Troubleshoot Reverse Transparent Caching for WCCP

Document ID: 9251

Contents

Introduction

This document describes how to troubleshoot Web Cache Communication Protocol (WCCP) when it is used to implement reverse transparent caching.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

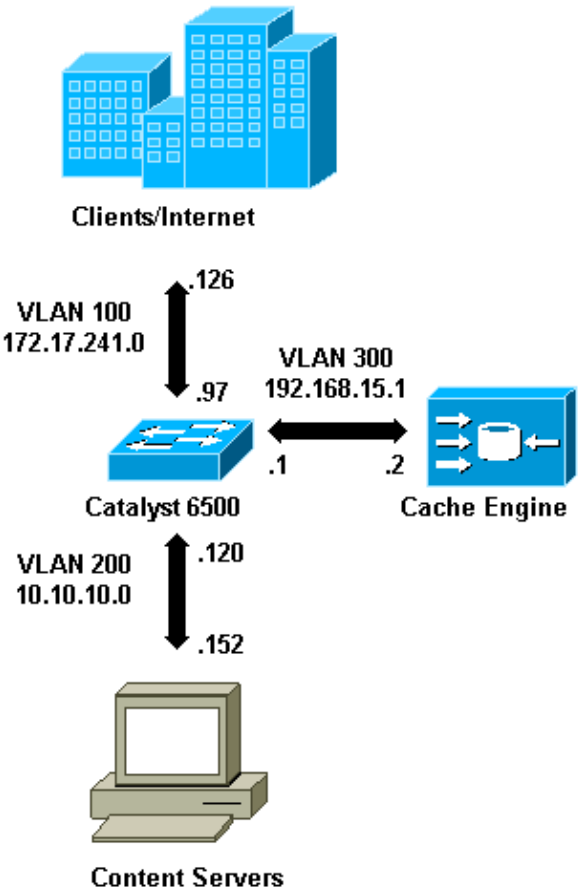
- Catalyst 6500 with Supervisor 1 and MSFC 1 configured in Native Mode
- Cisco IOS® Software Release 12.1(8a)EX (c6sup11−jsv−mz.121−8a.EX.bin)
- Cache Engine 550 with version 2.51

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, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for information on document conventions.

Configuration



When you install a Cache Engine, Cisco recommends that you configure only the commands necessary to implement WCCP. You can add other features, such as authentication to the router and clients redirection lists, at a later date.

On the Cache Engine, you must specify the IP address of the router and the version of WCCP you want to use.

```
wccp router-list 1 192.168.15.1
   wccp reverse-proxy router-list-num 1
   wccp version 2
```

Once the IP address and version of WCCP are configured, you might see a message that warns the service 99 should be activated in the router in order to implement reverse transparent caching. Service 99 is the WCCP service identifier for reverse transparent caching. The identifier for normal transparent caching is the word "web−cache" in the Cisco IOS. In order to activate service 99 (reverse transparent caching) on the router and in order to specify the port where the redirection will be performed, add these commands in the global configuration mode:

```
ip wccp 99
interface Vlan200
    ip address 10.10.10.120 255.255.255.0
    ip wccp 99 redirect out
```

When you configure reverse transparent caching, the router that runs WCCP service 99 intercepts requests directed to the web servers. The command **ip wccp 99 redirect out** is applied on the interface where you want to intercept the client HTTP packets in their path to your web server. Typically, this is the web server VLAN. This is normally not the VLAN where the Cache Engine is installed.

Once WCCP is active, the router listens on all ports that have WCCP redirect configured. To signal its presence, the Cache Engine continuously sends WCCP **Here I am** packets to the IP addresses that are configured in the router list.

A WCCP connection between the router and cache is formed. In order to view connection information, issue the **show ip wccp** command.

The router identifier is the IP address of the router as it is seen by the Cache Engines. This identifier is not necessarily the router interface used by the redirected traffic to reach the cache. The router identifier in this example is 192.168.15.1.

```
Router#show ip wccp
   Global WCCP information:
       Router information:
           Router Identifier:                  192.168.15.1
           Protocol Version:                   2.0
    Service Identifier: 99
           Number of Cache Engines:            1
           Number of routers:                  1
           Total Packets Redirected:           0
           Redirect access-list:               -none-
           Total Packets Denied Redirect:      0
           Total Packets Unassigned:           0
           Group access-list:                  -none-
           Total Messages Denied to Group:     0
           Total Authentication failures:      0
```

The **show ip wccp 99 detail** command provides detailed information about the caches.

```
Router#show ip wccp 99 detail
   WCCP Cache-Engine information:
           IP Address:               192.168.15.2
           Protocol Version:         2.0
           State:                    Usable
           Redirection:              GRE
           Initial Hash Info:        FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                                     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           Assigned Hash Info:       FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
                                     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           Hash Allotment:           256 (100.00%)
           Packets Redirected:       0
           Connect Time:             00:00:39
```

The `Redirection` field represents the method used to redirect the packets from the router to the Cache Engine. This method is either generic routing encapsulation (GRE) or Layer 2. With GRE, packets are encapsulated in a GRE packet. With Layer 2, packets are sent straight to the cache, but the Cache Engine and switch or router must be Layer 2 adjacent for Layer 2 redirection.

The `Hash Allotment` represented in hexadecimal in the `Initial Hash Info` and `Assigned Hash Info` fields is the number of hash buckets that are assigned to this cache. All possible source Internet addresses are divided into 64 equal sized ranges, one bucket per range, and each cache is assigned traffic from a number of these bucket source address ranges. This amount is managed dynamically by WCCP according to the load and load weighting of the cache. If you have only one cache installed, this cache might be assigned all buckets.

When the router starts to redirect packets to the Cache Engine, the number in the `Total Packets Redirected` field increases.

The `Total Packets Unassigned` field is the number of packets that were not redirected because they were not assigned to any cache. In this example, the number of packets is 5. Packets might be unassigned during initial discovery of caches or for a small interval when a cache is removed.

```
Router#show ip wccp
   Global WCCP information:
       Router information:
           Router Identifier:                    192.168.15.1
           Protocol Version:                     2.0
    Service Identifier: 99
           Number of Cache Engines:              1
           Number of routers:                    1
           Total Packets Redirected:             28
           Redirect access-list:                 -none-
           Total Packets Denied Redirect:        0
           Total Packets Unassigned:             5
           Group access-list:                    -none-
           Total Messages Denied to Group:       0
           Total Authentication failures:        0
```

If the cache does not get acquired by the router, it might be useful to debug the WCCP activity. Whenever the router receives a **Here I am** packet from the cache, it answers with an **I see you** packet, and this is reported in the debugs. The available **debug** commands are **debug ip wccp events** and **debug ip wccp packets**.

Note: Refer to Important Information on Debug Commands before you use **debug** commands.

This output provides a sample of normal WCCP debug messages:

```
Router#debug ip wccp event
   WCCP events debugging is on
  Router#debug ip wccp packet
   WCCP packet info debugging is on
   Router#
   2d18h: WCCP-EVNT:S00: Built new router view: 0 routers,
          0 usable web caches, change # 00000001
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to
          192.168.15.2 w/ rcv_id 00000001
   2d18h: WCCP-EVNT:S00: Redirect_Assignment packet from
          192.168.15.2 fails source check
   2d18h: %WCCP-5-SERVICEFOUND: Service web-cache
          acquired on Web Cache 192.168.15.2
   2d18h: WCCP-PKT:S00: Received valid Here_I_Am packet
          from 192.168.15.2 w/rcv_id 00000001
   2d18h: WCCP-EVNT:S00: Built new router view: 1
          routers, 1 usable web caches, change # 00000002
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2
          w/ rcv_id 00000002
   2d18h: WCCP-EVNT:S00: Built new router view: 1 routers,
          1 usable web caches, change # 00000002
   2d18h: WCCP-PKT:S00: Received valid Redirect_Assignment
          packet from 192.168.15.2 w/rcv_id 00000002
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2
          w/ rcv_id 00000003
   2d18h: WCCP-EVNT:S00: Built new router view: 1 routers,
          1 usable web caches, change # 00000002
   2d18h: WCCP-PKT:S00: Received valid Redirect_Assignment
          packet from 192.168.15.2 w/rcv_id 00000003
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2
          w/ rcv_id 00000004
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2
          w/ rcv_id 00000005
   2d18h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2
          w/ rcv_id 00000006
```

```
2d18h: WCCP-EVNT:S00: Built new router view: 1 routers,
       1 usable web caches, change # 00000002
2d18h: WCCP-PKT:S00: Received valid Redirect_Assignment
       packet from 192.168.15.2 w/rcv_id 00000006
```

In order to increase the debug level, you might want to trace the IP packet traffic in order to check whether the router receives packets from the Cache Engine. In order to avoid overloading a router in a production environment and in order to show only the interesting traffic, you can use an ACL to restrict the debugs only to the packets that have the IP address of the cache as source. A sample ACL is **access-list 130 permit ip host 192.168.15.2 host 192.168.15.1**.

```
Router#debug ip wccp event
   WCCP events debugging is on
  Router#debug ip wccp packet
   WCCP packet info debugging is on
  Router#debug ip packet 130
   IP packet debugging is on for access list 130
   2d19h: WCCP-EVNT:S00: Built new router view: 1 routers,  1 usable web caches,
          change # 00000002
   2d19h: WCCP-PKT:S00: Received valid Redirect_Assignment  packet from 192.168.15.2
          w/rcv_id 0000001B
   2d19h: datagramsize=174, IP 18390: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 0000001C
   2d19h: datagramsize=174, IP 18392: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 0000001D
   2d19h: datagramsize=174, IP 18394: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 0000001E
   2d19h: datagramsize=378, IP 18398: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 364, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-EVNT:S00: Built new router view: 1 routers,  1 usable web caches,
          change # 00000002
   2d19h: WCCP-PKT:S00: Received valid Redirect_Assignment  packet from 192.168.15.2
          w/rcv_id 0000001E
   2d19h: datagramsize=174, IP 18402: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 0000001F
   2d19h: datagramsize=174, IP 18404: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 00000020
   2d19h: datagramsize=174, IP 18406: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 00000021
   2d19h: datagramsize=378, IP 18410: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 364, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-EVNT:S00: Built new router view: 1 routers,  1 usable web caches,
          change # 00000002
   2d19h: WCCP-PKT:S00: Received valid Redirect_Assignment  packet from 192.168.15.2
          w/rcv_id 00000021
   2d19h: datagramsize=174, IP 18414: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
   2d19h: WCCP-PKT:S00: Sending I_See_You packet to 192.168.15.2  w/ rcv_id 00000022
   2d19h: datagramsize=174, IP 18416: s=192.168.15.2  (Vlan300), d=192.168.15.1
          (Vlan300), totlen 160, fragment 0, fo 0, rcvd 3
```

In the event that no caches are seen by the router and no WCCP activity is seen, check the basic connectivity. Try to ping the cache from the router or the router from the cache. If the ping works, an error might exist in the configuration.

If the cache is acquired, but no packets are redirected, verify that the router receives traffic and that the traffic is forwarded to the interface where the **ip wccp 99 redirect out** command is applied. Remember that the

traffic that is intercepted and redirected is only the traffic directed to the TCP port 80.

If the traffic is still not being redirected and the web content is coming straight from the servers, verify that the cache correctly passes the instruction on what to intercept. You must have some background information on WCCP in order to complete this action.

WCCP recognizes two different types of services: *standard* and *dynamic*. The router implicitly knows of a standard service. That is, the router does not need to be told to use port 80, because it already knows to do so. The normal transparent caching (web−cache − standard service 0) is a standard service.

In all the other cases (which includes transparent caching), the router is told which port to intercept. This information is passed in the **Here I am** packet.

You can issue the **debug ip packet dump** command in order to examine the packets themselves. Use the ACL created to debug only the packets sent by the Cache Engine.

```
Router#debug ip packet 130 dump
   2d19h: datagramsize=174, IP 19576: s=192.168.15.2 (Vlan300), d=192.168.15.1
      (Vlan300), totlen 160, fragment 0, fo 0,
   rcvd 3
   072C5120:                       0004 9B294800            ...)H.

!--- Start IP header.

   072C5130: 00500F0D 25360800 450000A0 4C780000  .P..%6..E.. Lx..
   072C5140: 3F118F81 C0A80F02 C0A80F01 08000800  ?...@(..@(......
   072C5150: 008CF09E 0000000A 0200007C 00000004  ..p........|....

!--- Start WCCP header.

   072C5160: 00000000 00010018 0163E606 00000515  .........cf.....
   072C5170: 00500000 00000000 00000000 00000000  .P..............

!--- Port to intercept (0x50=80).

   072C5180: 0003002C C0A80F02 00000000 FFFFFFFF  ...,@(..........

!--- Hash allotment (FFFF...).

   072C5190: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
   072C51A0: FFFFFFFF FFFFFFFF FFFF0000 00000000  ................
   072C51B0: 00050018 00000002 00000001 C0A80F01  ............@(..
   072C51C0: 0000000C 00000001 C0A80F02 00080008  ........@(......
   072C51D0: 00010004 00000001 30                 ........0
```

With this command, you can determine whether or not the port is advertised without the need to view the entire Request For Comments (RFC). If the port is not advertised, the problem is most likely in the configuration of the cache.

Refer to Web Cache Coordination Protocol V2.0 for more information.

If the cache is acquired and packets are redirected, but your Internet clients cannot browse your servers, check whether the cache has connectivity to the Internet and to your servers. Ping from the cache to various IP addresses on the Internet and to some of your internal servers. If you ping fully qualified domains (URLs) instead of IP addresses, be sure that you specify the DNS server to use in the cache configuration.

If you are unsure whether the cache processes the requests, you can debug the HTTP activity in the cache. In order to debug the HTTP activity in the cache, you must restrict the traffic to avoid overloading the cache. On the router, create an ACL with the source IP address of one client in the Internet that you can use as a device

for your tests and use the option **redirect−list** of the global command **ip wccp 99**.

```
Router(config)#access−list 50 permit 172.17.241.126
Router(config)#ip wccp 99 redirect−list 50
```

Once you create and apply the ACL, complete these steps:

1. Activate the HTTP debug in the cache with the command **debug http all all** (Cisco Cache Engine version 2.x) or **debug http all** (Cisco Cache Engine version 3 and ACNS version 4, 5).
2. Activate terminal monitoring (issue the **term mon** command).
3. Try to browse one of your servers from the client you configured in the ACL.

Here is an example of the output:

```
irq0#conf tcework_readfirstdata() Start the recv: 0xb820800 len 4096 timeout
   0x3a98 ms ctx 0xb87d800
   cework_recvurl() Start the request: 0xb20c800 0xb20c838 0xb20c8e0
   Http Request headers received from client:
   GET / HTTP/1.1
   Host: 10.10.10.152
   User−Agent: Links (0.92; Linux 2.2.16−22 i686)
   Accept: */*
   Accept−Charset: us−ascii, ISO−8859−1, ISO−8859−2, ISO−8859−4, ISO−8895−5,
      ISO−8859−13, windows−1250, windws−1251, windows−1257, cp437, cp850, cp852,
       cp866, x−cp866−u, x−mac−ce, x−kam−cs, x−koi8−r, x−koi8−u, utf8
   Connection: Keep−Alive

Protocol dispatch: mode=1 proto=2
   ValidateCode() Begin: pRequest=0xb20c800
   Proxy: CACHE_MISS: HealProcessUserRequest
   cework_teefile() 0xb20c800: Try to connect to server: CheckProxyServerOut():
      Outgoing proxy is not enable: 0xb20c800 (F)
   GetServerSocket(): Forwarding to server: pHost = 10.10.10.152, Port = 80
   HttpServerConnectCallBack : Connect call back socket = 267982944, error    = 0
   Http request headers sent to server:

   GET / HTTP/1.1
   Host: 10.10.10.152
   User−Agent: Links (0.92; Linux 2.2.16−22 i686)
   Accept: */*
   Accept−Charset: us−ascii, ISO−8859−1, ISO−8859−2, ISO−8859−4, ISO−8895−5,
      ISO−8859−13, windows−1250, windws−1251, windows−1257, cp437, cp850, cp852,
      cp866, x−cp866−u, x−mac−ce, x−kam−cs, x−koi8−r, x−koi8−u, utf8
   Connection: keep−alive
   Via: 1.1 irq0
   X−Forwarded−For: 172.17.241.126

cework_sendrequest: lBytesRemote = 386, nLength = 386 (0xb20c800)
   ReadResCharRecvCallback():  lBytesRemote = 1818, nLength = 1432 0xb20c800)
   IsResponseCacheable() OBJECTSIZE_IS_UNLIMITED, lContentLength = 3194
   cework_processresponse() : 0xb20c800 is cacheable
   Http response headers received from server:
   HTTP/1.1 200 OK
   Date: Tue, 20 Nov 2001 10:46:14 GMT
   Server: Apache/1.3.12 (Unix)  (Red Hat/Linux) mod_ssl/2.6.6 OpenSSL/0.9.5a
      mod_perl/1.24
   Last−Modified: Fri, 12 Oct 2001 12:55:23 GMT
   ETag: "5e23−c7a−3bc6e83b"
   Accept−Ranges: bytes
   Content−Length: 3194
   Keep−Alive: timeout=15, max=100
   Connection: Keep−Alive
   Content−Type: text/html
```

```
GetUpdateCode(): GET request from client, GET request to server.
   GetUpdateCode(): nRequestType = −1
   SetTChain() 0xb20c800: CACHE_OBJECT_CLIENT_OBJECT sendobj_and_cache
   Http response headers sent to client:
   HTTP/1.1 200 OK
   Date: Tue, 20 Nov 2001 10:46:14 GMT
   Server: Apache/1.3.12 (Unix)  (Red Hat/Linux) mod_ssl/2.6.6 OpenSSL/0.9.5a
      mod_perl/1.24
   Last−Modified: Fri, 12 Oct 2001 12:55:23 GMT
   ETag: "5e23−c7a−3bc6e83b"
   Content−Length: 3194
   Keep−Alive: timeout=15, max=100
   Content−Type: text/html
   Connection: keep−alive

cework_tee_sendheaders() 0xb20c800: sent 323 bytes to client
   cework_tee_send_zbuf() 0xb20c800: Send 1087 bytes to client (1087)
   UseContentLength(): Valid Content−Length (T)
   cework_tee_recv_zbuf() 0xb20c800: Register to recv 2107 bytes timeout 120 sec
   HttpServerRecvCallBack(): Recv Call Back socket 267982944, err 0, length  2107
   HttpServerRecvCallBack(): lBytesRemote = 3925, nLength = 2107 (186697728)
   cework_tee_send_zbuf() 0xb20c800: Send 2107 bytes to client (2107)
   UseContentLength(): Valid Content−Length (T)
   cework_setstats(): lBytesLocal = 0, lBytesRemote = 3925 (0xb20c800)
   cework_readfirstdata() Start the recv: 0xb84a080 len 4096 timeout 0x3a98
      ms ctx 0xb87d800
   cework_cleanup_final() End the request: 0xb20c800 0xb20c838 0xb20c8e0
```

The relevant information you might find in the debug is highlighted in **bold**.

These are the different phases of a Web page page transaction:

1. HTTP request headers received from client.
2. HTTP request headers sent to server.
3. HTTP response headers received from server.
4. HTTP response headers sent to client.

If the web page you browse contains multiple objects, multiple instances of this sequence of events exist. Use the simplest possible request to reduce the debug output.

On a Catalyst 6500 or a Cisco 7600 router, a feature manager handles all the features configured in the Cisco IOS in order to provide an added layer of troubleshooting. When a Layer 3 feature is configured in these devices, information that defines how to handled the received frames is passed to the Layer 2 control functions of the switch or router (the feature manager). For WCCP, this control information defines what packets are intercepted by IOS and WCCP and directed to the transparent cache.

The **show fm features** command displays the features that are enabled in the Cisco IOS. You can use this command in order to check whether the port to intercept is correctly advertised by the Cache Engine.

```
Router#show fm features
   Redundancy Status: stand−alone
   Interface: Vlan200 IP is enabled
     hw[EGRESS] = 1, hw[INGRESS] = 1
     hw_force_default[EGRESS] = 0, hw_force_default[INGRESS] = 0
     mcast = 0
     priority = 2
     reflexive = 0
     vacc_map :
     outbound label: 5
           merge_err: 0
           protocol: ip
```

```
              feature #: 1
              feature id: FM_IP_WCCP
              Service ID: 99
              Service Type: 1

The following are the used labels
     label 5:
           swidb: Vlan200
           Vlous:

The following are the features configured
     IP WCCP: service_id = 99, service_type = 1, state = ACTIVE
           outbound users:
             user_idb: Vlan200
           WC list:
             address: 192.168.15.2
           Service    ports:
             ports[0]:    80

The following is the ip ACLs port expansion information
     FM_EXP knob configured: yes

FM mode for WCCP: GRE (flowmask: destination-only)

FM redirect index base: 0x7E00

The following are internal statistics
     Number of pending tcam inserts: 0
     Number of merge queue elements: 0
```

The command **show fm int vlan 200** displays the exact content of the Ternary Content Addressable Memory (TCAM).

```
Router#show fm int vlan 200
 Interface: Vlan200 IP is enabled
  hw[EGRESS] = 1, hw[INGRESS] = 1
  hw_force_default[EGRESS] = 0, hw_force_default[INGRESS] = 0
  mcast = 0
  priority = 2
  reflexive = 0
  vacc_map :
  outbound label: 5
   merge_err: 0
   protocol: ip
    feature #: 1
    feature id: FM_IP_WCCP
    Service ID: 99
    Service Type: 1
     (only for IP_PROT) DestAddr SrcAddr          Dpt   Spt  L4OP TOS Est  prot  Rslt
     vmr IP value #1:   0.0.0.0  192.168.15.2     0     0    0    0   0    6     permit
     vmr IP mask  #1:   0.0.0.0  255.255.255.255  0     0    0    0   0    FF
     vmr IP value #2:   0.0.0.0  0.0.0.0          80    0    0    0   0    6     bridge
     vmr IP mask  #2:   0.0.0.0  0.0.0.0          FFFF  0    0    0   0    FF
     vmr IP value #3:   0.0.0.0  0.0.0.0          0     0    0    0   0    0     permit
     vmr IP mask  #3:   0.0.0.0  0.0.0.0          0     0    0    0   0    0
```

The `vmr IP value # 1:` line defines interception bypass on frames that come from the Cache Engine. Without this, there would be a redirection loop. The `vmr IP value # 2:` line defines interception of all the packets that have port 80 as their destination. If port 80 is not displayed in the second line, but WCCP is active and the cache is usable by the router, then there might be a problem in the cache configuration. Collect a dump of the **Here I am** packet in order to determine whether or not the port is sent by the cache.

If you are unable to solve the problem after you troubleshoot, report the problem to the Cisco Technical

Assistance Center (TAC).

Here is some basic information that you must provide to the Cisco TAC. From the router, collect this information:

- The output of the **show tech** command. The output of the **show running−config** and **show version output** commands can be substituted if there is difficulty with the size of the **show tech** output.
- The output of the **show ip wccp** command.
- The output of the **show ip wccp web−cache detail** command.
- If there seems to be a problem with communication between the router and the Web cache, provide output from the **debug ip wccp events** and **debug ip wccp packets** commands while the problem is occurring.

On the Cache Engine (Cisco Cache Engines only), collect the output of the **show tech** command.

When you contact the TAC, complete these steps:

1. Provide a clear description of the problem. You should include answers to these questions:
    - ♦ What are the symptoms?
    - ♦ Does it occur all the time or infrequently?
    - ♦ Did the problem start after a change in the configuration?
    - ♦ Are Cisco or 3rd party caches used?
2. Provide a clear description of the topology. Include a diagram if that will make it more clear.
3. Provide any other information that you think is useful in solving the problem.

Here is output of a sample configuration:

```
********************************** Router Configuration ****************************
Router#show running
   Building configuration...
Current configuration : 4231 bytes
   !
   version 12.1
   service timestamps debug uptime
   service timestamps log uptime
   no service password−encryption
   !
   hostname Router
   !
   boot buffersize 126968
   boot bootldr bootflash:c6msfc−boot−mz.120−7.XE1
   !
   redundancy
    main−cpu
     auto−sync standard
   ip subnet−zero
   ip wccp 99
   !
   !
   !
   interface FastEthernet3/1
    no ip address
    switchport
    switchport access vlan 100
    switchport mode access
   !
   interface FastEthernet3/2
    no ip address
    switchport
```

```
 switchport access vlan 200
 switchport mode access
!
interface FastEthernet3/3
 no ip address
 switchport
 switchport access vlan 300
 switchport mode access
!
interface FastEthernet3/4
 no ip address
!
!
interface Vlan100
 ip address 172.17.241.97 255.255.255.0
!
interface Vlan200
 ip address 10.10.10.120 255.255.255.0
 ip wccp 99 redirect out
!
interface Vlan300
 ip address 192.168.15.1 255.255.255.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.17.241.1
no ip http server
!
access-list 30 permit 192.168.15.2
!
!
line con 0
 exec-timeout 0 0
line vty 0 4
 login
 transport input lat pad mop telnet rlogin udptn    nasi
!
end
********************************** Cache Configuration ******************************
Cache#show running
Building configuration...
Current configuration:
!
!
logging disk /local/syslog.txt debug
!
user add admin uid 0  capability admin-access
!
!
!
hostname Cache
!
interface ethernet 0
 ip address 192.168.15.2 255.255.255.0
 ip broadcast-address 192.168.15.255
 exit
!
interface ethernet 1
 exit
!
ip default-gateway 192.168.15.1
ip name-server 172.17.247.195
ip domain-name cisco.com
ip route 0.0.0.0 0.0.0.0 192.168.15.1
cron file /local/etc/crontab
!
wccp router-list 1 192.168.15.1
```

```
wccp reverse-proxy router-list-num 1
wccp version 2
!
authentication login local enable
authentication configuration local enable
rule no-cache url-regex .*cgi-bin.*
rule no-cache url-regex .*aw-cgi.*
!
!
end
```

Related Information

- **Cisco Cache Software**
- **Cisco 500 Series Cache Engines**
- **Web Cache Communications Protocol (WCCP)**
- **Cisco Cache Engine 2.0 Software Download Page (registered customers only)**
- **Cisco Cache Engine 3.0 Software Download Page (registered customers only)**
- **Technical Support & Documentation − Cisco Systems**

Updated: Jan 02, 2007 Document ID: 9251
