Support for SNI Detection

This chapter describes the Server Name Indication (SNI) Detection feature in ADC, and provides detailed information on the following topics:

Feature Description

Server Name Indication (SNI) is an extension of the Transport Layer Security (TLS) protocol that allows multiple secure (HTTPS) websites (or any other service over TLS) to be served from the same IP address without requiring all those sites to use the same certificate. SNI provides a mechanism for the client to tell the server which hostname it is trying to connect to.

ADC detects encrypted traffic using the SNI field (signatures) of TLS/SSL (Secure Sockets Layer) traffic. These signatures are added along with other detection mechanisms and delivered as a plugin. If there are new SNI fields either in the already detected applications or new applications, then these new fields are added to the plugin and a new version of the plugin is released. This results in frequent releases of plugin versions causing delay in upgrading the new plugin in the network and leading to revenue leak to the operator. Due to increased number of applications moving towards TLS/SSL, an option is provided to configure the SNI in ruledef and classify traffic based on the configured SNI with this release.

Important The SNI Detection feature requires a valid Application Detection and Control license. Contact your Cisco Account representative for more information.

The SNI field in the TLS/SSL handshake is used to determine the type of TLS/SSL flow being setup. SNI rule variable is added as an optional field in EDRs for detection of TLS/SSL flows. When the “tls sni” rule variable is configured and a valid SNI name is detected in the flow, the SNI field is populated in the EDR.

QUIC SNI Detection

The SNI Detection feature is enhanced to support QUIC SNI configuration and classify traffic based on the configured SNI. The CLI commands added to configure the SNI are generic such that other application-identifiers added in the future are taken care of with the Plugin changes only.

Important This feature requires the latest ADC Plugin to be loaded from the adc_v2.x stream along with StarOS changes. The default plugin does not support this feature. Contact your Cisco account representative for more information.

When a QUIC flow is analyzed, the P2P rule match takes place on this flow with the configured SNI. If the flow/packet matches the rule, the custom-defined-protocol (CDP) name specified in the ruledef is taken and the flow is marked as CDP. If no CDP is configured in the rule, then the flow is treated as QUIC flow.

Within the P2P plugin, when a QUIC flow arrives with SNI that is already hardcoded and the same SNI is also configured in ruledef, the ruledef with higher priority takes precedence. Based on rule priority, rule matching is done and CDP statistics get incremented. When the new QUIC flow comes with SNI that is not hardcoded, the flow will be marked as a QUIC flow.

- The **p2p app-identifier** and **p2p set-app-proto** commands are added in the ACS Ruledef Configuration mode to configure QUIC-SNI and TLS-SNI that are dynamically populated from the P2P plugin and match the traffic against it.
- EDR attributes are added to support app-identifiers supplied from plugin. EDR logs the matched CDP in the "p2p-protocol" field for the flow. If QUIC-SNI is configured in the EDR field, then QUIC-SNI string will be populated.
- Analyzer statistics, ruledef statistics, and bulk statistics are updated for newly identified protocols
- Backward compatibility for the TLS-SNI feature is maintained such that this feature works seamlessly with new CLI commands added in this release.

Limitations

The limitations with this feature are listed in this section:

- The **quic-sni** identifier is configured in the EDR with the new plugin. When rolled back to old plugins, the EDR headers will print **p2p-unknown** since the old plugin does not support the configured app-identifier. When upgraded to a new plugin that supports QUIC-SNI identifier, **p2p-unknown** will be updated to display **p2p-quic-sni** in the EDR.
- The help strings related to the new CLI keywords will be updated in a later release.

Support for QUIC IETF Implementation

In the current framework, Deep Packet Inspection (DPI) is done for every packet in a flow when it reaches the plugin. The DPI is done by analyzing the packets and extracting deterministic patterns. The DPI is done in-order to detect the application and to classify its subtype. Plugin excludes the flow after the DPI. The flow is offloaded after the detection.

As part of QUIC IETF, the initial QUIC handshake packets (Client/Server Hello) are encrypted over the network. Hence, there are no deterministic patterns available for detection of the application. Support is added in p2p plugin to decrypt and obtain the SNI (Server Name Indication) for detection.

Relationships to Other Features

This section describes how the SNI Detection feature relates to other ADC features.

- **Analyzer Interworking**: In support of SNI Detection, HTTPS protocol support is added for ECS analysis of all SSL flows as part of the Analyzer Interworking feature. This feature is enabled by default for all analyzers including HTTPS if P2P detection/protocol is enabled.

  The different behaviors when Analyzer Interworking is enabled or disabled is listed in the table below.

Condition	HTTPS Routing Enabled	HTTPS Routing Disabled
With SNI feature in 17.5 and later releases:		
If SSL protocol is enabled, SNI/EDR features will work if any routing rule is configured	• Analyzer Interworking enabled for HTTPS: App-proto = HTTPS(6) and p2p protocol = SSL • Analyzer Interworking disabled for HTTPS: App-proto = P2P(29) and p2p protocol = SSL	App-proto = P2P(29) and p2p protocol = SSL
If SSL protocol is disabled, SNI/EDR features will not work.	App-proto = HTTPS(6) and p2p protocol = Unknown	App-proto = Unknown(0) and p2p protocol = Unknown
Without SNI feature in releases prior to 17.5:		
SNI/EDR features will not work and SSL will not be exposed to ASR 5500.	App-proto = HTTPS(6) and p2p protocol = Unknown	App-proto = Unknown(0) and p2p protocol = Unknown

- **SSL Renegotiation Tracking**: With the SNI Detection feature, the ADC plugin must be able to store Session ID in SSL renegotiated table, map the renegotiated flow to stored Session ID, and map the corresponding CDP name to the flow in the same way as it is done for SSL Renegotiation feature.

For more information on these features, refer to the *ADC Administration Guide*.

Configuring SNI Detection

This section describes how to configure the SNI Detection feature.

Configuring SNI in Ruledef

Use the following configuration to configure the TLS/SSL and QUIC Server Name Indication (SNI) and the corresponding custom defined protocol (CDP).

Important The QUIC SNI Detection feature requires the latest ADC Plugin to be loaded from the adc_v2.x stream along with StarOS changes. The default plugin does not support this feature. Contact your Cisco account representative for more information.

```
configure
  active-charging service service_name
    ruledef ruledef_name
     [ no ] tls { set-app-proto cdp_name_string | sni operator server_name_string
 }
     [ no ] p2p app-identifier { quic-sni operator quic_sni_string | tls-cname
 operator tls_cname_string | tls-sni operator tls_sni_string }
      [ no ] p2p set-app-proto cdp_name_string
      end
```

Notes:

- The **tls set-app-proto** command specifies the name of the custom defined protocol for TLS/SSL flows matching the ruledef.
- The **tls sni** command specifies the TLS/SSL Server Name Indication (SNI) field value in the Client Hello packet.
- In release 20.2, the **p2p app-identifier** command configures "quic-sni", "tls-sni", and "tls-cname" app-identifiers supported by the P2P dynamic library.
- In release 20.2, the **p2p set-app-proto** command configures the custom-defined protocol (CDP) name.
- The following commands must be configured for SNI rules to work:
  - Enable SSL protocol in the Active Charging Service configuration:

    ```
    [local]P2P_SSl(config-acs)# p2p-detection protocol ssl
    ```

    If the **p2p-detection protocol all** CLI command is enabled in the Active Charging Service configuration, then the **ssl** keyword need not be enabled again as it will be already enabled with the **all** keyword.

    The **ssl** protocol is available only in Plugin releases 1.142.526 and later.
  - Enable P2P in the ACS Rulebase configuration:

    ```
    [local]P2P_SSl(config-rule-base)# p2p dynamic-flow-detection
    ```
  - The action priority for SNI ruledef must be configured in the rulebase similar to other ruledefs.

For more information, refer to the *ACS Ruledef Configuration Mode* chapter of the *Command Line Interface Reference*.

Configuring the QUIC IETF

Use the following configuration to enable or disable the QUIC IETF decryption.

```
configure
  active-charging service service_name
    p2p-detection debug-param protocol-param p2p_quic_ietf_decrypt x
    end
```

Note By default, the CLI is disabled and there's minimal impact on the performance due to TLS decryption.

The CLI needs to be entered manually to boot configuration as plugin CLI is not part of show/save configuration.

Configuring SNI rule variable

Use the following configuration to configure the SNI rule variable for TLS/SSL and QUIC flows in EDR.

```
configure
  active-charging service acs_service_name
    edr-format format_name
      rule-variable tls sni priority priority
      rule-variable p2p app-identifier { quic-sni | tls-cname | tls-sni
 } priority priority
      end
```

Notes:

- The **tls sni** command specifies the TLS/SSL SNI rule variable configured for TLS/SSL flows in EDR.
- In release 20.2, the **p2p app-identifier** command specifies the QUIC-SNI, TLS-SNI, and TLS-CNAME application identifiers populated from the plugin.
- **priority** *priority*: Specifies the CSV position of the field (protocol rule) in the EDR. *priority* must be an integer from 1 through 65535.

For more information, refer to the *EDR Format Configuration Mode* chapter of the *Command Line Interface Reference*.

Enabling HTTPS Analyzer Interworking

Use the following configuration to enable or disable ECS analysis for HTTPS analyzer interworking.

```
configure
    active-charging service service_name
         [ no ] p2p-detection ecs-analysis { https }
         end
```

Notes:

- The Active Charging flows will have the app-proto marked as "HTTPS" instead of P2P for all SSL flows if analyzer interworking for HTTPS is enabled.
- The Active Charging flows will have the app-proto marked as "P2P" for all SSL flows if analyzer interworking for HTTPS is disabled.
- By default, analyzer interworking for all analyzers including HTTPS is enabled when P2P detection is enabled.

For more information on the commands listed above, refer to the *Command Line Interface Reference*.

Verifying the SNI Configuration

Executing the following command displays the application/protocol configured in the "set app-proto" string of TLS ruledef:

show active-charging analyzer statistics name cdp [application *app_name* **| instance** *instance_number* **| summary | verbose | wide]**

Executing the following command displays the fields for TLS/SSL SNI and CDP as configured in the TLS ruledef:

show active-charging ruledef name *ruledef_name*

Monitoring and Troubleshooting the SNI Detection

This section provides information on the show commands available to support this feature.

SNI Detection Show Command(s) and/or Outputs

show active-charging analyzer statistics name cdp

The following fields display the analyzer statistics for the custom defined protocol.

- CDP Summary:
  - Total Uplink Bytes
  - Total Downlink Bytes
  - Total Uplink Pkts
  - Total Downlink Pkts

For description of the fields listed above see, *Statistics and Counters Reference*.

show active-charging flows type cdp

The following fields display the flow-level statistics for the custom defined protocol.

- Session ID
- Flow-ID
- Application Protocol
- Transport Protocol
- Tethered Flow
- Bytes-Up
- Bytes-Down
- Pkts-Up
- Pkts-Down

Bulk Statistics

In support of the SNI detection feature, the "p2p-protocol" field in the P2P schema will display the application protocol configured in the "set app-proto" string of TLS ruledef.

For more information on bulk statistics, see the *P2P Schema Statistics* chapter in the *Statistics and Counters Reference*.