Configure Range Request for Microsoft Update Traffic in SWA

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Range Request

Range Request in the Proxy Environment

Enabling Range Request for Microsoft Updates

Steps to Enable Range Request only for Microsoft Updates

- Step 1. Enable the Range Request
- Step 2. Create a Custom URL Category for Microsoft Updates URLs
- Step 3. (Optional) Create an Identification Profile to exempt Microsoft Updates traffic from Authentication
- Step 4. (Optional) Create a Decryption Policy to Pass through Microsoft Updates Traffic
- Step 5. Create an Access Policy to Allow Range Request for Microsoft Updates Traffic

Modify Access Logs

Verification

Related Content

Introduction

This document describes the steps to allow Microsoft Updates Traffic to use Range Request in Secure Web Appliance (SWA).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- SWA Administration

Cisco recommends that you have these tools installed:

- Physical or Virtual SWA
- Administrative Access to the SWA Graphical User Interface

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Range Request

A range request is a feature of an HTTP protocol that allows a client (such as a web browser or download manager) to request only a specific portion of a file from a server, rather than downloading the entire file at once. This is useful for resuming interrupted downloads, streaming media, or accessing large files efficiently. A client specifies the desired byte range in the Range header of the HTTP request, and the server responds with a 206 Partial Content status code if it supports range requests, delivering only the requested segment of the file.

This mechanism enhances performance and user experience in several scenarios. For example, in video streaming, a range request allows a player to fetch only the segments needed for playback, which reduces the bandwidth usage and improves responsiveness. Similarly, download managers use range requests to split a file into chunks and download them in parallel, speeding up the process. Range requests also play a key role in caching and proxy systems, enabling partial updates and reducing redundant data transfers.

Range Request in the Proxy Environment

In a proxy environment, range requests play a crucial role in optimizing bandwidth usage and improving content delivery efficiency. When range requests are **enabled**, the proxy server can fetch only the required byte segments from the origin server and cache them locally. This allows clients to request partial content such as specific segments of a video or a large file and receive it quickly from the proxy cache, if available. It also enables parallel downloads and resume capabilities, which are beneficial in environments with limited bandwidth or high latency.

However, when range requests are **disabled**, the proxy must fetch the entire file from the origin server even if the client only needs a small portion. This leads to unnecessary data transfer, increased load on both the proxy and origin servers, and slower response times for clients. It also prevents efficient caching strategies, as the proxy cannot store or serve partial content. In streaming scenarios, this can result in buffering delays or a degraded user experience. Disabling range requests can be done for security or policy reasons, but it

often comes at the cost of performance and flexibility.

For example, consider a scenario where 10 users are attempting to download 1 MB each from a 100 MB file through a proxy server.

Range Requests Disabled:

- When range requests are disabled, the proxy cannot fetch just the 1 MB segment each user needs. Instead, it must download the entire 100 MB file from the origin server for each request. This results in:
  - Total traffic from origin to proxy: 10 × 100 MB = 1000 MB (1 GB)
  - Only 10 MB of that data is used by the clients.
  - The remaining 990 MB is wasted, leading to inefficient bandwidth usage and increased load on the proxy and origin servers.

Range Requests Enabled:

- With range requests enabled, the proxy fetches only the requested 1 MB per user:
  - Total traffic from origin to proxy: 10 × 1 MB = 10 MB
  - The proxy can cache these segments and serve them to other users if needed.
  - This results in 90 times less traffic, faster response times, and significantly better resource utilization.

Enabling Range Request for Microsoft Updates

Although range requests enhance performance, it hinders security scanning and policy enforcement within SWA environments, as these systems cannot fully inspect partial content. This article limits range request usage exclusively to Microsoft Update traffic.

⚠ **Caution**: Enabling range request forwarding can interfere with policy-based Application Visibility and Control (AVC) efficiency, and can compromise security.

Steps to Enable Range Request only for Microsoft Updates

Step 1. Enable the Range Request	**Step 1.1.** From the **GUI**, click **Security Services** and choose **Web Proxy**. **Step 1.2.** Click **Edit Settings**. **Step 1.3.** Select the check box **Enable Range Request Forwarding**. **Step 1.4.** Click **Submit**.

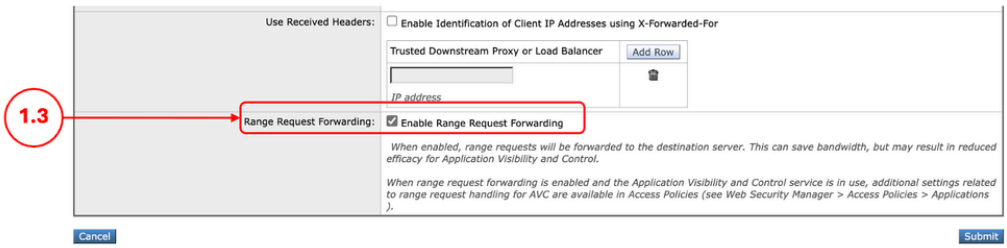


Image - Enable Range Request Forwarding

Step 2. Create a Custom URL Category for Microsoft Updates URLs

Step 2.1. From the **GUI**, choose **Web Security Manager**, and click **Custom and External URL Categories**.
Step 2.2. Click **Add Category** to add a Custom URL Category.
Step 2.3. Assign a unique **Category Name**.
Step 2.4. (Optional) Add **Description**.

Step 2.5. From **List Order**, choose the first category to position on top.

Step 2.6. From the **Category Type** drop-down list, choose **Local Custom Category**.

Step 2.7. Add Microsoft Updates URLs in the **Sites** Section.

Tip: You can check the list of Microsoft updates from this link: Step 2 - Configure WSUS | Microsoft Learn

Caution: Do not Copy/Paste the URLs as they are listed in the Microsoft Documents; format them properly as SWA format. For more information, please visit: Configure Custom URL Categories in Secure Web Appliance - Cisco

This is an example:

```
http://windowsupdate.microsoft.com     ===>  windowsupdate.microsoft.com
http://*.windowsupdate.microsoft.com   ===>  .windowsupdate.microsoft.com
```

Step 2.8. Click **Submit**.

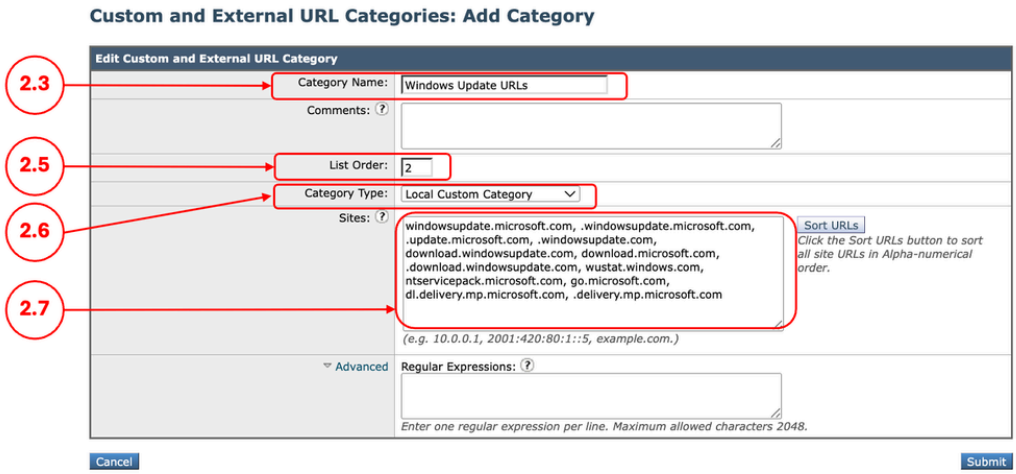


Image - Create a Custom URL Category

Step 3. (Optional) Create an Identification Profile to exempt Microsoft Updates traffic from Authentication

Note: This action is to reduce the authentication load on the SWA for the traffic to Microsoft Updates.

Step 3.1. From the **GUI**, choose **Web Security Manager**, and click **Identification Profiles**.
Step 3.2. Click **Add Profile** to add a profile.
Step 3.3. Ensure the **Enable Identification Profile** check box is selected.
Step 3.4. Assign a unique profile **Name**.
Step 3.5. (Optional) Add **Description**.
Step 3.6. From the **Insert Above** drop-down list, choose where this profile is to appear in the table.

Step 3.7. In the **User Identification Method** section, choose **Exempt from authentication/ identification**.

Step 3.8. In the **Define Members by Subnet**, If you want to Pass Through Microsoft traffic for specific users, enter the IP addresses or Subnets that apply, or leave this field blank to include all IP addresses.

Step 3.9. From the **Advanced** section, choose **Custom URL Categories**.

Step 3.10. Add the **Custom URL Category** that was created for Microsoft updates.

Step 3.11. Click **Done.**

Step 3.12. Click **Submit**.

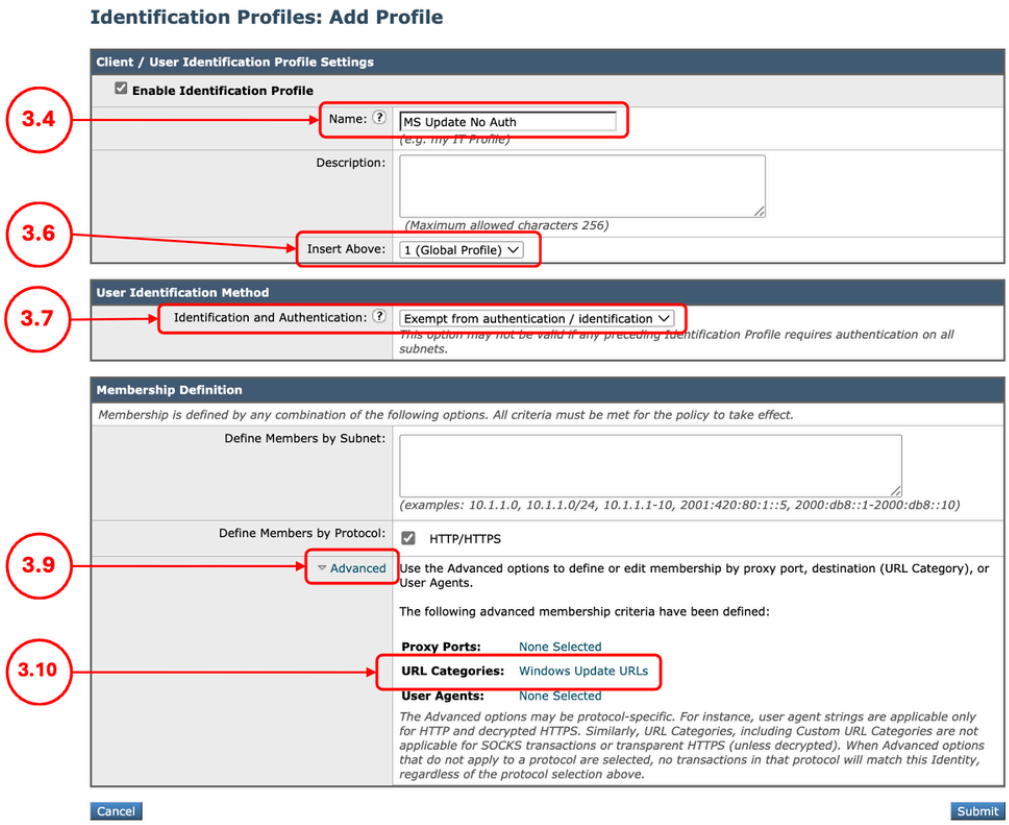


Image - Create Identification Profile

Step 4. (Optional) Create a Decryption Policy to Pass through Microsoft Updates Traffic

Note: Microsoft Updates use HTTP and HTTPS traffic to deliver update links. This action reduced the decryption load on the SWA.

Step 4.1. From the **GUI**, choose **Web Security Manager** and then click **Decryption Policy**.

Step 4.2. Click **Add Policy** to add a Decryption Policy.

Step 4.3. Assign a unique Policy **Name**.
Step 4.4. (Optional) Add **Description**.
Step 4.5. From the **Insert Above Policy** drop-down list, choose the first Policy.

Step 4.6. From the **Identification Profiles and Users**, choose **Select One or More Identification Profiles**.

Step 4.7. Select the Identification Profile that you created in **Step 3,** and skip to **Step 4.11.**

Step 4.8. If you did not create an ID profile for the Windows Updates, from the **Advanced** section, choose **Custom URL Categories**.

Step 4.9. Add the **Custom URL Category** that was created for the Microsoft Updates in **Step 2**.

Step 4.10. Click **Done**.

Step 4.11. Click **Submit**.

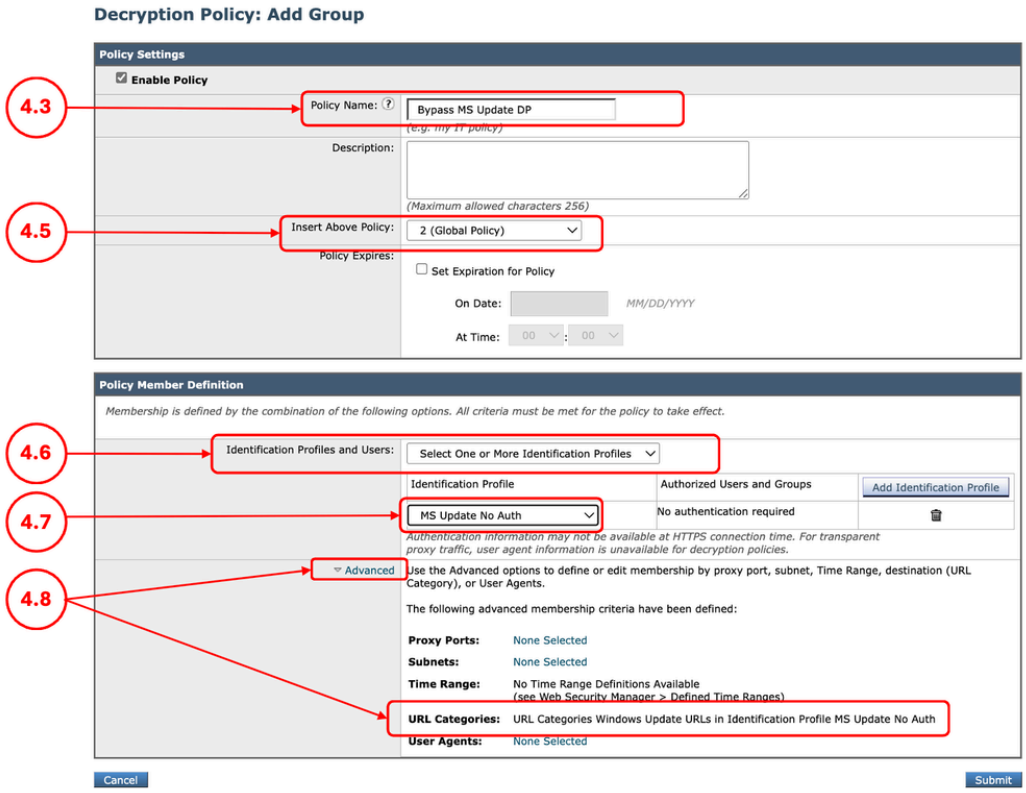


Image - Create a Decryption Policy

Step 4.12. In the **Decryption Policies** page under **URL Filtering**, click the link associated with the new Decryption Policy.

Step 4.13. Select **Pass Through** as the action for Microsoft Updates URL category.

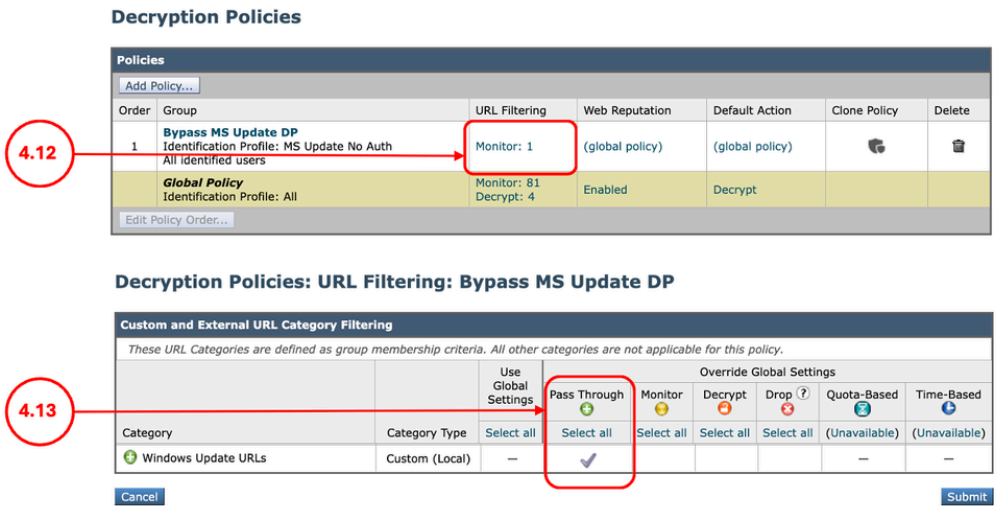


Image - Set the Action Pass Through for the URL Category

Step 4.14. Click **Submit**.

Step 5. Create an Access Policy to Allow Range Request for Microsoft Updates Traffic

Step 5.1. From the **GUI**, click **Web Security Manager** and choose **Access Policy**.

Step 5.2. Click **Add Policy** to add an Access Policy.

Step 5.3. Assign a unique Policy **Name**.
Step 5.4. (Optional) Add **Description**.
Step 5.5. From the **Insert Above Policy** drop-down list, choose the first Policy.

Step 5.6. From the **Identification Profiles and Users**, choose **Select One or More Identification Profiles**.

Step 5.7. Select the Identification Profile you created in **Step 3**, and skip to **Step 5.11**.

Step 5.8. If you did not create an ID profile for the Windows Update from the **Advanced** section, choose **Custom URL Categories**.

Step 5.9. Add the **Custom URL Category** that was created for the Microsoft Updates in **Step 2**.

Step 5.10. Click **Done**.

Step 5.11. Submit.

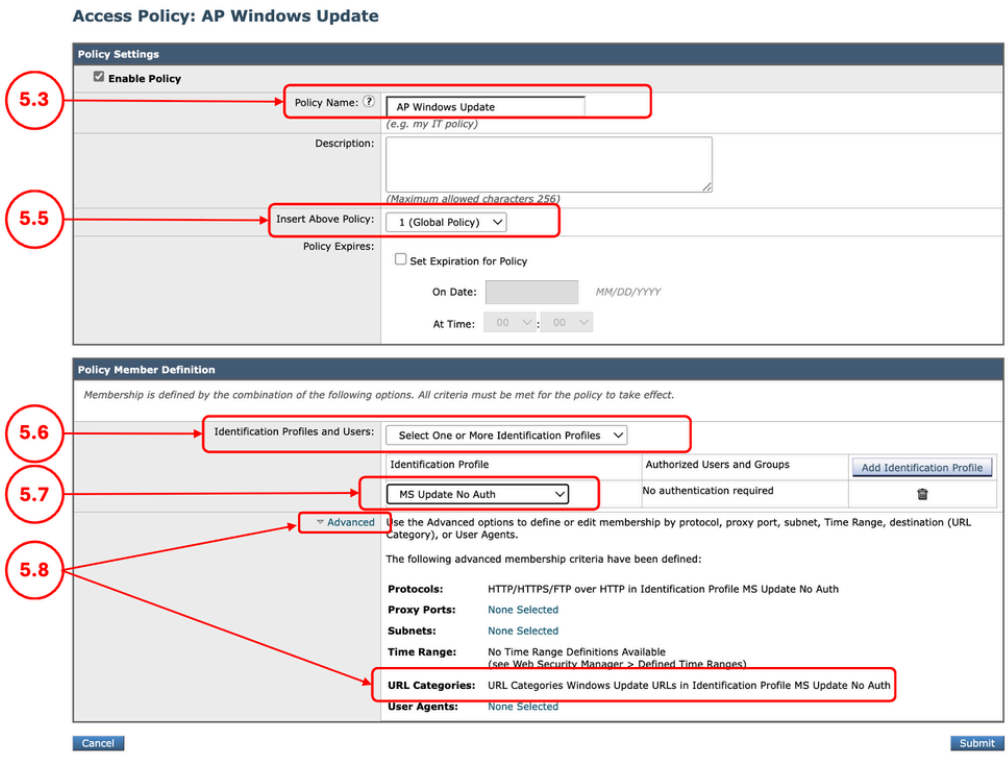


Image - Create the Access Policy

Step 5.12. On the **Access Policies** page under **URL Filtering**, click the link associated with the new Access Policy.

Step 5.13. Select **Allow** as the action for the Custom URL category created for the Microsoft Updates.

Step 5.14. Click **Submit**.

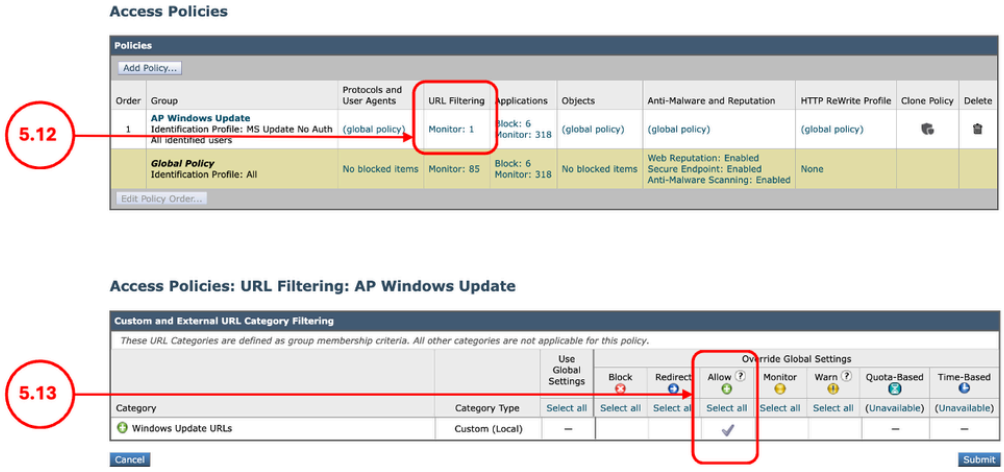


Image - Set the Action Allow for the URL Category

Step 5.15. On the **Access Policies** page under **Applications**, click the link associated with the new Access Policy.

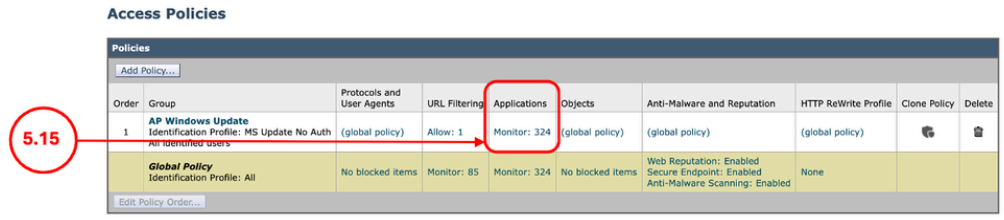


Image - Edit Application Visibility and Control

Step 5.16. On the **Edit Applications Settings** section, select **Define Applications Custom Settings**.

Step 5.17. From the Applications Settings section, click **Edit all** for **Games** application and set the action to **Block**.

Step 5.18. Click **Apply**.

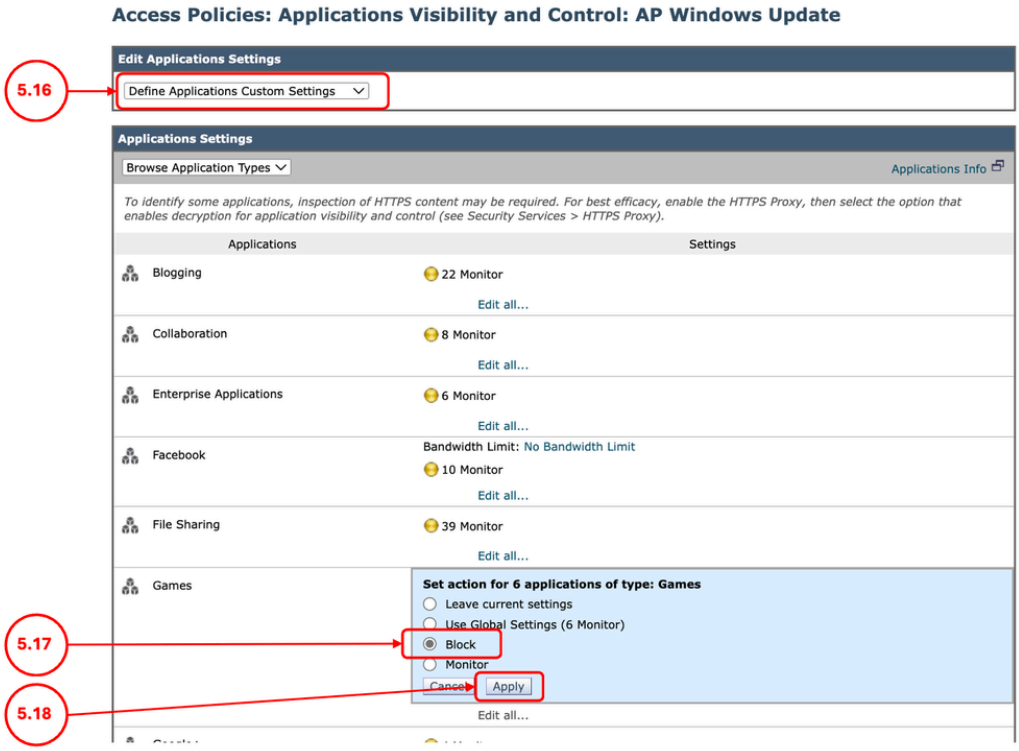


Image - Set One Application Action to Block

Step 5.19. Scroll down to the **Range Request Settings for Policy** section, ensure the **Forward range requests** is selected.

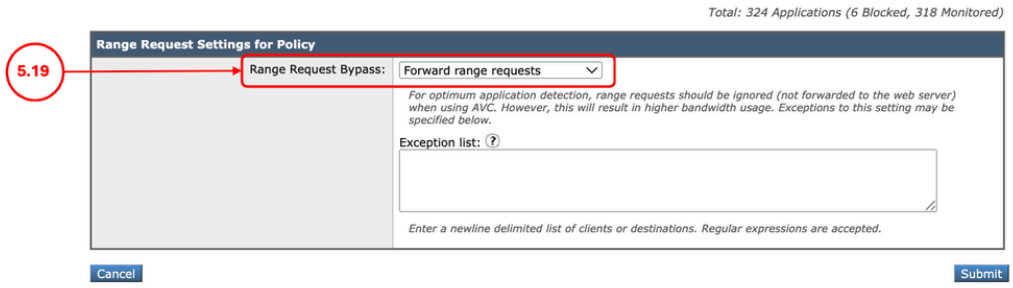


Image - Range Request Settings for Policy

Step 5.20. Click **Submit**.

Step 5.21. On the **Access Policies** page under **Applications**, click the link associated with the **Global Policy**.

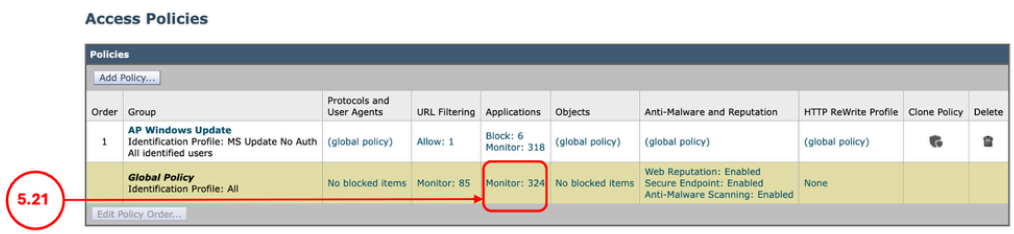


	Image - Default Access Policy Application Settings **Step 5.22.** Scroll down to the **Range Request Settings for Policy** section, ensure **Do Not Forward range requests** is selected, **Step 5.23. Commit c**hanges.

Modify Access Logs

For additional visibility on the Range Requests from the Access Logs, you can add these custom fields:

[Client Range = %<Range:]	Shows the range, requested by the client (Bytes)
[content= %>Content-Length:]	Shows the downloaded content size (Bytes)

For more information on adding a custom field to the SWA Access Logs, refer to the Configure Performance Parameter in Access Logs guide.

Verification

Run this CURL command to send a Range Request to the SWA:

```
curl -vvvk -H "Pragma: no-cache" -x 10.48.48.181:3128 -H 'Range: bytes=0-100' 'http://catalog.sf.dl.del
```

From the output of the CURL, you can see the HTTP response is **HTTP/1.1 206**:

```
> GET http://catalog.sf.dl.delivery.mp.microsoft.com/filestreamingservice/files/f263aa64-f367-42f0-9cad
> Host: catalog.sf.dl.delivery.mp.microsoft.com
> User-Agent: curl/8.7.1
> Accept: */*
> Proxy-Connection: Keep-Alive
> Pragma: no-cache
> Range: bytes=0-100
>
* Request completely sent off
< HTTP/1.1 206 Partial Content
```

From the Access Logs you can see the Action is **TCP_CLIENT_REFRESH_MISS/206**:

```
1773942471.096 14 10.190.0.206 TCP_CLIENT_REFRESH_MISS/206 860 GET http://catalog.sf.dl.delivery.mp.mic
```

Related Content

- User Guide for AsyncOS 15.0 for Cisco Secure Web Appliance - GD(General Deployment) - Classify End-Users for Policy Application [Cisco Secure Web Appliance] - Cisco
- Configure Custom URL Categories in Secure Web Appliance - Cisco
- How to Exempt Office 365 Traffic from Authentication and Decryption on Cisco Web Security Appliance (WSA) - Cisco
- Configure Performance Parameter in Access Logs
- Use Secure Web Appliance Best Practices - Cisco
- Bypass Authentication in Secure Web Appliance - Cisco