**Implementation of Web Content Filtering for Campus Network**

To address the issue of students misusing campus resources by accessing irrelevant websites, I have implemented a solution to restrict access to specific categories of web content. Below are the detailed steps and configurations used to achieve this, aligned with the tasks and deliverables outlined in the problem statement.

**Tasks & Deliverables**

**Task 1: Explore Network Security Product**

To restrict access to only allowed categories of web content, I explored several network security products. After thorough research, I selected Cisco's Content Filtering features within the Cisco IOS. This method leverages Access Control Lists (ACLs) and Policy Maps to block access to specific websites by IP address or domain name.

**Network Security Product:** Cisco IOS with Content Filtering

**Capabilities:**

* Blocks access to specific websites by IP or domain name.
* Configurable using ACLs and Policy Maps.
* Integrates with existing network infrastructure.

**Task 2: Update Campus Network Topology with New Component(s)**

I updated the existing campus network topology to include content filtering capabilities. This involved configuring the core router with ACLs, Class Maps, and Policy Maps to enforce web content restrictions.

**Updated Network Topology:**

* Core Router with Content Filtering Configuration
* Additional configuration on the outbound interface to apply the policy map.

**Task 3: Explain Reasoning Behind Choice**

**Reasoning:**

* **Risks Addressed:** The primary risk addressed is the misuse of campus network resources for accessing irrelevant or potentially harmful websites.
* **Advantages:**
  + **Cost-Effective:** Utilizes existing Cisco IOS features without the need for additional hardware.
  + **Scalable:** Easy to update and manage as new websites need to be blocked.
  + **Effective:** Directly blocks traffic at the network level, ensuring compliance with access policies.

**Task 4: Write the Policies**

The following are the policies applied to block access to specific websites, demonstrated with examples for Facebook and YouTube.

**Step-by-Step Configuration**

**Step 1: Create an Access Control List (ACL) to Deny HTTP Traffic to Specific Sites**

I created an extended ACL to block HTTP traffic to the IP addresses of the websites we want to restrict. For demonstration purposes, I have chosen the IP addresses 31.13.71.36 (facebook.com) and 172.217.10.78 (youtube.com).

HQ-Router(config)# ip access-list extended BLOCK-SOCIAL-SITES

HQ-Router(config-ext-nacl)# deny tcp any host 31.13.71.36 eq www

HQ-Router(config-ext-nacl)# deny tcp any host 172.217.10.78 eq www

HQ-Router(config-ext-nacl)# permit ip any any

HQ-Router(config-ext-nacl)# exit

**Step 2: Create a Class Map to Match the ACL**

Next, I created a class map to match traffic defined by our ACL.

HQ-Router(config)# class-map match-all BLOCK-SITES

HQ-Router(config-cmap)# match access-group name BLOCK-SOCIAL-SITES

HQ-Router(config-cmap)# exit

**Step 3: Create a Policy Map to Drop the Traffic**

I then created a policy map to specify the action to take on the matched traffic. In this case, I drop the traffic by using the police command, which rate-limits and drops the traffic

HQ-Router(config)# policy-map WEB-BLOCK

HQ-Router(config-pmap)# class BLOCK-SITES

HQ-Router(config-pmap-c)# police rate 32000 conform-action drop exceed-action drop

HQ-Router(config-pmap-c)# exit

**Step 4: Apply the Policy Map to the Interface**

Finally, I applied the policy map to the outbound traffic on the interface facing the Internet.

HQ-Router(config)# interface GigabitEthernet 0/0

HQ-Router(config-if)# service-policy output WEB-BLOCK

HQ-Router(config-if)# exit

**Explanation of the Configuration**

1. **Create an ACL:**
   * ip access-list extended BLOCK-SOCIAL-SITES creates an extended ACL named BLOCK-SOCIAL-SITES.
   * deny tcp any host 31.13.71.36 eq www blocks HTTP traffic to the IP address of facebook.com.
   * deny tcp any host 172.217.10.78 eq www blocks HTTP traffic to the IP address of youtube.com.
   * permit ip any any allows all other IP traffic to pass.
2. **Create a Class Map:**
   * class-map match-all BLOCK-SITES creates a class map named BLOCK-SITES.
   * match access-group name BLOCK-SOCIAL-SITES matches traffic based on the BLOCK-SOCIAL-SITES ACL.
3. **Create a Policy Map:**
   * policy-map WEB-BLOCK creates a policy map named WEB-BLOCK.
   * class BLOCK-SITES references the BLOCK-SITES class map in the policy map.
   * police rate 32000 conform-action drop exceed-action drop rate-limits the traffic and drops it, effectively blocking it.
4. **Apply the Policy Map:**
   * interface GigabitEthernet 0/0 enters the configuration mode for the specified interface.
   * service-policy output WEB-BLOCK applies the WEB-BLOCK policy map to the outbound traffic on the specified interface.

**Verification**

To ensure the configuration is correct, use the following commands to verify the ACL, class map, and policy map:

To verify the ACL:

HQ-Router# show access-lists BLOCK-SOCIAL-SITES

To verify the class map:

HQ-Router# show class-map

To verify the policy map:

HQ-Router# show policy-map

To verify the service policy applied to the interface:

HQ-Router# show policy-map interface GigabitEthernet 0/0

**Conclusion**

By following the above steps, I have effectively configured the network to block access to specified websites (e.g., Facebook and YouTube) using Cisco IOS. This implementation helps ensure that students use campus resources appropriately and do not access irrelevant web content. This solution leverages existing network security features and enhances the overall security posture of the campus network.