

# Troubleshooting SSL Configurations on the CSS 11000

Document ID: 23643

---

## Introduction

### Before You Begin

- Conventions

- Prerequisites

- Components Used

### Topics

- Configuration

- Server Side Issues

- Client Side Issues

- Debug Tools

- Workarounds

### Related Information

---

## Introduction

This document provides information on Secure Socket Layer (SSL) configuration on the Content Services Switch (CSS) 11000.

## Before You Begin

### Conventions

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

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

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

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

## Topics

### Configuration

Application SSL and advanced-balance SSL use Layer 5 (L5) sticky based SSL Session IDs (SIDs). Layer 4 (L4) sticky uses script and destination port. Layer 3 (L3) sticky uses scripts.

```

content L5rule1
vip address 192.168.222.182
application ssl
advanced-balance ssl
protocol tcp
port 443
url "/*"
add service x0lcol01
add service x0lcol02
add service x0lcol03
active

```

## Server Side Issues

The server may send a null SSL SID.

```

Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time Summary
28 [192.168.222.34] [192.168.222.251] 844 0:00:06.769 0.001.114 09/25/2001
    05:46:26 PM TCP: D=1272 S=443 ACK=84536369 SEQ=2714661520 LEN=790 WIN=5840
DLC: ----- DLC Header -----
DLC:
DLC: Frame 28 arrived at 17:46:26.7699; frame size is 844 (034C hex) bytes.
DLC: Destination = Station 0010A4A6FE51
DLC: Source = Station 00E01805C3A4
DLC: Ethertype = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 830 bytes
IP: Identification = 14832
IP: Flags = 4X
IP: .1.. .... = don't fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 64 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = BF5A (correct)
IP: Source address = [192.168.222.34]
IP: Destination address = [192.168.222.251]
IP: No options
IP:
TCP: Retransmitted in frame 42
TCP: ----- TCP header -----
TCP:
TCP: Source port = 443 (Https)
TCP: Destination port = 1272
TCP: Sequence number = 2714661520
TCP: Next expected Seq number= 2714662310
TCP: Acknowledgment number = 84536369
TCP: Data offset = 20 bytes
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)

```

```

TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 5840
TCP: Checksum = 1717 (correct)
TCP: No TCP options
TCP: [790 Bytes of data]
TCP:
ADDR HEX ASCII
0000: 00 10 a4 a6 fe 51 00 e0 18 05 c3 a4 08 00 45 00 | ..Q.....E.
0010: 03 3e 39 f0 40 00 40 06 bf 5a c0 a8 de 22 c0 a8 | .>9@.@.Z"
0020: de fb 01 bb 04 f8 a1 ce 72 90 05 09 ec 31 50 18 | ..r..lP.
0030: 16 d0 17 17 00 00 16 03 00 00 2a 02 00 00 26 03 | .....*...&.
0040: 00 3b b1 24 67 36 03 bd d3 fc 86 63 21 f1 d8 9a | .;$g6.c!
0050: 20 06 d0 f3 82 0d be 65 0c 61 1a 00 16 57 89 9c | ..e.a...W
0060: 4f 00 00 04 00 16 03 00 02 d9 0b 00 02 d5 00 02 | O.....
0070: d2 00 02 cf 30 82 02 cb 30 82 02 34 02 01 00 30 | ..0.0.4...0
0080: 0d 06 09 2a 86 48 86 f7 0d 01 01 04 05 00 30 81 | ...*H.....0
0090: ad 31 0b 30 09 06 03 55 04 06 13 02 55 53 31 13 | 1.0...U....US1.

```

Offset hex 62 is the SSL ID length field, which in this case is set to 0. The client is unable to cache this ID, and as a result, the client has to recreate its hashing string every time he reconnects to the backend server for a sticky connection. This will introduce performance degradation. The backend server needs to have the SSL ID set to cacheable and the length set to a valid value (6–32 bytes is the norm).

With a SID length of 0, you would not compute a hash value of zero. Instead, you would use a L4 hash consisting of the client Ip xor dest port.

## Client Side Issues

Internet Explorer (IE) 5.0/5.5 SSL SID changes approximately every two minutes.

The client will see different behavior with IE 5.x than with Netscape. IE will cause the client to change its SSL ID every two minutes and this will break stickyness with application SSL and advanced–balance SSL, as this is L5 stickyness based on SSL SID. A sniffer trace from the client will show the ID field change.

In this frame, starting at offset hex 62 for the length and offset 63 for the first byte of the SSL ID, you can see values of 20 and 29 respectively.

```

- - - - - Frame 28 - - - - -
Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time Summary
28 [161.44.175.145] [208.184.140.161] 150 0:00:03.371 0.016.261 10/19/2001 03:57:37 PM
    TCP: D=443 S=3406 ACK=121147614 SEQ=105456165 LEN=96 WIN=9520
----- DLC Header -----
DLC:
DLC:
DLC: Frame 28 arrived at 15:57:37.3792; frame size is 150 (0096 hex) bytes.
DLC: Destination = Station Cisco107AC01
DLC: Source = Station Xircm2229D27
DLC: Ethertype = 0800 (IP)
DLC:
----- IP Header -----
IP:
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit

```

```

IP: .... .0 = CE bit - no congestion
IP: Total length = 136 bytes
IP: Identification = 35210
IP: Flags = 4X
IP: .1.. .... = don't fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 128 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = C2CD (correct)
IP: Source address = [161.44.175.145]
IP: Destination address = [208.184.140.161]
IP: No options
IP:
----- TCP header -----
TCP:
TCP:
TCP: Source port = 3406
TCP: Destination port = 443 (Https)
TCP: Sequence number = 105456165
TCP: Next expected Seq number= 105456261
TCP: Acknowledgment number = 121147614
TCP: Data offset = 20 bytes
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 9520
TCP: Checksum = 9A67 (correct)
TCP: No TCP options
TCP: [96 Bytes of data]
TCP:
ADDR HEX ASCII
0000: 00 00 0c 07 ac 01 00 80 c7 22 9d 27 08 00 45 00 | ....."'..E.
0010: 00 88 89 8a 40 00 80 06 c2 cd a1 2c af 91 d0 b8 | .@.,,
0020: 8c a1 0d 4e 01 bb 06 49 22 25 07 38 90 de 50 18 | .N..I"%.8P.
0030: 25 30 9a 67 00 00 16 03 01 00 5b 01 00 00 57 03 | %0g.....[...W.
0040: 01 0c bd 94 4d 58 63 59 c2 e2 b7 7f 48 34 17 2b | ..MXcY H4.+
0050: cf f9 22 85 bc 00 fe 76 ad de 4d a0 f8 17 dc 3c | ".vM .<
0060: a1 20 29 57 08 7d de 30 23 c4 bc e2 fd 26 a5 8b | )W.}0#&
0070: f4 5b 48 a8 50 2d 8b be 37 76 cc b3 c2 48 5c 69 | [HP-7vH\i
0080: 71 02 00 10 00 04 00 05 00 0a 00 09 00 64 00 62 | q.....d.b
0090: 00 03 00 06 01 00 | .....

```

The frame below, sent by the client 2 minutes and 64 seconds later, has values of 40 and 01 for the same fields.

```

----- Frame 945 -----
Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time Summary
945 [161.44.175.145] [208.184.140.161] 153 0:02:35.533 0.001.228 10/19/2001 04:00:09
PM TCP: D=443 S=3464 ACK=1374357434 SEQ=105608315 LEN=99 WIN=9520
----- DLC Header -----
DLC:
DLC:
DLC: Frame 945 arrived at 16:00:09.5404; frame size is 153 (0099 hex) bytes.
DLC: Destination = Station Cisco107AC01
DLC: Source = Station Xircm2229D27
DLC: Ethertype = 0800 (IP)
DLC:

```

```

----- IP Header -----
IP:
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 139 bytes
IP: Identification = 63628
IP: Flags = 4X
IP: .1.. .... = don't fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 128 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = 53C8 (correct)
IP: Source address = [161.44.175.145]
IP: Destination address = [208.184.140.161]
IP: No options
IP:
----- TCP header -----
TCP:
TCP:
TCP: Source port = 3464
TCP: Destination port = 443 (Https)
TCP: Sequence number = 105608315
TCP: Next expected Seq number= 105608414
TCP: Acknowledgment number = 1374357434
TCP: Data offset = 20 bytes
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 9520
TCP: Checksum = E691 (correct)
TCP: No TCP options
TCP: [99 Bytes of data]
TCP:

```

ADDR HEX ASCII

```

0000: 00 00 0c 07 ac 01 00 80 c7 22 9d 27 08 00 45 00 | ....."'..E.
0010: 00 8b f8 8c 40 00 80 06 53 c8 a1 2c af 91 d0 b8 | .@..S,
0020: 8c a1 0d 88 01 bb 06 4b 74 7b 51 eb 07 ba 50 18 | ...Kt{Q.P.
0030: 25 30 e6 91 00 00 80 61 01 03 01 00 48 00 00 00 | %0..a....H...
0040: 10 8f 80 01 80 00 03 80 00 01 81 00 01 81 00 03 | .....
0050: 82 00 01 00 00 04 00 00 05 00 00 0a 83 00 04 84 | .....
0060: 80 40 01 00 80 07 00 c0 03 00 80 00 00 09 06 00 | @.....
0070: 40 00 00 64 00 00 62 00 00 03 00 00 06 83 00 04 | @..d.

```

## Debug Tools

Use tools below to debug your configuration.

1. **Sniffer trace:** Ideally, you would want a front side (between the CSS and the incoming client) and back side (between the CSS and the web server) sniffer traces. This will show what the client sends, what the CSS sends to the server, and how the server responds back to the CSS. You would also see what the CSS sends back to the client.
2. **Log mask for WCC:** To turn on debug messages for SSL transactions on the CSS, you would go into debug mode (type `llama` to get there), and set the mask for the Web Cache Communication (WCC) subsystem.

```
CS150(debug)# mask wcc 0x1000
```

To set this back to default, issue the command shown below.

```
CS150(debug)# mask wcc 0x0
OCT 19 11:34:51 5/1 2682406 WCC-7: SSL session started !!!
OCT 19 11:34:51 5/1 2682407 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:51 5/1 2682408 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:51 5/1 2682409 WCC-7: Hash insert for new server id accum = 48eab0e8,
server = 5, ruleIndex = 1
OCT 19 11:34:53 5/1 2682410 WCC-7: SSL session started !!!
OCT 19 11:34:53 5/1 2682411 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:53 5/1 2682412 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:53 5/1 2682413 WCC-7: Hash insert for new server id accum = a2550087,
server = 4, ruleIndex = 1
OCT 19 11:34:54 5/1 2682414 WCC-7: SSL session started !!!
OCT 19 11:34:54 5/1 2682415 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:54 5/1 2682416 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:54 5/1 2682417 WCC-7: Hash insert for new server id accum = 2de8de75,
server = 6, ruleIndex = 1
OCT 19 11:34:56 5/1 2682418 WCC-7: SSL session started !!!
OCT 19 11:34:56 5/1 2682419 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:56 5/1 2682420 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:56 5/1 2682421 WCC-7: Hash insert for new server id accum = 59001543,
server = 5, ruleIndex = 1
OCT 19 11:34:57 5/1 2682422 WCC-7: SSL session started !!!
OCT 19 11:34:57 5/1 2682423 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:57 5/1 2682424 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:57 5/1 2682425 WCC-7: Hash insert for new server id accum = a6b3fa70,
server = 4, ruleIndex = 1
OCT 19 11:34:59 5/1 2682426 WCC-7: SSL session started !!!
OCT 19 11:34:59 5/1 2682427 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:34:59 5/1 2682428 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:34:59 5/1 2682429 WCC-7: Hash insert for new server id accum = 8914d4a9,
server = 6, ruleIndex = 1
OCT 19 11:35:00 5/1 2682430 WCC-7: SSL session started !!!
OCT 19 11:35:00 5/1 2682431 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:35:00 5/1 2682432 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:35:00 5/1 2682433 WCC-7: Hash insert for new server id accum = 756a3627,
server = 5, ruleIndex = 1
OCT 19 11:35:02 5/1 2682434 WCC-7: SSL session started !!!
OCT 19 11:35:02 5/1 2682435 WCC-7: Sticky server not found, accum=8d8100d9
OCT 19 11:35:02 5/1 2682436 WCC-7: Sticky failover=0, csdStatus=17, ret=FALSE
OCT 19 11:35:02 5/1 2682437 WCC-7: Hash insert for new server id accum = 959af9b5,
server = 4, ruleIndex = 1
OCT 19 11:35:02 5/1 2682438 WCC-7: SSL session started !!!
OCT 19 11:35:02 5/1 2682439 WCC-7: sticky server is 5
OCT 19 11:35:02 5/1 2682440 WCC-7: Sticky failover=0, csdStatus=30, ret=FALSE
OCT 19 11:35:02 5/1 2682441 WCC-7: Hash insert for new server id accum = 25354271,
server = 5, ruleIndex = 1
OCT 19 11:35:03 5/1 2682442 WCC-7: SSL session started !!!
OCT 19 11:35:03 5/1 2682443 WCC-7: sticky server is 2
OCT 19 11:35:03 5/1 2682444 WCC-7: Sticky failover=0, csdStatus=30, ret=FALSE
OCT 19 11:35:04 5/1 2682445 WCC-7: SSL session started !!!
```

Issue debug mode commands that will display the sticky.

```
table status:
CSS11150(debug)# show sticky-table ssl-sticky collision
SSL Sticky List on Slot 5, subslot 1:
```

Entries for page 1.

```
Entry Hash Rule Rule Srv Srv Time(Sec) Hit Col Elem Inact
Number Value Indx State Indx State Elapsed Cnt Cnt Type Cfg(Min)
-----
```

Total number of entries found is 0.

```
CSS11150(debug)# show sticky-table ssl-sticky hash 0
Error found in hash input: 0.
```

```
CSS11150(debug)# show sticky-table ssl-sticky sid 0
Error found in SSL SID input
show sticky-table ssl-sticky rule
```

```
CS50(debug)# show sticky-table ssl-sticky rule 1
SSL Sticky List on Slot 5, subslot 1:
Rule index 1 SSL v3 hits: 525, v2 hits: 0.
```

```
Entry Hash Rule Rule Srv Srv Time(Sec) Hit Col Elem Inact
Number Value Indx State Indx State Elapsed Cnt Cnt Type Cfg(Min)
-----
```

```
1 e61346a4 1 ACT 6 ALIVE 354 1 0 SSL 0
2 f17ada8c 1 ACT 6 ALIVE 352 1 0 SSL 0
3 8fd8665d 1 ACT 6 ALIVE 350 1 0 SSL 0
4 a6060363 1 ACT 6 ALIVE 346 1 0 SSL 0
5 5dc12549 1 ACT 5 ALIVE 344 1 0 SSL 0
6 cd032afc 1 ACT 4 ALIVE 344 1 0 SSL 0
7 a12335f4 1 ACT 6 ALIVE 344 1 0 SSL 0
8 78e1ee82 1 ACT 6 ALIVE 343 1 0 SSL 0
9 1511bfd9 1 ACT 5 ALIVE 342 1 0 SSL 0
10 a77f05b9 1 ACT 4 ALIVE 342 1 0 SSL 0
11 a1fdf2ae 1 ACT 6 ALIVE 341 1 0 SSL 0
12 36f1dfae 1 ACT 4 ALIVE 340 1 0 SSL 0
13 997e35eb 1 ACT 5 ALIVE 340 1 0 SSL 0
14 5ca65a6b 1 ACT 4 ALIVE 339 1 0 SSL 0
15 c3d224a0 1 ACT 4 ALIVE 339 1 0 SSL 0
16 49e88e7e 1 ACT 6 ALIVE 338 1 0 SSL 0
17 4fcd3ed3 1 ACT 5 ALIVE 337 1 0 SSL 0
18 4278407 1 ACT 4 ALIVE 336 1 0 SSL 0
19 a227927d 1 ACT 4 ALIVE 336 1 0 SSL 0
20 9477a913 1 ACT 6 ALIVE 335 1 0 SSL 0
21 65e08044 1 ACT 5 ALIVE 335 1 0 SSL 0
22 142360f7 1 ACT 4 ALIVE 334 1 0 SSL 0
23 d36bc33a 1 ACT 5 ALIVE 334 1 0 SSL 0
24 addd1162 1 ACT 4 ALIVE 333 1 0 SSL 0
25 7219bec6 1 ACT 6 ALIVE 333 1 0 SSL 0
26 84e143a 1 ACT 5 ALIVE 332 1 0 SSL 0
27 59495466 1 ACT 5 ALIVE 332 1 0 SSL 0
28 601a7425 1 ACT 4 ALIVE 332 1 0 SSL 0
29 e6f4c380 1 ACT 5 ALIVE 331 1 0 SSL 0
30 d0a323d5 1 ACT 6 ALIVE 331 1 0 SSL 0
31 8940fdd2 1 ACT 5 ALIVE 330 1 0 SSL 0
32 f89a7fd1 1 ACT 5 ALIVE 330 1 0 SSL 0
```

```
33 30162b75 1 ACT 4 ALIVE 329 1 0 SSL 0
34 71ccale1 1 ACT 6 ALIVE 329 1 0 SSL 0
35 39dfc2c 1 ACT 5 ALIVE 329 1 0 SSL 0
36 1f9d2461 1 ACT 4 ALIVE 328 1 0 SSL 0
37 e4d06631 1 ACT 6 ALIVE 328 1 0 SSL 0
38 17c513db 1 ACT 4 ALIVE 327 1 0 SSL 0
39 2439b165 1 ACT 6 ALIVE 327 1 0 SSL 0
40 38aaa0c2 1 ACT 6 ALIVE 327 1 0 SSL 0
41 26bcb0a8 1 ACT 6 ALIVE 326 1 0 SSL 0
42 b29180e6 1 ACT 5 ALIVE 326 1 0 SSL 0
43 31fd8307 1 ACT 4 ALIVE 325 1 0 SSL 0
44 8edaa23f 1 ACT 6 ALIVE 324 1 0 SSL 0
45 81e639ce 1 ACT 4 ALIVE 324 1 0 SSL 0
46 ec250204 1 ACT 6 ALIVE 324 1 0 SSL 0
47 26fb7401 1 ACT 6 ALIVE 323 1 0 SSL 0
48 21a06660 1 ACT 6 ALIVE 322 1 0 SSL 0
49 42e4e314 1 ACT 5 ALIVE 322 1 0 SSL 0
50 ee9355a2 1 ACT 4 ALIVE 321 1 0 SSL 0
51 10367156 1 ACT 4 ALIVE 321 1 0 SSL 0
52 4925f113 1 ACT 6 ALIVE 321 1 0 SSL 0
53 ce4ac9e8 1 ACT 6 ALIVE 320 1 0 SSL 0
54 e0f9f062 1 ACT 6 ALIVE 320 1 0 SSL 0
55 b4aea7fb 1 ACT 4 ALIVE 319 1 0 SSL 0
56 368f8b85 1 ACT 5 ALIVE 319 1 0 SSL 0
57 49783633 1 ACT 6 ALIVE 319 1 0 SSL 0
58 fb69a234 1 ACT 4 ALIVE 318 1 0 SSL 0
59 7dbf0998 1 ACT 6 ALIVE 318 1 0 SSL 0
60 2dl5ba72 1 ACT 6 ALIVE 318 1 0 SSL 0
61 ae789c99 1 ACT 4 ALIVE 318 1 0 SSL 0
62 1c858091 1 ACT 5 ALIVE 317 1 0 SSL 0
63 5b62edb4 1 ACT 6 ALIVE 317 1 0 SSL 0
64 e1839156 1 ACT 4 ALIVE 316 1 0 SSL 0
65 40ec0a88 1 ACT 4 ALIVE 316 1 0 SSL 0
66 2a2221a8 1 ACT 6 ALIVE 316 1 0 SSL 0
67 f2ab04c5 1 ACT 6 ALIVE 315 1 0 SSL 0
68 5c8bacfe 1 ACT 4 ALIVE 315 1 0 SSL 0
69 3b700308 1 ACT 6 ALIVE 314 1 0 SSL 0
70 c05d25ed 1 ACT 6 ALIVE 313 1 0 SSL 0
71 cabbal22 1 ACT 4 ALIVE 313 1 0 SSL 0
72 98e064b9 1 ACT 6 ALIVE 312 1 0 SSL 0
73 b7e1fffb4 1 ACT 4 ALIVE 311 1 0 SSL 0
74 fa185595 1 ACT 6 ALIVE 311 1 0 SSL 0
75 f0427fbc 1 ACT 5 ALIVE 311 1 0 SSL 0
76 4d4510ce 1 ACT 6 ALIVE 310 1 0 SSL 0
77 610cfb7 1 ACT 4 ALIVE 310 1 0 SSL 0
78 42dlbbd0 1 ACT 4 ALIVE 310 1 0 SSL 0
79 d6dd743a 1 ACT 6 ALIVE 309 1 0 SSL 0
80 37ba95c6 1 ACT 6 ALIVE 308 1 0 SSL 0
81 d3286f00 1 ACT 4 ALIVE 308 1 0 SSL 0
82 6ccac424 1 ACT 6 ALIVE 308 1 0 SSL 0
83 a213bf2 1 ACT 5 ALIVE 307 1 0 SSL 0
84 75302f8a 1 ACT 4 ALIVE 307 1 0 SSL 0
85 eda8e833 1 ACT 4 ALIVE 307 1 0 SSL 0
86 38c3b816 1 ACT 6 ALIVE 307 1 0 SSL 0
87 44be9577 1 ACT 6 ALIVE 306 1 0 SSL 0
88 237f4c9b 1 ACT 4 ALIVE 305 1 0 SSL 0
89 9278fb56 1 ACT 6 ALIVE 305 1 0 SSL 0
90 e9448c4c 1 ACT 5 ALIVE 305 1 0 SSL 0
91 627239 1 ACT 6 ALIVE 304 1 0 SSL 0
92 ade7902f 1 ACT 4 ALIVE 304 1 0 SSL 0
93 94f7d05a 1 ACT 4 ALIVE 304 1 0 SSL 0
94 fe165f0b 1 ACT 6 ALIVE 303 1 0 SSL 0
95 79c33908 1 ACT 6 ALIVE 303 1 0 SSL 0
96 f3524248 1 ACT 4 ALIVE 302 1 0 SSL 0
```

```

3.      97 acff59b8 1 ACT 5 ALIVE 302 1 0 SSL 0
        98 f987d512 1 ACT 6 ALIVE 302 1 0 SSL 0
        99 4324951c 1 ACT 4 ALIVE 301 1 0 SSL 0
       100 964a48f1 1 ACT 4 ALIVE 301 1 0 SSL 0

```

Total number of entries found is 528.

```

CS50(debug)# show sticky-table ssl-sticky rule 3
SSL Sticky List on Slot 5, subslot 1:
Rule index 3 SSL v3 hits: 319, v2 hits: 0.

```

```

Entry Hash Rule Rule Srv Srv Time(Sec) Hit Col Elem Inact
Number Value Indx State Indx State Elapsed Cnt Cnt Type Cfg(Min)
-----
1 ce3976e2 3 ACT 3 ALIVE 259 9 0 SSL 0
2 1239113a 3 ACT 2 ALIVE 210 163 0 SSL 0
3 3ffac770 3 ACT 3 ALIVE 115 129 0 SSL 0
4 3b5c75ee 3 ACT 2 ALIVE 48 18 0 SSL 0

```

Total number of entries found is 4.

## Workarounds

For the server side, make sure that the SSL server allows re-use of SIDs (caching).

For the client side, use the HTTP to HTTPS redirect configuration on the CSS for pointing to a content rule with one server in each rule.

```

service Redirect_WW1
type redirect
keepalive type none
no prepend-http
ip address 192.168.222.100
redirect-string https://ww1.test.net/reporter/
active

service Redirect_WW2
type redirect
keepalive type none
no prepend-http
ip address 192.168.222.101
redirect-string https://ww2.test.net/reporter/
active

service SSL_Dummy
ip address 192.168.255.1
active

service SSL1
ip address 192.168.222.34
keepalive uri "/"
keepalive type script ap-kal-ssl "192.168.222.34"
keepalive port 443
active

service SSL2
ip address 192.168.222.50
keepalive type script ap-kal-ssl "192.168.222.50"
keepalive port 443
keepalive uri "/"

```

```
active

content redirect
vip address 192.168.222.102
protocol tcp
port 80
url "/*"
add service Redirect_WW2
add service SSL_Dummy
add service Redirect_WW1
active

content vipWW1
vip address 192.168.222.90
add service SSL1
protocol tcp
port 443
url "/*"
application ssl
active

content vipWW2
vip address 192.168.222.91
add service SSL2
protocol tcp
port 443
url "/*"
application ssl
active
```

The redirect services will only be used when all other services are down, which is why you need to create a dummy service and assign it an IP address that is never reachable. This will keep the service down and the redirect services will be used. You need to make sure that the redirect IP addresses are accessible to the outside world. You need to make sure that the upstream DNS server also has the entries for WW1.test.net and WW2.test.net added to them.

The client comes in, hits the redirect rule, and is load-balanced between the two DNS entries (the URLs). Each one of these entries will issue a 302 redirect to the client to an address which will be either content rule vipWW1 or vipWW2. In each of these rules, there is only one web server so the client cannot get unstuck and sent to another server if it gets re-loadbalanced.

The long term solution to this and most SSL issues is the Cisco Secure Content Accelerator (SCA) (formerly SonicWall). For more information on SCA, refer to SCA 11000 Series Secure Content Accelerators Technical Support.

---

## Related Information

- [Technical Support – Cisco Systems](#)

---

All contents are Copyright © 1992–2006 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

---

Updated: Sep 29, 2006

Document ID: 23643

---