[RETIRED] Cribl HF Workflow

NOTE: This document has been retired and moved to a ServiceNow Knowledge Base (KB) article. Please find the latest documentation at the following link:

ServiceNow

Purpose

To support engineers with configuring Splunk Heavy Forwarders (`HF`s) to route data to Cribl Cloud, from which the data will be routed to the customers Splunk Index Cluster (`IDX Cluster`).

Supporting Documentation/References

deepwatch/dw-hf_outputs

deepwatch's Official Stance on Cribl

MD Owned by Micah Donald (Deactivated) • Updated on May 31, 2022

Cribl is a third-party software that provides a way to streamline log ingestion and help enable integration teams to quickly tune data sources. Cribl currently doesn't provide much more functionality than the Heavy Forwarders do in a customer's environment. Because of our current architecture utilizing Heavy Forwarders, the value prop is greatly di

1 4

Confluence Open preview

Splunk TCP Source | Cribl Docs

Receive Splunk data from Universal or Heavy Forwarders

docs.cribl.io

#

We couldn't load this link for an unknown reason.

Configuration

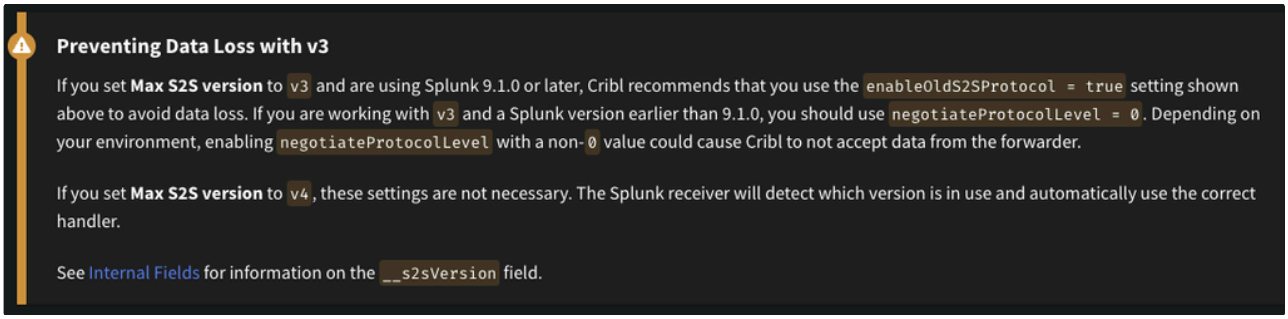
Data forwarding should be initially performed as a duplicate outputs configuration. Cribl Support can configure data streams to drop to a Null Queue, which will prevent those duplicate events from reaching Splunk and subsequently incurring additional license usage. Once the `HF -> Cribl -> IDX cluster` data ingestion pipeline is intact and validated, the `HF -> IDX Cluster` pipeline can be disabled with the exception of Splunk indexes (e.g. `_internal, _audit, _introspection`, etc). Test

> The following was written in mind for a Deepwatch Cloud customer. Should the customer leverage Splunk Cloud, or have a fully customer on-premise environment, reach out to a Principal SIEM Engineer for additional guidance.

Requirements

Customer

- `HF` has a route to the public Internet
- `HF -> Cribl Cloud` 9997/tcp communication is allowed on any potential interfering customer firewalls
- S2S v4 is selected in the Cribl SplunkTCP input configuration



Deepwatch

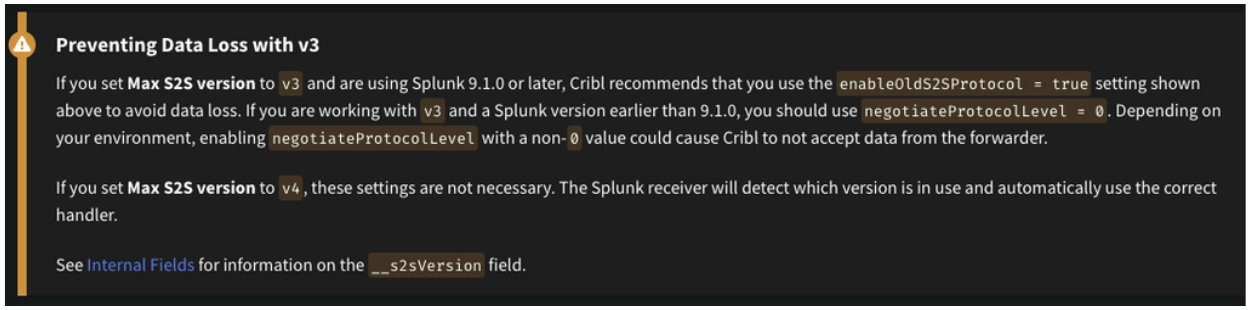
- `Cribl Cloud -> IDX Cluster` 9997/tcp communication is allowed in AWS.
- Provide certificates used to encrypt `HF -> IDX Cluster` 9997/tcp communication to Cribl Support.

Heavy Forwarder

outputs.conf

- Reference Splunk TCP Source | Cribl Docs to add an additional outputs stanza on the desired Splunk heavy forwarder, modifying the `<tenant-ID>` (provided by the customer or Cribl cloud support representative) to the tenant Cribl cloud instance as necessary:

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/outputs.conf




```
[tcpout:cribl_cloud]
server = in.logstream.<tenant-ID>.cribl.cloud:9997
```

```
useSSL = true
sslRootCAPath = $SPLUNK_HOME/etc/auth/cacert.pem
sendCookedData = true

# Edit below as necessary for compatability with any variation of Splunk HF Version / Cribl Max S2S Version
#enableOldS2SProtocol = true
#negotiateProtocolLevel = 0
```

- Ensure `defaultGroup` is set to null/blank in `dw-hf_outputs/local/outputs.conf`

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/outputs.conf

```
[tcpout]
defaultGroup =
```

This will force Splunk services to force it's data forwarding decision to leverage the props/transforms.

Props.conf

Scenario 1: Complete Data Duplication

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/props.conf

```
[default]
TRANSFORMS-routing = replicate_to_cribl
```

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/transforms.conf

```
[replicate_to_cribl]
REGEX = (.)
DEST_KEY = _TCP_ROUTING
FORMAT = cribl_cloud,primary_indexers
```

Scenario 2: All Production Events to only Cribl

Once the `HF -> Cribl -> IDX Cluster` log ingestion pipeline has been validated and all data sources have been vetted/tuned by Cribl cloud support, the `HF -> IDX Cluster` pipeline can be disabled **with the exception** of Splunk-native indexes:

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/props.conf

```
[default]
TRANSFORMS-routing = dw_cloud, cribl_cloud
```

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/transforms.conf

The below configuration will . This architecture is recommended to ensure Deepwatch retains the integrity of Splunk-native indexes, which is crucial for troubleshooting. It will require that the customer retain network configurations to allow `HF -> IDX Cluster` 9997/tcp communication.

```
[dw_cloud]
### Keep "_*" and "deepwatch" indexes forwarding directly to Splunk ###
SOURCE_KEY = _MetaData:Index
REGEX = (^deepwatch$|^_.*$)
DEST_KEY = _TCP_ROUTING
FORMAT = primary_indexers

[cribl_cloud]
### Route Production indexes to Cribl ###
SOURCE_KEY = _MetaData:Index
```

```
REGEX = ^(?!^deepwatch$|^_.+$)(^.+$)$
DEST_KEY = _TCP_ROUTING
FORMAT = cribl_cloud
```

Scenario 3: Selective Routing

Customers may wish to iteratively clone data to both Deepwatch Cloud and Cribl Cloud. The easiest known and tested method to selectively clone data is to leverage three transforms stanzas.

> The below example defines a situation in which both the Deepwatch Detection engineers and the customer have come to an agreement that the data contained in the `wineventlog` index (routing through Cribl) is healthy and can be consumed by the MDR service for alerting.

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/props.conf

```
[default]
TRANSFORMS-routing = cribl_dwcloud,cribl_cloud,dw_cloud
```

$SPLUNK_HOME/etc/apps/dw-hf_outputs/local/transforms.conf

```
[cribl_dwcloud]
# Clone data with the exception of 'wineventlog', 'deepwatch', and '_*' indexes
SOURCE_KEY = _MetaData:Index
REGEX = ^(?!^wineventlog$|^deepwatch$|^_.*$)(^.+$)$
DEST_KEY = _TCP_ROUTING
FORMAT = cribl_cloud, primary_indexers

[cribl_cloud]
### Route validated production indexes only to Cribl ###
SOURCE_KEY = _MetaData:Index
REGEX = (^wineventlog$)
DEST_KEY = _TCP_ROUTING
FORMAT = cribl_cloud

[dw_cloud]
### Keep "_*" and "deepwatch" indexes forwarding directly to Splunk ###
SOURCE_KEY = _MetaData:Index
REGEX = (^deepwatch$|^_.*$)
DEST_KEY = _TCP_ROUTING
FORMAT = primary_indexers
```

AWS

The Deepwatch Cloud AWS Indexer security group will need to be modified to allow inbound 9997/tcp communication from their Cribl environment. Reach out to the customer and/or a Cribl support representative to acquire the expected source IPs of Cribl data, and add the AWS Indexer security group entries as necessary

> Cribl Cloud does not maintain static egress IP address for forwarding data. Cribl has acknowledged this issue, and is taking steps to ensure static IP egress addresses. Until Cribl solves for this, bear in mind these IP addresses may change without notice.

Health Monitor Whitelist

The latest updates to https://bitbucket.org/deepwatch/ta-dw-infra/src/main/ Can't find link https://bitbucket.org/deepwatch/ta-sc-infra/src/main/?search_id=b88d48ee-05ed-4f44-88aa-781025468328 Can't find link have a default enabled macro to omit Cribl instances from triggering `dwo_infa_00007: Missing Instance`. No additional whitelisting is required.

```
[dwo_infa_00007_cribl_exclusions]
definition = version IN ("4.*", "5.*", "6.*")
iseval = 0
```

Validation

To ensure that the changes implemented are as intended, perform some searches on the customer's search head to surrounding the `splunkd` sourcetype, the state of the HF's queues, and `tcpout` metrics data.

tcpout Connections

```
index=_internal host="<hf>" source="metrics.log" sourcetype=splunkd "group=tcpout_connections"
| rex field=name "^(?<output_group>[^:])"
| eval tcp_MBps = round((tcp_KBps/1024),2)
| timechart useother=false max(tcp_MBps) by output_group
```

Queues 1

```
index=_internal host=* sourcetype=splunkd group=queue (name=)
| eval blocked=if(blocked=="true",1,0), queued_host=host." - ".name
| stats sparkline sum(blocked) as blocked,count by queued_host
| eval blocked_ratio=round(blocked/count100,2)
| sort - blocked_ratio
| eval
Finding=case(blocked_ratio>50.0,"Critical",blocked_ratio>40.0,"Warning",blocked_ratio>20.0,"Low",1=1,"Health
y")
```

Queues 2 (Use Trellis Chart)

```
index=_internal host=* sourcetype=splunkd group=queue
| timechart span=30s useother=false max(current_size_kb) by name
```