**Displays contents of the switch's CAM table.**
`Switch# show interfaces`	**Displays information about *every* interface on the switch. With this command, you can find the status of the port here if you think it might be in error-disabled mode.**
`Switch# show interface int`	**Displays the same information as the *show interfaces* command, but for a specific port.**
`Switch# show interface int status`	**Same as the previous command, but it only shows the status of the port. Handy if you just need to check if it is in error-disabled mode.**

`Switch(config)# interface int`
`Switch(config-if)# shutdown`
`Switch(config-if)# no shutdown`
	To bring port back up from error-disabled mode

Configuring SVI and VLANs on a Switch

`Switch# configure terminal`
`Switch(config)# interface vlan 99`
`Switch(config-if)# ip address 172.17.99.11 255.255.255.0`
`Switch(config-if)# ipv6 address ipv6 address/prefix`
`Switch(config-if)# no shutdown`
`Switch(config-if)# end`
`Switch# copy running-config startup-config`
 	
`** If running a catalyst 2960 switch, in order to have an IPv6 VLAN, do:`
`Switch(config)#sdm prefer dual-ipv4-and-ipv6 default`
	Configure the Management VLAN

Switch# configure terminal Switch(config)# ip default-gateway *172.17.99.1* Switch(config)# end Switch# copy running-config startup-config	Configure the Default Gateway
Switch# configure terminal Switch(config)# vlan *vlan-id* Switch(config-vlan)# name *vlan-name* Switch(config-vlan)# end	Add a VLAN to a switch and give it a name. Naming each VLAN is considered a best practice in switch configuration.
Switch(config)# vlan *VLAN-ID* Switch(config-vlan)# name *VLAN-name*	Gives a descriptive name to the VLAN that displays in the vlan.dat file.
Switch(config)# vlan *100,102,105-107*	Creates VLANs, but won't have names.
Switch# configure terminal Switch(config)# interface *interface-id* OR Switch(config)# interface range *int-int,int* Switch(config-if)# switchport mode access Switch(config-if)# switchport access vlan *vlan-id* Switch(config-if)# end	VLAN port assignment switchport access vlan command automatically creates a VLAN if it had not been created before assigning.
Switch(config-if)# mls qos trust cos Switch(config-if)# switchport voice vlan *vlan-id*	Voice VLAN assignment
Switch(config)# interface *int* Switch(config-if)# no switchport access vlan	Assigns port back to default.
Switch(config)# no vlan *vlan-id*	Deletes a VLAN. Be sure to reassign the accesports before shutting down, or there would be a black hole.
Switch# delete flash:vlan.dat OR Switch# delete vlan.dat	Deletes the entire vlan.dat file.
Switch(config)# interface *interface-id* Switch(config-if)# switchport mode trunk	Configure a trunk link that would only allow particular

`Switch(config-if)# switchport trunk native vlan vlan-id`
`Switch(config-if)# switchport trunk allowed vlan vlan-list`
`Switch(config-if)# end`
	VLANs and define the Native VLAN.

`Switch(config)# interface interface-id`
`Switch(config-if)# no switchport trunk allowed vlan vlan-list`
`Switch(config-if)# no switchport trunk native vlan vlan-id`
`Switch(config-if)# end`
	Reset the trunk link to default.

`Switch(config-if)# switchport mode trunk` `Switch(config-if)# switchport nonegotiate`	**Enable trunking from a Cisco switch to a device that does not support DTP.** *****For trunking to work properly, DTP has to be set as it is a Cisco proprietary protocol for 2960 and 3650 switches.**
`Switch(config-if)# switchport mode dynamic auto`	**To re-enable dynamic trunking protocol**
`Switch(config-if)# switchport mode dynamic desirable`	**To enable Trunk link between switches, do not do dynamic auto.**
`Switch# show interfaces trunk`	**Verify the active trunk links on a Layer 2 switch and troubleshoot misconfiguration**
`Switch# show dtp interface interface`	**Verify DTP** *****A general best practice is to set the interface to trunk and nonegotiate when a trunk link is required.**
`Switch# show ip interface brief` `Switch# show ipv6 interface brief`	**Verify Configuration**
`Switch# show vlan brief`	**Verifies Name, Status and ports assigned to VLAN.**

`Switch# show vlan id vlan-id`	**Display information about the identified VLAN ID number. For vlan-id, the range is 1 to 4094.**
`Switch# show vlan name vlan-name`	**Display information about the identified VLAN name. The vlan-name is an ASCII string from 1 to 32 characters.**
`Switch# show vlan summary`	**Display VLAN summary information, such as the number of existing VLANs and the number of existing VTP VLANS and the existing extended VLANS.**
`Switch# show interfaces int switchport`	**Shows interface switchport information such as VLANs and if they were correctly assigned.**

Legacy InterVLAN Routing

`Switch(config)# vlan number` `Switch(config-vlan)# name name`	**Name the VLANs required.**
Assuming router IP addresses have already been configured.	
`Switch(config)# interface int` `Switch(config)# switchport mode access`	**Configure Access Ports. Access ports are ports connected to host devices** **Ports connected to the router are also considered access ports.**

`Switch(config)# interface int`
`Switch(config)# switchport mode trunk`
`Switch(config)# switchport trunk native vlan native-vlan`
`Switch(config)# switchport trunk allowed vlan vlan,vlan-vlan`
	Configure trunk ports-- or ports connected to other switches that will be transmitting data.

Modifying and Verifying VLAN.dat file

`Switch# delete flash:vlan.dat`
`Switch# erase startup-config`
`Switch# reload`
	Restores switch to factory defaults

`Switch# show vlan [brief | name vlan-name| id vlan-num| summary]`	Option 1: Shows all VLANs, VLAN names, and port assignments.' Option 2: Show a single VLAN by name. Option 3: Show a single VLAN by VLAN number. Option 4: Show a summary of VLAN stats such as how many VLANs are on the switch.
`Switch# show interfaces [int-id | int-id switchport | vlan vlan-num|trunk]`	Option 1: Show detailed information about a specific port, limited VLAN information. Option 2: Show detailed VLAN settings for an interface, all VLAN information. Particularly good for interface trunking information. Option 3: Show detailed information about a virtual interface. Option 4: Show an easy to read list of trunk ports configured.

Layer 3 Switch Configuration

`Switch(config)# ip routing`	Enable IP Routing.

`Switch# configure terminal`
`Switch(config)# interface vlan vlan-id`
`Switch(config)# ip address ip-address subnet-mask`
`Switch(config)# no shut`
	Configure the SVI VLAN interfaces. The IP addresses that are configured will serve as the default gateways to the hosts in the respective VLANs.

`Switch# configure terminal`
`Switch(config)# vlan vlan-id`
`Switch(config-vlan)# name vlan-name`
`Switch(config-vlan)# end`
	Add a VLAN to a switch and give it a name. Naming each VLAN is considered a best practice in switch configuration.

`Switch(config)# interface interface`
`Switch(config-if)# switchport mode access`
`Switch(config-if)# switchport access vlan vlan-id`
`Switch(config-if)# no shut`
	Configure Access Ports.

`Switch(config)# interface interface`
`Switch(config-if)# switchport trunk encapsulation dot1q`
`Switch(config-if)# switchport mode trunk`
`Switch(config-if)# switchport trunk native vlan vlan-id`
	Enable trunking on a layer 3 switch.

`Switch(config)# interface interface`
`Switch(config-if)# no switchport`
`Switch(config-if)# ip address ip-address subnet-mask`
	If VLANs are to be reachable by other Layer 3 devices, then they must be advertised using static or dynamic routing. To enable routing on a Layer 3 switch, a routed port must be configured.

`Switch(config)# router ospf num` `Switch(config-if)# network network-address wildcard-mask area 0`	To configure routing--OSPF on a routed port (previous command)
`Switch# show ip route | begin Gateway`	Verify the routing table

Configuring the Physical Layer of the Switch

Switch# configure terminal Switch(config)# interface FastEthernet 0/1 Switch(config-if)# duplex full Switch(config-if)# speed 100 Switch(config-if)# end	Enter global configuration mode. Enter interface configuration mode. Configure the interface duplex. Configure the interface speed. Return to the privileged EXEC mode.
Switch(config-if)# mdix auto	Configure Auto-MDIX which means a straight-through or crossover cable can be used. Ports would negotiate.
Switch# show controllers	To examine the auto-MDIX setting.
Switch# show controllers ethernet-controller *fa0/1* phy | include MDIX	To examine the auto-MDIX setting for a specific interface.

Useful Switch Verification Commands

Switch# show interfaces [interface-id]	Display interface status and configuration.
Switch# show startup-config	Display current startup configuration.
Switch# show running-config	Display current running configuration.
Switch# show flash	Display information about the flash file system.
Switch# show version	Display system hardware and software status. An IOS file with "k9' supports encryption.
Switch# show history	Display history of command entered.
Switch#show ip interface [interface-id] OR Switch#show ipv6 interface [interface-id]	Display IP information about an interface.
Switch# show mac-address-table OR Switch# show mac address-table	Display the MAC address table.
Switch# show spanning-tree vlan *vlan ID*	Check STP (Spanning Tree Protocol) to gather information about the status.

Maintaining Router and Switch Files

`Switch# show file systems`	**Shows all file systems on the device, including flash and nvram. The default file system, flash, is indicated with an asterisk. The bootable file system, also flash, is indicated with a #.**
`Switch# dir`	**"Change Directory", navigates to the directory indicated in the command. In this example, nvram.**
`Switch# cd nvram:`	**Verifies current directory.**
`Switch# pwd`	**Display information about the flash file system.**
`Switch# show version` *Not specifically related to file systems, is a very important command needed to see basic information about the device. It can be used when updating from one IOS version to another. For instance, in order to update the IOS, you might ask yourself: what version of IOS am I currently running and do I have enough memory to support a newer version? The Show version command can answer those questions.If you were to display the contents of the flash directory, you would find the Cisco IOS image. The name of the file varies depending on the version you are running and is quite long and somewhat coded, but it is typically the only.bin file there.*	**Not a command directly related to the file system. Shows basic information about the device, such as hardware specs and IOS version.**
`Switch# copy running-config tftp`	**Copies the file indicated in the command *to* the TFTP server. (Source: router or switch, destination: TFTP server)**
`Switch# copy tftp running-config`	**Copies the file indicated in the command *from* the TFTP server. (Source: TFTP server, destination: router or switch)**

CDP

`Router(config)# cdp run` `Router(config)# no cdp run`	Enables CDP for the entire router or switch. Using the *no* keyword before the command disables CDP for the entire device. It is enabled by default.
`Router(config-if)# cdp enable` `Router(config-if)# no cdp enable`	Enables CDP for the interface. Disable for the interface by using *no* keyword.
`Router# show cdp`	Shows the status of CDP.
`Router# show cdp interface`	Displays which interfaces are CDP enabled.
`Router# show cdp neighbors`	Shows a list of known neighbors with basic information about each one such as its hostname, the local port and remote port ID, and what type of device it is (router vs switch).
`Router# show cdp neighbors detail`	Shows more detailed information, such as known IP addresses and IOS version.

LLDP

`Router(config)# lldp run` `Router(config)# no lldp run`	Enables CDP for the entire router or switch. Using the *no* keyword before the command disables CDP for the entire device. It is enabled by default.

`Router(config-if)# lldp transmit`
`Router(config-if)# lldp receive`
 	
`Router(config-if)# no lldp transmit`
`Router(config-if)# no lldp receive`
	Enables transmitting or receiving LLDP frames for the interface. Disable for the interface by using the *no* keyword.

`Router# show lldp`	Shows the status of LLDP.
`Router# show lldp neighbors`	Shows a list of known neighbors with basic information about each one such as its hostname, the local port and remote port ID, and what type of device it is (router vs switch).
`Router# show lldp neighbors detail`	Shows more detailed information, such as known IP addresses and IOS version.

RIPv2 Configuration

`Router(config)# router rip`
`Router(config-router)# version 2`
`Router(config-router)# no auto-summary`
	Enables RIPv2, and disables auto summary

`Router(config-router)# network` *network address*	Enables RIP on all interfaces belonging to the specified network.
`Router(config-router)#` `passive-interface` *int*	Disables sending RIP updates out of the specified interface.
`Router(config-router)#` `default-information originate`	Adds a router's default route to the RIP updates sent out by the router.
`Router(config)# no router rip`	Disables RIP and erases all existing RIP configurations
`Router(config-router)# version 1`	Enables RIPv1 only
`Router(config-router)# no version`	Restores router to default version settings: the router will use RIPv1 updates but will listen for both RIPv1 and RIPv2 updates
`Router(config-router)#` `passive-interface default`	Makes passive interface the default setting--convenient if you have mostly passive interfaces and you'd rather indicate which interfaces should not be passive.
`Router(config-router)no` `passive-interface` *int*	Re-enables RIP updates on an interface that has been marked as passive.
`Router# show ip protocols`	Displays IPv4 routing protocol settings; use when verifying the settings of any configured routing protocol, including RIP, EIGRP, or OSPF
`Router# show ip route`	Display the IPv4 routing table

Single Area OSPFv2

`Router(config)# router ospf process ID`
`Router(config-router)# version 2`
`Router(config-router)# no auto-summary`
	Enable OSPF. Process ID is any number between 1 and 65,535

`Router(config)# router-id #.#.#.#`
`or`
`Router(config)# interface loopback 0`
`Router(config-if)# ip address ip`
	Configure the Router ID. It assigns the router an ID to use with OSPF. Values are between 0.0.0.0 and 255.255.255.255. The Router ID can be the lowest loopback address automatically.

`Router(config-if)# interface loopback 0` `Router(config-if)# ip ospf network point-to-point`	To simulate a real LAN, the Loopback 0 interface is configured as a point-to-point network so that R1 will advertise the full network to connected LANs.

`Router(config-router)# network network wildcard mask area area-id`
 	
`Router(config-router)# network interface-IP-address 0.0.0.0 area 0`
 	
`Router(config-if)# ip ospf process-id area area-id`
	-Specifies the directly connected interface that will participate in OSPF and the area to which it belongs.
	-Alternative to indicating interfaces that will participate in OSPF that does not require the calculation of wildcard masks. Uses the router interface address with a 0.0.0.0 wildcard mask (exact match).
	-Configure OSPF directly on the interface instead of using the network command.

`Router(config-router)# passive-interface int`	Specifies which interface will be passive.
`Router(config-router)# passive-interface default`	Makes *passive* the default for all interfaces on the router. Active interfaces must be configured.
`Router(config-router)# no passive-interface int`	Makes a router interface active.
`Router(config)# interface g0/0` `Router(config)# ip ospf priority 255`	Configure OSPF priority for DR or BDR election. Range from 1-255. DR=255 ; BDR=1 ; others=0
`Router# show ip ospf neighbor`	Displays neighbor adjacency table and the state of each adjacency. Routing issues will occur if neighbor adjacencies aren't in FULL state. FULL state indicates the two neighbors have identical LSDBs, as they should.

`Router# show ip ospf database`	Displays contents of LSDB, which contains the entire network topology and should be identical on all routers in the area.
`Router# show ip route`	Displays routing table. Only the best routes from the LSDB will appear in the routing table.
`Router# show ip protocols`	Handy for use with any dynamic routing protocol. Shows basic information about the routing protocol configuration, such as OSPF router ID, process ID, advertised networks and areas on which they are operating, and neighbors.
`Router# show ip ospf`	Shows detailed information such as process ID, router ID, area information, and events such as SPF calculations.
`Router# show ip ospf interface brief`	Shows basic information relating to OSPF for each OSPF enabled interface, including area, IP configuration, cost, and state.
`Router# show ip ospf interface int`	Shows detailed information relating to OSPF for the specified interface.

Configure Standard Numbered ACLs

`Router(config)# access-list acl# [permit | deny] source-ip source-wildcard-mask`	Configure a Standard Numbered ACL if it doesn't already exist and adds a permit or deny entry. ACL number is an identification number, must be between 1 and 99.

`Router(config)# access-list acl# permit | deny] host-ip-address 0.0.0.0`
 	
or
 	
`Router(config)# access-list acl# permit | deny] host host-ip-address`
	Only specific host is [permitted | denied] entry

`Router(config)# access-list acl# permit | deny] 0.0.0.0 255.255.255.255`	**255.255.255.255 indicates that no bits have to match in the IP address for this statement to be a match. That means that this statement will match all IP addresses.**
`Router(config)# access-list acl# permit | deny] any`	**The *any* keyword can and should be used with any permit or deny statement that is meant to be a match to all IP addresses. These types of statements are commonly found at the end of ACLs. Remember, there is an implicit *deny any* statement at the end of every ACL.**
`Router(config)# access-list acl# remark comment`	**Creates the ACL if it doesn't already exist and adds a remark entry. A remark entry is not executed. It is for documentation purposes only and allows you to put comments in the ACL to make them easier to understand.**
`Router(config-if)# ip access-group acl# [in | out]`	**Applies the ACL specified to [incoming | outgoing] traffic on the interface.**
`Router# show access-lists`	**Displays all ACLs configured on the router along with statistics for how many packets have matched each ACE.**

Modify ACLs

`Router(config)# ip access-list standard acl#`	**Navigates to standard named ACL mode for the ACL number specified. From here, you can add or remove entries.**
`Router(config-std-nacl)# no ACE#`	**Removes the ACE specified from the ACL you have navigated to.**
`Router(config-std-nacl)# ACE# [permit | deny] source-ip source-wildcard`	**Adds a new statement to the ACL to which you have navigated. Can be *permit* or *deny*, and can also use *any* or *host* keywords. Must specify an ACE number.**
`Router(config)# no access-list ACL#`	**Deletes the *entire* ACL. be sure to remove any interfaces to which it has been applied before deleting.**

`Router(config-if)# no access-group ACL#`	Removes interface applied to ACL. Do this before deleting ACL
`Router# show ip interface int`	Can be used to check what ACL has been applied to an interface.
`Router# clear access-list counters`	Used during testing to reset all counters displayed using the *show access-lists* command discussed above.

Configure Standard Named ACLs

`Router(config)# ip access-list standard ACL-NAME`	Creates an ACL using the name specified and navigates to named ACL configuration mode for the ACL.

`Router(config-std-nacl)# [permit | deny] host-ip-address 0.0.0.0`
 	
or
 	
`Router(config)# access-list acl# permit | deny] host host-ip-address`
	Only specific host is [permitted | denied] entry

`Router(config)# access-list acl# permit | deny] 0.0.0.0 255.255.255.255`	255.255.255.255 indicates that no bits have to match in the IP address for this statement to be a match. That means that this statement will match all IP addresses.

Configure SSH

`Switch(config)# interface vlan vlan-id`
`Switch(config-if)# ip address ip-address subnet-mask`
`Switch(config-if)# no shutdown`
	Configure an IP address for the switch on a VLAN

`Switch(config)# enable secret password`	Configure a secret password for Privileged EXEC mode
`Switch(config)# line vty 0 15`	Configure the VTY lines to use SSH

`Switch(config-line)# transport input ssh` `Switch(config-line)# login local`	
`Switch(config)# hostname hostname` `hostname(config)# ip domain-name domain.com`	**Configure a hostname and domain name**

`Switch(config)# crypto key generate rsa`
 	
`Will prompt for a modulus length. Recommended length is 1024`
	Generate RSA keys

`Switch(config)# ip ssh version 2`	**Enables SSH version 2. May or may not be the default, depending on the IOS version you are using.**
`Switch(config)# username username secret password`	**Configure users**
`Switch# show ip ssh`	**Verify SSH support. If the switch is not running an IOS that supports cryptographic features, this command is unrecognized.**

Configure Telnet

`Switch(config)# enable secret password`	**Configure a secret password for Privileged EXEC mode**

`Switch(config)# interface vlan vlan-id`
`Switch(config-if)# ip address ip-address subnet-mask`
`Switch(config-if)# no shutdown`
	Configure an IP address for the switch on a VLAN

`Switch(config)# ip default-gateway ip-address`	**Configure a default gateway.**

`Switch(config)# line vty 0 15`
`Switch(config-line)# transport input telnet`
`Switch(config-line)# login local`
`Switch(config-line)# exit`
	Configure the vty lines.

`Switch(config-line)# exec-timeout min`	**Specifies how long until the VTY connection times out due to inactivity and the user is automatically logged out. Default is 10 minutes.**

Secure VTY Lines with Standard ACLs

`Router(config)# access-list acl# permit | deny] ip address range`	Defines the ACL with the permitted IP addresses allowed to access the VTY Lines.
`Router(config)# line vty low-num high-num`	Navigates to VTY line(s) specified. Can be a range or a single line.

`Router(config-line)# access-class acl-name [in | out]`
 	
or
 	
`Router(config-line)# access-class acl-num [in | out]`
	Applies ACL specified to either incoming or outgoing VTY traffic.

Network Management

`Router(config)# ntp server source-ip-address`	Configures the device as an NTP server. The IP address should be that of an authoritative time source.
`Router# show clock`	Displays current time setting.
`Router# show clock detail`	Includes time source in output. Can be used to verify if NTP is in use.
`Router# show ntp status`	Privileged EXEC command. Shows current status of NTP synchronization, stratum, and IP address of server being used.
`Router# show ntp associations`	Displays a list of time sources in use.
`Router# clock set hh:mm:ss month day year`	Manually sets the clock with time and date.

Link Aggregation (EtherChannel) Commands

`Switch(config)# interface range f0/1-2`	Greate the range of interfaces you want to aggregate.

`Switch(config-if-range)# channel-group <1-6>`
`Switch(config-if)# channel-group <1-6> mode mode`
 	
`***Modes to choose from:`
`active` **Enable LACP unconditionally**
`auto` **Enable PAgP only if a PAgP device is detected**
`desirable` **Enable PAgP unconditionally**
`on` **Enable Etherchannel only**
`passive` **Enable LACP only if a LACP device is detected**
	Can have up to 6 channels to aggregate.

`Switch(config)# interface port channel <1-6>`
`Switch(config-if)# switchport mode trunk`
`Switch(config-if)# switchport trunk allowed vlan 1,2,20`
	Configure the switchport settings through the port channel. All interfaces in the port channel will be affected.

`Switch# show interfaces port-channel <1-6>`	Verify the general status of the port channel interface.
`Switch# show etherchannel summary`	The output of the command shows the complete summary of the port channel with flags.
`Switch# show run | begin interface port-channel`	The configuration shown for port-channels in the running-config file.

Mitigate VLAN Hopping Attacks

`Switch(config)# interface range f0/1-16` `Switch(config-if-range)# switchport mode access`	**Step 1: Disable DTP (auto trunking) negotiations on non-trunking ports.**

`Switch(config)# interface range f0/17-20`
`Switch(config-if)# switchport mode access`
`Switch(config-if)# switchport access vlan 1000`
`Switch(config-if)# shutdown`
	Step 2: Disable unused ports and put them in an unused VLAN.

`Switch(config)# interface range f0/21-24`
`Switch(config-if)# switchport mode trunk`
`Switch(config-if)# switchport nonegotiate`
`Switch(config-if)# switchport trunk native vlan 999`
	Step 3: Manually enable the trunk link on a trunking port by using the switchport mode trunk command.
	
	Step 4: Disable DTP (auto trunking) negotiations on trunking ports by using the switchport nonegotiate command.
	
	Step 5: Set the native VLAN to a VLAN other than VLAN 1 by using the switchport trunk native vlan vlan_number command.

Mitigate DHCP Attacks

`Switch(config)# ip dhcp snooping`	**Step 1. Enable DHCP snooping**
`Switch(config)# interface range f0/1` `Switch(config-if)# ip dhcp snooping trust`	**Step 2. On trusted ports, use the ip dhcp snooping trust interface configuration command.**
`Switch(config)# interface range f0/21-24` `Switch(config-if)# ip dhcp snooping limit rate 6`	**Step 3. Limit the number of DHCP discovery messages that can be received per second on untrusted ports.**

`Switch(config)# ip dhcp snooping vlan 5,6,50-52`	**Step 4. Enable DHCP snooping by VLAN, or by a range of VLANs.**
`Switch# show ip dhcp snooping`	**Command to verify DHCP snooping.**
`Switch# show ip dhcp binding`	**Command to view the clients that have received DHCP information.**

Mitigate ARP Attacks

`Switch(config)# ip dhcp snooping`
`Switch(config)# ip dhcp snooping vlan 10`
`Switch(config)# ip arp inspection vlan 10`
`Switch(config)# interface f0/24`
`Switch(config-if)# ip dhcp snooping trust`
`Switch(config-if)# ip arp inspection trust`
	Configuring Dynamic ARP Inspection. In order to have DAI, you need to configure DHCP Snooping.

`Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}`	**Configure DAI to drop ARP packets when the IP addresses are invalid. It can be used when the MAC addresses in the body of the ARP packets do not match the addresses that are specified in the Ethernet header.**

Mitigate STP Attacks

`Switch(config)# spanning-tree portfast default` `Switch(config-if)# spanning-tree portfast`	**Configure PortFast**

`Switch# show running-config | begin span`
`Switch# show spanning-tree summary`
`Switch# show running-config interface int`
`Switch# show spanning-tree interface detail`
	Verify configuration of PortFast.

`Switch(config)# spanning-tree portfast bpduguard default` `Switch(config-if)# spanning-tree bpduguard enable`	**Configure BPDU Guard**

Switch(config)# errdisable recovery cause bpduguard	**To manually re-enable an error-disabled BPDU Guard Port**
Switch# show spanning-tree summary	**To display information about the state of spanning tree**

Configure Extended ACLs

Router(config)# access-list *access-list-number* **{deny | permit | remark** *text***}**
protocol source source-wildcard [operator {port}] destination
destination-wildcard [operator {port}] **[established] [log]**

Parameter	**Description**
access-list-number	This is the decimal number of the ACL. Extended ACL number range is 100 to 199 and 2000 to 2699.
deny	This denies access if the condition is matched.
permit	This permits access if the condition is matched.
remark *text*	(Optional) Adds a text entry for documentation purposes. Each remark is limited to 100 characters.
protocol	Name or number of an internet protocol. Common keywords include **ip**, **tcp**, **udp**, and **icmp**. The **ip** keyword matches all IP protocols.
source	This identifies the source network or host address to filter. Use the **any** keyword to specify all networks. Use the **host** ip-address keyword or simply enter an ip-address (without the host keyword) to identify a specific IP address.
source-wildcard	(Optional) A 32-bit wildcard mask that is applied to the source.

destination	This identifies the destination network or host address to filter. Use the **any** keyword to specify all networks. Use the **host** ip-address keyword or ip-address.
destination-wildcard	(Optional) This is a 32-bit wildcard mask that is applied to the destination.
operator	(Optional) This compares source or destination ports. Some operators include **lt** (less than), **gt** (greater than), **eq** (equal), and **neq** (not equal).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	(Optional) For the TCP protocol only. This is a 1st generation firewall feature.
log	*(Optional) This keyword generates and sends an informational message whenever the ACE is matched. This message includes ACL number, matched condition (i.e., permitted or denied), source address, and number of packets.* *This message is generated for the first matched packet.* *This keyword should only be implemented for troubleshooting or security reasons.*

Router(config-std-nacl)# [permit | deny] *host-ip-address* **0.0.0.0**

or

Router(config)# access-list *acl#* **permit | deny] host** *host-ip-address*

Router(config)# access-list *acl#* **permit | deny]** *0.0.0.0 255.255.255.255*

COMMAND PROMPT COMMANDS

C:\>ipconfig

IP address configuration of the computer. Works in the Command Prompt of the computer.

C:\>ipconfig /all

Displays all layer 2 and layer 3 addressing information.

C:\>nslookup

To initiate a DNS request manually.

C:\>ipconfig /displaydns

To display a PC's DNS cache.

C:\>ipconfig /flushdns

To flush a PC's cache.

C:\>ipconfig /release

Terminates an IP address lease before it expires

C:\>ipconfig /renew

Prompts PC to broadcast a DHCPDISCOVER message to get an IP address release.

Wireless Controller Navigation

CONFIGURE Wireless LANs:

1. **Configure VLAN interface**
2. **Configure RADIUS Server**
3. **Configure Wireless LAN**
4. **Configure DHCP Scope**
5. **Configure SNMP Server**

Configure VLAN interface

Log into Wireless Controller
Go to the CONTROLLER tab
Click the Interfaces tab on the left side of the page
Click New
Enter the interface name and VLAN ID
Hit Apply
Select the VLAN which is now included in the interfaces tab
Fill out the DHCP information

Configure RADIUS Server

Go to the SECURITY tab
Click the RADIUS tab on the left side of the page
Click Authentication under the Radius tab
Click New
Fill out the Server information

Configure Wireless LAN

Go to the WLANs Tab
Create NEW
Fill out the information
Hit APPLY
On the edit page of the newly created WLAN, Click Enabled on the GENERAL tab
On the General tab, select the WLAN created in the interfaces section
In the Security section of the WLAN tab, select the security policy you want to use and fill out the permissions as followed
Go to the AAA Servers section in the WLAN page, and select the IP address of the RADIUS server in the Server 1 tab
Open the Advanced section of the WLAN tab and select both Flexconnect sessions

Configure DHCP Scope

Go to the CONTROLLER Tab
Click Internal DHCP Servers tab in the left hand side
Click DHCP Scope under the Internal DHCP Servers tab
Click New
Name the Scope
Click Apply
Click the Scope again
Fill in the Parameters

Configure SNMP Server

Hit the MANAGEMENT tab
Go to the SNMP right side tab
Click Trap Receivers
Add the information-- the IP address being that of the Radius Server, Click Apply
Then Save the Configuration.

FTP Server Setup Windows 2012-16

Go to Server Manager:

Add Server Role ⟶ IIS Web Server

Get FTP Server

Go to Administrative Tools:

Active Directory Users and Computers

Look at what computer names are

If you can, set up users.

Use : or ; to add more users (AD).

Go to IIS Manager

Create site (FTP)

Name site

Choose Physical Path

Go to Server Manager

Configuration

Firewall

Allow inbound FTP

FTP Server DNS == DNS Manager

New Zone

Primary zone

Zone name: company.com

New host: ftp.company.com

Use IP address of server

Fix Corrupted SD Card

1. diskpart
2. list disk
3. select disk
4. clean
5. create partition primary
6. format fs=fat32 ***or*** fs=ntfs

HSRP Configuration

`Router(config)# interface g0/1`	Interface used to configure HSRP virtual router.
`Router(config-if)# standby version 2`	Specify the HSRP protocol version number. The most recent version is version 2. * Standby version 1 only supports IPv4 addressing.
`Router(config-if)# standby group# ip virtual-ip-address`	The group number ranges from 0-4095 for version 2. For version 1, it ranges from 0-255. The group number should be consistent between the routers in this group.
`Router(config-if)# standby group# priority 150`	This designates the active router for the HSRP group. Specify the priority for the router interface. This number must be consistent between the routers in the group. The default value is 100. The higher value is the active router. The priority value ranges from 0-255.
`Router(config-if)# standby group# preempt`	By default, after a router becomes the active router, it will remain the active router even if another router comes online with a higher HSRP priority. Preemption is the ability of an HSRP router to trigger the re-election process.

`Router# show standby`	**Verify the values for HSRP role, group, virtual IP address of the gateway, preemption, and priority. HSRP also identifies the active and standby router IP addresses for the group.**
`Router# show standby brief`	**Used to view an HSRP status summary.**