Schema: Correlation Tables

This chapter contains information on the schema and supported joins for correlation-related events, including remediation status and allow list events. For more information, see the sections listed in the following table.

Table 9-1 Schema for Correlation Tables

See...	For the table that stores information on...	Version
compliance_event, page 9-1	Correlation events, which are generated when a correlation rule within an active correlation policy triggers.	4.10.x+
remediation_status, page 9-6	Remediation status events, which are generated when an active correlation policy triggers a remediation as a response.	4.10.x+
white_list_event, page 9-7	Allow list events, which are generated when the system detects a host out of compliance with a allow list in an active allow list compliance policy.	4.10.x+
white_list_violation, page 9-9	Allow list violations, which track the ways that the hosts on your network violate the compliance allow lists in active compliance policies.	4.10.x+

compliance_event

The **compliance_event** table contains information about the correlation events that your Secure Firewall Management Center generates.

For more information, see the following sections:

- compliance_event Fields, page 9-2
- compliance_event Joins, page 9-5
- compliance_event Sample Query, page 9-5

compliance_event Fields

Keep in mind that many of the fields in the table can be blank, depending on what type of event triggered the correlation rule. For example, if the Secure Firewall Management Center generates a correlation event because the system detects a specific application protocol or web application running on a specific port, that correlation event does not include intrusion-related information. Fields in this table can also be blank depending on your Secure Firewall configuration. For example, if you do not have a Control license, correlation events do not include user identity information.

Note that starting in Version 5.0, the Secure Firewall records the detection of network and user activity at the managed device level, rather than by detection engine. The `detection_engine_name` and `detection_engine_uuid` fields in the **`compliance_event`** table now return only blanks, and queries that join on those fields return zero records. You must query on the `sensor_uuid` field instead of `detection_engine_uuid` for information about the location of an event's detection.

The following table describes the fields you can access in the **`compliance_event`** table.

Table 9-2 compliance_event Fields

Field	Description
`blocked`	Value indicating what happened to the packet that triggered the intrusion event: • `0` — Packet not dropped • `1` — Packet dropped (inline, switched, or routed deployments) • `2` — Packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in an inline, switched, or routed deployment
`description`	Information about the correlation event and how it was triggered.
`detection_engine_name`	Field deprecated in Version 5.0. Returns `null` for all queries.
`detection_engine_uuid`	Field deprecated in Version 5.0. Returns `null` for all queries.
`domain_name`	Name of the domain on which the event was detected.
`domain_uuid`	UUID of the domain on which the event was detected. This is presented in binary.
`dst_host_criticality`	The user-assigned host criticality of the destination host involved in the correlation event: `None`, `Low`, `Medium`, or `High`.
`dst_host_type`	The destination host type: `Host`, `Router`, `Bridge`, `NAT Device`, or `Load Balancer`.
`dst_ip_address`	Field deprecated in Version 5.2. Due to backwards compatibility the value in this field is not set to `null`, but it is not reliable.
`dst_ip_address_v6`	Field deprecated in Version 5.2. Due to backwards compatibility the value in this field is not set to `null`, but it is not reliable.
`dst_ipaddr`	A binary representation of the IPv4 or IPv6 address for the destination host involved in the triggering event.
`dst_os_product`	The operating system name on the destination host.
`dst_os_vendor`	The operating system's vendor on the destination host.
`dst_os_version`	The operating system's version number on the destination host.
`dst_port`	The port number for the host receiving the traffic if the event protocol type is TCP or UDP. The ICMP code if the protocol type is ICMP.

Table 9-2 compliance_event Fields (continued)

Field	Description
dst_rna_service	If identified, the application protocol on the source host that is associated with the triggering event. If not identified, one of the following: • none or blank - no application protocol traffic • unknown - the server cannot be identified based on known server fingerprints • pending - the system needs more information
dst_user_dept	The department of the destination user.
dst_user_email	The email address of the destination user.
dst_user_first_name	The first name of the destination user.
dst_user_id	The internal identification number for the destination user; that is, the user who last logged into the destination host before the event occurred.
dst_user_last_name	The last name of the destination user.
dst_user_last_seen_sec	The UNIX timestamp of the date and time the system last reported a login for the destination user.
dst_user_last_updated_sec	The UNIX timestamp of the date and time the destination user's information was last updated.
dst_user_name	The user name for the destination user.
dst_user_phone	The destination user's phone number.
dst_vlan_id	The destination host's VLAN identification number, if applicable.
event_id	The identification number of the triggering intrusion event generated by the device.
event_time_sec	The UNIX timestamp of the date and time of the triggering event.
event_time_usec	The microsecond increment of the triggering event timestamp.
event_type	The type of underlying event that triggered the correlation rule or caused the Secure Firewall Management Center to generate the correlation event. Values are: • ids, for intrusion event triggers • rna, for discovery event, host input event, connection event, or traffic profile change triggers • rua, for user discovery event triggers • whitelist, for compliance allow list violation triggers
host_event_type	The event type, for example, New Host or Identity Conflict.
id	An internal identification number for the correlation event.
impact	The impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact) Set only when the correlation rule was triggered by an intrusion event.

Table 9-2 compliance_event Fields (continued)

Field	Description
interface_egress_name	The ingress interface associated with the connection.
interface_ingress_name	The egress interface associated with the connection.
policy_name	The correlation policy that was violated.
policy_rule_name	The correlation rule that triggered the policy violation.
policy_rule_uuid	A unique identifier for the correlation rule.
policy_time_sec	The UNIX timestamp of the date and time the correlation event was generated.
policy_uuid	A unique identifier for the correlation policy.
priority	The priority for the correlation event, which is set in the user interface. The event priority is determined by the priority of either the triggered rule or the violated correlation policy.
protocol_name	The protocol associated with the event, if available.
protocol_num	The IANA-specified protocol number, if available.
rna_event_type	Field deprecated in Version 5.0. Returns null for all queries.
rua_event_type	Field deprecated in Version 5.0. Returns null for all queries.
rule_generator_id	The generator ID number (GID) of the component that generated the triggering intrusion event.
rule_message	Explanatory text about the intrusion event that triggered the correlation rule. For rule-based events, the message is generated from the rule. For decoder- and preprocessor-based events, the message is hard coded.
rule_signature_id	The signature ID (SID) for the event. Identifies the specific rule or rules, decoder message, or preprocessor message that caused the triggering intrusion event to be generated.
security_zone_egress_name	The egress security zone in the correlation event.
security_zone_ingress_name	The ingress security zone in the correlation event.
sensor_address	The IP address of the managed device that generated the underlying event that triggered the compliance event. Format is *ipv4_address,ipv6_address*.
sensor_name	The managed device that generated the underlying event that triggered the compliance event.
sensor_uuid	A unique identifier for the managed device, or 0 if sensor_name is null.
src_host_criticality	The user-assigned host criticality of the source host involved in the compliance event: None, Low, Medium, or High.
src_host_type	The source host type: Host, Router, Bridge, NAT Device, or Load Balancer.
src_ip_address	Field deprecated in Version 5.2. Due to backwards compatibility the value in this field is not set to null, but it is not reliable.
src_ip_address_v6	Field deprecated in Version 5.2. Due to backwards compatibility the value in this field is not set to null, but it is not reliable.
src_ipaddr	A binary representation of the IPv4 or IPv6 address for the source host involved in the triggering event.
src_os_product	The operating system's name on the source host.
src_os_vendor	The operating system's vendor on the source host.

Table 9-2 compliance_event Fields (continued)

Field	Description
src_os_version	The operating system's version number on the source host.
src_port	The port number on the source host. For ICMP traffic, the ICMP type appears instead.
src_rna_service	If identified, the application protocol on the source host that is associated with the triggering event. If not identified, one of the following: • none or blank - no application protocol traffic • unknown - the server and application protocol cannot be identified based on known server fingerprints • pending - the system needs more information
src_user_dept	The department of the source user.
src_user_email	The email address of the source user.
src_user_first_name	The first name of the source user.
src_user_id	The internal identification number for the source user; that is, the user who last logged into the source host before the event occurred.
src_user_last_name	The last name of the source user.
src_user_last_seen_sec	The UNIX timestamp of the date and time the system last reported a login for the source user.
src_user_last_updated_sec	The UNIX timestamp of the date and time the source user's information was last updated.
src_user_name	The login user name for the source user.
src_user_phone	The source user's phone number.
src_vlan_id	The source host's VLAN identification number, if applicable.
user_event_type	The type of triggering user event, for example, New User Identity or User Login.

compliance_event Joins

The following table describes the joins you can perform on the **compliance_event** table.

Table 9-3 compliance_event Joins

You can join this table on...	And...
dst_ipaddr or src_ipaddr	rna_host_ip_map.ipaddr user_ipaddr_history.ipaddr

compliance_event Sample Query

The following query returns up to 25 correlation event records from a week, with event information such as the event time, source and destination IP addresses, source and destination ports, policy information, and so on.

```
SELECT event_id, policy_time_sec, impact, blocked, src_ipaddr, dst_ipaddr, src_port,
dst_port, description, policy_name, policy_rule_name, priority, src_host_criticality,
dst_host_criticality, security_zone_egress_name, security_zone_ingress_name,
sensor_name, interface_egress_name, interface_ingress_name
FROM compliance_event WHERE event_type!="whitelist"
AND policy_time_sec
BETWEEN UNIX_TIMESTAMP("2011-10-01 00:00:00")
AND UNIX_TIMESTAMP("2011-10-07 23:59:59")
domain_name= "Global \ Company B \ Edge"
ORDER BY policy_time_sec
DESC LIMIT 0, 25;
```

remediation_status

The **remediation_status** table contains information about remediation events, which are generated when the Secure Firewall Management Center launches a remediation in response to a correlation policy violation.

For more information, see the following sections:

- remediation_status Fields, page 9-6
- remediation_status Joins, page 9-7
- remediation_status Sample Query, page 9-7

remediation_status Fields

The following table describes the database fields you can access in the **remediation_status** table.

Table 9-4 remediation_status Fields

Field	Description
id	The identification number of the policy that was violated and triggered the remediation.
policy_name	The correlation policy that was violated and triggered the remediation.
policy_rule_name	The specific correlation rule that triggered the remediation.
policy_rule_uuid	A unique identifier for the correlation rule.
policy_time_sec	The UNIX timestamp of the date and time that the correlation event that triggered the remediation was generated.
policy_uuid	A unique identifier for the correlation policy that triggered the correlation event.
remediation_name	The remediation that was launched.
remediation_time_sec	The UNIX timestamp of the date and time the Secure Firewall Management Center launched the remediation.
status_text	A message that describes what happened when the remediation was launched, such as "successful completion of remediation."

remediation_status Joins

You cannot perform joins on the **remediation_status** table.

remediation_status Sample Query

The following query returns up to 25 records generated before a given date. These records include remediation status information such as the remediation timestamp, the status message, and so on.

```
SELECT policy_time_sec, remediation_time_sec, remediation_name, policy_name,
policy_rule_name, status_text

FROM remediation_status WHERE remediation_time_sec <= UNIX_TIMESTAMP("2011-10-01
00:00:00")

ORDER BY policy_time_sec

DESC LIMIT 0, 25;
```

white_list_event

The **white_list_event** table contains allow list events that are generated when the system detects a host not compliant with a allow list in an active allow list compliance policy.

Note that starting in Version 5.0, the Secure Firewall records the detection of network and user activity at the managed device level, no longer by detection engine. The `detection_engine_name` and `detection_engine_uuid` fields in the **white_list_event** table now return only `null`, and queries that join on those fields return zero records. Querying on the `sensor_uuid` field instead of `detection_engine_uuid` provides the equivalent information.

For more information, see the following sections:

- white_list_event Fields, page 9-7
- white_list_event Joins, page 9-9
- white_list_event Sample Query, page 9-9

white_list_event Fields

The following table describes the database fields you can access in the **white_list_event** table.

Table 9-5 white_list_event Fields

Field	Description
`description`	A description of how the allow list was violated.
`detection_engine_name`	Field deprecated in Version 5.0. Returns `null` for all queries.
`detection_engine_uuid`	Field deprecated in Version 5.0. Returns `null` for all queries.

Table 9-5 white_list_event Fields (continued)

Field	Description
host_criticality	The user-assigned criticality of the host that is out of compliance with the allow list: None, Low, Medium, or High.
host_type	The host type: Host, Router, Bridge, NAT Device, or Load Balancer.
id	An internal unique identifier for the allow list event.
ip_address	Field deprecated in Version 5.2. Returns null for all queries.
ip_address_v6	Field deprecated in Version 5.2. Returns null for all queries.
ipaddr	A binary representation of the IP address of the non-compliant host.
os_product	The operating system's product name.
os_vendor	The operating system's vendor.
os_version	The operating system's version number.
policy_name	The violated compliance policy that includes the allow list.
policy_time_sec	The UNIX timestamp of the date and time the event was generated.
policy_uuid	A unique identifier for the compliance policy that includes the allow list event.
port	The port, if any, associated with the event that triggered a service allow list violation (that is, when a violation occurs as a result of a non-compliant service). For other types of allow list violations, the field is blank.
priority	The priority for the allow list event, which is set in the user interface.
protocol_name	The protocol associated with the event, if available.
protocol_num	The IANA-specified protocol number, if available.
rna_service	The service that triggered the allow list violation, if available.
sensor_address	IP address of the managed device that detected the traffic. Format is *ipv4_address,ipv6_address*.
sensor_name	The device that generated the allow list event.
sensor_uuid	A unique identifier for the managed device, or 0 if sensor_name is null.
user_dept	The department of the user.
user_email	The email address for the user.
user_first_name	The first name for the user.
user_id	Internal identification number of the user who last logged into the host before the event occurred.
user_last_name	The last name for the user.
user_last_seen_sec	The UNIX timestamp of the date and time the system last reported a login for the user.
user_last_updated_sec	The UNIX timestamp of the date and time the user's information was last updated.
user_name	The login user name for the user.
user_phone	The phone number for the user.
vlan_id	The VLAN identification number, if applicable.
white_list_name	The allow list that was violated.
white_list_uuid	A unique identifier for the allow list.

white_list_event Joins

The following table describes the joins you can perform on the **white_list_event** table.

Table 9-6 white_list_event Joins

You can join this table on...	And...
ipaddr	rna_host_ip_map.ipaddr user_ipaddr_history.ipaddr

white_list_event Sample Query

The following query returns up to 25 records generated before a specified time. The records include allow list event information such as the compliance policy name, timestamp the event was generated, allow list name, and so on.

```
SELECT policy_name, policy_time_sec, ipaddr, user_name, port, description,
white_list_name, priority, host_criticality, sensor_name

FROM white_list_event WHERE policy_time_sec <= UNIX_TIMESTAMP("2011-10-01 00:00:00")

ORDER BY policy_time_sec DESC LIMIT 0, 25;
```

white_list_violation

The **white_list_violation** table track compliance allow list violations, which track the ways that the hosts on your network violate the compliance allow lists in active compliance policies.

For more information, see the following sections:

- white_list_violation Fields, page 9-9
- white_list_violation Joins, page 9-10
- white_list_violation Sample Query, page 9-10

white_list_violation Fields

The following table describes the database fields you can access in the **white_list_violation** table.

Table 9-7 white_list_violation Fields

Field	Description
host_id	ID number of the host in violation of the allow list.
info	Any available vendor, product, or version information associated with the allow list violation. For protocols that violate an allow list, the field also indicates whether the violation is due to a network or transport protocol.
ip_address	Field deprecated in Version 5.2. Returns null for all queries.

Table 9-7 white_list_violation Fields (continued)

Field	Description
port	The port, if any, associated with the event that triggered a service allow list violation (that is, when a violation occurs as a result of a non-compliant service). For other types of allow list violations, the field is blank.
protocol_name	The protocol associated with the event.
type	The type of allow list violation, indicating whether the violation occurred due to a non-compliant: • operating system (os) • service (service) • client application (client app) • protocol (protocol)
violation_time_sec	The UNIX timestamp of the date and time the violation was logged.
white_list_name	The allow list that was violated.
white_list_uuid	A unique identifier for the allow list.

white_list_violation Joins

You cannot perform joins on the **white_list_violation** table.

white_list_violation Sample Query

The following query returns up to 25 records with allow list violation information such as the host IP address violating the allow list, the violated allow list name, and the count of violations.

```
SELECT host_id, white_list_name, count(*)
FROM white_list_violation
GROUP BY white_list_name, host_id
ORDER BY white_list_name
DESC LIMIT 0, 25;
```