Security Incidents Connector Configuration and Usage

Armor Knowledge Base

In this document:

Overview of Security Incidents Connectors

Security incidents, detections, and events are supported by the Armor webhook platform. Two configuration APIs are needed to set subscriptions of events, detections, and incidents: Transform Detections API and Notifications API. Subscription types which are defined can be enriched with default values, values from the Armor tags implementation, partner level overrides, and related to an Armor search direct link URL (if applicable). The platform also supports limited self service debugging through the use of an “email on failure” address whenever a target destination fails. Armor maintains a history of all webhook actions and can be contacted through a support ticket if further troubleshooting is required.

How to use the Detections Transform Config API

The Detections Transform Config API is used for each customer account which is subscribing to security detections, incidents, and/or events. For partners: All of the child accounts are automatically subscribed when creating a transform API request. The purpose of this API to two fold: to create the subscription in the Armor platform so that Armor will begin to process detections, incidents, and/or events for subscribed account (or child accounts) and to allow the customer to define additional property transforms to the outgoing webhook payload which may be relevant to the customer or partner environment (e.g. customer unique IDs or foreign keys). After the webhook payload processed it is sent to the delivery handler which will process the request through up to 25 defined destinations. Note: Only one transform can be defined per account.

> Whitelisting or blacklisting of child accounts at the parent level is not supported, all child accounts are automatically processed and require the end user to filter accounts or events if required.

The API reference is available at https://developer.armor.com/webhooks.

The API POST request body is broken into 6 parts: `org_id`, `name`, `format`, `events`, `default_label`, and `transform`.

- `name` - A short name describing the transform configuration
- `format` - Can be default, splunk, or ecs
- `events` - Contains a string array of types of data which will be added to the payload and sent to the webhook endpoints. For event ingestion platforms use “security event” to receive all correlated events which belong to a given offense, this could trigger multiple subsequent webhooks as more events could be correlated to a detection over time. For ticketing platforms it is common to only leverage incidents and detections to trigger ticket creation. All events are stored and searchable at https://amp.armor.com.
    - `security incident` - Armor incidents are security detections which have been processed by Armor to be escalated as a full incident.
    - `security detection`- Security detections are any correlated detection in the Armor platform which hasn’t been escalated to an incident.
    - `security event`- The raw events which belong to the correlated detection or incident.
- `default_label` - Use this section to define properties which are added to event, detection, or incident that is processed by the webhook platform. NOTE: This would apply to partner child account events, detections, or incidents which are processed by the partner parent account.
    - `"key": "[nested].key"` - Defines the name of the property that’s added to the payload object
    - `"value": "Alpha"` - The value that is placed in the property or nested property.
- `transform` - Use this section to define transforms which require the lookup of additional data to complete.
    - `add_dynamic_tag` - Supports the Armor Tags API to enrich events.
    - `remove_dynamic_tag` - Supports the ability to exclude properties or tags which may be defined by the customer or Armor and should not be populated during the transform process. (i.e. prevent customers from accidentally overwriting important partner defined values, exclude a built in property by Armor)
    - `add_log_search_url` - Supports events which have the “eventId” property and can add linkable Kibana queries directly to the correlated event.

```
{
  "name": "Hello world config",
  "format": "splunk",
  "event_type": [
    "security_incident",
    "security_detection",
    "security_event"
  ],
  "default_label": [
    {
      "key": "idx",
      "value": "Alpha"
    },
    {
      "key": "provider_id",
      "value": "Lotlux"
    },
    {
      "key": "customer_id",
      "value": "Cardify"
    },
    {
      "key": "account_id",
      "value": "Home"
    }
  ],
  "transform": [
    {
      "type": "add_dynamic_tag",
      "key": "index",
      "value": "{{index}}"
    },
    {
      "type": "remove_dynamic_tag",
      "key": "classificationTag",
      "value": null
    },
    {
      "type": "add_dynamic_tag",
      "key": "[event].provider_id",
      "value": "{{provider_id}}"
    },
    {
      "type": "add_dynamic_tag",
      "key": "[event].provider_type",
      "value": "{{provider_type}}"
    },
    {
      "type": "add_log_search_url",
      "key": "link_event_kibana",
      "value": "event_uuid"
    }
  ]
}
```

How to use the Notifications API

The Notifications API is used to define where events, detection, and incidents should be sent. There are several properties available to be configured for custom URL targets, headers, and email preferences in the event of an error. The notifications API also allows up to 25 notification configs to be installed. Each config can subscribe to one or more types defined in the API to allow for customization of target destinations based on the requirements of the customer.

The configuration is broken into 4 parts: `name`, `email_for_error`, `event_type` and properties. At release, email_for_error supports 'email' type only and properties only supports a single uri and up to 25 defined header properties . All headers are added to the webhook request automatically.

- `name` A short name describing the webhook configuration
- `on_error` Use this section to define how errors can be received if the destination is unavailable or inaccessible for delivery.
- `event_type` An array of security_incident, security_detecetion, and/or security_event. These event types are also defined in the Detection Transform config and are processed with each Notification config individually.

- `properties` This section contains the uri and header(s) for the target destination webhook.
    - `"type": "uri"` The full URI (with port and scheme) of the target destination webhook. NOTE Only one URI can be defined per Notification Config
    - `"type": "header"` Allows the control of headers to be placed in the request to the target destination. Headers are loaded “as is” with no attempt sanitize what gets placed in the request to allow for full customizability of any header formats.

```
{
  "name": "Test Create Notification Liza 1",
  "email_for_error": "test@armor.com",
  "event_type": ["security_incident"],
  "properties":
  [
          {
        "type": "uri",
        "key": "endpoint",
        "value": "http://www.google.com"
      },
      {
        "type": "header",
        "key": "Authorization",
        "value": "Splunk hec_access_token"
      }
  ]
}
```

Example Security Incidents Connector Payloads

Splunk Payload - Incident/Detection Only

```
{
    "time": 1590603925,
    "sourcetype": "_json",
    "event": {
        "accountId": 1000,
        "categories": [
            "Suspicious Activity",
            "System Informational"
        ],
        "credibility": 1,
        "description": "1156482 Armor - External - FIM - Task Scheduler Entries Modified",
        "documentType": "OFFENSE_DETECTION_1590603925345",
        "eventCount": 4,
        "flowCount": 0,
        "incidentId": "IR1000000",
        "lastUpdatedTime": 1590603925345,
        "magnitude": 3,
        "offenseId": 1000000,
        "offenseSource": "48905726-13c5-4b4e-92e3-3e59aa829f77",
        "offenseStatus": "OPEN",
        "offenseType": "Detection",
        "relevance": "1",
        "ruleGroups": null,
        "rules": [
            {
                "Id": "104208",
                "Name": "Armor - External - FIM - Task Scheduler Entries Modified",
                "Type": "CRE_RULE"
            }
        ],
        "severity": 3,
        "ticketId": null,
        "ticketStatus": null,
        "ticketUrl": null,
        "sourceIps": null,
        "tags": null,
        "classificationTag": null,
        "closeReason": null,
        "closeTime": null,
        "closingUser": null,
        "startTime": 1590603925
    }
}
```


Splunk Payload - Event(s) Only

```
]
    {
        "time": 1590673504,
        "sourcetype": "_json",
        "event": {
            "accountId": 1000,
            "category": "System Informational",
            "documentType": "EVENT_1590673504500_F97B30AC-3BD6-4040-B5B5-93C11825F231",
            "coreInstanceId": "eeeeee1-66f4-4344-bc76-229f2805a95f",
            "correlationDescription": "Task Scheduler Entries Modified",
            "destIp": null,
            "destPort": null,
            "eventId": "f97b30ac-3bd6-4040-b5b5-93c11825f231",
            "eventName": "Task Scheduler Entries Modified",
            "logSourceName": "Trend Micro Deep Security @ trend-fim",
            "logSourceType": "Trend Micro Deep Security",
            "payload": "<13>May 28 13:45:02 trend-fim LOGSTASH[-]: LEEF:2.0|Trend Micro|Deep Security Agent|12.
5.613|2006076|cat=Integrity Monitor\tname=Task Scheduler Entries Modified (ATT&CK T1168)\tdesc=Task Scheduler
Entries Modified (ATT&CK T1168)\tsev=6\tcn1=500\tcn1Label=Host ID\tdvchost=1000__eeeeeee1-66f4-4344-bc76-
229f2805a95f\tTrendMicroDsTenant=Primary\tTrendMicroDsTenantId=0\tact=created\tfilePath=cron
/21339\tsuser=root\tsproc=/usr/sbin/cron/794\tmsg=No description is available. AMD:1000:eeeeee1-66f4-4344-bc76-
229f2805a95f::f97b30ac-3bd6-4040-b5b5-93c11825f231:",
            "offenseId": 10000000,
            "sourceIp": "10.128.49.107",
            "sourcePort": 0,
            "startTime": 1590673504,
            "username": null
        }
    }
]
```


Default Payload - Incident/Detection Only

```
{
    "accountId": 1000,
    "categories": [
        "Suspicious Activity",
        "System Informational"
    ],
    "credibility": 1,
    "description": "1156482 Armor - External - FIM - Task Scheduler Entries Modified",
    "documentType": "OFFENSE_DETECTION_1590603925345",
    "eventCount": 4,
    "flowCount": 0,
    "incidentId": "IR1000000",
    "lastUpdatedTime": 1590603925345,
    "magnitude": 3,
    "offenseId": 1000000,
    "offenseSource": "48905726-13c5-4b4e-92e3-3e59aa829f77",
    "offenseStatus": "OPEN",
    "offenseType": "Detection",
    "relevance": "1",
    "ruleGroups": null,
    "rules": [
        {
            "Id": "104208",
            "Name": "Armor - External - FIM - Task Scheduler Entries Modified",
            "Type": "CRE_RULE"
        }
    ],
    "severity": 3,
    "ticketId": null,
    "ticketStatus": null,
    "ticketUrl": null,
    "sourceIps": null,
    "tags": null,
    "classificationTag": null,
    "closeReason": null,
    "closeTime": null,
    "closingUser": null,
    "startTime": 1590603925
}
```


Default Payload - Event(s) Only

```
[
    {
        "accountId": 1000,
        "category": "System Informational",
        "documentType": "EVENT_1590673504500_F97B30AC-3BD6-4040-B5B5-93C11825F231",
        "coreInstanceId": "eeeeee1-66f4-4344-bc76-229f2805a95f",
        "correlationDescription": "Task Scheduler Entries Modified",
        "destIp": null,
        "destPort": null,
        "eventId": "f97b30ac-3bd6-4040-b5b5-93c11825f231",
        "eventName": "Task Scheduler Entries Modified",
        "logSourceName": "Trend Micro Deep Security @ trend-fim",
        "logSourceType": "Trend Micro Deep Security",
        "payload": "<13>May 28 13:45:02 trend-fim LOGSTASH[-]: LEEF:2.0|Trend Micro|Deep Security Agent|12.5.613
|2006076|cat=Integrity Monitor\tname=Task Scheduler Entries Modified (ATT&CK T1168)\tdesc=Task Scheduler
Entries Modified (ATT&CK T1168)\tsev=6\tcn1=500\tcn1Label=Host ID\tdvchost=1000__eeeeee1-66f4-4344-bc76-
229f2805a95f\tTrendMicroDsTenant=Primary\tTrendMicroDsTenantId=0\tact=created\tfilePath=cron
/21339\tsuser=root\tsproc=/usr/sbin/cron/794\tmsg=No description is available. AMD:1000:eeeeee1-66f4-4344-bc76-
229f2805a95f::f97b30ac-3bd6-4040-b5b5-93c11825f231:",
        "offenseId": 10000000,
        "sourceIp": "10.128.49.107",
        "sourcePort": 0,
        "startTime": 1590673504,
        "username": null
    }
]
```


ECS Payload - Detection


```
{
  "index_type": "security-detections",
  "event": {
    "category": [
      "access denied",
      "misc suspicious event"
    ],
    "id": "117242",
    "kind": "alert",
    "code": "detection",
    "start": 1591799449,
    "end": 1591799452,
    "action": "AWS Cloud: Multiple Failed API Requests From Same Source IP",
    "severity": 3,
    "risk_score": 3,
    "type": "Detection",
    "EventCount": 17,
    "FlowCount": 0,
    "Modified": 1591799449433,
    "provider": "10.10.10.10",
    "url": "https://test.com"
  },
  "tenant_id": 997242,
  "source": {
    "ip": "192.168.1.1",
    "port": "80",
    "bytes": "2334",
    "packets": "3444"
  },
  "destination": {
    "ip": "27.7.108.187",
    "port": "80",
    "bytes": "2334",
    "packets": "3444"
  },
  "organization": {
    "id": 997242
  },
  "user": "username",
```

```
  "Description": "description",
  "rule": [
    {
      "id": "123",
      "name": "test name",
      "category": "test category"
    }
  ],
  "tags": [
    "tag_1",
    "tag_2"
  ],
  "related": {
    "Source": [
      "test1",
      "test2"
    ],
    "RuleGroup": [
      {
        "Id": "12223",
        "Name": "test name",
        "Type": "test type"
      }
    ],
    "ip": [
      "192.168.1.1",
      "27.7.108.187"
    ],
    "user": [
      "username"
    ]
  }
}
```

ECS Payload – Incident

```
{
  "index_type": "security-detections",
  "event": {
    "category": [
      "access denied",
      "misc suspicious event"
    ],
    "id": "117242",
    "kind": "alert",
    "code": "incident",
    "start": 1591799449,
    "end": 1591799452,
    "action": "AWS Cloud: Multiple Failed API Requests From Same Source IP",
    "severity": 3,
    "risk_score": 3,
    "type": "Incident",
    "EventCount": 17,
    "FlowCount": 0,
    "Modified": 1591799449433,
    "provider": "10.10.10.10",
    "url": "https://test.com"
  },
  "tenant_id": 997242,
  "source": {
    "ip": "192.168.1.1",
    "port": "80",
    "bytes": "2334",
    "packets": "3444"
  },
  "destination": {
    "ip": "27.7.108.187",
    "port": "80",
```

```
    "bytes": "2334",
    "packets": "3444"
  },
  "organization": {
    "id": 997242
  },
  "user": "username",
  "trace": {
    "Description": "description"
  },
  "rule": [
    {
      "id": "123",
      "name": "test name",
      "category": "test category"
    }
  ],
  "tags": [
    "tag_1",
    "tag_2"
  ],
  "related": {
    "Source": [
      "test1",
      "test2"
    ],
    "RuleGroup": [
      {
        "Id": "12223",
        "Name": "test name",
        "Type": "test type"
      }
    ],
    "ip": [
      "192.168.1.1",
      "27.7.108.187"
    ],
    "user": [
      "username"
    ]
  }
}
```

ECS Payload – Event(s)

```
[
  {
    "index_type": "security-detections",
    "event": {
      "category": [
        "authentication"
      ],
      "id": "d538cbeb-df7e-4bec-848c-9f8357342454",
      "kind": "event",
      "code": null,
      "start": 1576776880446,
      "reference": "23334",
      "original": "testing",
      "module": "module name",
      "action": "Failure Audit: An account failed to log on"
    },
    "tenant_id": 1024,
    "source": {
      "ip": "192.168.1.1",
      "port": "80",
      "bytes": "2334",
      "packets": "3444"
    },
    "destination": {
      "ip": "27.7.108.187",
      "port": "80",
      "bytes": "2334",
      "packets": "3444"
    },
    "organization": {
      "id": 997242
    },
    "external_id": "3e5c90a6-1472-4b1f-8038-ccf1711e4759",
    "user": {
      "name": "username"
    },
    "trace": {
      "Description": "failure audit: an account failed to log on."
    },
    "related": {
      "Source": [
        "test1",
        "test2"
      ],
      "RuleGroup": [
        {
          "Id": "12223",
          "Name": "test name",
          "Type": "test type"
        }
      ],
      "ip": [
        "192.168.1.1",
        "27.7.108.187"
      ],
      "user": [
        "username"
      ]
    }
  }
]
```

Was this helpful?

Your Rating: 

Results:  14 rates