DNS-Based Access Control Lists

Information About DNS-Based Access Control Lists

The DNS-based ACLs are used for wireless client devices. When using these devices, you can set pre-authentication ACLs on the Cisco Catalyst 9800 Series Wireless Controller to determine the data requests that are allowed or blocked.

To enable DNS-based ACLs on the controller , you need to configure the allowed URLs or denied URLs for the ACLs. The URLs need to be pre-configured on the ACL.

With DNS-based ACLs, the client when in registration phase is allowed to connect to the configured URLs. The controller is configured with the ACL name that is returned by the AAA server. If the ACL name is returned by the AAA server, then the ACL is applied to the client for web-redirection.

At the client authentication phase, the AAA server returns the pre-authentication ACL (url-redirect-acl, which is the attribute name given to the AAA server). The DNS snooping is performed on the AP for each client until the registration is complete and the client is in SUPPLICANT PROVISIONING state. When the ACL configured with the URLs is received on the controller , the CAPWAP payload is sent to the AP enabling DNS snooping for the URLs to be snooped.

With URL snooping in place, the AP learns the IP address of the resolved domain name in the DNS response. If the domain name matches the configured URL, then the DNS response is parsed for the IP address, and the IP address is sent to the controller as a CAPWAP payload. The controller adds the IP address to the allowed list of IP addresses and thus the client can access the URLs configured.

During pre-authentication or post-authentication, DNS ACL is applied to the client in the access point. If the client roams from one AP to another AP, the DNS learned IP addresses on the old AP is valid on the new AP as well.

Note URL filter needs to be attached to a policy profile in case of the local mode. In the flex mode, the URL filter is attached to the flex profile and it is not need to be attached to a policy profile.

Note DNS based URLs work with active DNS query from the client. Hence, for URL filtering, the DNS should be setup correctly.

Note URL filter takes precedence over punt or redirect ACL, and over custom or static pre-auth ACL.s

Defining ACLs

Extended ACLs are like standard ACLs but identifies the traffic more precisely.

The following CLI allows you to define ACLs by name or by an identification number.

```
Device(config)#ip access-list extended ?
<100-199> Extended IP access-list number
<2000-2699> Extended IP access-list number (expanded range)
WORD Access-list name
```

The following is the structure of a CLI ACL statement:

```
<sequence number> [permit/deny] <protocol> <address or any> eq <port number> <subnet>
<wildcard>
```

For example:

```
1 permit tcp any eq www 192.168.1.0 0.0.0.255
```

The sequence number specifies where to insert the Access Control list Entry (ACE) in the ACL order of ACEs. You can define your statements with sequences of 10, 20, 30, 40, and so on.

The controller GUI allows you to write a complete ACL going to the **Configuration** > **Security** > **ACL** page. You can view a list of protocols to pick from, and make changes to an existing ACL.

Applying ACLs

The following are the ways to apply ACLs:

- **Security ACL**: A security ACL defines the type of traffic that should be allowed through the device and that which should be blocked or dropped.

  A security ACL is applied:

  - **On SVI interfaces**: The ACL will only be evaluated against the traffic that is routed through the interface.

    ```
    Device(config)# interface Vlan<number>
    Device(config-if)# ip access-group myACL in/out
    ```

- **On a physical interface of the controller**: The ACL will be evaluated against all traffic that passes through the interface. Along with applying ACLs on SVI, this is another option for restricting traffic on the controller management plane.

  ```
  Device(config)#interface GigabitEthernet1
  Device(config-if)#ip access-group myACL in/out
  ```

- **In the wireless policy profile or WLAN**: This option includes several places where you can configure an ACL that will be applied to the wireless client traffic, in case of central switching or local switching of traffic. Such ACLs are only supported in the inbound direction.

- **On the AP**: In case of FlexConnect local switching, the ACL is configured and applied from the policy profile on the controller. This ACL has to be downloaded on to the AP through the Flex profile. ACLs must be downloaded to the AP before they can be applied. As an exception, fabric mode APs (in case of Software Defined Access) also use Flex ACLs even though the AP is not operating in Flex mode.

- **Punt ACL or Redirect ACL**: Punt ACL or redirect ACL refers to an ACL that specifies as to which traffic will be sent to the CPU (instead of its normal expected handling by the dataplane) for further processing. For example, the Central Web Authentication (CWA) redirect ACL defines as to which traffic is intercepted and redirected to the web login portal. The ACL does not define any traffic to be dropped or allowed, but follows the regular processing or forwarding rules, and what will be sent to the CPU for interception.

  A redirect ACL has an invisible last statement which is an implicit deny. This implicit deny is applied as a security access list entry (and therefore drops traffic that is not explicitly allowed through or sent to the CPU).

Types of URL Filters

The following are the two types of URL filters:

- **Standard**: Standard URL filters can be applied before client authentication (pre-auth) or after a successful client authentication (post-auth). Pre-auth filters are extremely useful in the case of external web authentication to allow access to the external login page, as well as, some internal websites before authentication takes place. Post-auth, they can work to block specific websites or allow only specific websites while all the rest is blocked by default. This type of URL filtering post-auth is better handled by using Cisco DNS Layer Security (formerly known as Umbrella) for more flexibility. The standard URL filters apply the same action (permit or deny) for the whole list of URLs. It is either all permit or all deny.

- **Enhanced**: Enhanced URL filters allow specification of a different action (deny or permit) for each URL inside the list and have per-URL hit counters.

In both types of URL filters, you can use a wildcard sub-domain such as **`*.cisco.com`**. URLfilters are standalone but always applied along with an IP-based ACL. A maximum of 20 URLs are supported in a given URL filter. Considering one URL can resolve multiple IP addresses, only up to 40 resolved IP addresses can be tracked for each client. Only DNS records are tracked by URL filters. The controller or APs do not track the resolved IP address of a URL if the DNS answer uses a CNAME alias record.

Note In a scenario where you have a URL filter of type POST and an ACL applied to a policy profile, traffic to the URL is blocked by the ACL if there are no permit statements regarding the URLs. This can occur if the URL filter is POST with permit statement and within the ACL there is no permit statement for the URLs. Therefore, we recommend that you create permit statements within the ACL, regarding the IP address of the URLs, instead of using the POST URL filter.

Restrictions on DNS-Based Access Control Lists

The restriction for DNS-based ACLs is as follows:

- Pre-authentication and Post-authentication filters are supported in local modes. Only Pre-authentication filter is supported in Flex (Fabric) mode.
- ACL override pushed from ISE is not supported.
- FlexConnect Local Switching with External Web authentication using URL filtering is not supported until Cisco IOS XE Gibraltar 16.12.x.
- Fully qualified domain name (FQDN) or DNS based ACLs are not supported on Cisco Wave 1 Access Points.
- The URL filter considers only the first 20 URLs, though you can add more.
- The URL filter employs regular regex patterns and permits wildcard characters only at the beginning or at the end of an URL.
- The URL ACLs are defined and added to the FlexConnect policy profile in which they associate with a WLAN. The URL ACL creation follows a similar mechanism as that of local mode URL ACLs.
- In FlexConnect mode, the URL domain ACL works only if they are connected to a FlexConnect policy profile.
- The ACL can be attached to a WLAN by associating a policy profile with a WLAN or local policies. However, you can override it using "url-redirect-acl".
- For the Cisco AV pair received from ISE, the policy that needs to be applied for a particular client is pushed as part of ADD MOBILE

  message.
- When an AP joins or when an existing URL ACL is modified and applied on FlexConnect profile, the ACL definition along with mapped URL filter list is pushed to the AP.
- The AP stores the URL ACL definition with mapped ACL name and snoops the DNS packets for learning the first IP address for each URL in the ACL. When the AP learns the IP addresses, it updates the controller of the URL and IP bindings. The controller records this information in the client database for future use.
- When a client roams to another AP during the pre-authentication state, the learned IP addresses are pushed to a new AP. Otherwise, these learned IP addresses are purged when a client moves to a post-authentication state or when the TTL for the learned IP address expires.

Flex Mode

Applying URL Filter List to Flex Profile

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless profile flex** *default-flex-profile* **Example:** `Device(config)# wireless profile flex default-flex-profile`	Creates a new flex policy. The default flex profile name is *default-flex-profile*.
Step 3	**acl-policy** *acl policy name* **Example:** `Device(config-wireless-flex-profile)# acl-policy acl_name`	Configures ACL policy.
Step 4	**urlfilter list** *name* **Example:** `Device(config-wireless-flex-profile-acl)# urlfilter list urllist_flex_preauth`	Applies the URL list to the Flex profile.
Step 5	**end** **Example:** `Device(config-wireless-flex-profile-acl)# end`	Returns to privileged EXEC mode.

Configuring ISE for Central Web Authentication (GUI)

Perform the following steps to configure ISE for Central Web Authentication.

Procedure

Step 1 Login to the Cisco Identity Services Engine (ISE).

Step 2 Click **Policy** and then click **Policy Elements**.

Step 3 Click **Results**.

Step 4 Expand **Authorization** and click **Authorization Profiles**.

Step 5 Click **Add** to create a new authorization profile for URL filter.

Step 6 Enter a name for the profile in the **Name** field. For example, CentralWebauth.

Step 7 Choose **ACCESS_ACCEPT** option from the **Access Type** drop-down list.

Step 8 Alternatively, in the **Common Tasks** section, check **Web Redirection**..

Step 9 Choose the **Centralized Web Auth** option from the drop-down list.

Step 10 Specify the ACL and choose the ACL value from the drop-down list.

Step 11 In the **Advanced Attributes Setting** section, choose **Cisco:cisco-av-pair** from the drop-down list.

Note Multiple ACL can be applied on the controller based on priority. In L2 Auth + webauth multi-auth scenario, if the ISE returns ACL during L2 Auth then ISE ACL takes precedence over the default webauth redirect ACL. This leads to traffic running in webauth pending state, if ISE ACL has permit rule. To avoid this scenario, you need to set the precedence for L2 Auth ISE returned ACL. The default webauth redirect ACL priority is 100. To avoid traffic issue, you need to configure the redirect ACL priority above 100 for ACL returned by ISE.

Step 12 Enter the following one by one and click (+) icon after each of them:

- url-redirect-acl=<sample_name>
- url-redirect=<sample_redirect_URL>

  For example,

```
Cisco:cisco-av-pair = priv-lvl=15
Cisco:cisco-av-pair = url-redirect-acl=ACL-REDIRECT2
Cisco:cisco-av-pair = url-redirect=
https://9.10.8.247:port/portal/gateway?
sessionId=SessionIdValue&portal=0ce17ad0-6d90-11e5-978e-005056bf2f0a&daysToExpiry=value&action=cwa
```

Step 13 Verify contents in the **Attributes Details** section and click **Save**.

Local Mode

Defining URL Filter List

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**urlfilter list** *list-name* **Example:** `Device(config)# urlfilter list urllist_local_preauth`	Configures the URL filter list. Here, *list-name* refers to the URL filter list name. The list name must not exceed 32 alphanumeric characters.

	Command or Action	Purpose
Step 3	**action permit** **Example:** `Device(config-urlfilter-params)# action permit`	Configures the action: permit (allowed list) or deny (blocked list).
Step 4	**filter-type post-authentication** **Example:** `Device(config-urlfilter-params)# filter-type post-authentication`	**Note** This step is applicable while configuring post-authentication URL filter only. Configures the URL list as post-authentication filter.
Step 5	**redirect-server-ip4** *IPv4-address* **Example:** `Device(config-urlfilter-params)# redirect-server-ipv4 9.1.0.101`	Configures the IPv4 redirect server for the URL list. Here, *IPv4-address* refers to the IPv4 address.
Step 6	**redirect-server-ip6** *IPv6-address* **Example:** `Device(config-urlfilter-params)# redirect-server-ipv6 2001:300:8::82`	Configures the IPv6 redirect server for the URL list. Here, *IPv6-address* refers to the IPv6 address.
Step 7	**url** *url* **Example:** `Device(config-urlfilter-params)# url url1.dns.com`	Configures an URL. Here, *url* refers to the name of the URL.
Step 8	**end** **Example:** `Device(config-urlfilter-params)# end`	Returns to privileged EXEC mode.

Applying URL Filter List to Policy Profile (GUI)

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **Policy**.

Step 2 Click on the **Policy Name**.

Step 3 Go to **Access Policies** tab.

Step 4 In the **URL Filters** section, choose the filters from the **Pre Auth** and **Post Auth** drop-down lists.

Step 5 Click **Update & Apply to Device**.

Applying URL Filter List to Policy Profile

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless profile policy** *profile-policy* **Example:** `Device(config)# wireless profile policy default-policy-profile`	Configures wireless policy profile. Here, *profile-policy* refers to the name of the WLAN policy profile.

Step 3	**urlfilter list** {**pre-auth-filter** *name* | **post-auth-filter** *name*}
	Example:
	`Device(config-wireless-policy)# urlfilter list pre-auth-filter urllist_local_preauth`
	`Device(config-wireless-policy)# urlfilter list post-auth-filter urllist_local_postauth`
		Applies the URL list to the policy profile.
		Here, *name* refers to the name of the pre-authentication or post-authentication URL filter list configured earlier.
		Note During the client join, the URL filter configured on the policy will be applied.

Step 4	**end** **Example:** `Device(config-wireless-policy)# end`	Returns to privileged EXEC mode.

Configuring ISE for Central Web Authentication

Creating Authorization Profiles

Procedure

Step 1 Login to the Cisco Identity Services Engine (ISE).
Step 2 Click **Policy**, and click **Policy Elements**.
Step 3 Click **Results**.
Step 4 Expand **Authorization**, and click **Authorization Profiles**.
Step 5 Click **Add** to create a new authorization profile for URL filter.
Step 6 In the Name field, enter a name for the profile. For example, *CentralWebauth*.
Step 7 Choose **ACCESS_ACCEPT** from the Access Type drop-down list.
Step 8 In the Advanced Attributes Setting section, choose **Cisco:cisco-av-pair** from the drop-down list.
Step 9 Enter the following one by one and click (+) icon after each of them:

- url-filter-preauth=<preauth_filter_name>
- url-filter-postauth=<postauth_filter_name>

For example,

```
Cisco:cisco-av-pair = url-filter-preauth=urllist_pre_cwa
Cisco:cisco-av-pair = url-filter-postauth=urllist_post_cwa
```

Step 10 Verify contents in the Attributes Details section and click **Save**.

Mapping Authorization Profiles to Authentication Rule

Procedure

Step 1 In the **Policy** > **Authentication** page, click **Authentication**.

Step 2 Enter a name for your authentication rule.

For example, *MAB*.

Step 3 In the If condition field, select the plus (+) icon.

Step 4 Choose **Compound condition**, and choose **WLC_Web_Authentication**.

Step 5 Click the arrow located next to **and ...** in order to expand the rule further.

Step 6 Click the + icon in the Identity Source field, and choose **Internal endpoints**.

Step 7 Choose **Continue** from the 'If user not found' drop-down list.

This option allows a device to be authenticated even if its MAC address is not known.

Step 8 Click **Save**.

Mapping Authorization Profiles to Authorization Rule

Procedure

Step 1 Click **Policy** > **Authorization**.

Step 2 In the Rule Name field, enter a name.

For example, *CWA Post Auth*.

Step 3 In the Conditions field, select the plus (+) icon.

Step 4 Click the drop-down list to view the Identity Groups area.

Step 5 Choose **User Identity Groups** > **user_group**.

Step 6 Click the plus (+) sign located next to **and ...** in order to expand the rule further.

Step 7 In the Conditions field, select the plus (+) icon.

Step 8 Choose **Compound Conditions**, and choose to create a new condition.

Step 9 From the settings icon, select **Add Attribute/Value** from the options.

Step 10 In the Description field, choose **Network Access** > **UseCase** as the attribute from the drop-down list.

Step 11 Choose the **Equals** operator.

Step 12 From the right-hand field, choose **GuestFlow**.

Step 13 In the Permissions field, select the plus (+) icon to select a result for your rule.

You can choose **Standard** > **PermitAccess** option or create a custom profile to return the attributes that you like.

Viewing DNS-Based Access Control Lists

To view details of a specified wireless URL filter, use the following command:

```
Device# show wireless urlfilter details <urllist_flex_preauth>
```

To view the summary of all wireless URL filters, use the following command:

```
Device# show wireless urlfilter summary
```

To view the URL filter applied to the client in the resultant policy section, use the following command:

```
Device# show wireless client mac-address <MAC_addr> detail
```

Configuration Examples for DNS-Based Access Control Lists

Flex Mode

Example: Defining URL Filter List

This example shows how to define URL list in Flex mode:

```
Device# configure terminal
Device(config)# urlfilter list urllist_flex_pre
Device(config-urlfilter-params)# action permit
Device(config-urlfilter-params)# redirect-server-ipv4 8.8.8.8
Device(config-urlfilter-params)# redirect-server-ipv6 2001:300:8::81
Device(config-urlfilter-params)# url url1.dns.com
Device(config-urlfilter-params)# end
```

Example: Applying URL Filter List to Flex Profile

This example shows how to apply an URL list to the Flex profile in Flex mode:

```
Device# configure terminal
Device(config)# wireless profile flex default-flex-profile
Device(config-wireless-flex-profile)# acl-policy acl_name
Device(config-wireless-flex-profile-acl)# urlfilter list urllist_flex_preauth
Device(config-wireless-flex-profile-acl)# end
```

Local Mode

Example: Defining Preauth URL Filter List

This example shows how to define URL filter list (pre-authentication):

```
Device# configure terminal
Device(config)# urlfilter list urllist_local_preauth
Device(config-urlfilter-params)# action permit
Device(config-urlfilter-params)# redirect-server-ipv4 9.1.0.101
Device(config-urlfilter-params)# redirect-server-ipv6 2001:300:8::82
Device(config-urlfilter-params)# url url1.dns.com
Device(config-urlfilter-params)# end
```

Example: Defining Postauth URL Filter List

This example shows how to define URL filter list (post-authentication):

```
Device# configure terminal
Device(config)# urlfilter list urllist_local_postauth
Device(config-urlfilter-params)# action permit
Device(config-urlfilter-params)# filter-type post-authentication
Device(config-urlfilter-params)# redirect-server-ipv4 9.1.0.101
Device(config-urlfilter-params)# redirect-server-ipv6 2001:300:8::82
Device(config-urlfilter-params)# url url1.dns.com
Device(config-urlfilter-params)# end
```

Example: Applying URL Filter List to Policy Profile

This example shows how to apply an URL list to the policy profile in local mode:

```
Device# configure terminal
Device(config)# wireless profile policy default-policy-profile
Device(config-wireless-policy)# urlfilter list pre-auth-filter urllist_local_preauth
Device(config-wireless-policy)# urlfilter list post-auth-filter urllist_local_postauth
Device(config-wireless-policy)# end
```

Verifying DNS Snoop Agent (DSA)

To view details of the DNS snooping agent client, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client
```

To view details of the DSA enabled interface, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client enabled-intf
```

To view the pattern list in uCode memory, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client
hw-pattern-list
```

To view the OpenDNS string for the pattern list, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client
hw-pattern-list odns_string
```

To view the FQDN filter for the pattern list, use the following command:

```
Device#
show platform hardware chassis active qfp feature dns-snoop-agent client hw-pattern-list
fqdn-filter <fqdn_filter_ID>
```

Note The valid range of *fqdn_filter_ID* is from 1 to 16.

To view details of the DSA client, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client info
```

To view the pattern list in CPP client, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client pattern-list
```

To view the OpenDNS string for the pattern list, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client pattern-list
 odns_string
```

To view the FQDN filter for the pattern list, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent client pattern-list
 fqdn-filter <fqdn_filter_ID>
```

Note The valid range of *fqdn_filter_ID* is from 1 to 16.

To view details of the DSA datapath, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath
```

To view details of the DSA IP cache table, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath ip-cache
```

To view details of the DSA address entry, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath ip-cache
 address {ipv4 <IPv4_addr> | ipv6 <IPv6_addr>}
```

To view details of all the DSA IP cache address, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath ip-cache
 all
```

To view details of the DSA IP cache pattern, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath ip-cache
 pattern <pattern>
```

To view details of the DSA datapath memory, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath memory
```

To view the DSA regular expression table, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath
regexp-table
```

To view the DSA statistics, use the following command:

```
Device# show platform hardware chassis active qfp feature dns-snoop-agent datapath stats
```