

Table of Contents

<u>Transparent Caching with the Content Switching Module Configuration Example</u>	1
<u>Document ID: 50381</u>	1
<u>Introduction</u>	1
<u>Before You Begin</u>	1
<u>Requirements</u>	1
<u>Components Used</u>	1
<u>Conventions</u>	1
<u>Configure</u>	2
<u>Network Diagram</u>	2
<u>Configurations</u>	2
<u>Verify</u>	4
<u>Troubleshoot</u>	5
<u>Related Information</u>	5

Transparent Caching with the Content Switching Module Configuration Example

Document ID: 50381

Introduction

Before You Begin

Requirements

Components Used

Conventions

Configure

Network Diagram

Configurations

Verify

Troubleshoot

Related Information

Introduction

This document provides a sample configuration for transparent caching using Cisco Cache Engines and the Content Switching Module (CSM). Transparent caching is the technique used to transparently intercept traffic from a Web browser and redirect it to a cache device to retrieve the content that was previously cached.

Another method to do transparent caching is Web Cache Communications Protocol (WCCP). The advantage of transparent caching over WCCP is that the CSM looks at the URL requested by the client and decides if the traffic should be sent to the cache or not. Requests for static files such as gif or jpeg images are retrieved from the cache, while dynamic pages (result of a script) are retrieved directly from the server without going to the cache.

Before You Begin

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these hardware and software versions:

- CSM version 3.x
- Application Content Networking Software (ACNS) version 5.1

Conventions

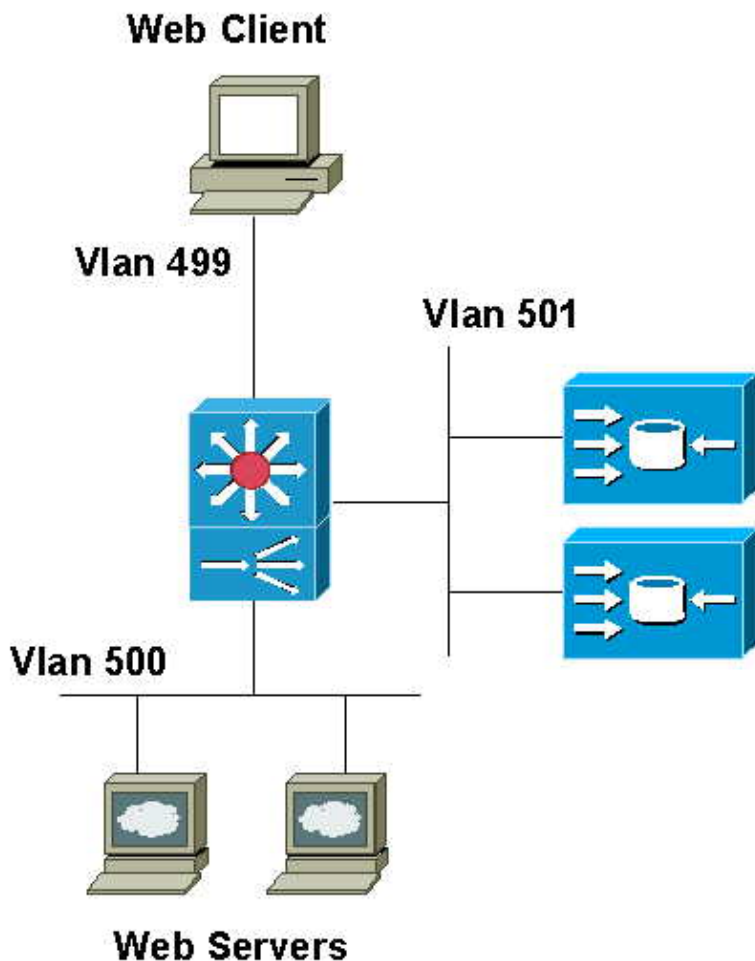
For more information on document conventions, see the Cisco Technical Tips Conventions.

Configure

In this section, you are presented with the information to configure the features described in this document.

Network Diagram

This document uses the network setup shown in the diagram below.



Configurations

This document uses this configuration:

```
module ContentSwitchingModule 4
  vlan 501 server
    ip address 192.168.30.97 255.255.254.0
  !
  vlan 499 client
    ip address 192.168.10.97 255.255.254.0
    gateway 192.168.10.1
  !
  vlan 500 server
    ip address 192.168.20.97 255.255.254.0
  !
  serverfarm CACHES
```

```

no nat server

!--- This is a transparent redirect; do not change the destination IP address.

no nat client
  predictor hash url

!--- Use URL hashing to make sure the request for a specific URL always goes to the same s

real 192.168.30.200
  inservice
  real 192.168.30.201
  inservice
!

serverfarm FORWARD
  no nat server
  no nat client
  predictor forward

!--- This serverfarm tells the CSM not to load balance.
!--- The CSM instead uses its routing table to forward the traffic.

!

map CACHEABLE url

!--- In this example, you want to only redirect requests for certain file types.
!--- This is not mandatory.
!--- You can also adjust this to something more realistic.

match protocol http url *.html
  match protocol http url *.gif
  match protocol http url *.jpg
  match protocol http url *.exe
  match protocol http url *.zip
!

policy CACHEABLE

!--- The policy is the way to link the map with a serverfarm.

url-map CACHEABLE
  serverfarm CACHES
!

vserver FROMCACHE

!--- This rule is for traffic originating from the caches (when they have
!--- to retrieve content from the origin server).

virtual 0.0.0.0 0.0.0.0 any
  vlan 501

```

!--- The VLAN command guarantees that you limit this vserver to the cache VLAN.

```
serverfarm FORWARD
```

*!--- Use the **serverfarm FORWARD** command to disable load balancing for this traffic.
!--- In this example, you need forward requests from the caches to the origin server.
!--- You could, however, load balance this traffic to a series of Web servers, that is,
!--- when doing reverse proxy caching.*

```
persistent rebalance  
  inservice  
!
```

```
vserver INTERCEPT
```

!--- This is the rule to transparently redirect requests from the client to the caches.

```
virtual 0.0.0.0 0.0.0.0 tcp www  
  vlan 499  
  serverfarm FORWARD
```

*!--- The default action is forward; no load balancing.
!--- This is for requests that do not match the policy.*

```
persistent rebalance  
  slb-policy CACHEABLE
```

!--- Traffic matching the policy is load balanced to the caches.

```
inservice  
!
```

```
vserver NONHTTP
```

!--- Non-HTTP traffic from the clients is forwarded.

```
virtual 0.0.0.0 0.0.0.0 any  
  vlan 499  
  serverfarm FORWARD  
  persistent rebalance  
  inservice  
!
```

Verify

This section provides information you can use to confirm your configuration is working properly.

- **show mod csm X vserver name *name* detail**
- **show mod csm X conns detail**

```

EOMER#show mod csm 4 vser name intercept det
INTERCEPT, type = SLB, state = OPERATIONAL, v_index = 22
  virtual = 0.0.0.0/0:80 bidir, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 499, pending = 30, layer 4
  max parse len = 2000, persist rebalance = TRUE
  ssl sticky offset = 0, length = 32
  conns = 0, total conns = 3
  Default policy:
    server farm = FORWARD, backup = <not assigned>
    sticky: timer = 0, subnet = 0.0.0.0, group id = 0
  Policy          Tot matches  Client pkts  Server pkts
  -----
  CACHEABLE      2           410         926
  (default)      5           20          17

```

Verify that the traffic matched the policy (traffic redirected to the caches), or if the traffic was forwarded (match on the default policy).

```

EOMER#show mod csm 4 conn det

      prot vlan source                destination                state
-----
In  ICMP 499 192.168.11.41          192.168.21.4             ESTAB
Out ICMP 500 192.168.21.4            192.168.11.41           ESTAB
    vs = NONHTTP, ftp = No, csrp = False

In  ICMP 501 192.168.10.107          10.48.66.102            ESTAB
Out ICMP 499 10.48.66.102          192.168.10.107          ESTAB
    vs = FROMCACHE, ftp = No, csrp = False

In  TCP 499 192.168.11.41:4402       192.168.21.4:80         REQ_WAIT
Out TCP 501 192.168.21.4:80     192.168.11.41:4402      REQ_WAIT
    vs = INTERCEPT, ftp = No, csrp = False

In  TCP 501 192.168.11.41:32784     192.168.21.4:80         ESTAB
Out TCP 500 192.168.21.4:80     192.168.11.41:32784     ESTAB
    vs = FROMCACHE, ftp = No, csrp = False

```

The cache was configured for IP spoofing. You can see in the output above that there is a connection from client 192.168.11.41 to server 192.168.21.4 seen on VLAN 499, and a similar connection seen on VLAN 501. The first one is the real connection from the client that was redirected to the cache (the out VLAN is 501), and the second one is the connection from the cache (spoofing client IP address) to the origin server.

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Related Information

- [Configuring Secure \(Router\) Mode on the CSM](#)
- [Content Switching Module Hardware Support](#)
- [Cisco Cat 6000 Other Intelligent Module SW Download \(registered customers only\)](#)
- [Technical Support – Cisco Systems](#)

All contents are Copyright © 1992–2005 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.
