C8300シリーズでのFQDN ACLパターンマッチングを使用したZBFWの設定

内容

はじめに

このドキュメントでは、C8300プラットフォームで自律モードのFQDN ACLパターンマッチングを使用してZBFWを設定する手順について説明します。

前提条件

要件

次の項目に関する専門知識があることが推奨されます。

- ゾーンベースポリシーファイアウォール(ZBFW)
- 仮想ルーティングおよび転送(VRF)
- ネットワーク アドレス変換（NAT）

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- C8300-2N2S-6T 17.12.02

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

ゾーンベースポリシーファイアウォール(ZBFW)は、Cisco IOS®およびCisco IOS XEデバイス上でファイアウォールを設定するための高度な方法であり、ネットワーク内にセキュリティゾーンを作成できます。

ZBFWにより、管理者はインターフェイスをゾーンにグループ化し、ゾーン間を移動するトラフィックにファイアウォールポリシーを適用できます。

FQDN ACL(完全修飾ドメイン名(FQDN)アクセスコントロールリスト)は、CiscoルータのZBFWとともに使用され、管理者がIPアドレスだけでなくドメイン名に基づいてトラフィックを照合するファイアウォールルールを作成できるようにします。

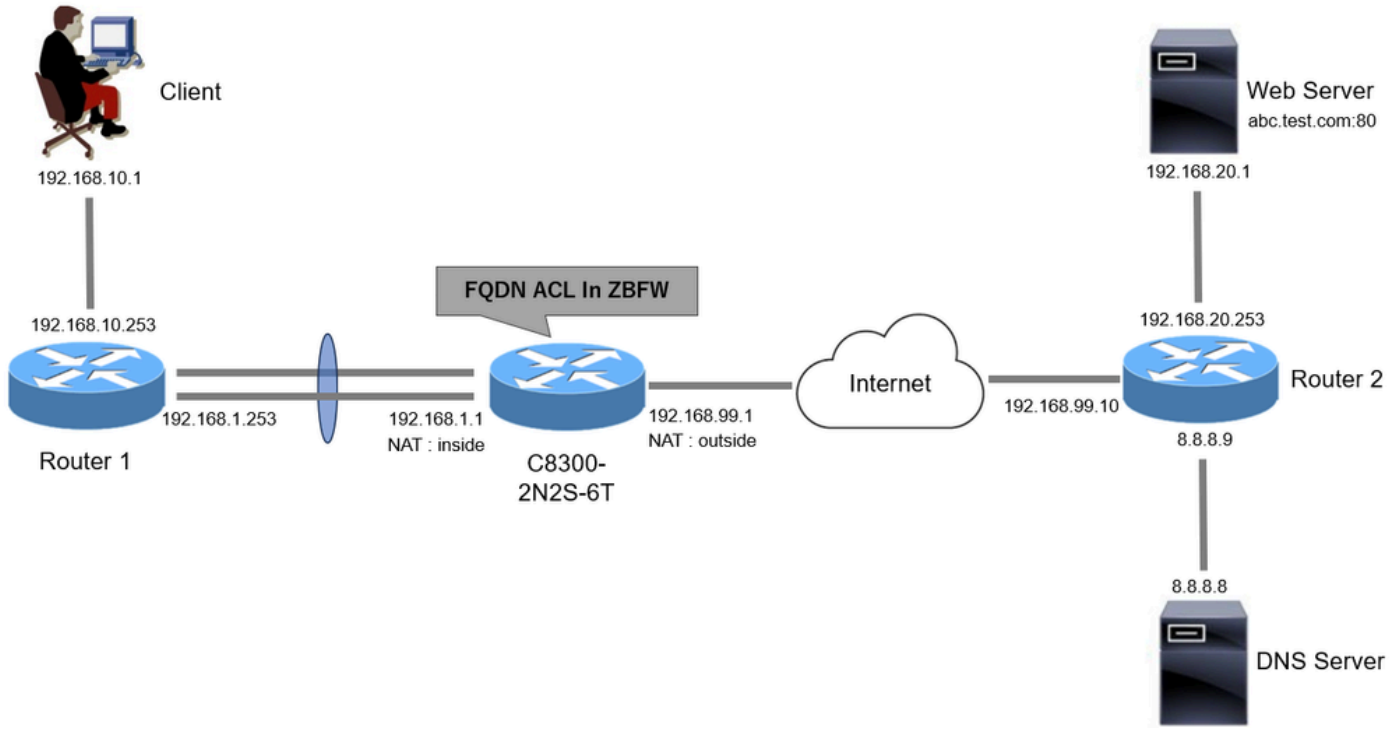
この機能は、サービスに関連付けられているIPアドレスが頻繁に変更される可能性がある、AWSやAzureなどのプラットフォームでホストされているサービスを扱う場合に特に便利です。

アクセスコントロールポリシーの管理を簡素化し、ネットワーク内のセキュリティ設定の柔軟性を向上させます。

設定

ネットワーク図

このドキュメントでは、この図に基づくZBFWの設定と検証を紹介します。 これは、BlackJumboDogをDNSサーバとして使用するシミュレーション環境です。



ネットワーク図

コンフィギュレーション

これは、クライアントからWebサーバへの通信を許可する設定です。

ステップ1: (オプション) VRFの設定

VRF(Virtual Routing and Forwarding)機能を使用すると、単一のルータ内に複数の独立したルーティングテーブルを作成して管理できます。この例では、WebVRFという名前のVRFを作成し、関連する通信のルーティングを実行します。

```
vrf definition WebVRF
rd 65010:10
!
address-family ipv4
route-target export 65010:10
route-target import 65010:10
exit-address-family
!
address-family ipv6
route-target export 65010:10
route-target import 65010:10
exit-address-family

ip route vrf WebVRF 8.8.8.8 255.255.255.255 GigabitEthernet0/0/3 192.168.99.10
ip route vrf WebVRF 192.168.10.0 255.255.255.0 Port-channel1.2001 192.168.1.253
ip route vrf WebVRF 192.168.20.0 255.255.255.0 GigabitEthernet0/0/3 192.168.99.10
```

ステップ 2：インターフェイスの設定

ゾーンメンバー、VRF、NAT、InsideおよびOutsideインターフェイスのIPアドレスなどの基本情報を設定します。

```
interface GigabitEthernet0/0/1
no ip address
negotiation auto
lacp rate fast
channel-group 1 mode active

interface GigabitEthernet0/0/2
no ip address
negotiation auto
lacp rate fast
channel-group 1 mode active

interface Port-channel1
no ip address
no negotiation auto

interface Port-channel1.2001
encapsulation dot1Q 2001
vrf forwarding WebVRF
ip address 192.168.1.1 255.255.255.0
ip broadcast-address 192.168.1.255
no ip redirects
no ip proxy-arp
ip nat inside
zone-member security zone_client

interface GigabitEthernet0/0/3
vrf forwarding WebVRF
ip address 192.168.99.1 255.255.255.0
ip nat outside
zone-member security zone_internet
speed 1000
no negotiation auto
```

ステップ3: (オプション) NATの設定

InsideおよびOutsideインターフェイス用にNATを設定します。 この例では、クライアントからの送信元IPアドレス(192.168.10.1)が192.168.99.100に変換されます。

```
ip access-list standard nat_source
10 permit 192.168.10.0 0.0.0.255

ip nat pool natpool 192.168.99.100 192.168.99.100 prefix-length 24
ip nat inside source list nat_source pool natpool vrf WebVRF overload
```

ステップ 4：FQDN ACLの設定

ターゲットトラフィックに一致するようにFQDN ACLを設定します。この例では、FQDNオブジェクトグループのパターンマッチングでワイルドカード'*'を使用して、宛先FQDNに一致させます。

```
object-group network src_net
192.168.10.0 255.255.255.0

object-group fqdn dst_test_fqdn
pattern .*\.test\.com

object-group network dst_dns
host 8.8.8.8

ip access-list extended Client-WebServer
1 permit ip object-group src_net object-group dst_dns
5 permit ip object-group src_net fqdn-group dst_test_fqdn
```

ステップ 5：ZBFWの設定

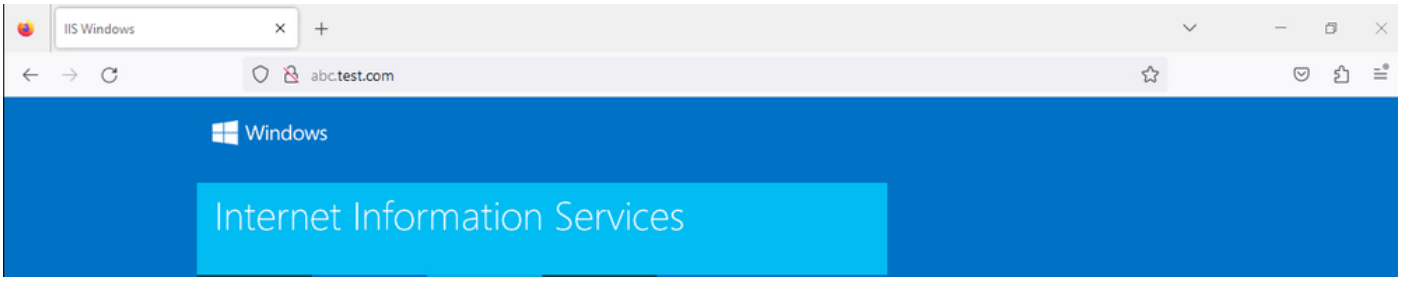
ZBFWのゾーン、クラスマップ、ポリシーマップを設定します。 この例では、パラメータマップを使用して、トラフィックがZBFWによって許可されたときに口グが生成されます。

```
zone security zone_client
zone security zone_internet

parameter-map type inspect inspect_log
audit-trail on

class-map type inspect match-any Client-WebServer-Class
match access-group name Client-WebServer

policy-map type inspect Client-WebServer-Policy
class type inspect Client-WebServer-Class
inspect inspect_log
class class-default
drop log

zone-pair security Client-WebServer-Pair source zone_client destination zone_internet
service-policy type inspect Client-WebServer-Policy
```

確認

ステップ 1：クライアントからのHTTP接続の開始

クライアントからWEBサーバへのHTTP通信が正常であることを確認します。



HTTP接続

ステップ 2：IPキャッシュの確認

show platform hardware qfp active feature dns-snoop-agent datapath ip-cache allコマンドを実行して、ターゲットFQDNに対するIPキャッシュがC8300-2N2S-6Tで生成されていることを確認します。

<#root>

02A7382#

show platform hardware qfp active feature dns-snoop-agent datapath ip-cache all

```
IP Address Client(s) Expire RegexId Dirty VRF ID Match
-----------------------------------------------------------------------------------------------------------
192.168.20.1 0x1 117 0xdbccd400 0x00 0x0 .*\.test\.com
```

ステップ 3：ZBFWログの確認

IPアドレス(192.168.20.1)がFQDN(.*\.test\.com)と一致していることを確認し、ステップ1のHTTP通信がZBFWによって許可されていることを確認します。

*Mar 7 11:08:23.018: %IOSXE-6-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:003 TS:00000551336606461468 %FW-6-SESS_AUDIT_TRAIL_START

*Mar 7 11:08:24.566: %IOSXE-6-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:002 TS:00000551338150591101 %FW-6-SESS_AUDIT_TRAIL: (target:

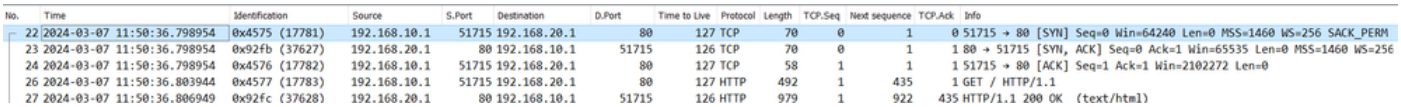
ステップ 4：パケットキャプチャの確認

ターゲットFQDNのDNS解決と、クライアントとWEBサーバ間のHTTP接続が成功したことを確認します。

内部でのパケットキャプチャ：

No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.Seq	Next sequence	TCP.Ack	Info
15	2024-03-07 11:50:36.775945	0x0511 (1297)	192.168.10.1	64078	8.8.8.8	53	127	DNS	76				Standard query 0xa505 A abc.test.com
18	2024-03-07 11:50:36.782949	0xe036 (57398)	8.8.8.8	53	192.168.10.1	64078	126	DNS	92				Standard query response 0xa505 A abc.test.com A 192.168.20.1

内部の*DNS*パケット



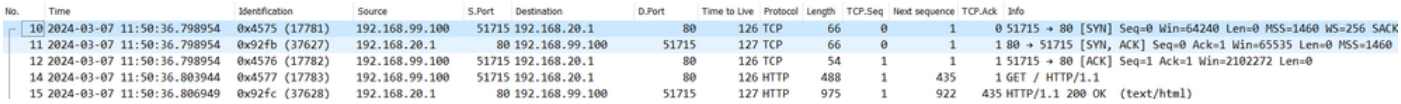
No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.Seq	Next sequence	TCP.Ack	Info
22	2024-03-07 11:50:36.798954	0x4575 (17781)	192.168.10.1	51715	192.168.20.1	80	127	TCP	70	0	1	0	51715 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
23	2024-03-07 11:50:36.798954	0x92fb (37627)	192.168.20.1	80	192.168.10.1	51715	126	TCP	70	0	1	1	80 → 51715 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256
24	2024-03-07 11:50:36.798954	0x4576 (17782)	192.168.10.1	51715	192.168.20.1	80	127	TCP	58	1	1	1	51715 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
26	2024-03-07 11:50:36.803944	0x4577 (17783)	192.168.10.1	51715	192.168.20.1	80	127	HTTP	492	1	435	1	GET / HTTP/1.1
27	2024-03-07 11:50:36.806949	0x92fc (37628)	192.168.20.1	80	192.168.10.1	51715	126	HTTP	979	1	922	435	HTTP/1.1 200 OK (text/html)

内部の*HTTP*パケット

オンサイドでのパケットキャプチャ(192.168.10.1は192.168.19.100に対してNATです):

No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.Seq	Next sequence	TCP.Ack	Info
3	2024-03-07 11:50:36.775945	0x0511 (1297)	192.168.99.100	64078	8.8.8.8	53	126	DNS	72				Standard query 0xa505 A abc.test.com
6	2024-03-07 11:50:36.782949	0xe036 (57398)	8.8.8.8	53	192.168.99.100	64078	127	DNS	88				Standard query response 0xa505 A abc.test.com A 192.168.20.1

外部の*DNS*パケット



No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.Seq	Next sequence	TCP.Ack	Info
10	2024-03-07 11:50:36.798954	0x4575 (17781)	192.168.99.100	51715	192.168.20.1	80	126	TCP	66	0	1	0	51715 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
11	2024-03-07 11:50:36.798954	0x92fb (37627)	192.168.20.1	80	192.168.99.100	51715	127	TCP	66	0	1	1	80 → 51715 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
12	2024-03-07 11:50:36.798954	0x4576 (17782)	192.168.99.100	51715	192.168.20.1	80	126	TCP	54	1	1	1	51715 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
14	2024-03-07 11:50:36.803944	0x4577 (17783)	192.168.99.100	51715	192.168.20.1	80	126	HTTP	488	1	435	1	GET / HTTP/1.1
15	2024-03-07 11:50:36.806949	0x92fc (37628)	192.168.20.1	80	192.168.99.100	51715	127	HTTP	975	1	922	435	HTTP/1.1 200 OK (text/html)

外部の*HTTP*パケット

トラブルシュート

FQDN ACLパターンマッチングを使用したZBFWに関連する通信の問題のトラブルシューティングを行うには、問題時にログを収集して、Cisco TACに提供できます。トラブルシューティングのログは、問題の性質によって異なることに注意してください。

収集するログの例：

```
!!!! before reproduction
!! Confirm the IP cache
show platform hardware qfp active feature dns-snoop-agent datapath ip-cache all

!! Enable packet-trace
debug platform packet-trace packet 8192 fia-trace
debug platform packet-trace copy packet both
debug platform condition ipv4 access-list Client-WebServer both
debug platform condition feature fw dataplane submode all level verbose

!! Enable debug-level system logs and ZBFW debug logs
debug platform packet-trace drop
debug acl cca event
debug acl cca error
debug ip domain detail
!! Start to debug
debug platform condition start

!! Enable packet capture on the target interface (both sides) and start the capture
monitor capture CAPIN interface Port-channel1.2001 both
monitor capture CAPIN match ipv4 any any
monitor capture CAPIN buffer size 32
monitor capture CAPIN start

monitor capture CAPOUT interface g0/0/3 both
monitor capture CAPOUT match ipv4 any any
monitor capture CAPOUT buffer size 32
monitor capture CAPOUT start
```

!! (Optional) Clear the DNS cache on the client
ipconfig/flushdns
ipconfig /displaydns

!! Run the show command before reproduction
show platform hardware qfp active feature firewall drop all
show policy-map type inspect zone-pair Client-WebServer-Pair sessions
show platform packet-trace statistics
show platform packet-trace summary
show logging process cpp_cp internal start last boot
show platform hardware qfp active feature dns-snoop-agent client hw-pattern-list
show platform hardware qfp active feature dns-snoop-agent client info
show platform hardware qfp active feature dns-snoop-agent datapath stats
show ip dns-snoop all
show platform hardware qfp active feature dns-snoop-agent datapath ip-cache all
show platform software access-list F0 summary

!!!! Reproduce the issue - start

!! During the reproductionof the issue, run show commands at every 10 seconds
!! Skip show ip dns-snoop all command if it is not supported on the specific router
show ip dns-snoop all
show platform hardware qfp active feature dns-snoop-agent datapath ip-cache all

!!!! After reproduction
!! Stop the debugging logs and packet capture
debug platform condition stop
monitor capture CAPIN stop
monitor capture CAPOUT stop

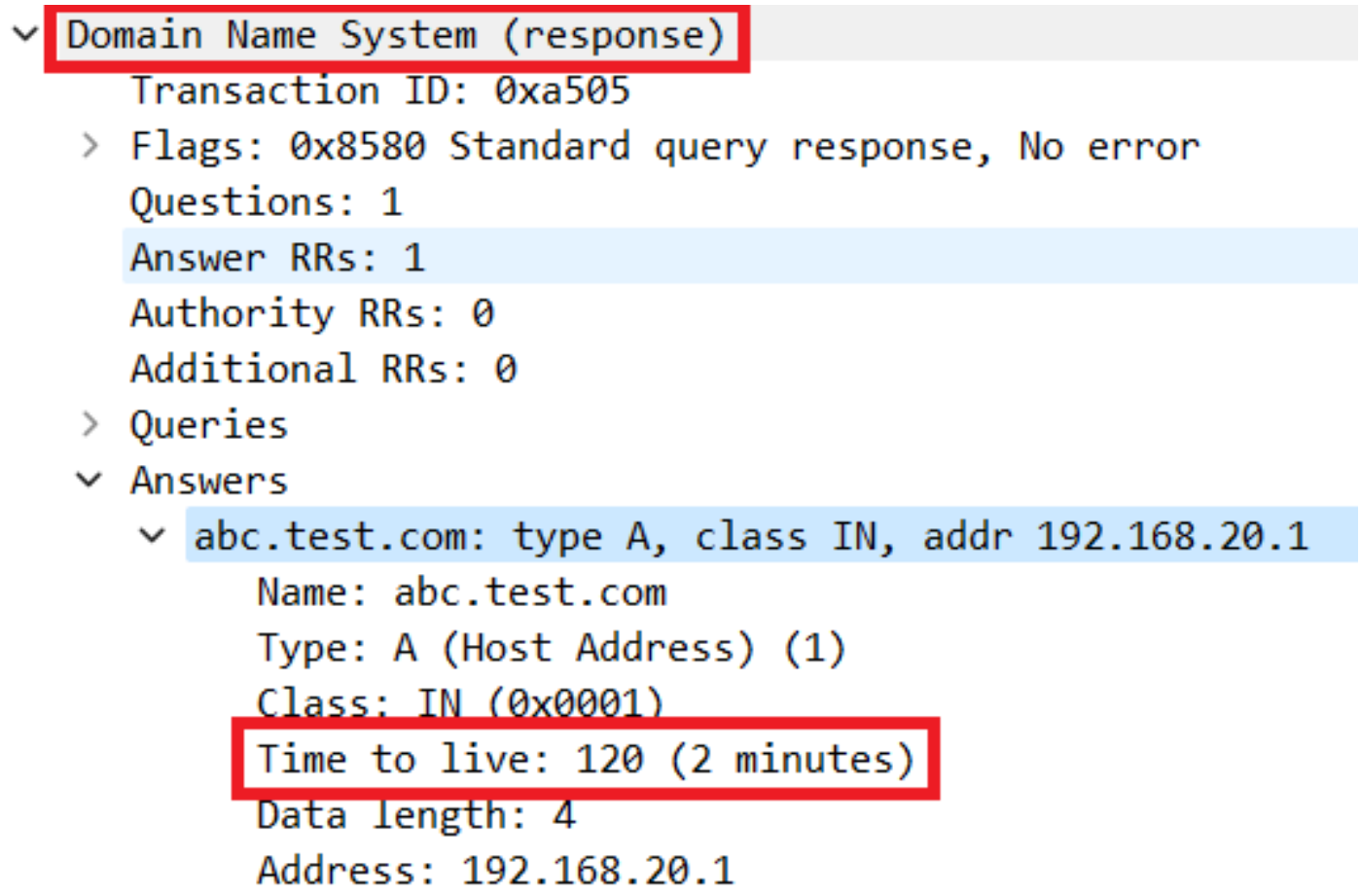
!! Run the show commands
show platform hardware qfp active feature firewall drop all
show policy-map type inspect zone-pair Client-WebServer-Pair sessions
show platform packet-trace statistics
show platform packet-trace summary
show logging process cpp_cp internal start last boot
show platform hardware qfp active feature dns-snoop-agent client hw-pattern-list
show platform hardware qfp active feature dns-snoop-agent client info
show platform hardware qfp active feature dns-snoop-agent datapath stats
show ip dns-snoop all
show platform hardware qfp active feature dns-snoop-agent datapath ip-cache all
show platform software access-list F0 summary

show platform packet-trace packet all decode
show running-config

よく寄せられる質問（FAQ）

Q：ルータでIPキャッシュのタイムアウト値はどのように決定されるのですか。

A:IPキャッシュのタイムアウト値は、DNSサーバから返されるDNSパケットのTTL（存続可能時間）値によって決まります。この例では、120秒です。IPキャッシュがタイムアウトすると、自動的にルータから削除されます。パケットキャプチャの詳細を次に示します。



*DNS*解決のパケットの詳細

Q: DNSサーバがAレコードではなくCNAMEレコードを返す場合に問題はありませんか。

A：はい、問題ありません。DNSサーバからCNAMEレコードが返されると、DNS解決とHTTP通信は問題なく行われます。パケットキャプチャの詳細を次に示します。

No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.Seq	Next sequence	TCP.Ack	Info
350	2024-03-07 12:09:55.625959	0x0bc5 (3013)	192.168.10.1	63777	8.8.8.8	53	127	DNS	76				Standard query 0x6bd8 A abc.test.com
352	2024-03-07 12:09:55.629957	0xe4fe (58622)	8.8.8.8	53	192.168.10.1	63777	126	DNS	114				Standard query response 0x6bd8 A abc.test.com CNAME def.test.

内部の*DNS*パケット

```
Domain Name System (response)
    Transaction ID: 0x6bd8
  > Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  v Answers
    v abc.test.com: type CNAME, class IN, cname def.test.com
        Name: abc.test.com
        Type: CNAME (Canonical NAME for an alias) (5)
        Class: IN (0x0001)
        Time to live: 120 (2 minutes)
        Data length: 6
        CNAME: def.test.com
    v def.test.com: type A, class IN, addr 192.168.20.1
        Name: def.test.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 120 (2 minutes)
        Data length: 4
        Address: 192.168.20.1
```

*DNS*解決のパケットの詳細

No.	Time	Identification	Source	S.Port	Destination	D.Port	Time to Live	Protocol	Length	TCP.S	Next :	TCP.A	Info
356	2024-03-07 12:09:55.644955	0x4589 (17801)	192.168.10.1	51801	192.168.20.1	80	127	TCP	70	0	1	0	51801 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
357	2024-03-07 12:09:55.644955	0x9349 (37705)	192.168.20.1	80	192.168.10.1	51801	126	TCP	70	0	1	1	80 → 51801 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
358	2024-03-07 12:09:55.644955	0x458a (17802)	192.168.10.1	51801	192.168.20.1	80	127	TCP	58	1	1	1	51801 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
359	2024-03-07 12:09:55.645962	0x458b (17803)	192.168.10.1	51801	192.168.20.1	80	127	HTTP	492	1	435	1	GET / HTTP/1.1
362	2024-03-07 12:09:55.646954	0x934a (37706)	192.168.20.1	80	192.168.10.1	51801	126	HTTP	979	1	922	435	HTTP/1.1 200 OK (text/html)

内部の*HTTP*パケット

Q: C8300ルータで収集されたパケットキャプチャをFTPサーバに転送するコマンドは何ですか。

A:monitor capture <capture name> export bootflash:<capture name>.pcapおよびcopy bootflash:<capture name>.pcap ftp://<user>:<password>@<FTP IP Address>コマンドを使用して、パケットキャプチャをFTPサーバに転送します。次に、CAPINをFTPサーバに転送する例を示します。

<#root>

monitor capture CAPIN export bootflash:CAPIN.pcap

copy bootflash:CAPIN.pcap ftp://<user>:<password>@<FTP IP Address>

参考

ゾーンベースポリシーファイアウォール設計について

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。