



# Tenable.ot Syslog Integration Guide

**Version 3.6**

Copyright © Tenable 2020

All Rights Reserved

# Revision History

Product version: 3.6

Document revision history:

Document Revision	Date	Description
1.0	August 7, 2019	Created first version of documentation for Version 2.7
1.1	October 25, 2019	Updated for product version 3.1
1.2	December 22, 2019	Updated for product version 3.3
1.3	July 14, 2020	Updated for product version 3.6

# Table of Contents

Introduction .....	4
Document Purpose and Audience .....	4
Tenable.ot Events .....	4
Policy-based Detection.....	4
Anomaly Detection.....	5
Events.....	5
Syslog Notifications .....	5
Syslog Integration .....	6
Connecting Tenable.ot to a Syslog Server .....	6
Configuring Policies to Send Syslog Notifications.....	7
Interpreting Tenable.ot Log Entries .....	12
Header Parameters.....	13
Event Class IDs .....	14
Extension Parameters – CEF Keys.....	16
Standard CEF Keys.....	17
Tenable.ot Custom CEF Keys .....	20
Appendix – Syslog Sample .....	25

# Introduction

The Tenable.ot platform addresses operational blind-spots and security gaps by providing unmatched visibility into the OT segment of ICS network activity, with a unique focus on interpreting critical commands transmitted using industrial, vendor specific protocols. Tenable.ot is an agent-less solution that is deployed as an appliance (or as a virtual appliance) in the network, and offers tools and reports that benefit both security personnel and operations personnel.

Tenable.ot enables ICS engineering and security professionals to configure their own Policies to alert for specific unauthorized/important activities as well as for various anomalies in the ICS network.

You can configure Tenable.ot to send notifications using CEF protocol to an external Syslog server. This makes the pertinent information available in real-time to the relevant personnel who are responsible for implementing the necessary mitigation measures.

## Document Purpose and Audience

This document explains the information transmitted in the Syslog messages that are generated by Tenable.ot. This document helps the control room supervisors and other relevant personnel to accurately interpret the Syslog notifications. This document is also helpful for SIEM integration developers responsible for integrating the Tenable.ot notifications into their existing SIEM structure.

## Tenable.ot Events

Tenable.ot sends out Syslog notifications for Events that are detected by Tenable.ot. Events are generated when a policy-hit occurs for a *Policy* that is configured in the system. Therefore, it is important to understand how Tenable.ot's system of Policies and Events works. This is explained in the following sections.

## Policy-based Detection

For Policy-based detection, you configure the specific conditions for what events in the system trigger Event notifications. Policy-based Events are triggered only when the precise conditions of the policy are met. This ensures zero false positives, as the system alerts for actual events that take place in the ICS network, while providing meaningful detailed information about the 'who', 'what', 'when', 'where' and 'how'. The policies can be based on various Event types and descriptors. The following, are some examples of possible policy configurations:

- **Anomalous or unauthorized ICS control-plane activity (engineering):** for example, an HMI should not query the firmware version of a controller (may indicate reconnaissance), and a controller

should not be programmed during operational hours (may indicate unauthorized, potentially malicious activity).

- **Change to controller's code** – a change to the controller logic was identified (“Snapshot mismatch”).
- **Anomalous or unauthorized network communications:** for example, an un-allowed communication protocol was used between two network assets or a communication took place between two assets that have never communicated before.
- **Anomalous or unauthorized changes to the asset inventory:** for example, a new asset was discovered or an asset stopped communicating in the network.
- **Anomalous or unauthorized changes in asset properties:** for example, the asset firmware or state has changed.
- **Abnormal writes of set-points:** Events are generated for changes made to specific parameters. The user can define the allowed ranges for a parameter and generate Events for deviations from that range.

## Anomaly Detection

Anomaly Detection policies discover suspicious behavior in the network based on the system's built-in capabilities for detecting deviations from 'normal' activity. The following anomaly detection policies are available.

- **Deviations from a network traffic baseline:** the user defines a baseline of 'normal' network traffic based on the traffic map during a specified time range and generates alerts for deviations from the baseline. The baseline can be updated at any time.
- **Spike in Network Traffic:** a dramatic increase in the volume of network traffic or number of conversations is detected.
- **Potential network reconnaissance/cyber-attack activity:** Events are generated for activities indicative of reconnaissance or cyber-attack activity in the network, such as IP conflicts, TCP port scans and ARP scans.

## Events

When an event occurs that matches the conditions of a Policy, an Event is generated in the system. Events are configured as part of the Policy configuration with the following parameters:

- **Event Severity** – the severity of the alert. For example, Low severity Events may require attention at any time, while High severity Events should be addressed immediately.
- **Alert Notification** – Events can be configured to notify a Syslog server. In addition, Events can be configured to send email notifications to a recipient or a group of recipients.

## Syslog Notifications

In summary, a Syslog notification is sent under the following circumstances:

- An Event occurred that constituted a policy-hit for a Policy that is activated in Tenable.ot, and
- The configuration for that Policy was set to send notifications to your Syslog server.

# Syslog Integration

## Connecting Tenable.ot to a Syslog Server

In order to enable sending of log events to an external server you must set up a *Syslog Server* in the system. The following procedure describes how to set up a Syslog Server using the Management Console UI.

### ➔ To set up a Syslog Server:

1. In the **Management Console UI**, under **Local Settings**, go to **Servers > Syslog Servers** screen.
2. Click **+ Add Syslog Server**.

The **Syslog Servers** configuration window is displayed.

The screenshot shows a configuration window titled "Syslog Servers". It contains the following fields and controls:

- Server Name \***: A text input field.
- Hostname / IP \***: A text input field.
- Port \***: A text input field containing the value "25".
- Transport \***: A dropdown menu with "Select" and a downward arrow.
- At the bottom of the form are two buttons: "Cancel" and "Create".
- Below the form is a green link: "+ Add Syslog Server".

3. In the **Server Name** field, enter the name of a Syslog Server to be used for logging system events.
4. In the **Hostname/IP** field, enter a host name or an IP address of the Syslog server.
5. In the **Port** field, enter the port number on the Syslog server to which the Events will be sent. (default: 514)
6. In the **Transport** field, select from the dropdown list the transport protocol to be used. Options are: *TCP* or *UDP*.
7. If you would like to send a test message to verify that the configuration was successful, click **Send Test Message**, and check if the message has arrived. If the message did not arrive, then troubleshoot to discover the cause of the problem and correct it.
8. Click **Save**.  
You can set up additional Syslog Servers by repeating the procedure described above.

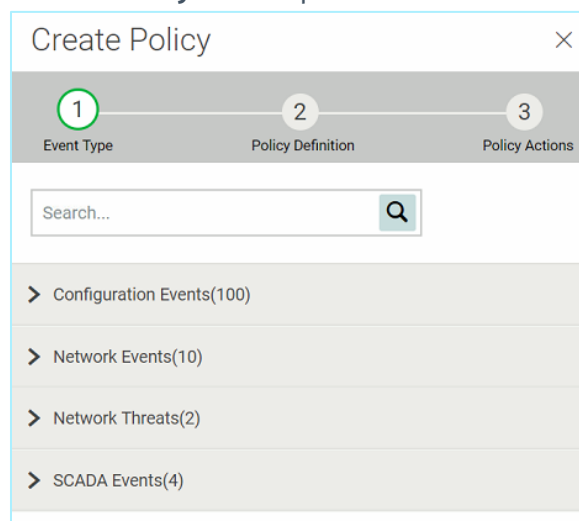
## Configuring Policies to Send Syslog Notifications

Once a Syslog server has been configured in Tenable.ot, you can configure specific Policies to send notifications to the Syslog server. When creating new Policies, you can select the Syslog server as a destination for the alert notifications. You can also edit existing Policies to add the Syslog server as a destination for alert notifications. **You must ensure that the relevant Policy is turned ON in order for alerts to be generated.**

### ➡ To add a Syslog server during Policy creation:

1. In the **Management Console UI**, go to **Policies** screen.
2. Click **Create Policy**.

The **New Policy** wizard opens.



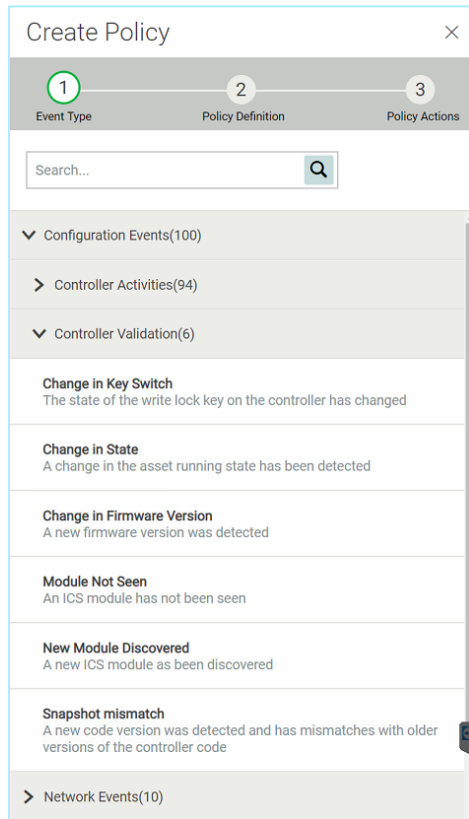
Create Policy

1 Event Type 2 Policy Definition 3 Policy Actions

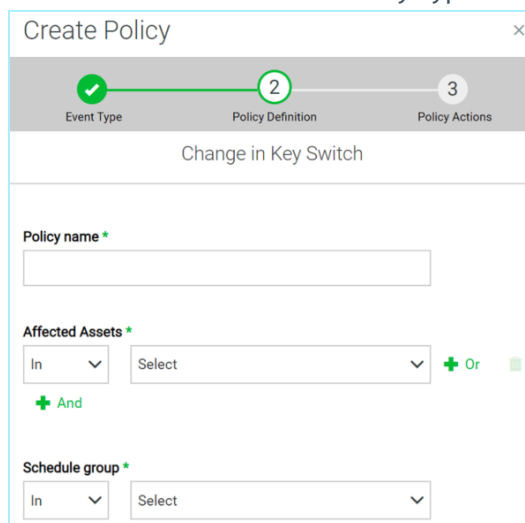
Search...

- > Configuration Events(100)
- > Network Events(10)
- > Network Threats(2)
- > SCADA Events(4)

- Click on a **Policy Category** to show the sub-categories and/or Policy Types. A list of all sub-categories and/or Types included in that category are displayed.



- Select a **Policy Type**.
- Click **Next**. A Series of parameters for defining the Policy are displayed. This includes all relevant Policy conditions for the selected Policy Type.



- Fill in all relevant fields according to your specifications. (See *Tenable.ot User Guide*)



- Once all fields have been filled in, click **Next**.  
A series of Policy Action parameters (i.e. the actions taken by the system when a Policy hit occurs) are shown.

Create Policy

Event Type Policy Definition Policy Actions 3

Change in Key Switch

Event severity \*

High Medium Low None

Syslog

Plant A

Plant B

Email group

SMTP servers are not configured

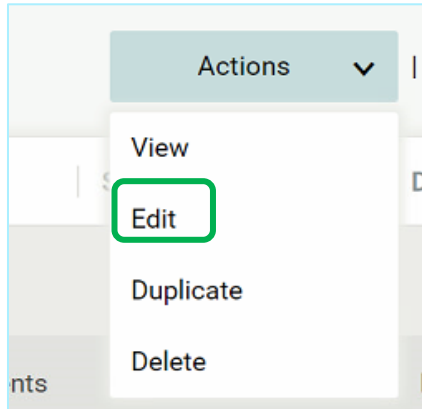
- Fill in all relevant fields according to your specifications. (See *Tenable.ot User Guide*)
- To send the Event logs to one or more Syslog servers, in the **Syslog** section, select the checkbox next to each server to which you would like to send the Event logs.
- Once all fields have been filled in, click **Create**.



Once a new Policy is created it is automatically activated so that policy-hits will generate alerts.

➔ To add a Syslog server to an existing Policy:

1. In the **Management Console UI**, on the **Policies** screen, select the desired Policy. (See *Tenable.ot User Guide*)
2. Click on the **Actions** menu and select **Edit** from the dropdown list.



The **Edit Policy** screen is shown with the current configuration filled in.

A screenshot of the 'Edit Policy' configuration screen. The screen is titled 'Edit Policy' and has a close button (X) in the top right corner. Below the title bar, there are two tabs: 'Policy Definition' (marked with a circled '1') and 'Policy Actions' (marked with a circled '2'). The current policy name is 'Modbus Illegal Data Value'. The configuration fields are as follows:

- Policy name \***: Text input field containing 'Modbus Exception Occurred: Illegal Data Value'.
- Source \***: Two dropdown menus. The first is set to 'In' and the second is set to 'Any Asset'.
- Destination \***: Two dropdown menus. The first is set to 'In' and the second is set to 'Any Asset'.
- Schedule \***: Two dropdown menus. The first is set to 'In' and the second is set to 'Any Time'.

3. Click **Next**.

A series of Policy Action parameters (i.e. the actions taken by the system when a Policy hit occurs) are shown.

Edit Policy

Policy Definition Policy Actions

SIMATIC Firmware Download

Event severity \*

High Medium Low None

Syslog

Plant A

Plant B

Email group

SMTP servers are not configured

Additional Actions \*

Take snapshot after policy hit

< Back Cancel Save

4. To send the event logs to one or more Syslog servers, in the **Syslog** section, select the checkbox next to each server to which you would like to send the event logs.5. Click **Save**.

The Policy is saved with the new configuration.

# Interpreting Tenable.ot Log Entries

The following sample shows the breakdown of the elements included in a Syslog log entry received from Tenable.ot.

## Syslog Prefix

Timestamp

```
2019-09-22T11:31:27.791Z
```

Source IP

```
10.100.20.10
```

## Message

Header

```
<12>Sep 22 11:32:56 10.100.20.10 CEF:0|Indegy|Indegy Security  
Platform|3.0.32|114|Open Port|7|
```

Extension

```
dvchost=indegy rt=Sep 22 2019 11:32:56 suser=longrun1.indegy.local  
proto=TCP src=10.100.20.200 spt=80 cs6Label=policy_name  
cs6=Commonly exploited ports cat=NetworkEvents
```

Each Log entry consists of the **Syslog Prefix** which is generated by the Syslog Server and the **Message** body which is given using CEF protocol.

- The **Syslog Prefix** consists of the following elements:
  - **Timestamp** – the date and time that the log entry was generated, given in the following format: yyyy-MM-dd'T'HH:mm:ss.ZZZ'Z'
  - **Source IP** – the IP of the Tenable.ot appliance that generated the log entry.
- The **Message** body consists of the following elements:
  - **Header** – a fixed set of fields that indicate the source of the log and the nature of the Event that generated the log. The information returned for each of these parameters is described in **HEADER PARAMETERS**.
  - **Extension** – this field provides additional details about the precise nature of the Event. This field consists of a series of key-value pairs which vary according to the type of alert. A list of fields included in the extension for each Event type is shown in **EVENT CLASS IDS**.

Each data element is separated by a pipe (|) character. You can see a few sample Tenable.ot Syslog messages in **APPENDIX – SYSLOG SAMPLE**.

## Header Parameters

The following table describes the values shown in the Header section of CEF messages generated by Tenable.ot.

Parameter	General Description	Tenable.ot Value										
Timestamp	The date and time that the log entry was generated, given in the following format: MMM dd yyyy HH:mm:ss	e.g. <12>Sep 24 14:03:44										
Source IP	The IP of the host that sent the Syslog message.	The IP of the Tenable.ot appliance that sent the log entry.										
CEF:Version	The mandatory prefix 'CEF:' followed by the CEF version number.	CEF:0										
Device Vendor	The vendor name for the sending device.	Indegy										
Device Product	The product name of the sending device.	Indegy Security Platform (Note: this refers to the <i>Tenable.ot appliance</i> )										
Device Version	The product version of the sending device.	The version of the Tenable.ot appliance that is in use. For example, 3.0.32										
Device Event Class ID	A unique identifier for each Event type. This can be a string or an integer. Device Event Class ID identifies the type of Event reported.	Tenable.ot produces log entries with distinct Event Classes for each type of Event that generates alerts. The meaning of each Event Class ID is explained in section <b>EVENT CLASS IDS.</b>										
Name	The name of the Event Class.	Tenable.ot provides a descriptive name corresponding to each of the Event Class IDs, see <b>EVENT CLASS IDS.</b>										
Severity	A string or integer that reflects the importance of the Event. The valid string values are Unknown, Low, Medium, High, and Very-High. The valid integer values are 0-3=Low, 4-6=Medium, 7-8=High, and 9-10=Very-High.	<table border="1"> <thead> <tr> <th>Tenable.ot Event Severity</th> <th>CEF Value</th> </tr> </thead> <tbody> <tr> <td>None</td> <td>0</td> </tr> <tr> <td>Low</td> <td>3</td> </tr> <tr> <td>Medium</td> <td>7</td> </tr> <tr> <td>High</td> <td>9</td> </tr> </tbody> </table>	Tenable.ot Event Severity	CEF Value	None	0	Low	3	Medium	7	High	9
Tenable.ot Event Severity	CEF Value											
None	0											
Low	3											
Medium	7											
High	9											

## Event Class IDs

The following table describes the meaning of each Event Class ID and the data included in the extension for that Event type. For an explanation of how Tenable.ot uses each of the CEF fields included in the Extensions see **EXTENSION PARAMETERS – CEF KEYS**.

ID	Name	Description
0	Unknown	Unidentified alert Event.
01-94	[the type of activity that occurred]	A command to execute a particular type of activity was sent to a controller in the network. For example, the Event type 'Rockwell Code Download' indicates that a code download command was sent to a Rockwell controller.
95	New Asset Discovered	A new asset was detected in the network.
96	New Module	A new module was added to a backplane in the network.
97	IP Conflict	Multiple assets in the network are using the same IP address.
98	ARP Scan Detected	An ARP scan (indicative of reconnaissance activity) was detected in the network.
99	SYN Scan Detected	A SYN scan (indicative of reconnaissance activity) was detected in the network.
100-107	Inactive Asset for [time period] Minutes	An asset was inactive in the network for the specified period of time. Possible time periods are: 15 minutes, 30 minutes, one hour, 3 hours, 12 hours, one day or one week.
108	Snapshot Mismatch	The most recent Snapshot (which captures the current state of the program deployed on a controller) of a controller was not identical to the previous Snapshot of that controller.
109	Unauthorized Conversation	A conversation was detected (between the specified assets) in the network.
110	Change in State	The controller changed from one operational state (e.g. running, stopped, test etc.) to another.
111	Change in Key State	A change was made to the controller state by adjusting the physical key position.
112	Change in FW Version	A change was made to the firmware running on the controller.

ID	Name	Description
113	Unauthorized Tag Write	An unauthorized CIP write of tag values on a controller was detected.
114	Open Port	A new open port was detected in your network.
115	Baseline Deviation	A new connection was detected between assets that did not communicate with each other during the Network Baseline sampling period.
116	Module not Seen	A previously identified module is no longer detected on its backplane.
117	RDP Connection With Authentication	An RDP (Remote Desktop Protocol) connection was made between assets in your network, using authentication credentials.
118	RDP Connection Without Authentication	An RDP (Remote Desktop Protocol) connection was made between assets in your network, without using authentication credentials.
119	Intrusion Detected	Network traffic indicative of intrusion threats was detected, based on Suricata Emerging Threat rules.
120	Modbus Exception Occurred: Illegal Function	An "illegal function" error code was detected in Modbus protocol.
121	Modbus Exception Occurred: Illegal Data Address	An "illegal data address" error code was detected in Modbus protocol.
122	Modbus Exception Occurred: Illegal Data Value	An "illegal data address" error code was detected in Modbus protocol.
123	Traffic Data Spike Detected	A dramatic increase in the volume of network traffic was detected.
124	Traffic Conversation Count Spike Detected	A dramatic increase in the number of conversations was detected.

ID	Name	Description
125	Change in Windows USB State	A USB device was connected to or removed from a Windows based workstation.
126-143	[the type of activity that occurred]	A command to execute a particular type of activity was sent to a controller in the network. For example, the Event type 'Rockwell Code Download' indicates that a code download command was sent to a Rockwell controller.
144-166	DNP3 [the type of Event that occurred]	A command was sent using DNP3 protocol, e.g. select, operate, warm/cold restart etc. Also detects errors originating from internal indicators such as unsupported function codes and parameter errors.
167-169	Honeywell [the type of Event that occurred]	A command was sent to a Honeywell PLC causing a state change: warm restart, cold restart or stop.
170, 171	FTP [the type of Event that occurred]	An attempt to login using FTP protocol (failed or successful).
172-174	Telnet [the type of Event that occurred]	An attempt to login using Telnet protocol (successful, failed or not detected).

## Extension Parameters – CEF Keys

There are two types of CEF Keys:

- **Standard Keys** - which have a fixed meaning for log entries from any source that uses CEF protocol.
- **Custom Keys** – which are customized for use by each source that generates CEF log entries.

The following sections describe the meaning of each CEF key used by Tenable.ot.



## Standard CEF Keys

The following table explains the meaning of the standard CEF keys that are used by Tenable.ot. For a complete listing of CEF keys see [CommonEventFormatV25](#).

CEF Key Name	Full Name	Data Type	Length	Description
cat	deviceEventCategory	String	1023	<p>Shows the general category of the Event.</p> <p>The following is a description of the possible Event Categories.</p> <ul style="list-style-type: none"> <li>• Configuration Events – this includes two sub-categories</li> <li>• Controller Validation Events – These policies detect changes that take place in the controllers in the network.</li> <li>• Controller Activity Events – Activity Policies relate to the Activities that occur in the network (i.e. the “commands” implemented between assets in the network).</li> <li>• SCADA Events – policies that identify changes made to the data plane of controllers.</li> <li>• Network Threats Events – these Policies identify network traffic that is indicative of intrusion threats.</li> <li>• Network Events – Policies that relate to the assets in the network and the communication streams between assets.</li> </ul>
duser	destinationUserName	String	1023	The name of the destination asset which received the activity. This value can be any name by which Tenable.ot identifies the asset, such as a user defined name, the DNS name, the IP address etc.
dvchost	devicehostname	String	100	The device that sent the log entry. For Tenable.ot logs the value is 'Indegy'.

CEF Key Name	Full Name	Data Type	Length	Description
<b>dst</b>	destinationAddress	IPv4 Address		The IP address of the destination asset which received the activity. The format is an IPv4 address. Example: "192.168.10.1"
<b>dpt</b>	destinationPort	Integer		The port on the destination asset which received the activity. Valid port numbers are between 0 and 65535.
<b>externalId</b>	externalId	String	40	The Log ID used by Tenable.ot to refer to the Event.
<b>in</b>	bytesIn	Integer		The volume of data transferred from the source asset to the destination asset during the Event (in bytes).
<b>outcome</b>	eventOutcome	String	63	Displays the outcome of the Event. For example, "success" or "failure".
<b>proto</b>	transportProtocol	String	31	Identifies the Layer-4 protocol used for the activity. The possible values are protocols such as TCP or UDP.
<b>rt</b>	deviceRecipientTime	Time Stamp		The date and time at which the Event was registered in Tenable.ot. The format is MMM dd yyyy HH:mm:ss.
<b>smac</b>	sourceMacAddress	MAC address		The MAC address of the source <sup>1</sup> asset that initiated the activity. The format is six colon-separated hexadecimal numbers. Example: "00:0D:60:AF:1B:61"

---

<sup>1</sup> For Events that do not involve communication between a source and destination asset, all "source" fields refer to the asset affected by the Event.

CEF Key Name	Full Name	Data Type	Length	Description
<b>dmac</b>	destinationMacAddress	MAC address		The MAC address of the destination asset that received the activity. The format is six colon-separated hexadecimal numbers. Example: "00:0D:60:AF:1B:61"
<b>spt</b>	sourcePort	Integer		The port involved in the Event. Used in Open Port Events to show the open port that was discovered. Valid port numbers are 0 to 65535.
<b>src</b>	sourceAddress	IPv4 Address		The IP address of the source <sup>1</sup> asset which initiated the activity. The format is an IPv4 address. Example: "192.168.10.1"
<b>suser</b>	sourceUserName	String	1023	The name of the source <sup>1</sup> asset which initiated the activity. This value can be any name by which Tenable.ot identifies the asset, such as a user defined name, the DNS name, the IP address etc.
<b>msg</b>	message	String	1023	A message with additional details about the Event. Used for anomaly detection Events. Example: msg= :: ET SCAN NMAP -sS window 1024

## Tenable.ot Custom CEF Keys

The following table explains the meaning of the custom CEF Keys that are used by Tenable.ot in the Extension section of log entries. For each custom key there is a corresponding *Label* key which describes the purpose of that key as used by Tenable.ot.

CEF Key Name	Full Name	Data Type	Length	Description
cn1	deviceCustomNumber1	Long		A custom number field. Tenable.ot uses this field for 'Snapshot Diff detected' Events, to show which revision number didn't match the previous revision. Format: "cn1=%d"
cn1Label	deviceCustomNumber1Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cn1Label="revision".
cn2	deviceCustomNumber2	Long		A custom number field. Tenable.ot uses this field for 'Firmware Version Change detected' Events, to show which backplane slot the firmware change occurred on. Format: "cn2=%d"
cn2Label	deviceCustomNumber2Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cn2Label="bpslot".

CEF Key Name	Full Name	Data Type	Length	Description
cn3	deviceCustomNumber2	Long		A custom number field. Tenable.ot uses this field for "Intrusion Detection" Events to show the ID of the Vulnerability (in the CVE listing) that was detected. Format: "cn3=%d"
cn3Label	deviceCustomNumber3Label	String	1023	The label field that describes the purpose of the corresponding custom field. For tenable.ot, cn3Label="rule_sid".
cs1	deviceCustomString1	String	4000	A custom string field. Tenable.ot uses this field for 'Controller State Change detected' and 'Controller Key State Change detected' Events, to show the old and new states of the controller. Format: "cs1=%s->%s" (old status->new status, e.g. "running->stopped")
cs1Label	deviceCustomString1Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs1Label="value_change".

CEF Key Name	Full Name	Data Type	Length	Description
cs2	deviceCustomString2	String	4000	A custom string field. Tenable.ot uses this field for 'Tag Write Values detected' Events, to show the tags that were written to and the values that were written. Format: "cs2=%s:%s", (tag name:tag value)
cs2Label	deviceCustomString2Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs2Label="tag".
cs3	deviceCustomString3	String	4000	A custom string field. Tenable.ot uses this field for 'New Module detected' Events, to show the name of the Backplane to which the module was added. Format: "cs3=%s"
cs3Label	deviceCustomString3Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs3Label="Bpname".
cs4	deviceCustomString4	String	4000	A custom string field. Tenable.ot uses this field for 'IP Conflict detected' and 'ARP Scan detected' Events, to show the IP addresses involved. Format: "cs4=%s"

CEF Key Name	Full Name	Data Type	Length	Description
cs4Label	deviceCustomString4Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs4Label="addresses".
cs5	deviceCustomString5	String	4000	A custom string field. Tenable.ot uses this field for 'SYN Scan detected' Events, to show the involved ports. Format: "cs5=%s"
cs5Label	deviceCustomString5Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs5Label="ports".
cs6	deviceCustomString6	String	4000	A custom string field. Tenable.ot uses this field to show the name of the Policy that generated the Event. Format: "cs6=%s"
cs6Label	deviceCustomString6Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, cs6Label="policy_name".

CEF Key Name	Full Name	Data Type	Length	Description
deviceCustomDate1	deviceCustomDate1	TimeStam p		A custom TimeStamp field. Tenable.ot uses this field for 'inactive asset' Events, to show the date and time that the asset was last active. Format: "last deviceCustomDate1=%s"
deviceCustomDate1Label	deviceCustomDate1Label	String	1023	The label field that describes the purpose of the corresponding custom field. For Tenable.ot, deviceCustomDate1Label = "last".



## Appendix – Syslog Sample

The following table shows a few sample listings of Tenable.ot's Syslog messages.

2019-09-24T13:19:39.847Z	10.100.20.10	<10>Sep 24 13:19:53 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 99 SYN Scan Detected 9 dvchost=indegy rt=Sep 24 2019 13:19:53 duser=ICS Device #104 suser=Eng. Station #236 proto=TCP externalId=15970 dst=10.100.103.20 src=10.100.20.117 cs5Label=ports cs5=6,497,23502,135,445,5900,25,110,80,993,22,3306,554,443,23 (423 more) cs6Label=policy_name cs6=SYN Scan Detected cat=NetworkThreats
2019-09-24T11:32:16.366Z	10.100.20.10	<12>Sep 24 11:32:43 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 109 Unauthorized Conversation 7 dvchost=indegy rt=Sep 24 2019 11:32:43 duser=8.8.8.8 suser=HIS0864 proto=DNS externalId=15751 dst=8.8.8.8 src=10.100.108.150 dpt=53 cs6Label=policy_name cs6=Communication to External Network cat=NetworkEvents
2019-09-24T08:33:51.623Z	10.100.20.10	<14>Sep 24 08:35:19 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 24 Modicon Online Session 3 dvchost=indegy rt=Sep 24 2019 08:35:19 duser=Controller #11 suser=INDEGY-XP outcome=success dst=10.100.105.24 dmac=00:00:54:10:5d:f8 src=10.100.105.152 smac=00:50:56:83:b4:bf cs6Label=policy_name cs6=Modicon Online Session cat=ConfigurationEvents
2019-09-24T13:41:39.931Z	10.100.20.10	<10>Sep 24 13:42:19 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 99 SYN Scan Detected 9 dvchost=indegy rt=Sep 24 2019 13:42:19 duser=CPU 412-2 PN/DP,S7-400 station_1 suser=robocop.indegy.local proto=TCP externalId=16015 dst=10.100.102.33 src=10.100.20.37 cs5Label=ports cs5=6,497,23502,1025,993,53,443,23,995,1723,25,445,3306,554,5900 (423 more) cs6Label=policy_name cs6=SYN Scan Detected cat=NetworkThreats
2019-09-24T13:19:39.861Z	10.100.20.10	<10>Sep 24 13:19:52 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 99 SYN Scan Detected 9 dvchost=indegy rt=Sep 24 2019 13:19:52 duser=SD_PLC suser=INDGY_MAIN_FW.il.indegy.com proto=TCP externalId=15971 dst=10.4.0.10 src=10.4.0.1 cs5Label=ports cs5=62078,1271,7937,1783,1864,1060,416,636,5298,6,497,23502,25,3306,995 (485 more) cs6Label=policy_name cs6=SYN Scan Detected cat=NetworkThreats
2019-09-24T11:31:21.583Z	10.100.20.10	<12>Sep 24 11:32:49 10.100.20.10 CEF:0 Indegy Indegy Security Platform 3.0.32 114 Open Port 7 dvchost=indegy rt=Sep 24 2019 11:32:49 suser=Eng. Station #112 proto=TCP src=10.100.20.25 spt=443 cs6Label=policy_name cs6=Commonly exploited ports cat=NetworkEvents