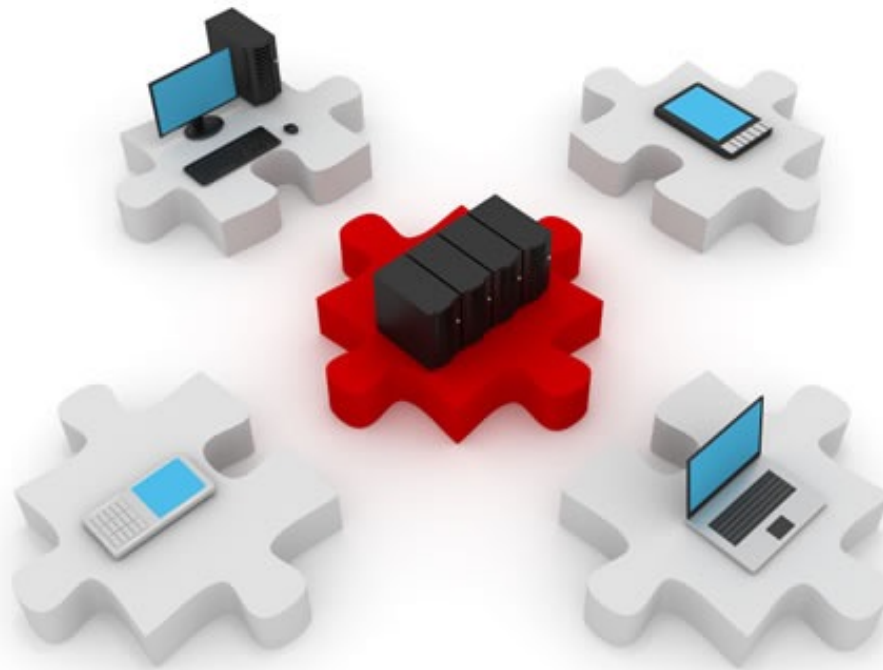


Endpoint Security

November 25, 2014

What this lecture is about:

- ▶ Securing hosts inside a network
 - ▶ Hosts & servers
- ▶ Layer 2 attacks
 - ▶ On the network
 - ▶ On its devices



Endpoint Security

A secure network is only as strong as its weakest link

Securing “inside” and “outside”

- ▶ You know about securing the perimeter of a network against outside threats:
 - ▶ ACLs
 - ▶ CBAC
 - ▶ ZPF
 - ▶ TCP intercept
 - ▶ IDS/IPS: NIPS

- ▶ Internal threats are there, too
 - ▶ How well can you control who accesses your internal network?

Big-picture strategy for endpoint security

- ▶ **NAC (Network Access Control)**
 - ▶ A solution that requires every endpoint to comply with company policies
 - ▶ Non-compliant endpoints are denied access
- ▶ **Endpoint protection using HIPS**
 - ▶ Implemented using CSA (Cisco Security Agent)
 - ▶ Complemented by IronPort Perimeter Security Appliances
- ▶ **Network infection containment**
 - ▶ Before stopping an attack, containment is required
 - ▶ Must be an automated process
 - ▶ Implemented as a NAC or IPS service

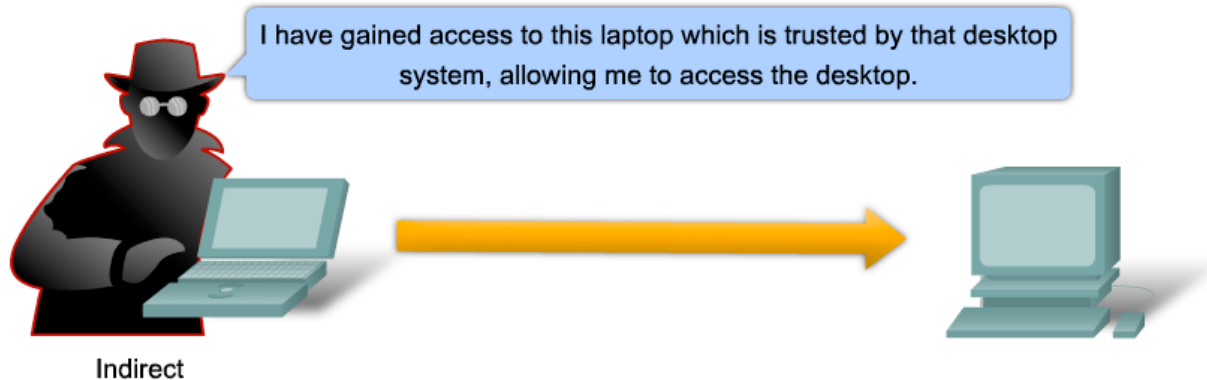
Operating system security

- ▶ Protecting an endpoint is protecting its services and applications
- ▶ Ultimately, an operating system has full access over a host
- ▶ Protecting the operating system becomes a priority

- ▶ OS'es have basic security features like:
 - ▶ Processes - independent address spaces
 - ▶ Privileges - execution must be made from a user account with sufficient privileges
 - ▶ Least privilege concept
 - ▶ Applies to processes and users, as well

Gaining access: directly and indirectly

- ▶ Security must be viewed from a network perspective, too
- ▶ Hosts have privileges and can be “trusted”, too



Strategy: NAC - Network Access Control

- ▶ NAC provides several features in order to enforce a network security policy:
 - ▶ Authentication and authorization
 - ▶ Evaluating a foreign device against the policies of the network
 - ▶ Quarantining of non-compliant systems
 - ▶ Remediation of non-compliant systems

Strategy: NAC - Network Access Control

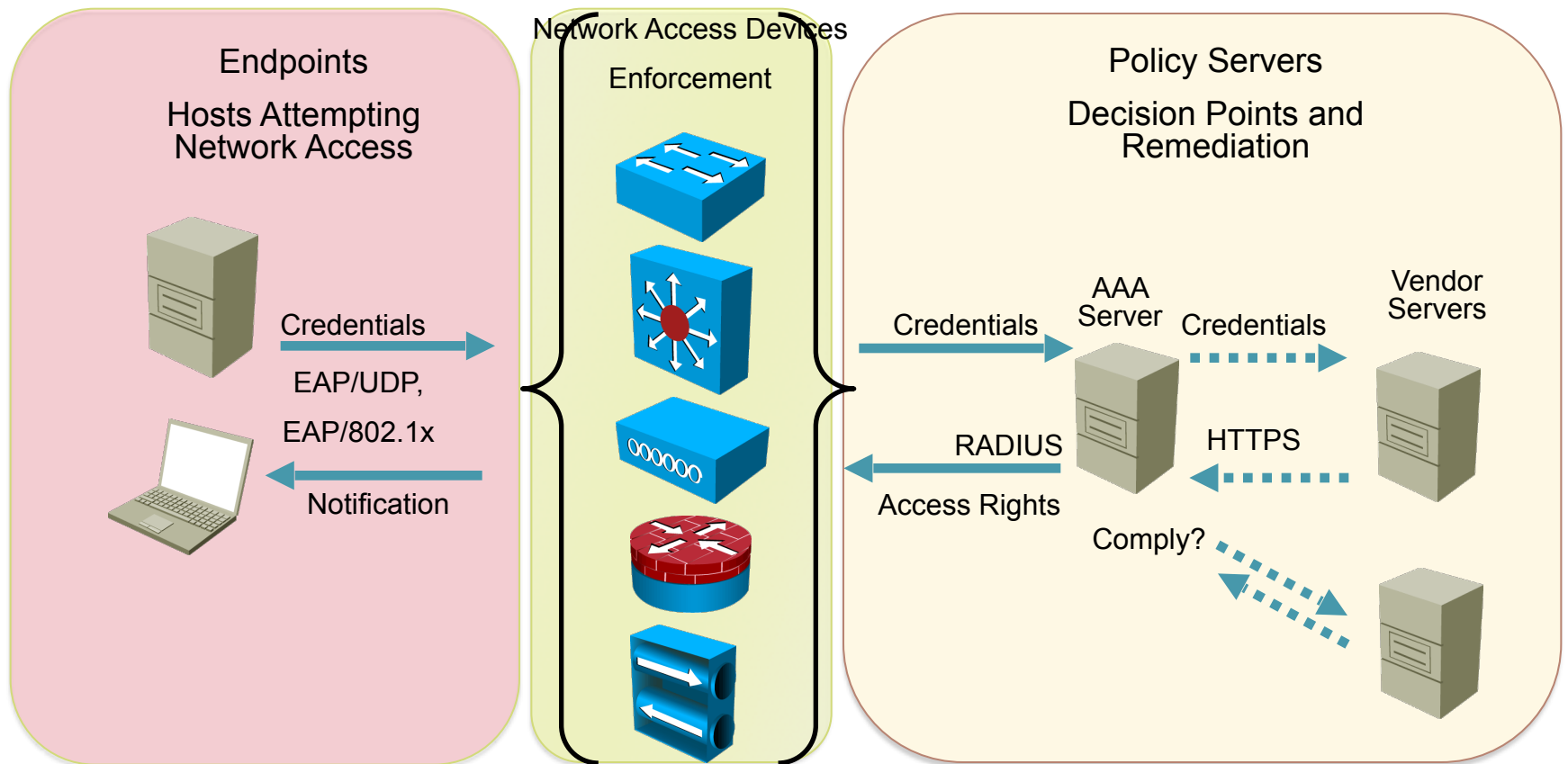
- ▶ **Purpose:** ensure that only authenticated and policy-compliant hosts are given access to the network.
 - ▶ Protects against foreign devices such as laptops, PDAs, smartphones
 - ▶ Not only “guest” devices, but also devices from your company that have gone off-site and might have become infected
 - ▶ These devices can infect a network from inside the perimeter
 - ▶ Network Intrusion Prevention System (NIPS) doesn't help here

Cisco NAC

- ▶ Cisco implements NAC in two logical models:
 - ▶ NAC framework
 - ▶ Distributed solution, for large networks, many network connections and many endpoints.
 - ▶ Suited for remote access solutions, too
 - ▶ NAC appliance
 - ▶ Simplified solution, self-contained
 - ▶ Anti-virus and vulnerability updates
 - ▶ Can be used on any Cisco platform
 - ▶ Turnkey solution

Cisco NAC framework - distributed solution

- ▶ Several devices enforcing different security policies



Cisco NAC appliance

▶ Cisco NAC components:

- ▶ NAS (NAC Appliance Server)
 - ▶ Stores network security policies
 - ▶ Performs device-compliance checks
- ▶ NAM (NAC Appliance Manager)
 - ▶ Administration interface used by support personnel
 - ▶ Allows configuration of NAS
- ▶ NAA (NAC Appliance Agent)
 - ▶ Software client, runs on endpoint machines
 - ▶ Read-only rights over the operating system
 - ▶ Performs constant deep inspection and analysis



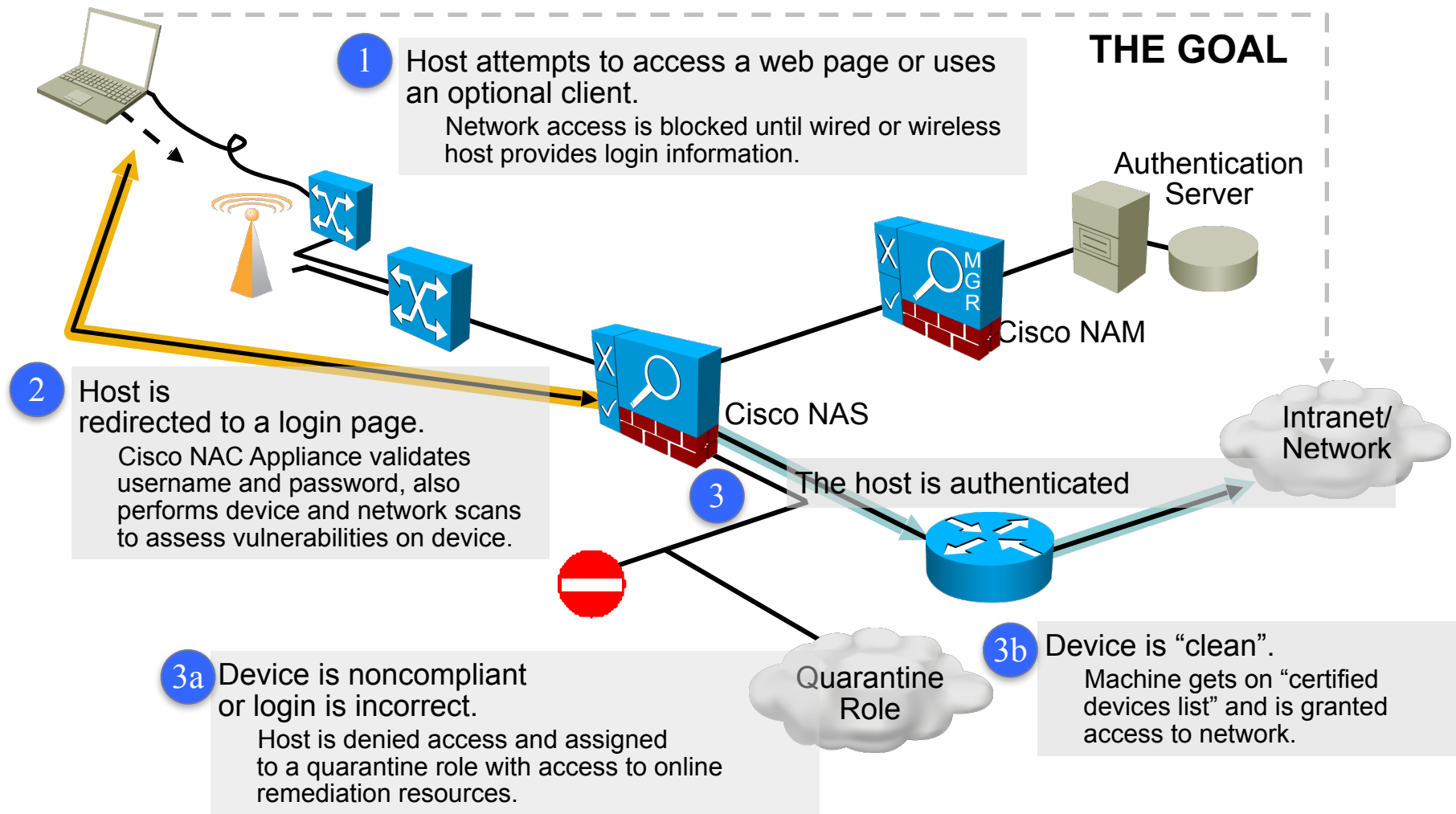
NAS



NAM



How does Cisco NAC appliance work?

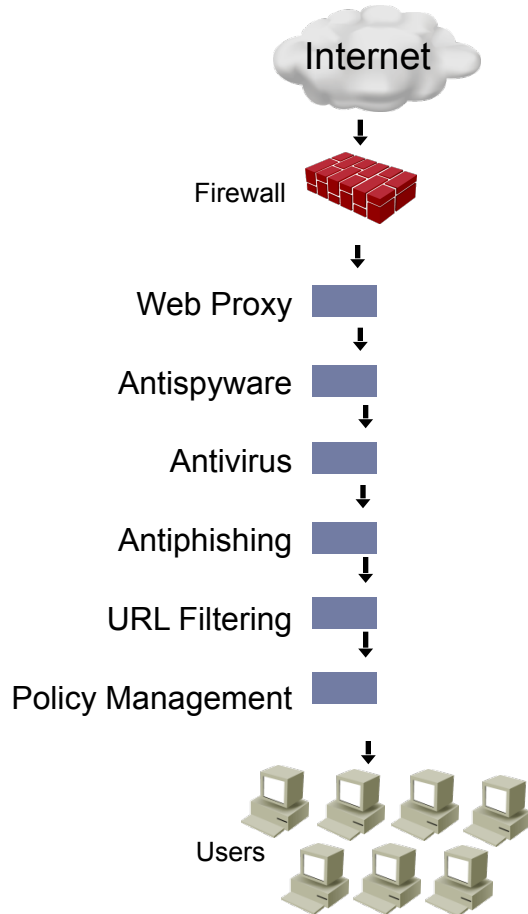


Strategy: IronPort

- ▶ Acquired by Cisco in 2007
- ▶ Leading provider of anti-spam, anti-virus, anti-spyware appliances
- ▶ C-series: *e-mail security*, virus and spam control
- ▶ S-series: *web security*, anti-spyware, anti-malware
- ▶ M-series: e-mail, web and *organization-specific policies*

Strategy: IronPort combined functionality

Before IronPort

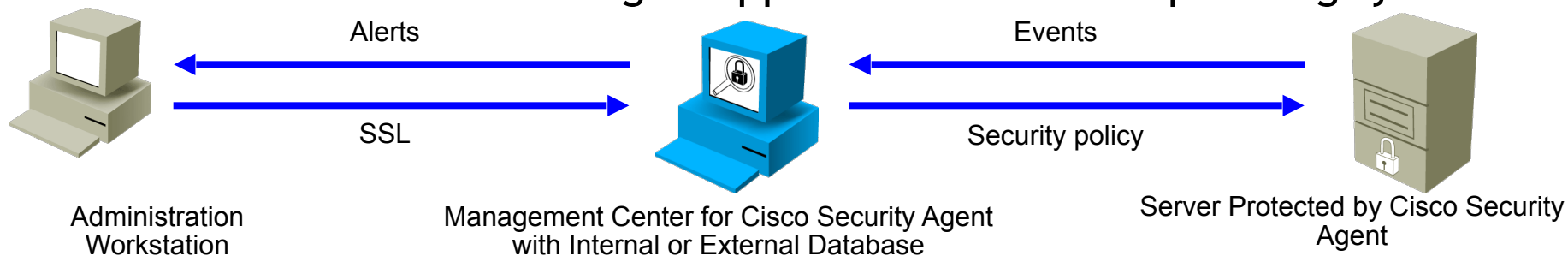


After IronPort



HIPS solution: Cisco Security Agent (CSA)

- ▶ CSA - HIPS solution providing endpoint security
 - ▶ Installed on desktop and server systems
- ▶ Components:
 - ▶ Management center for CSA
 - ▶ Administrative interface, maintains logs for alerts sent by clients
 - ▶ Cisco Security Agent
 - ▶ Installed on host system
 - ▶ Continuous monitoring of applications and the operating system



CSA functionality

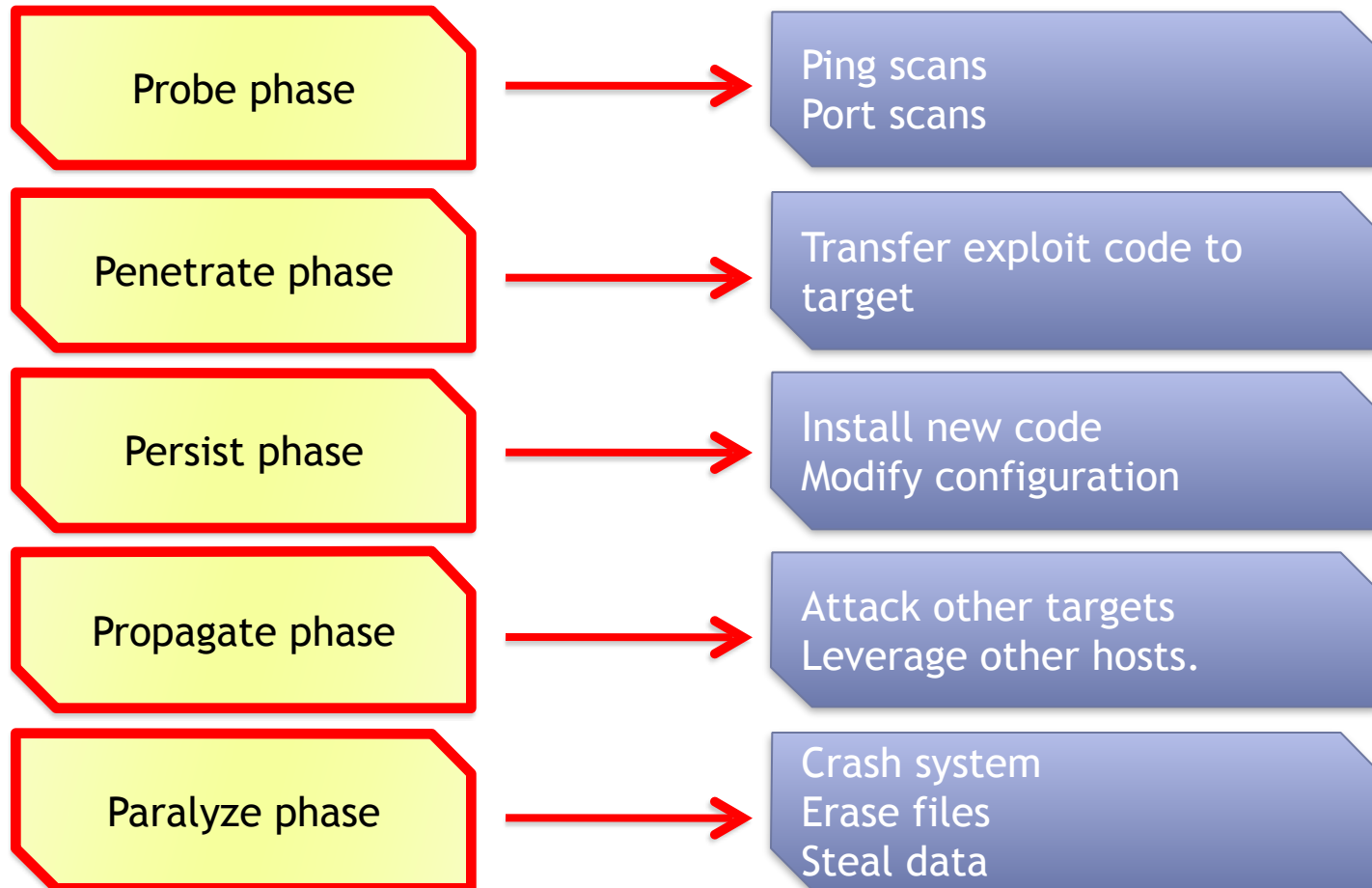
- ▶ When applications require system resources, they make a **system call** to the **kernel**
- ▶ CSA intercepts system calls and compares them to the system policy
- ▶ If the request violates the policy:
 - ▶ CSA blocks it
 - ▶ Sends an appropriate error message to the application
 - ▶ Sends an alert to the Management Center

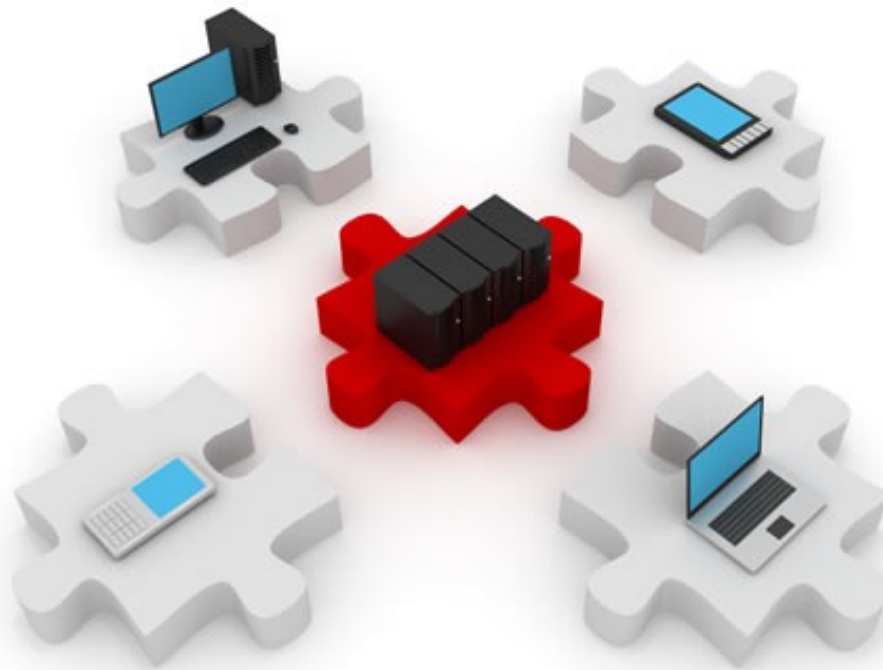
CSA intercept feature

- ▶ CSA intercepts operating system calls using four specialized interceptors:
 - ▶ *File system interceptor*: read/write requests to all file systems
 - ▶ *Network interceptor*: inspects network traffic; can force limitations to protect from DoS attacks
 - ▶ *Configuration interceptor*: read/write requests to the operating system's configuration (like the registry)
 - ▶ *Execution space interceptor*: protects the dynamic runtime environment
 - ▶ Blocks requests to memory that is not owned by an application

CSA security features

- ▶ CSA enables protection against all phases of an attack:





Layer 2 security

The lowest link that can prove to be the weakest.

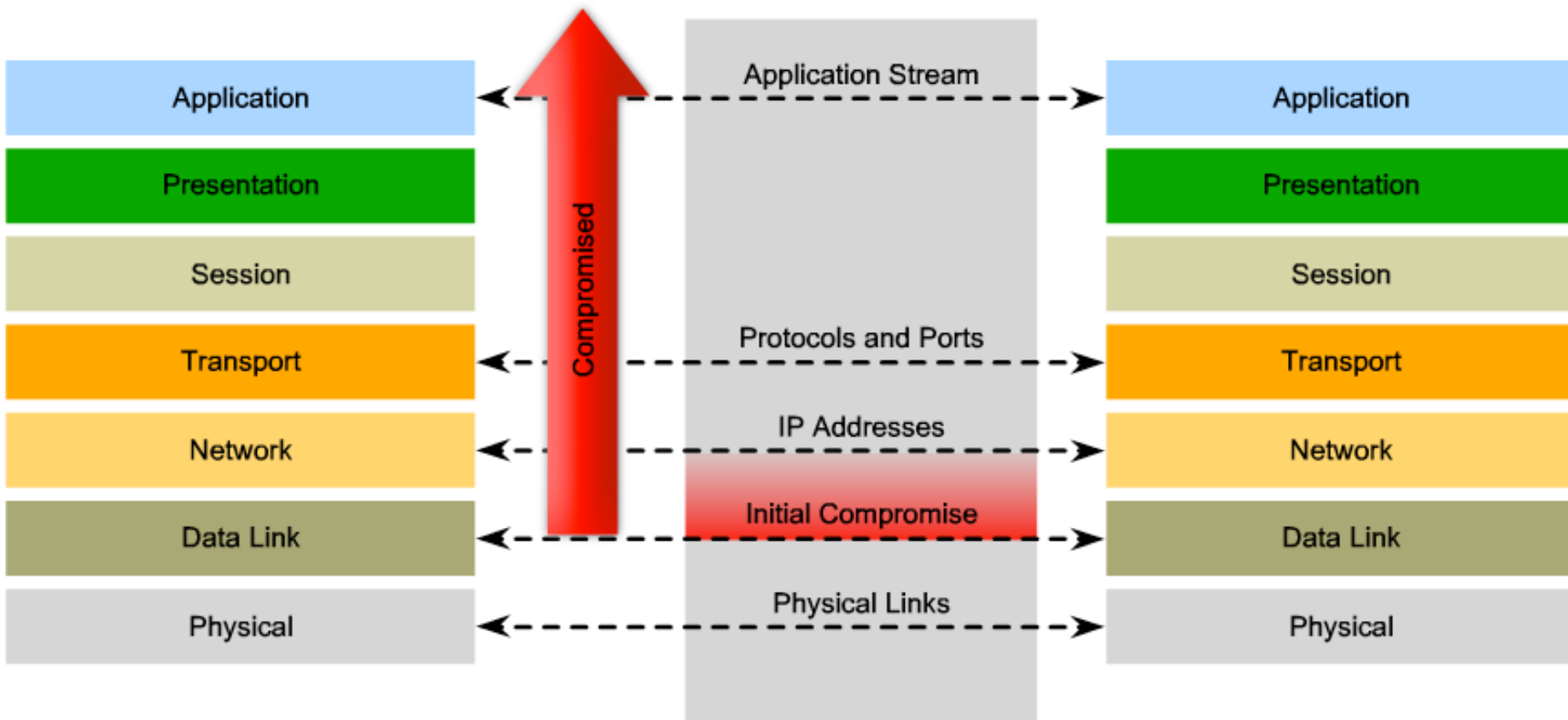
This section will cover:

- ▶ Layer 2 attack methods:
 - ▶ MAC address spoofing
 - ▶ STP manipulation
 - ▶ MAC table overflow
 - ▶ LAN storms
 - ▶ VLAN attacks

- ▶ Also, a little brief recap of the LAN technologies

Compromising layer 2 compromises all layers

- ▶ If the data link layer is hacked, the other layers will not be aware



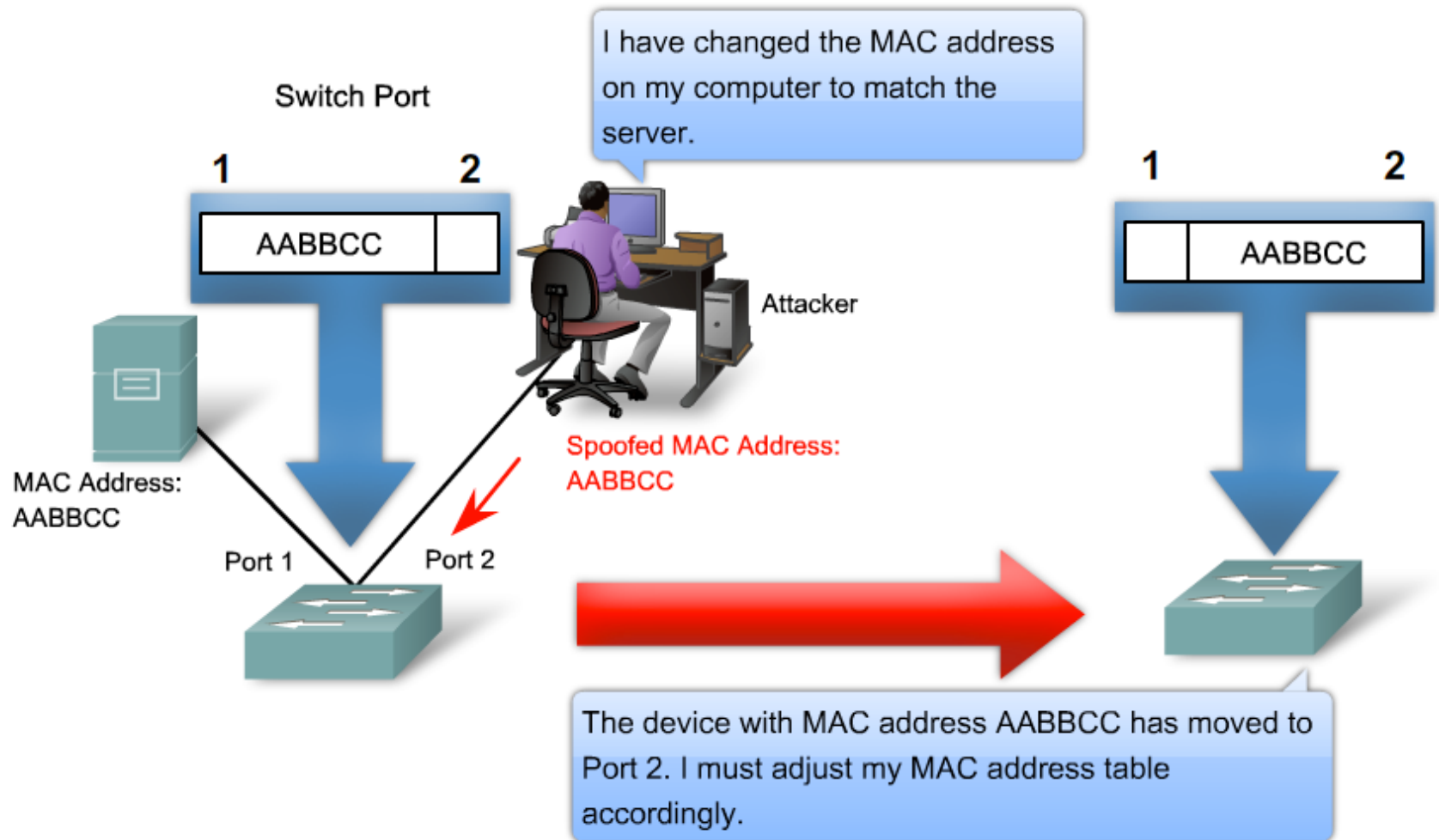
Switched networks

- ▶ A hub is an intermediary device that forwards data to all ports except the one it was received
- ▶ Switches optimize this behaviour:
 - ▶ They forward data on specific ports, based on destination MAC addresses
 - ▶ So, switches must learn on which port is each MAC address located. The CAM memory stores these mappings
 - ▶ How does a switch learn about MAC addresses?
- ▶ Can a switch learn a MAC address on more than one port?
- ▶ Can a switch learn more than one MAC address per port?
 - ▶ Give an example. When?

MAC spoofing

- ▶ The way switches learn MAC addresses is a vulnerability by itself
- ▶ Hosts can use another MAC address to impersonate another device and “fool” the switch
 - ▶ The switch receives frames with the spoofed MAC address
 - ▶ It look at the source address and learns it on a different port.
 - ▶ The switch updates its CAM table and maps the old MAC address on the new port
 - ▶ Frames destined to the target host are now sent to the attacking host

Example: MAC address spoofing



MAC address table overflow attack

- ▶ A switch stores MAC-port mappings in its CAM memory
 - ▶ Which, of course, is limited...
- ▶ Flooding a switch with many fake (spoofed) source MAC addresses will fill up this memory
 - ▶ Having its memory full, the switch cannot learn new MAC addresses
 - ▶ What does a switch do when it does not have the destination MAC address in its memory? (what would YOU do?)
- ▶ The switch will start acting like a hub
- ▶ Any attacker will be able to sniff traffic between any two hosts in the network

STP quick recap

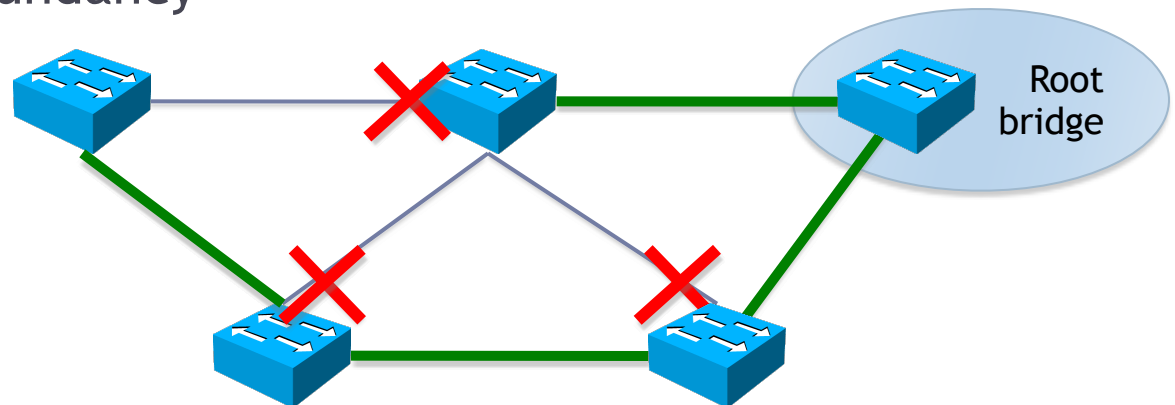
- ▶ STP = Spanning Tree Protocol
- ▶ A Protocol that Spans Trees over your local network 😊

- ▶ The MAC address learning method used by switches does not work if there is a loop in the network
- ▶ Also, loops in a LAN can cause:
 - ▶ Data cycling indefinitely in the loop
 - ▶ MAC address table inconsistency

- ▶ STP creates a loop-free topology (a tree) covering all your switches
- ▶ Traffic will flow only on the tree's links

STP facts

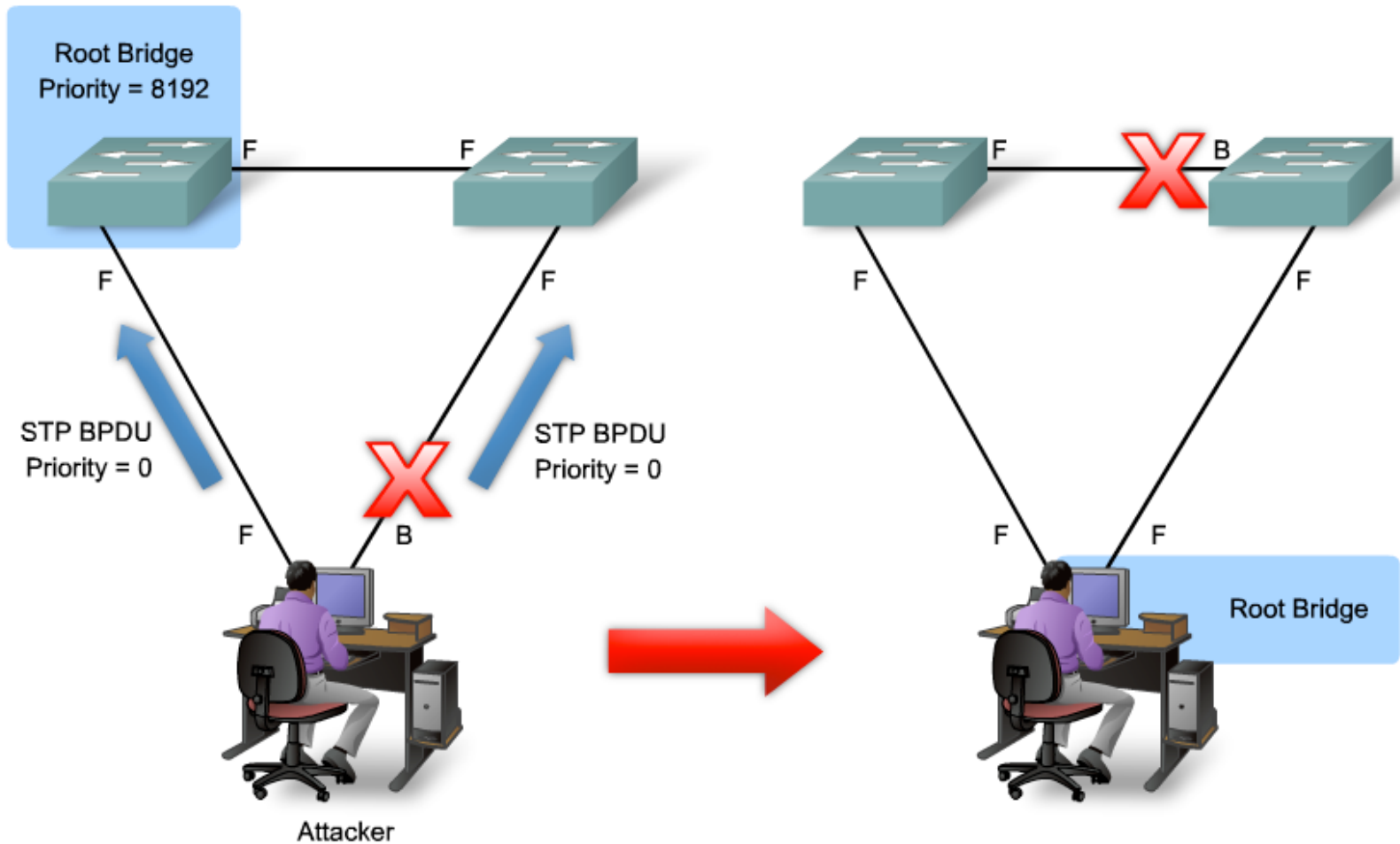
- ▶ Switches in STP are called “bridges”
- ▶ A **root bridge** is elected based on:
 - ▶ Lowest configured priority
 - ▶ If the above are equal, the lowest MAC address is the tie breaker
(now, if THOSE are equal you’ve got bigger problems...)
- ▶ STP shuts down switch ports in order to create a loop-free path
- ▶ In case of failures, closed ports will be opened again
 - ▶ Thus, ensuring redundancy



STP manipulation attacks

- ▶ The election process for the root bridge is carried out using BPDUs
 - ▶ A BPDU (Bridge Protocol Data Unit) is a small piece of data exchanged between adjacent switches every 2 seconds
 - ▶ It contains relevant data for STP election and stability
- ▶ Sending false BPDUs can change the logical topology of the network
 - ▶ Attackers can send false BPDUs to make themselves the root bridges and to be able to receive all traffic traversing the network

STP attack: sending the lowest priority: 0



LAN storm attack

- ▶ Broadcasts are vital for a network to function properly.
 - ▶ Example protocols: ARP, DHCP
- ▶ But flooding a network with broadcast traffic degrades network performance
- ▶ **Broadcast storm**: flooding the network with excessive broadcast traffic
 - ▶ Why is this possible?
 - ▶ Because switches forward broadcasts out on all their ports

Mitigating LAN storms

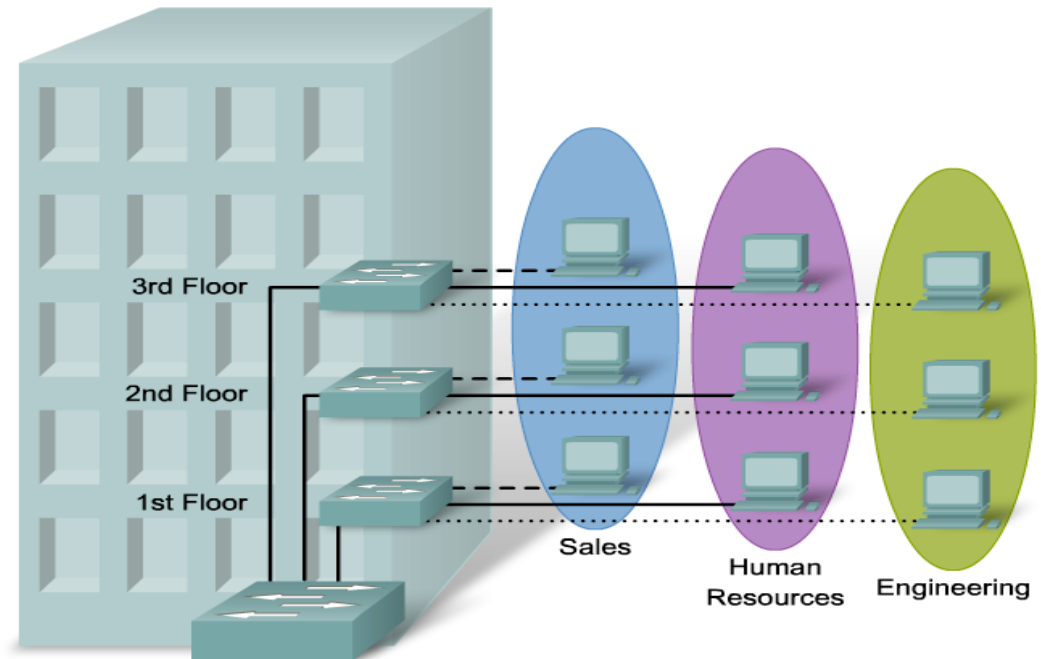
- ▶ Broadcasts cannot be eliminated from the network
- ▶ Solution: **Storm control** (traffic suppression)
 - ▶ Monitors unicast, multicast and broadcast traffic on an interface
 - ▶ Compares the amount of traffic to a predefined threshold
 - ▶ If the number of incoming packets is too high, traffic is blocked
 - ▶ Storm control unblocks traffic after a period of time

VLANs and VLAN attacks

- ▶ A **VLAN** (Virtual LAN) is a logical broadcast domain within a switched network
 - ▶ Multiple VLANs appear as different subnets
 - ▶ Allow segmentation of the LAN without using routers
 - ▶ Hosts cannot communicate between VLANs without a routing-capable device (router, layer 3 switch)
- ▶ VLANs are a simple way to securely isolate groups of hosts inside a LAN
- ▶ Attempting to gain access to another VLAN is a type of a VLAN attack

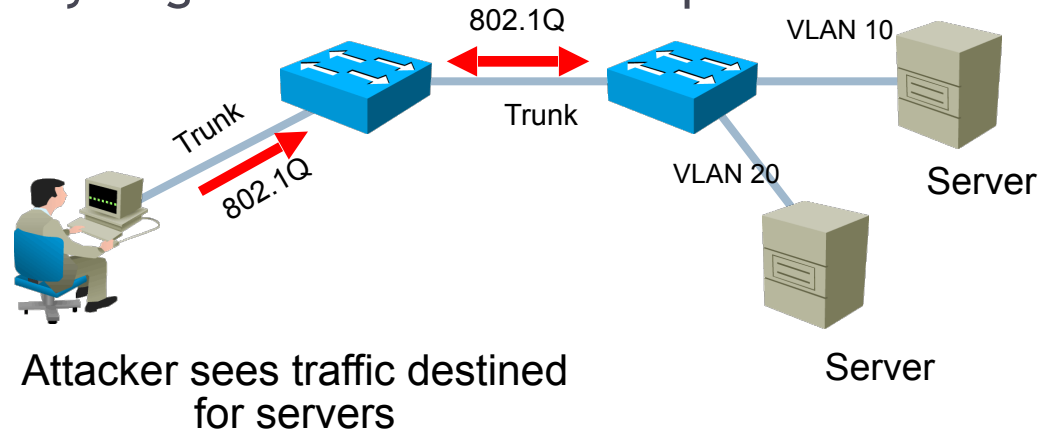
VLAN extensibility

- ▶ VLANs are not restricted to a single geographical location
 - ▶ Inter-switch links that carry more than one VLANs are called **trunk links**
 - ▶ Common trunking protocol: IEEE 802.1q (“dot1q”)
 - ▶ Ports that connect hosts to a single VLAN are called **access ports**



VLAN hopping attacks

- ▶ End-users (their hosts) are always members of a single VLAN
- ▶ Accessing another VLAN, other than the one assigned to your switch port, is called **VLAN hopping**
- ▶ Method: establish your own trunk link with the switch
 - ▶ The trunk link can transport any VLAN
 - ▶ DTP (Dynamic Trunking Protocol) is active by default and will automatically negotiate a trunk when possible



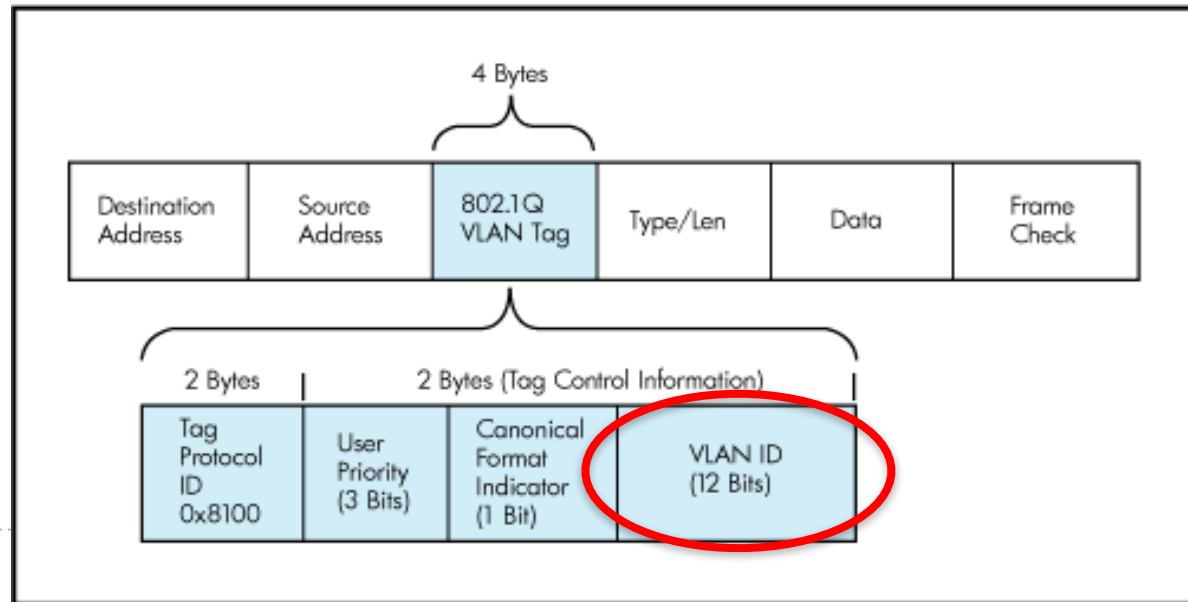
VLAN hopping mitigation

- ▶ **Unwanted trunk links can be created using:**
 - ▶ A host that acts like a switch and sends DTP negotiation frames
 - ▶ A normal switch, owned by the attacker

- ▶ **Solution:**
 - ▶ Disable DTP on ports that do not require trunking
 - ▶ Negotiation of a trunk will not be possible any more
 - ▶ Preferably, manually enable trunking where needed

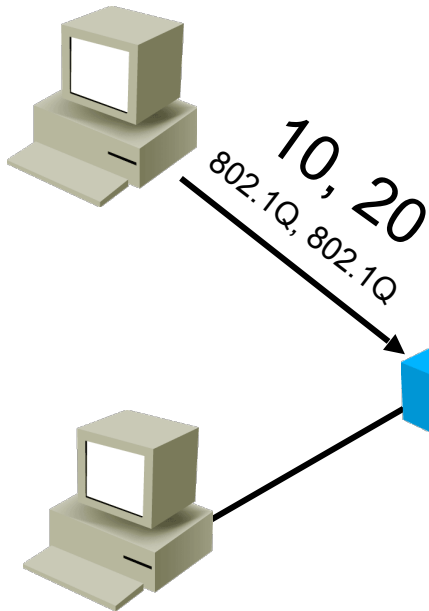
Trunks and native VLANs

- ▶ When passing over a trunk, a frame must retain its VLAN information
 - ▶ So that the end switch will know to which VLAN it belongs to
 - ▶ “Tagging” a frame with its VLAN information is done using the dot1q protocol
 - ▶ If a frame does not have a tag, it is considered to belong to the “native VLAN” of the trunk link

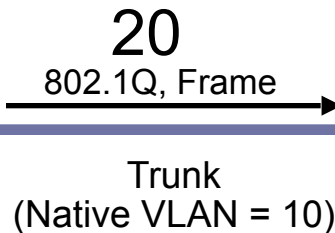


VLAN hopping: double-tagging

1 Attacker is on VLAN 10 but also puts a 20 tag in the packet

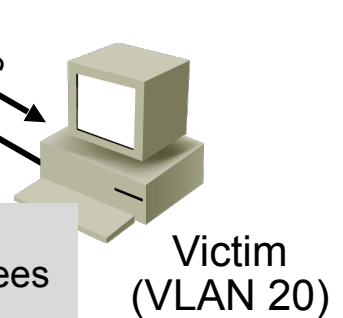


2 The first switch strips off the first tag and does not retag it (native traffic is not retagged). It then forwards the packet to switch 2.

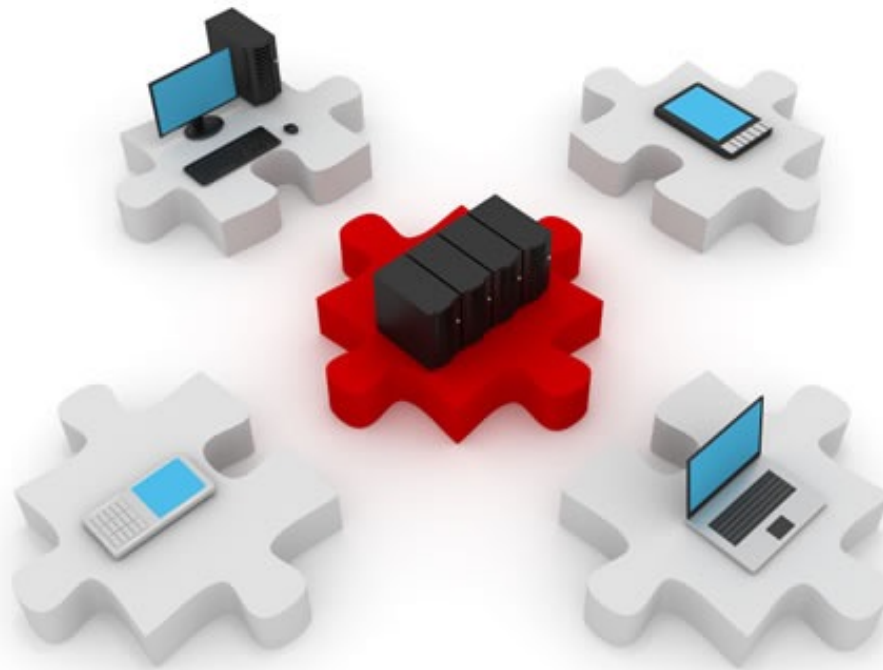


3 The second switch receives the packet on the native VLAN

4 The second switch examines the packet, sees the VLAN 20 tag and forwards it accordingly.



Note: This attack works only if the trunk has the same native VLAN as the attacker's access VLAN.



Configuring Layer 2 Security

Here come the commands...

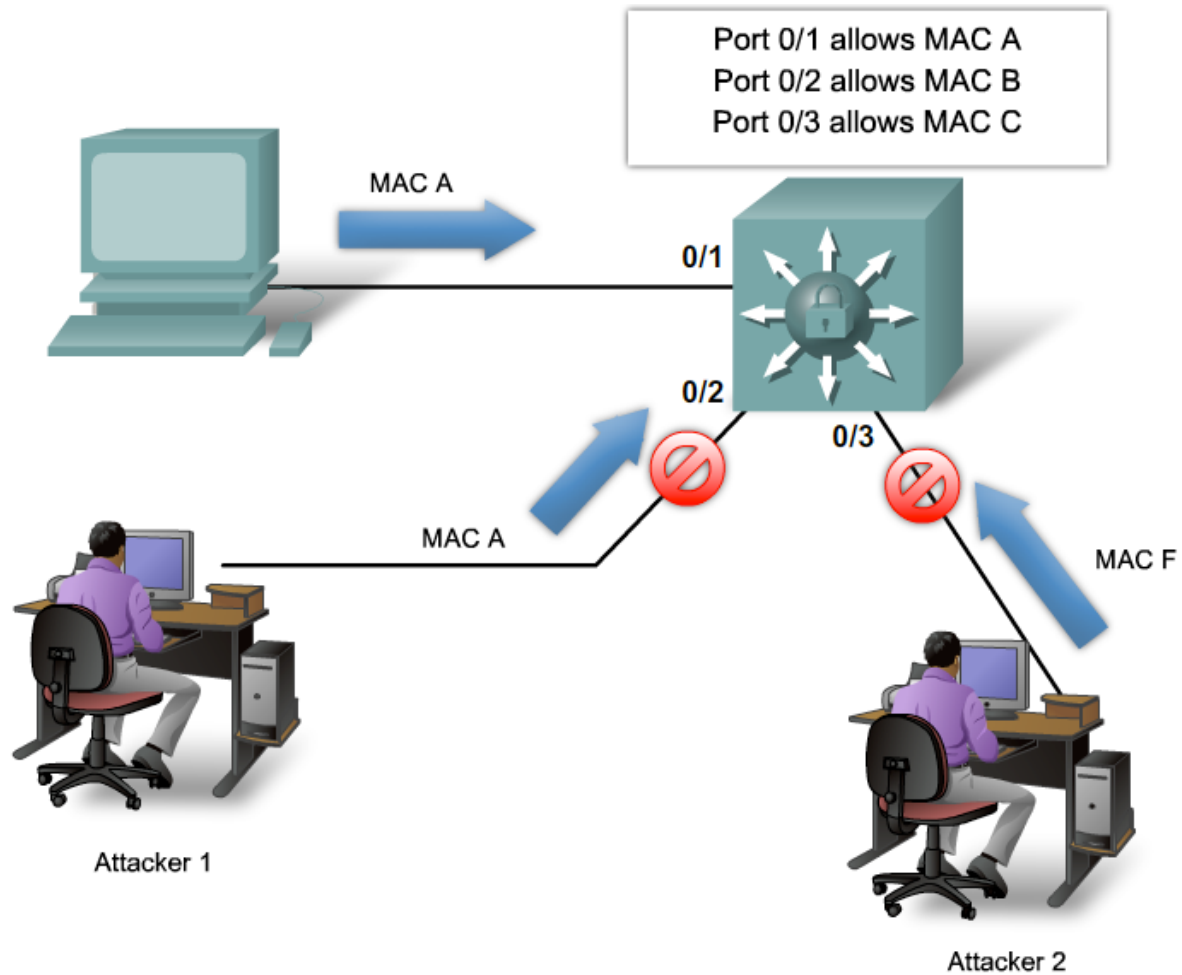
Overview

- ▶ **Overview of this section:**
 - ▶ Configuring port security
 - ▶ Verifying port security
 - ▶ Configuring BPDU Guard and Root Guard
 - ▶ Configuring Storm Control

Port security

- ▶ **Port security is a feature that allows you to:**
 - ▶ Configure the maximum number of MAC addresses a switch can learn on a certain port
 - ▶ Statically configure the allowed MAC addresses
- ▶ **All incoming frames using disallowed MAC addresses are dropped by default**
- ▶ **Protects against:**
 - ▶ Unauthorized expansion of the network
 - ▶ Foreign hosts or switches becoming members of your network

Port security example



Configuring port security

- ▶ Changing the interface mode to access: [access != trunk]

```
S1(config)#interface FastEthernet 0/24
```

```
S1(config-if)#switchport mode access
```

- ▶ The default mode on a switch port (interface) is **dynamic auto**, which will use DTP to try and dynamically negotiate a trunk on the link
- ▶ Port security cannot be enabled on dynamic auto ports

- ▶ Activating port security on the interface:

```
S1(config-if)#switchport port-security
```

- ▶ Set the maximum number of MAC addresses that can be learned on the interface:

```
S1(config-if)#switchport port-security maximum 3
```

Configuring port security MAC addresses

- ▶ Specifying one or more MAC addresses that are associated on the interface:

```
S1 (config-if)#switchport port-security mac-address  
0026.08de.f22e
```

- ▶ We still have only a maximum of 3 MAC addresses on the interface
- ▶ Configuring one MAC address leaves the other 2 to be dynamically learned
- ▶ The first MAC address of a sending host will be recorded

Configuring port security actions

- ▶ The action that is to be taken by the switch port when an invalid source MAC address is detected on the port is called **violation**
- ▶ Setting the violation mode:

```
S1(config-if)#switchport port-security violation ?  
protect    Security violation protect mode  
restrict   Security violation restrict mode  
shutdown   Security violation shutdown mode
```
- ▶ Setting the violation mode is optional
 - ▶ The default is to shut down the port

Port security violation modes

▶ **Protect**

- ▶ Frames with unknown source addresses are dropped.
- ▶ Until you remove some secure MAC addresses or increase the maximum allowed number of addresses to let them pass.
- ▶ No notifications are sent.

▶ **Restrict**

- ▶ Frames with unknown source addresses are dropped.
- ▶ Until you remove some secure MAC addresses or increase the maximum allowed number of addresses to let them pass.
- ▶ Security Violation counter is incremented
- ▶ SNMP traps are sent, syslog messages as well.

▶ **Shutdown**

- ▶ Completely shuts down the interface.
- ▶ The port is set in the error-disabled state.
- ▶ The port has to be manually brought up.
- ▶ Sends the same notifications as in Restrict mode.

Witnessing a “shutdown” violation 😊

- ▶ A port shutting down after receiving one more MAC address than the maximum allowed:

```
2d17h: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in err-disable state
```

```
2d17h: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 0019.e792.8321 on port FastEthernet0/1.
```

```
2d17h: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
```

```
2d17h: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
```

- ▶ Checking port state:

```
SW3(config-if)#do sh ip int brief | incl 0/1
FastEthernet0/1          unassigned      YES unset  down  down
```

- ▶ Checking for the error-disabled state:

```
SW3#show int fa 0/1
FastEthernet0/1 is down, line protocol is down (err-disabled)
  Hardware is Fast Ethernet, address is 001a.6cf8.8c01 (bia 001a.6cf8.8c01)
```

Recovering from a “shutdown” violation

- ▶ Do not attempt the following:

```
SW3(config)#int FastEthernet0/1
SW3(config-if)#no shutdown
```

- ▶ ...as it will have the following “effect”:

```
SW3(config-if)#do sh ip int brief | inc 0/1
FastEthernet0/1  unassigned  YES unset  down  down
```

- ▶ Err-disabled state is not **really** a “shutdown” mode of the interface. Recover by shutting down the interface and bringing it up again:

```
SW3(config-if)#shutdown
SW3(config-if)#no shutdown
2d17h: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
2d17h: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
```

- ▶ Or, even better, automate this to recover after 60 minutes:

```
SW3(config)#errdisable recovery cause psecure-violation
SW3(config)#errdisable recovery interval 60
```


The “sticky” ones...

- ▶ Dynamically learned MAC addresses are lost after the switch reloads
 - ▶ They will be learned again but this could be a security risk
- ▶ You can make all dynamically learned MAC addresses “sticky”
 - ▶ MAC addresses will still be dynamically learned
 - ▶ But they will be automatically saved in the running config

```
SW3(config-if)#switchport port-security mac-address sticky
```

- ▶ The running-config will automatically include:

```
SW3#sh run | include sticky
```

```
switchport port-security mac-address sticky
```

```
switchport port-security mac-address sticky 0019.e792.8321
```

Aging port security entries

- ▶ Configuring aging for static entries:

```
SW3(config-if)#switchport port-security aging static
```

- ▶ Setting the number of minutes after which the entries will age out:

```
SW3(config-if)#switchport port-security aging time 15
```

- ▶ Setting the type of aging:

```
SW3(config-if)#switchport port-security aging type ?
```

```
absolute Absolute aging (default)
```

```
inactivity Aging based on inactivity time period
```

- ▶ Absolute: entries will age out after 15 minutes
- ▶ Inactivity: entries will age out after 15 minutes of inactivity from the specific MAC address

Verifying port security

```
SW3#show port-security
```

```
Secure Port    MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
              (Count)        (Count)      (Count)
-----
      Fa0/1          3             3             0             Shutdown
      Fa0/22         3             1             0             Protect
```

```
Total Addresses in System (excluding one mac per port) : 2
```

```
Max Addresses limit in System (excluding one mac per port) : 8320
```

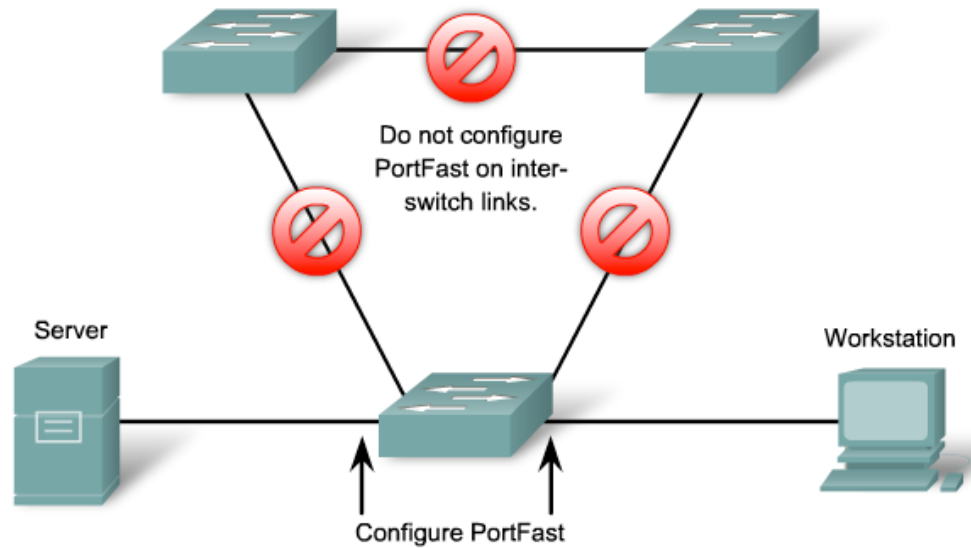
► Showing all learned or configured addresses:

```
SW3#show port-security address
```

```
Secure Mac Address Table
```

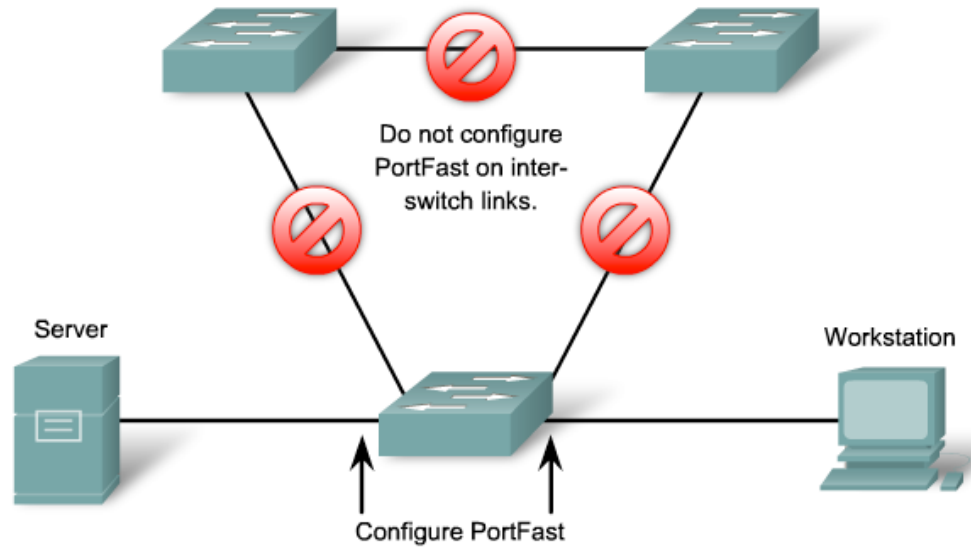
```
-----
Vlan    Mac Address          Type                Ports    Remaining Age
        (mins)
-----
      1    0019.e792.8321      SecureSticky        Fa0/1    -
      1    0025.bcdc.17b6      SecureConfigured    Fa0/1    11
      1    001b.9035.f118      SecureDynamic        Fa0/22    -
-----
```

PortFast



- ▶ STP takes time to converge and goes through several states
- ▶ The PortFast feature can be enabled on access links to avoid STP calculations on them
 - ▶ They are not included in the STP tree, anyway

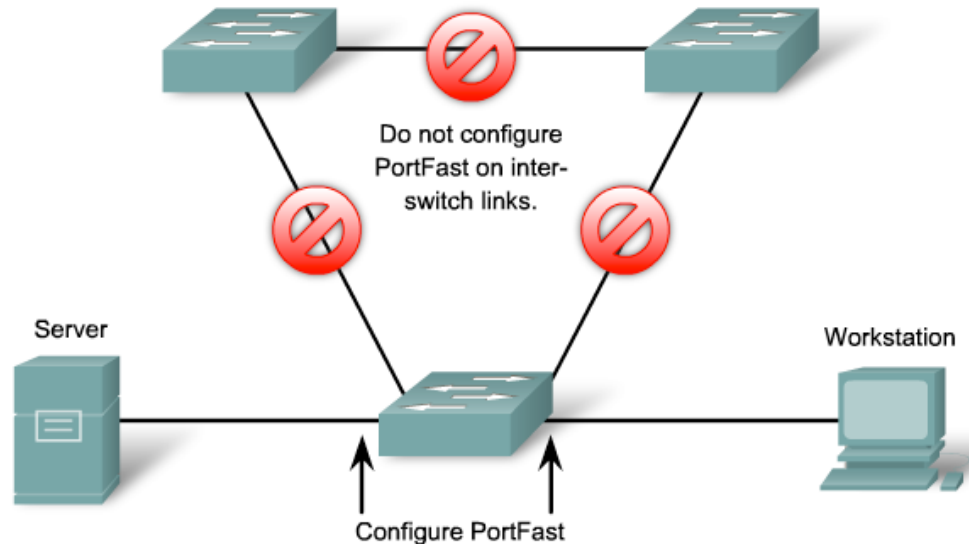
Configuring PortFast globally



- ▶ Configuring PortFast on all non-trunking ports at once:

```
SW3 (config) #spanning-tree portfast default
```

Configuring PortFast at interface level



- ▶ Enabling PortFast for a specific interface:

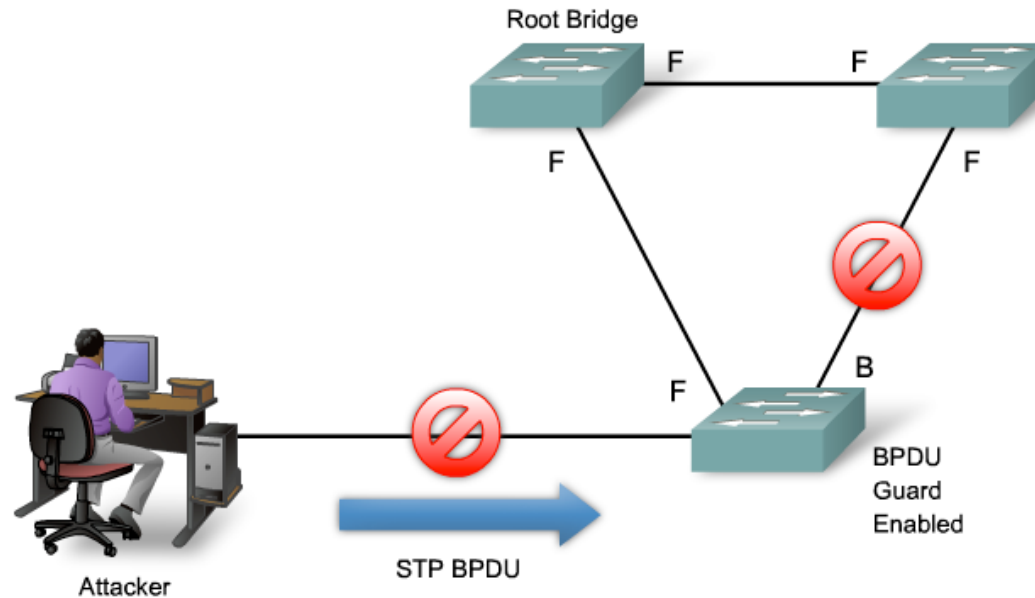
```
SW3 (config-if) #spanning-tree portfast
```

%Portfast has been configured on FastEthernet0/1 but will only have effect when the interface is in a non-trunking mode.

- ▶ Proof that PortFast cannot work on trunk links

Configuring BPDU guard

- ▶ BPDU guard protects the network by blocking BPDUs on ports where they should not be received
 - ▶ This way the network topology remains predictable
 - ▶ Intruders cannot alter the root bridge of the STP tree
 - ▶ Access ports should have BPDU guard enabled



Configuring BPDU guard

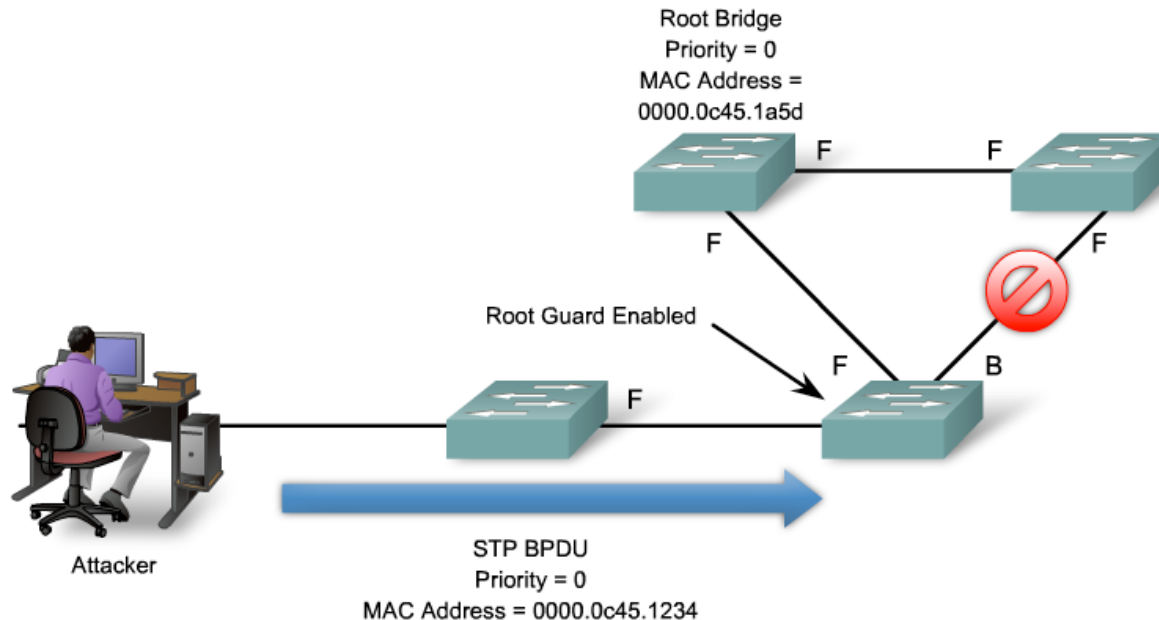
- ▶ BPDU guard is a PortFast feature
 - ▶ By default, BPDU guard will shut down the port if a BPDU is received
- ▶ Enabling BPDU guard globally on the switch:

```
SW3(config)#spanning-tree portfast bpduguard default
```

- ▶ Verifying:

```
SW3#show spanning-tree summary
Switch is in pvst mode
Root bridge for: VLAN0001
Extended system ID          is enabled
Portfast Default            is enabled
PortFast BPDU Guard Default is enabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
[...output ommited...]
```


Root guard



- ▶ An attacker can send spoofed BPDUs in an attempt to become the root
- ▶ The device connected to the switch port can participate in STP as long as it does not try to become the root
- ▶ Root guard puts the port in the root-inconsistent state
 - ▶ It automatically recovers when the offending BPDUs stop

Configuring Root guard

- ▶ Configuring Root guard on the interface:

```
Switch(config-if)#spanning-tree guard root
00:16:27: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard
enabled on port FastEthernet0/1.
```

- ▶ Root guard should be configured on ports that do not lead to the root switch
- ▶ To view port states use:

```
Switch#show spanning-tree inconsistentports
```
- ▶ BTW: sending BPDUs with a priority of 0 does not guarantee that you will become the root bridge
 - ▶ Some other switches might exist, with 0 priority and a lower MAC address

Configuring storm control

- ▶ Example scenarios for configuring storm control:
 - ▶ Block broadcast packets over 75.55% of the interface's capacity:
`sW(config-if)#storm-control broadcast level 75.55`
 - ▶ You can specify upper and lower threshold levels
 - ▶ Block multicast packets that go over 5Mbps:
`sW(config-if)#storm-control multicast level bps 5000000`
 - ▶ Configure the interface to shut down (err-disabled) when either storm control violation occurs:
`sW(config-if)#storm-control action shutdown`
 - ▶ The default is to filter broadcasts

Verifying storm control

► Show storm control status:

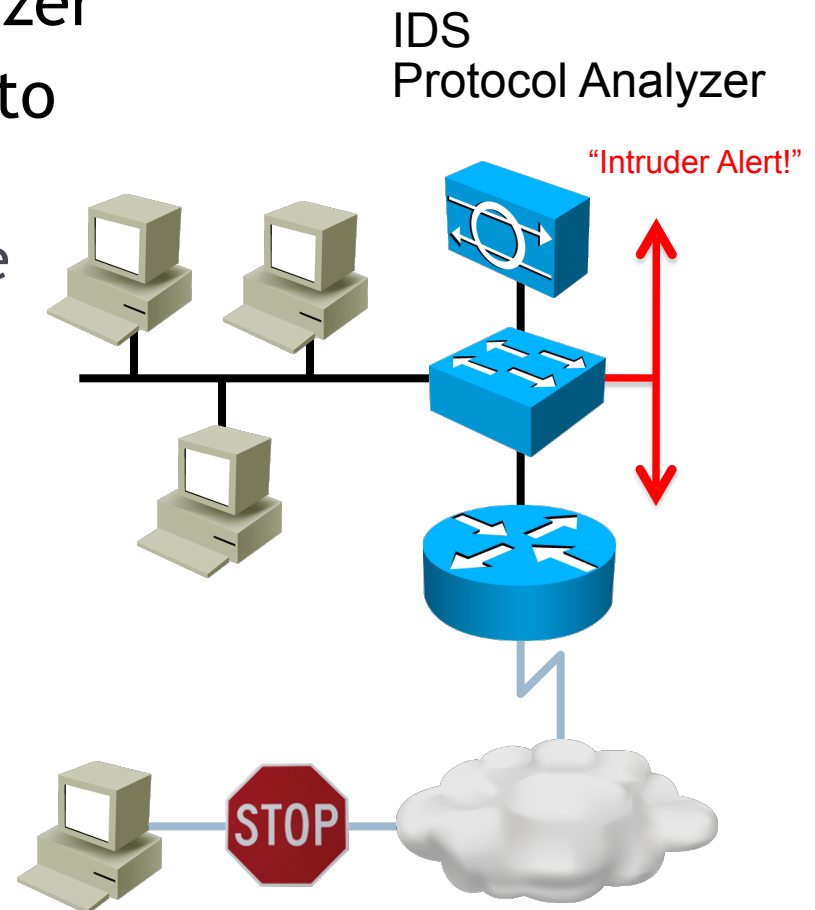
```
SW3# show storm-control
```

Interface	Filter State	Upper	Lower	Current
-----	-----	-----	-----	-----
Gi0/1	Forwarding	20 pps	10 pps	5 pps
Gi0/2	Forwarding	50.00%	40.00%	0.00%

```
<output omitted>
```

Monitoring with SPAN

- ▶ SPAN = Switched Port ANalyzer
- ▶ A SPAN port mirrors traffic to another port
 - ▶ Monitors the entire interface (port) or a single VLAN
 - ▶ Monitors inbound and/or outbound traffic
- ▶ Ideal deployment for IDS systems
- ▶ Does not affect normal switching operation



Configuring SPAN - Example #1

- ▶ The “monitor session” command:

- ▶ Setting the source interface to monitor:

```
Switch(config)# monitor session 1 source interface  
gigabitethernet0/1
```

- ▶ Setting the destination interface, where traffic will be mirrored:

```
Switch(config)# monitor session 1 destination interface  
gigabitethernet0/2 encapsulation replicate
```

- ▶ The “encapsulation” parameter tells the switch to mirror traffic while retaining the same encapsulation method

Configuring SPAN - Example #2

- ▶ The following example monitors only two VLANs:

- ▶ Mirror only received traffic on VLAN 10:

```
Switch(config)# monitor session 1 source vlan 10 rx
```

- ▶ Mirror only sent traffic on VLAN 20:

```
Switch(config)# monitor session 1 source vlan 20 tx
```

- ▶ The destination is still an interface:

```
Switch(config)# monitor session 1 destination interface  
FastEthernet 0/1
```

Viewing SPAN configuration

- ▶ Use the “show monitor” command to view configuration info about all monitor sessions:

```
#show monitor session 1
Session 1
-----
Source Ports:
    RX Only:      None
    TX Only:      None
    Both:         Fa0/2
Destination Ports: Fa0/3
```

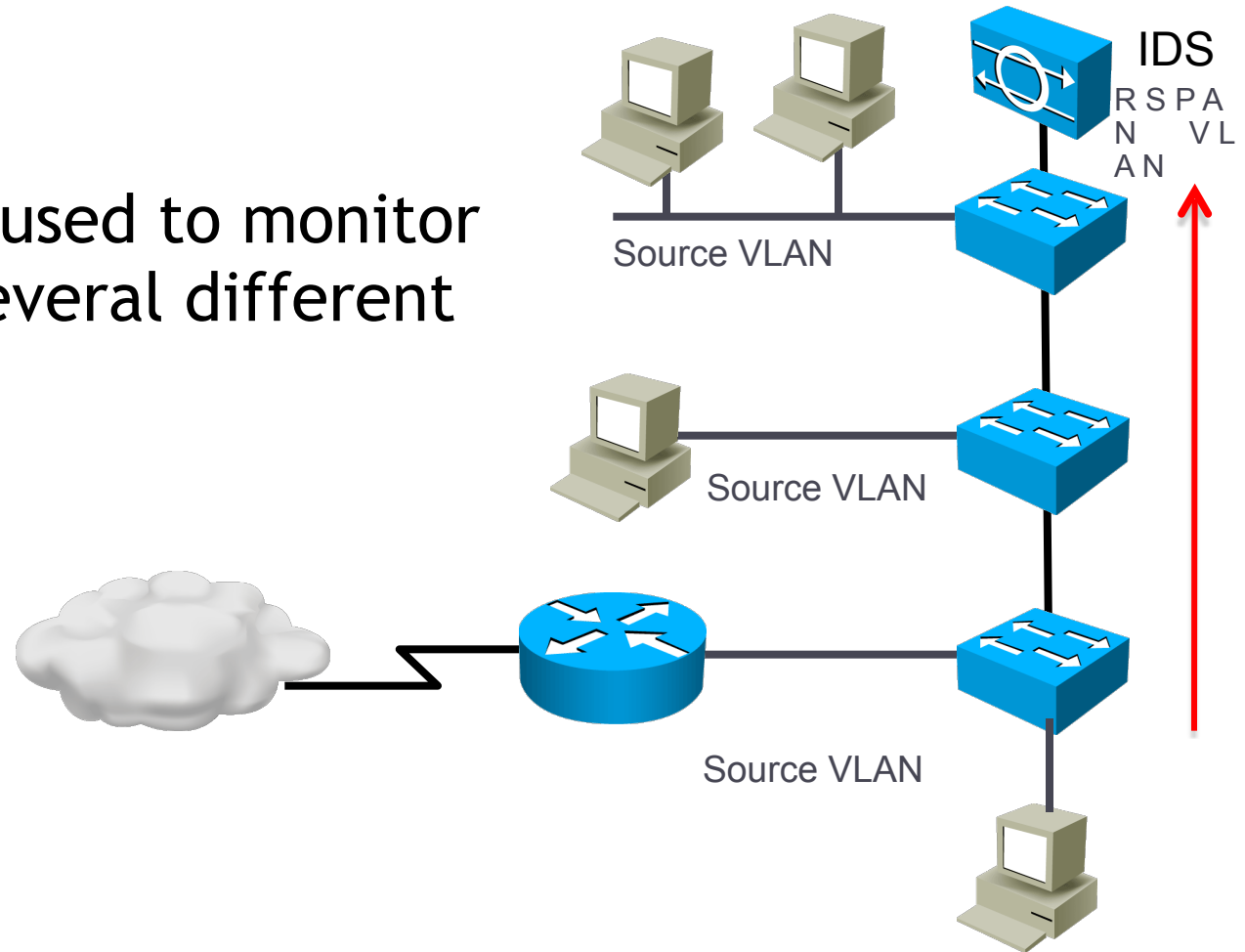

Monitoring with RSPAN

- ▶ RSPAN = Remote SPAN
- ▶ SPAN mirrors traffic between ports on the same switch
- ▶ RSPAN mirrors traffic to a port on a different switch

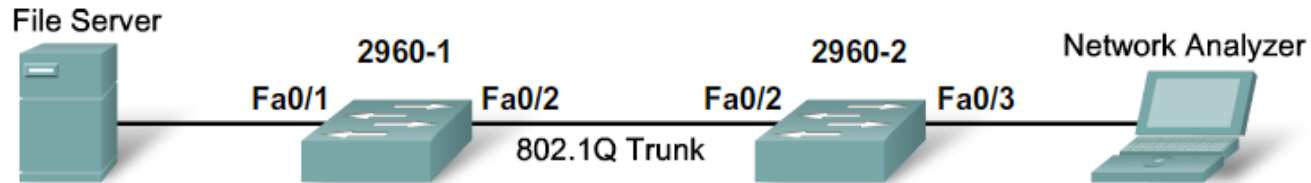
- ▶ This way, traffic from multiple switches can be mirrored to a single destination
 - ▶ Multiple traffic flows can be monitored at the same time
 - ▶ Using the same IDS

RSPAN deployment

- ▶ RSPAN can be used to monitor traffic from several different VLANs



Configuring RSPAN



- ▶ Create the RSPAN VLAN on both switches:

```
2960-1(config)# vlan 100
2960-1(config-vlan)# remote-span
2960-1(config-vlan)# exit
```

- ▶ Configure RSPAN source ports and VLANs:

```
2960-1(config)# monitor session 1 source interface FastEthernet 0/1
2960-1(config)# monitor session 1 destination remote vlan 100
reflecter-port FastEthernet 0/2
2960-1(config)# interface FastEthernet 0/2
2960-1(config-if)# switchport mode trunk
```

- ▶ Configure RSPAN traffic to be forwarded:

```
2960-2(config)# monitor session 2 source remote vlan 100
2960-2(config)# monitor session 2 destination interface FastEthernet
0/3
2960-2(config)# interface FastEthernet 0/2
2960-2(config-if)# switchport mode trunk
```

“Security depends not so much upon how much you have, as upon how much you can do without..”

Joseph Wood Crutch