Firepower Data Path Troubleshooting Phase 4: Access Control Policy

Contents

[Introduction](#)
[Troubleshooting the Access Control Policy (ACP) Phase](#)
[Check for Connection Events](#)
[Quick Mitigation Steps](#)
[Debugging the ACP](#)
[Example 1: Traffic Matches a Trust Rule](#)
[Example 2: Traffic Matching a Trust Rule is Blocked](#)
[Scenario 3: Traffic Blocked by Application Tag](#)
[Data to Provide to TAC](#)
[Next Step: Troubleshoot the SSL Policy Layer](#)

Introduction

This article is part of a series of articles which explain how to systematically troubleshoot the data path on Firepower systems to determine whether components of Firepower may be affecting traffic. Please refer to the Overview article for information about the architecture of Firepower platforms and links to the other Data Path Troubleshooting articles.

This article covers the fourth stage of the Firepower data path troubleshooting, the Access Control Policy (ACP). This information is applicable to all of the currently supported Firepower platforms and versions.

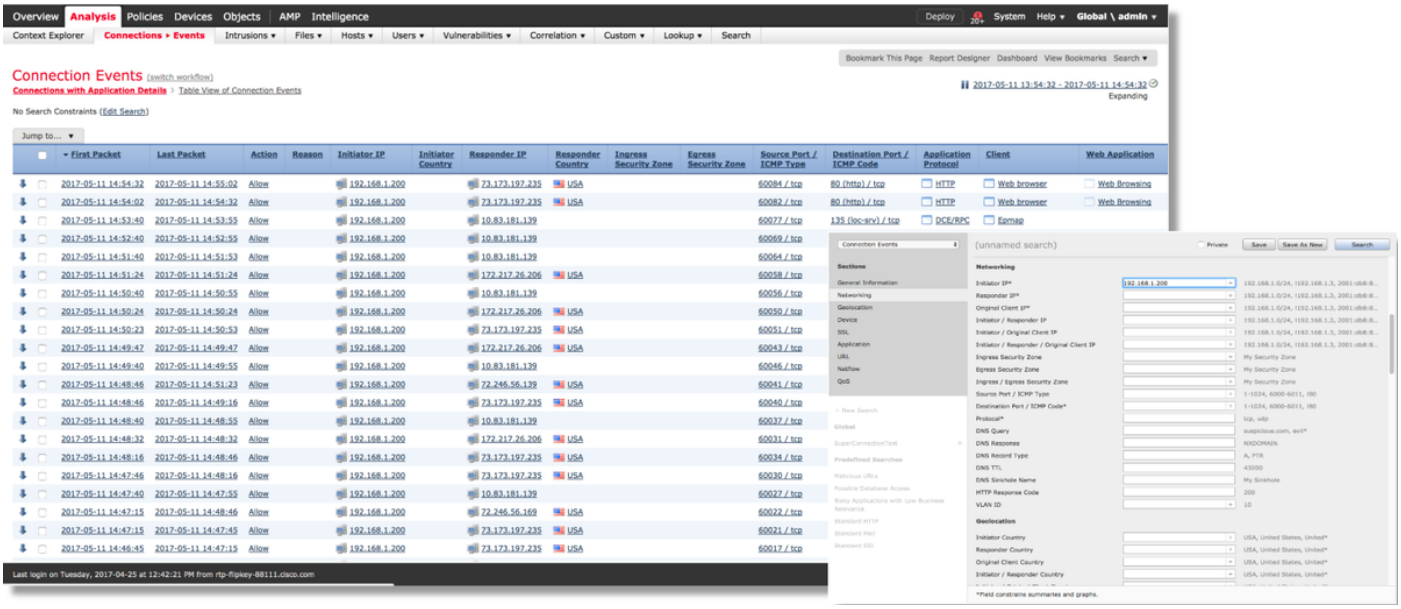
Troubleshooting the Access Control Policy (ACP) Phase

Generally speaking, determining which ACP rule a flow is matching should be pretty straight-forward. The Connection Events can be reviewed to see which rule/action is being enforced. If that does not clearly show what the ACP is doing with the traffic, debugging can be performed on the Firepower Command Line Interface (CLI).

Check for Connection Events

After getting an idea of the ingress and egress interface the traffic should be matching as well as the flow information, the first step to identifying whether Firepower is blocking the flow would be to check the Connection Events for the traffic in question. These can be viewed in the Firepower Management Center under **Analysis > Connections > Events.**

Note: Prior to checking Connection Events, ensure that logging is enabled in your ACP rules. Logging is configured in the "Logging" tab within each Access Control Policy rule as well as the Security Intelligence tab. Make sure the suspect rules are configured to send the logs to the "Event Viewer". This also applies to the default action.



By clicking on "Edit Search" and filtered by a unique source (Initiator) IP you can see the flows which were being detected by Firepower. The Action column shows "Allow" for this host's traffic.

If Firepower is intentionally blocking traffic, the Action would contain the word "Block". Clicking on "Table View of Connection Events" provides more data. The following fields in the Connection Events can be reviewed if the action is "Block":

- Reason

- Access Control Rule

Quick Mitigation Steps

In order to quickly mitigate an issue which is believed to be caused by the ACP rules, the following can be performed:

- Create a rule with the action of "Trust" or "Allow" for the traffic in question and place it at the very top of the ACP, or above all block rules.
- Temporarily disable any rules with an action containing the word "Block"
- If the Default Action is set to "Block All Traffic", temporarily switch it to "Network Discovery Only"

Note: These quick mitigations require policy changes which may not be possible in all environments. It is recommended to first try to use system support trace to determine which rule the traffic is matching before making policy changes.

Debugging the ACP

Further troubleshooting can be performed against the ACP operations via the **> system support firewall-engine-debug** CLI utility.

> **Note**: On the Firepower 9300 and 4100 platforms, the shell in question can be accessed via the following commands:
>
> # **connect module 1 console**
> Firepower-module1> **connect ftd**
> >
>
> For multi-instances, the logical device CLI can be accessed with the following commands.
> **# connect module 1 telnet**
> Firepower-module1> **connect ftd ftd1**
> Connecting to container ftd(ftd1) console... enter "exit" to return to Boot CLI
> >

The **system support firewall-engine-debug** utility has an entry for each packet being evaluated by the ACP. It shows the rule evaluation process taking place, along with why a rule is matched or not matched.

> **Note**: In version 6.2 and above, the **system support trace** tool can be run. It uses the same parameters but includes more details. Be sure to enter 'y' when prompted with "**Enable firewall-engine-debug too?"**.

Example 1: Traffic Matches a Trust Rule

In the example below, the establishment of an SSH session is evaluated using **system support firewall-engine-debug**.

This is the ACP which is running on the Firepower device.

#	Name	Source Zones	Dest Zones	Source Networks	Dest Networks	VLAN ...	Users	Applic...	Sourc...	Dest P...	URLs	ISE/S... Attrib...	Acti...
Mandatory - JG AC (all) (1-6)													
1	Trust ssh for host	Any	Any	192.168.0.7	Any	Any	Any	Any	Any	SSH	Any	Any	Trust
2	inspect	Any	Any	10.0.0.0/8	Any	Any	Any	Any	Any	Any	Any	Any	Allow
3	trust server backup	Any	Any	192.168.62.3	10.123.175.22	Any	Any	Any	Any	Any	Any	Any	Trust

The ACP has three rules.

1. The first rule is trusting any traffic from 192.168.0.7 with destination ports used by SSH.
2. The second rule inspects all traffic sourced from 10.0.0.0/8 in which the network criteria matches based on the XFF header data (as indicated by the icon next to the network object)
3. The third rule trusts all traffic from 192.168.62.3 to 10.123.175.22

In the troubleshooting scenario, an SSH connection from 192.168.62.3 to 10.123.175.22 is being analyzed.

The expectation is that the session matches AC rule 3 "trust server backup". The question is, how many packets should it take for this session to match this rule. Is all of the information needed in the first packet to determine the AC rule or multiple packets are required, and if that is so, how

many?

On the Firepower CLI, the following is entered to see what the ACP rule evaluation process.

```
>system support firewall-engine-debug

Please specify an IP protocol: tcp
Please specify a client IP address: 192.168.62.3
Please specify a client port:
Please specify a server IP address: 10.123.175.22
Please specify a server port: 22
Monitoring firewall engine debug messages
```

> **Tip**: It is best to fill out as many parameters as possible when running **firewall-engine-debug**, so that only the interesting debug messages are printed to screen.

In the debug output below, you see the first four packets of the session being evaluated.

SYN

SYN,ACK

ACK

First SSH Packet (client to server)

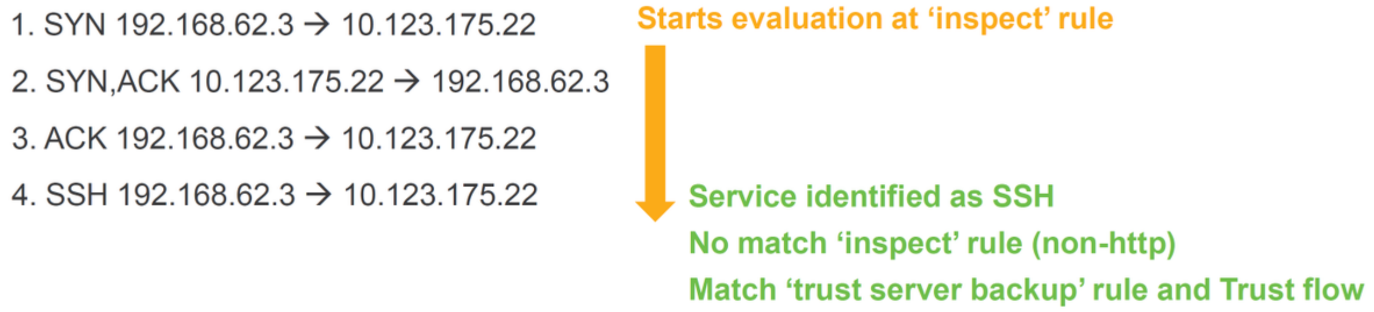
```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId

192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId

192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId

192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 846, payload -1, client 2000000846, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust
```

This is a chart further illustrating the debug logic.



For this flow, it takes 4 packets for the device to match the rule.

This is a detailed explanation of the debug output.

- The ACP evaluation process starts at the "inspect" rule because the "trust ssh for host" rule was not matched as the IP address did not match the requirement. This is a quick match because of all of the information needed to determine if this rule should match is present in the first packet (IPs and ports)
- It cannot be determined whether the traffic matches the "inspect" rule until the application is identified, since X-Forwarded-For (XFF) information is found in HTTP application traffic, the application isn't known yet, so this puts the session into a pending state for rule 2, pending application data.
- Once the application is identified in the fourth packet, the "inspect" rule results in a non-match, since the application is SSH, rather than HTTP
- The "trust server backup" rule is then matched, based on the IP addresses.

In summary, the connection takes 4 packets to match the session because it has to wait for the firewall to identify the application since rule 2 has an application constraint in it.

If rule 2 had only had source networks and it was not XFF, then this would have taken 1 packet to match the session.

You should always place layers 1-4 rules above all other rules in the policy when possible as these rules typically require 1 packet to make a decision. However, you may also notice that even with just layers 1-4 rules it may more than just 1 packet to match an AC rule, and the reason for this is URL/DNS security intelligence. If you have either of these enable, the firewall has to determine the application for all sessions being evaluated by the AC policy because it has to determine if they are HTTP or DNS. Then, it must determine if it should allow the session based on the blacklists.

Below is a truncated output of the **firewall-engine-debug** command, which has the relevant fields highlighted in red. Note the command used to obtain the name of the application which is identified.

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0,
inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId

[…omitted for brevity]

192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0,
inline sgt tag: untagged, ISE sgt id: 0, svc 846, payload -1, client 2000000846, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust


[! How to map service/application ID to name]
> expert
$ grep "^846[^0-9]" /var/sf/appid/odp/appMapping.data
846 SSH 32 0 0 ssh
```

Example 2: Traffic Matching a Trust Rule is Blocked

In some scenarios, traffic can be blocked despite matching a Trust rule in the ACP. The example below evaluates traffic with the same Access Control Policy and hosts.

```
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0,
inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Deleting session

[!Session was deleted because we hit a drop IPS rule and blacklisted the flow.
This happened before AC rule was matched (Intrusion policy before AC rule match dropped).
Firewall engine will re-evaluate from top of AC policy to find a rule for logging decision]

192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 0, id 0 and IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline
sgt tag: 0, ISE sgt id: 0, svc -1, payload -1, client -1, misc -1, user 9999997, icmpType 102, icmpCode 22
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 3, 'Trust ssh for host', src network and GEO
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust
```

Action	Reason	Initiator IP	Responder IP	Source Port / ICMP Type	Destination Port / ICMP Code	Application Protocol	Client	Intrusion Events	Access Control Policy	Access Control Rule
Block	Intrusion Block	192.168.62.3	10.123.175.22	55654 / tcp	22 (ssh) / tcp				JG AC (all)	trust server backup

As seen above, the **firewall-engine-debug** output shows that the traffic matches a "Trust", whereas the Connection Events show action of **Block** due to an Intrusion Policy rule (determined because the Reason column shows **Intrusion Block**).

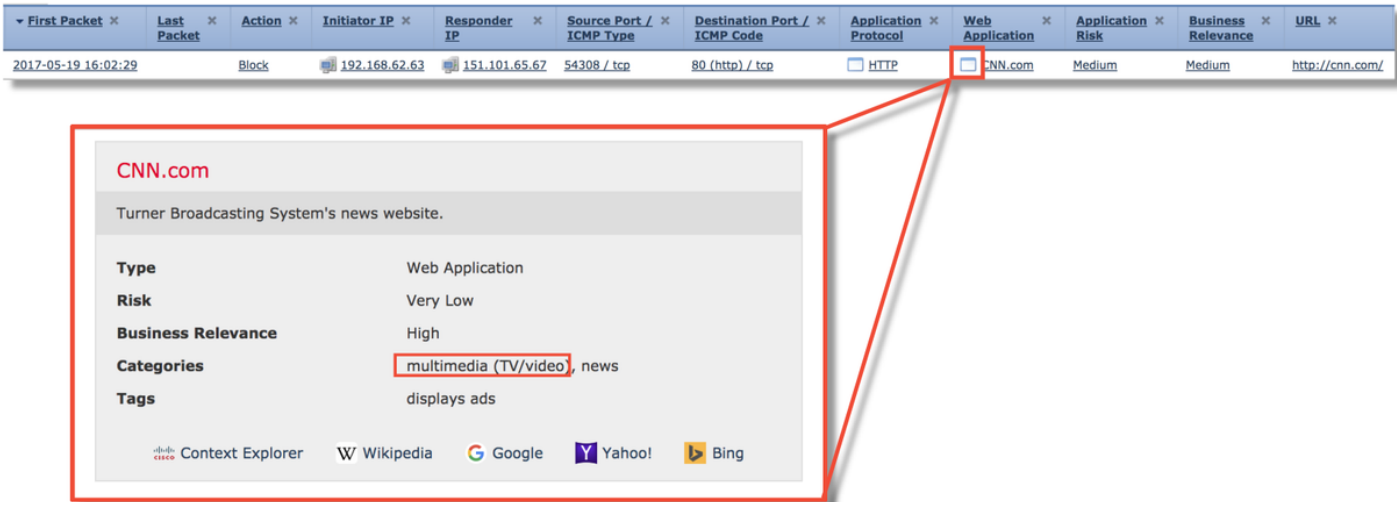
The reason this can occur is due to the **Intrusion Policy used before Access Control rule is determined** Setting in the **Advanced** tab on the ACP. Before the traffic could be Trusted per the rule action, the Intrusion Policy in question identifies a pattern match and drops the traffic. However, the ACP rule evaluation results in a match of the Trust rule, since the IP addresses did match the criteria of the "trust server backup" rule.

In order to have the traffic not undergo the Intrusion Policy inspection, the Trust rule can be placed above the "inspect" rule, which would be a best practice in either case. Since application identification is necessary for a match and non-match of the "inspect" rule, the **Intrusion Policy used before Access Control rule is determined** is used for traffic which gets evaluated by the same. Placing the "trust server backup" rule above the "inspect" rule causes the traffic to match the rule when the first packet is seen since the rule is based on IP address, which can be determined in the first packet. Therefore, the **Intrusion Policy used before Access Control rule is determined** doesn't need to be used.

Scenario 3: Traffic Blocked by Application Tag

In this scenario, users report that cnn.com is being blocked. However, there is no specific rule which blocks CNN. The Connection Events, in conjunction with **firewall-engine-debug** output, shows the reason for the block.

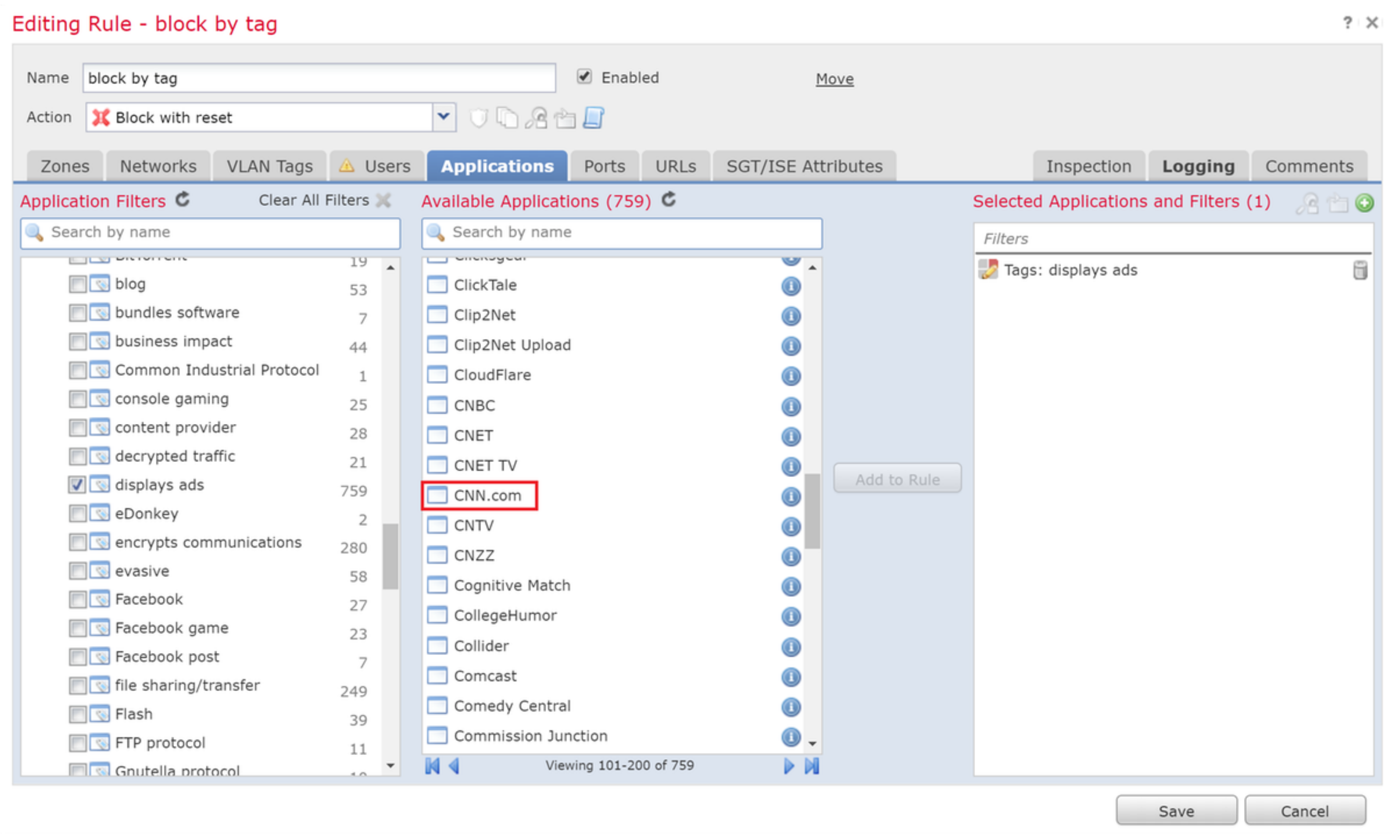
First, the Connection Events has an information box next to the application fields which shows information about the application as well as how Firepower categorizes said application.



With this information in mind, **firewall-engine-debug** is run. In the debug output, the traffic is blocked based on the application tag.

```
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 New session
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by tag', and SrcZone first with zones 1 -> 2, geo 0 -> 0,
vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by tag', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by tag', and SrcZone first with zones 1 -> 2, geo 0 -> 0,
vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by tag', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by tag', and SrcZone first with zones 1 -> 2, geo 0 -> 0,
vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by tag', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 URL SI: ShmDBLookupURL("http://cnn.com/") returned 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by tag', and SrcZone first with zones 1 -> 2, geo 0(0) -> 0,
vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 676, payload 1190, client 638, misc 0, user 9999997, url http://cnn.com/, xff
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 match rule order 4, 'block by tag', action Block
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 sending block response of 605 bytes
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Deleting session
```

Even though there isn't a rule which explicitly blocked http://cnn.com, the **displays ads** tagged is being blocked within the **Applications** tab of an ACP rule.



Data to Provide to TAC

Data	Instructions
Troubleshoot file from the Firepower device inspecting the traffic	http://www.cisco.com/c/en/us/support/docs/security/sourcefire-defense-center/11
system support firewall-engine-debug and **system-support-trace** output	See this article for instructions
Access Control Policy export	Navigate to **System > Tools > Import / Export**, select the Access Control Policy

> **Caution**: If the ACP contains an SSL Policy, remove the SSL Policy from the ACP before exporting to avoid disclosing sensitive PKI information

Next Step: Troubleshoot the SSL Policy Layer

If an SSL Policy is in use and the Access Control Policy troubleshooting did not reveal the issue, the next step would be to troubleshoot the SSL Policy.