### Logger Models

#### Overview

Logger models serve as a roadmap that defines a series of data processing steps from data collection to normalization. They specify how the logger should collect original data, how the collected data should be parsed, and how the parsed data should be normalized.



In some cases, a parser may not be specified according to the logger model. If the log data is already structured, no separate parsing process is necessary. For example, data stored in a database is already separated by fields, so additional parsing is not required.

App Logger Models and Basic Logger Models

Logger models can be categorized into basic logger models and app logger models. It is not common for administrators to need to configure logger models, similar to parsers.

**Basic Logger Models**

Logger models that can be configured in Logpresso Sonar without installing an app. Initially, only two basic logger models are provided:

|  |  |
| --- | --- |
| Name | Description |
| WTMP | Logger model defining the collection and normalization method for wtmp log files |
| Windows XML Event Log | Logger model defining the method for collecting/normalizing Windows event logs |

The Windows XML Event Log logger model can be used in environments with Windows Sentries.

Below are the types of basic logger models that can be configured in Logpresso Sonar. The type determines the logger's [collection settings](https://docs.logpresso.comnull).

|  |  |
| --- | --- |
| Logger Type | Description |
| CEP Event | Data collection using Complex Event Processing (CEP) features |
| DNS Sniffer | Collects DNS packets using a PCAP device |
| FTP Directory Watcher | Collects text log files from a specific directory on a remote host via FTP communication |
| FTP Rotation Log File | Collects a single text log file that rotates on a remote host via FTP communication |
| FTP Daily Directory | Collects text log files that match patterns and names from directories created daily via FTP communication |
| GZIP Directory Watcher | Collects text log files compressed in GZIP format from the local host |
| HTTP POST | Collects data sent by a remote host using the HTTP POST method |
| HTTP Monitor | Collects response codes and statistical information from a remote host's web service |
| HTTP Sniffer | Collects HTTP packets using a PCAP device |
| JDBC Logger | Collects data from a database using JDBC |
| NetFlow | Collects NetFlow v5/v9 packets |
| PCAP Directory Watcher | Collects PCAP files from a local directory |
| PCAP Packet | Collects all packets using a PCAP device |
| RSS Logger | Collects data from an RSS feed specified by a URL |
| SFTP WTMP File | Collects WTMP logs from a remote host via SFTP communication |
| SFTP Directory Watcher | Collects text log files from a specific directory on a remote host via SFTP communication |
| SFTP Rotation Log File | Collects a single text log file that rotates on a remote host via SFTP communication |
| SFTP Daily Directory | Collects text log files that match patterns and names from directories created daily via SFTP communication |
| SNMP GET | Collects data from SNMP agents |
| SNMP Interface Statistics | Collects traffic statistics per network interface from SNMP agents |
| SNMP Trap Receiver | Collects SNMP messages received via SNMP traps |
| SNMPv3 GET | Collects data from SNMPv3 agents |
| SNMPv3 Interface Statistics | Collects traffic statistics per network interface from SNMPv3 agents |
| SSH Execution | Collects standard output of commands executed in an SSH shell |
| TCP Port Status Logger | Checks the open status of TCP ports on local/remote hosts and collects results |
| WTMP | Collects WTMP logs from the local host |
| sFlow | Collects sFlow v5 packets |
| Directory Watcher | Collects text log files from a specific directory on the local host at regular intervals |
| Disk Usage | Periodically monitors and collects disk usage of Logpresso Sonar nodes or sentries |
| Rotation Log File | Collects a single text log file that rotates on the local host |
| Linux Account Event Detection | Collects changes to the account list file passwd on the local host |
| Recursive GZIP Directory Watcher | Collects GZIP log files that match patterns and names from a specific directory on the local host |
| Recursive Directory Watcher | Collects text log files that match patterns and names from a specified directory on the local host |
| Multi Rotation Log File | Collects log files that are backed up to different paths at regular intervals and then deleted and rewritten |
| Configuration File Change Detection | Collects changes to files that match patterns and names on the local host |
| Sentry Performance Log | Collects Sentry performance logs in bulk from Logpresso Sonar |
| Stream Query Output | Collects output data from stream queries in Logpresso Sonar |
| Syslog Logger | Collects Syslog messages sent by remote hosts |
| External Programs | Collects standard output of commands and scripts executed on the local host |
| Web Keeper | Collects data from older versions of Web Keeper that use MSSQL |
| Daily Directory | Collects text log files that match patterns and names from directories created daily on the local host |
| Forwarder Performance Log | Collects performance logs of forwarding nodes in bulk from Logpresso Sonar |

**App Logger Models**

Most logger models are provided when installing an [app](https://docs.logpresso.comnull). App logger models are pre-configured with all the settings necessary to extract fields from data collected from the integrated target system.

Do not modify or delete app logger models. This may cause issues with the app's functionality.

Normalization Rules

Logger models provide rules for normalizing logs according to the log schema. Normalization rules consist of the log schema, queries necessary for processing the input log data, and refresh cycles.

Since multiple types of log records may be mixed within a single log data entry, normalization rules must be written to be mutually exclusive and collectively exhaustive (MECE principle). For instance, if log data contains three types of records A, B, and C, the logger model must include three normalization rules corresponding to each type. Palo Alto Networks firewall logs consist of session logs and intrusion detection logs. Different types of log records require normalization according to the appropriate log schema for each record.

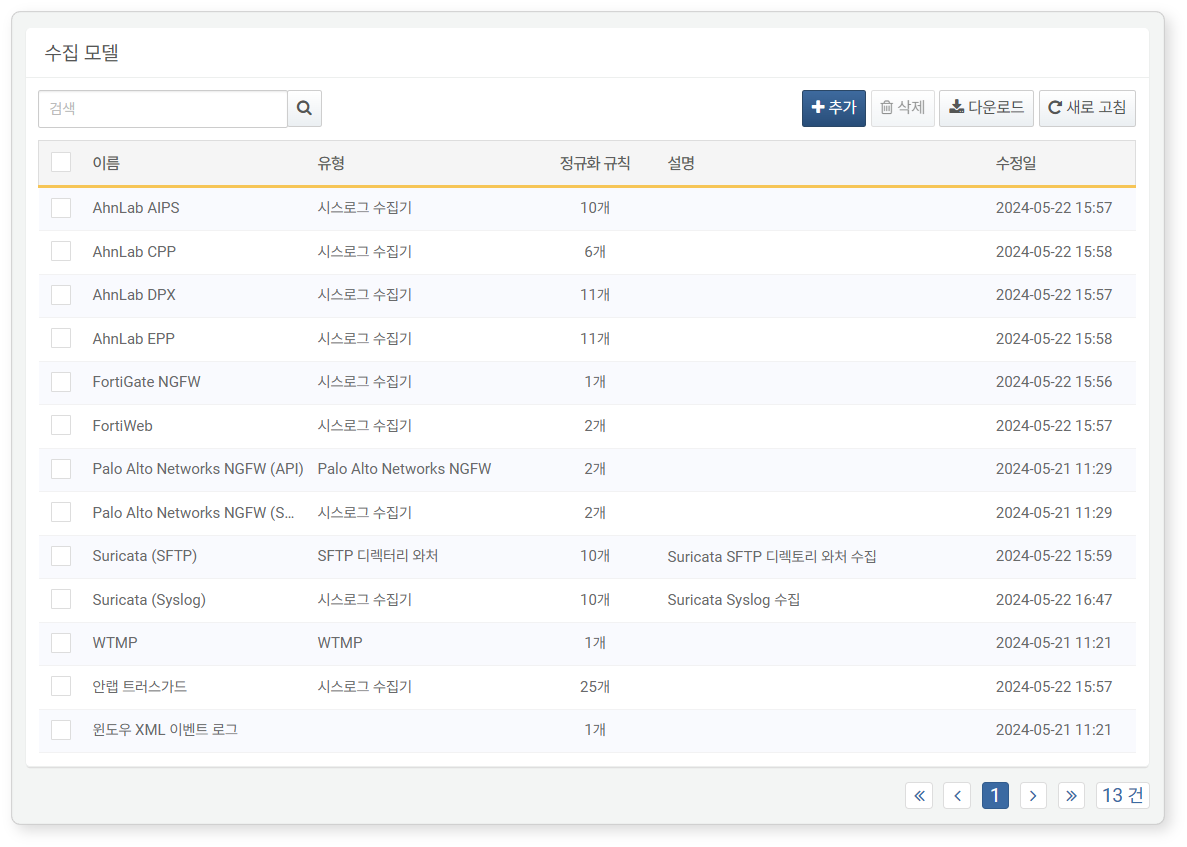
Additionally, considering the possibility of changes in log formats and parsing errors, normalization rules must also include provisions for preserving log records that do not match any normalization rules or cannot be normalized.



The image presented shows the collection model screen for the [Suricata](https://logpresso.store/ko/apps/suricata) app. You can see that in addition to normalization rules such as **Alert**, **Anomaly**, **DHCP**, **DNS**, **Fileinfo**, **Flow**, **HTTP**, **SSH**, and **TLS**, there is also an **Unclassified** rule. The **Stream Query** of the **Unclassified** rule is a query that identifies records among the input data that either lack a value for the event\_type field or do not match the defined values in each normalization rule (alert, anomaly, dhcp, dns, fileinfo, flow, http, ssh, tls) and classifies them as unclassified.

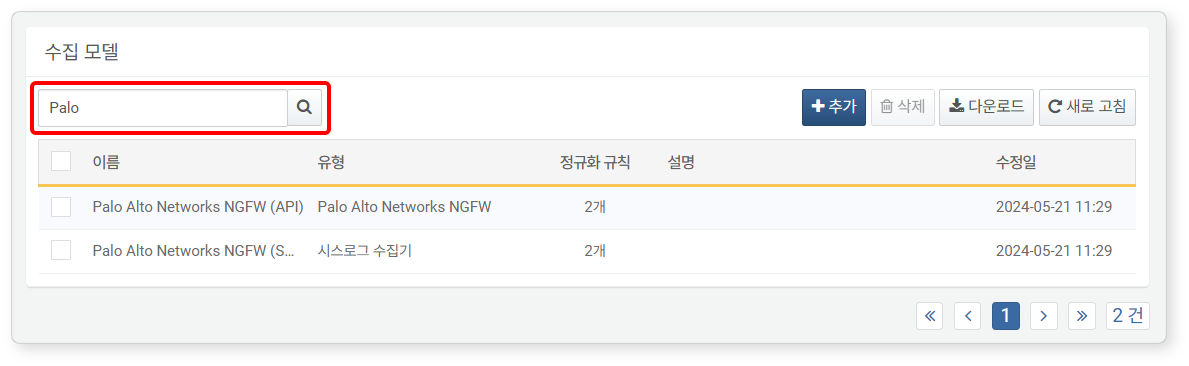
#### Viewing/Search Logger Model List

You can view or search the logger model list under **Collection > Logger Models**.



* **Name**: The name used to identify the logger model
* **Type**: The type of logger to be executed
* **Normalization Rules**: The number of normalization rules defined in the logger model
* **Description**: Additional information about the logger model
* **Modification Date**: The date the logger model was created or last modified

To find a specific logger model in the list, use the search tool in the toolbar. The search tool will display logger models that contain the word entered in the **Name** field. The logger model search tool is case-insensitive.

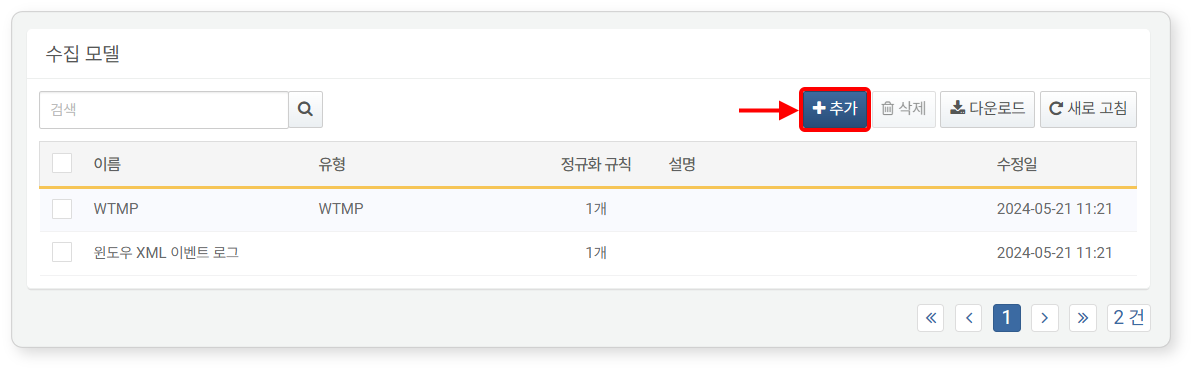


#### Adding a Logger Model

In most operational environments, using app logger models is sufficient. However, if a new logger model is needed, you can add a logger model.

To add a new logger model:

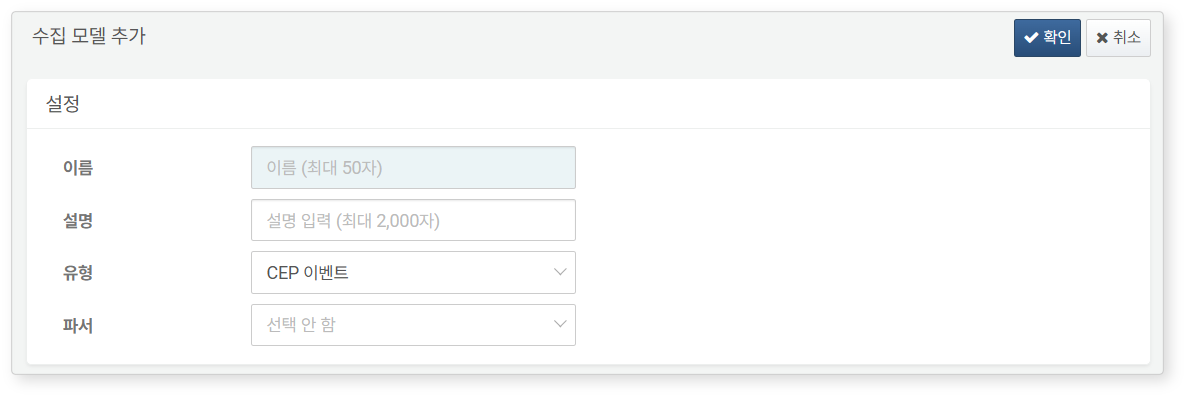
Click **Add** in the toolbar under **Collection > Logger Models**.



Enter or select the necessary values for [Settings](https://docs.logpresso.comnull) and [Normalization Rules](https://docs.logpresso.comnull), then click the confirmation button in the upper right corner.

Settings

In **Settings**, enter or select the **Name**, **Description**, **Type**, and **Parser** for the logger model.

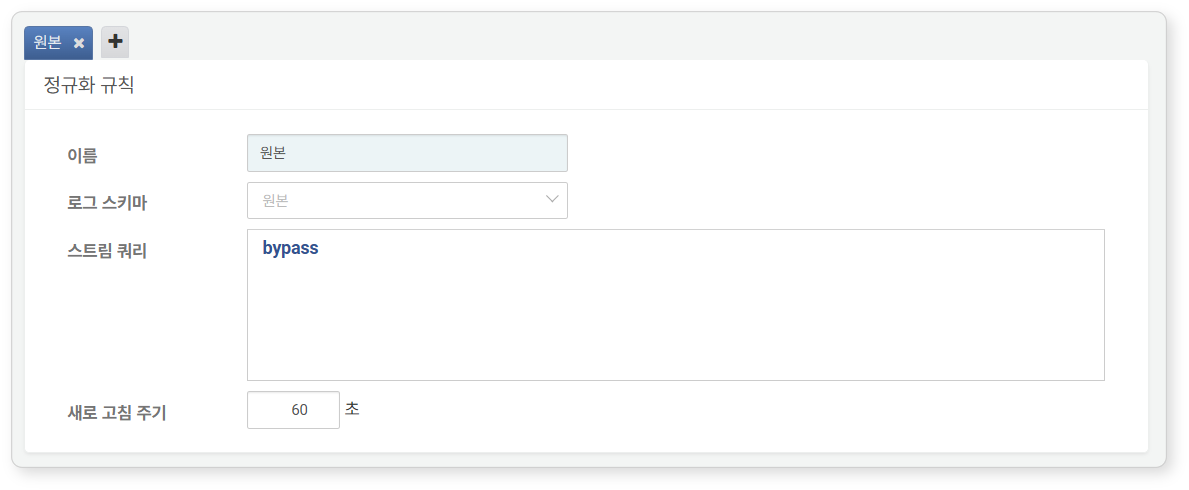


* **Name**: A name assigned for users to identify the logger model in the web console
* **Description**: A description of the logger model. Use this field to provide additional information about the logger model.
* **Type**: The type of logger to be executed (default: None selected).
* **Parser**: The name of the parser that will process the data received from the logger. If the input data is already structured, you may not need to select a parser (e.g., if the type is JDBC logger).

Normalization Rules

You can set one or more normalization rules. Normalization rules must be written to classify input records without overlap and ensure no records are omitted (mutually exclusive, collectively exhaustive).

* To add a normalization rule, click the **+** tab.
* To delete a normalization rule, click the **X** on the tab.



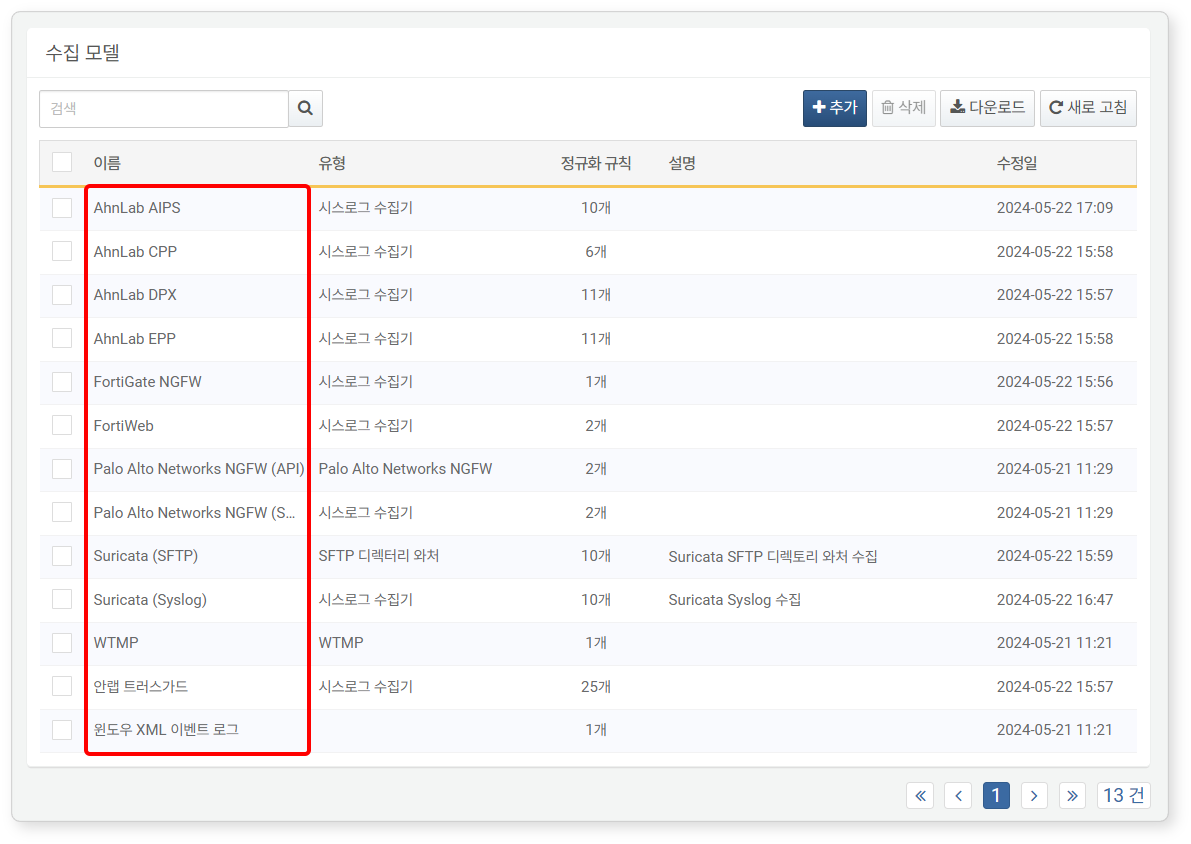
* **Name**: A name to identify the normalization rule
* **Log Schema**: The log schema to be applied to the normalization rule
* **Stream Query**: A preprocessing query necessary for mapping part or all of the data processed by the parser to the log schema
* Typically, a query using the [search](https://docs.logpresso.comnull) command is used to filter input data based on specific field values.
* Enter queries that change the field names of input records to match the log schema or assign values to empty fields.
* It can also be used for mapping fields like employee number (emp\_key) as shown in the example below (CONNECT\_PROFILE is the name of the connect profile).
* streamjoin emp\_key [ dbquery CONNECT\_PROFILE select emp\_key, emp\_name from emp ]
* **Refresh Cycle**: The cycle for resetting the state of the stream query (default: 60 seconds; specify 0 for real-time reset)

#### Modifying a Logger Model

Do not modify the built-in normalization rules of basic/app logger models. This may cause issues with the app's functionality.

To modify a logger model:

Click the **Name** of the logger model you wish to modify from the list.



In the **Modify Logger Model** screen, modify the information and click **Confirm**.

* For descriptions of the properties to be modified, refer to [Adding a Logger Model](https://docs.logpresso.comnull).
* The **Type** cannot be modified. To change the type, you must delete the logger model and recreate it.

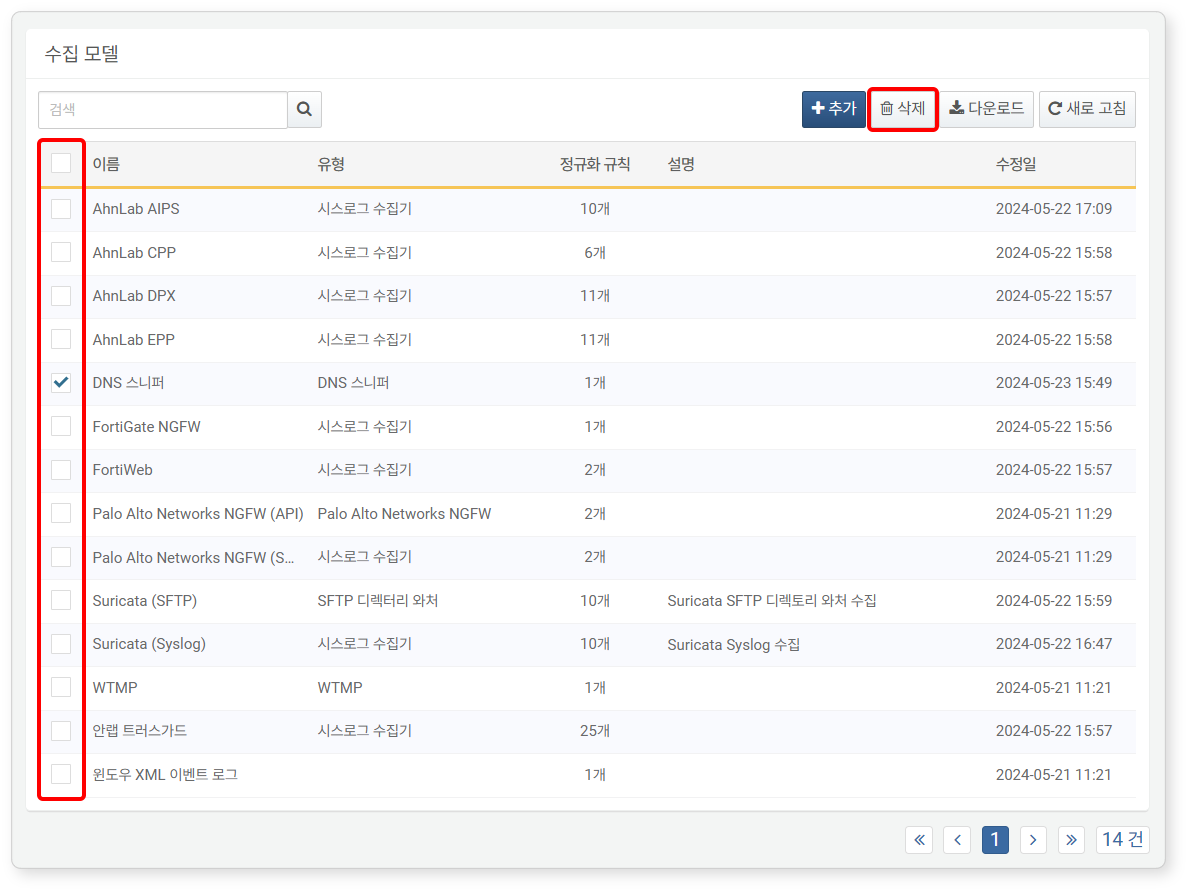
#### Deleting a Logger Model

Do not delete basic/app logger models. This may cause issues with the app's functionality.

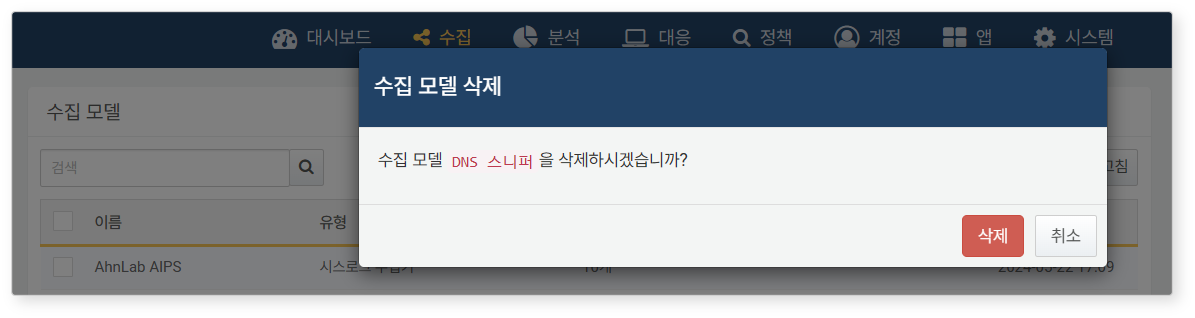
To delete a logger model:

Select the checkbox for the row containing the logger model information you wish to delete.

Click **Delete** in the toolbar.



In the **Delete Logger Model** dialog, review the list of logger models to be deleted and click **Delete**. Click **Cancel** if you do not wish to delete.



If you are unable to delete the logger model, you can check the reason for the failure in the **Logger Model Deletion Failed** dialog.

