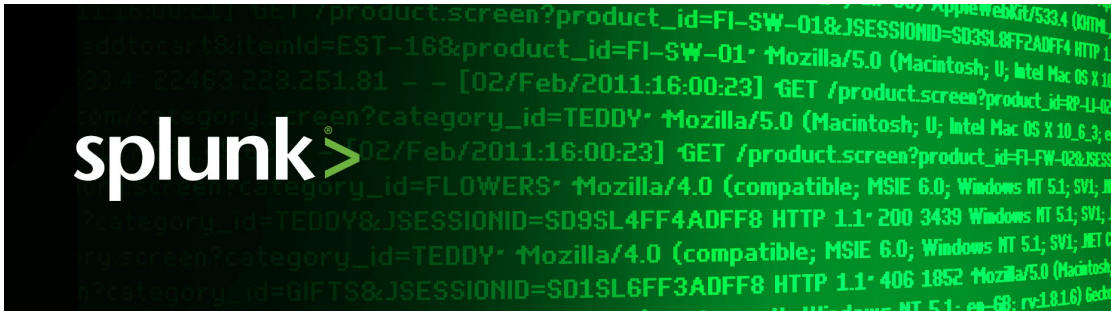


Splunk® Enterprise 7.2.0

转发数据

生成时间：2018 年 10 月 17 日，上午 11:15

Table of Contents

转发简介

关于转发和接收

您可从一个 Splunk Enterprise 实例转发数据到另一个 Splunk Enterprise 实例，甚至转发数据到非 Splunk 系统。执行**转发**的 Splunk 实例称为**转发器**。

有几种类型的转发器。请参阅“转发器类型”以了解每一种转发器。

从一个或多个转发器**接收**数据的 Splunk 实例称为**接收器**。接收器通常为 Splunk **索引器**，但也可以使用另一个转发器。

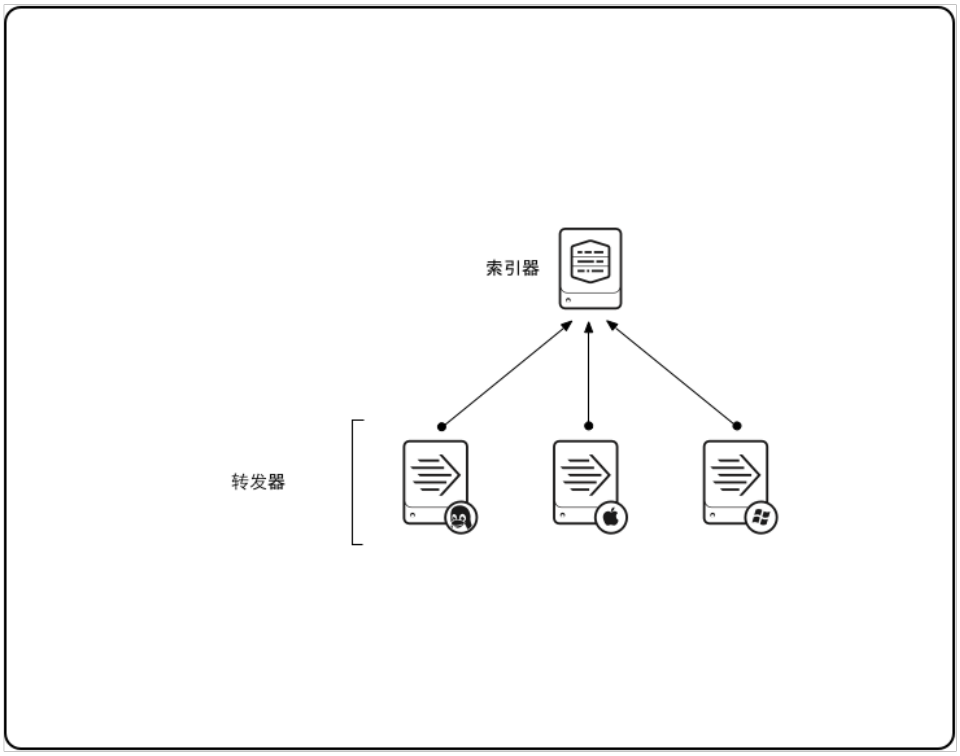
安装说明

如果您已经了解转发器并想获取转发器的安装说明，请查看以下内容：

- 设置转发和接收：重型或轻型转发器
- 设置通用转发器

示例转发布局

本图介绍了发送数据到单个接收器（索引器）的三个转发器，然后索引数据并使其对搜索可用：



转发器代表了较原始网络传输数据更加强大的数据转发解决方案，它们能够：

- 标记元数据（数据来源、来源类型和主机）
- 可配置的缓冲
- 数据压缩
- SSL 安全性
- 使用任何可用的网络端口

转发和接收功能允许各种有趣的拓扑。您可以构建一些环境来处理**数据整合**、**负载均衡**和**数据路由**等功能。

了解有关转发和接收的更多信息

- 要了解有关 Splunk Enterprise 分布式部署基础的更多信息，请参阅《分布式部署手册》。
- 有关可以使用转发器新建的部署拓扑类型的更多信息，请参阅本手册中的“转发器部署拓扑”。
- 要了解什么是中间转发，请参阅本手册中的“中间转发”。
- 要了解可用转发器的不同类型，请参阅“转发器类型”。

- 如要了解通用转发器，请参阅*《通用转发器》*手册。

转发器类型

这里有三种转发器类型：

- **通用转发器**仅包含转发数据所需的组件。如要了解通用转发器的更多内容，请参阅*《通用转发器》*手册。
- **重型转发器**是完整的 Splunk Enterprise 实例，能够索引、搜索和更改数据，同时也可以转发。为减少系统资源使用情况，系统禁用了重型转发器的一些功能。
- **轻型转发器**也是完整 Splunk Enterprise 实例，其中禁用了大部分功能以实现尽可能小的资源空间。轻型转发器已在 Splunk Enterprise 6.0 版本中弃用。通用转发器可替换几乎所有用途的轻型转发器，并代表了发送数据到索引器的最佳工具。

通用转发器

通用转发器的唯一目的是转发数据。不同于完整的 Splunk 实例，您无法使用通用转发器搜索数据或为数据建立索引。为实现更高性能和更小空间，它具有几个限制：

- 通用转发器无法通过数据搜索、索引或发出警告。
- 通用转发器不**分析**数据。您无法根据通用转发器内容使用其向不同的 Splunk 索引器路由数据。
- 与完整 Splunk Enterprise 不同的是，通用转发器不包含捆绑的 Python 版本。

通用转发器可以获取来自各种输入的数据，并把数据转发给 Splunk 部署，以供索引和搜索。它还可以转发数据给另一个转发器，作为向上发送数据至索引器之前的中间步骤。

通用转发器是可单独下载的软件。与重型和轻型转发器不同的是，您不会从完整 Splunk Enterprise 实例启用它。如要了解通用转发器的更多内容，请参阅*《通用转发器》*手册。

如要了解如何下载、安装和部署通用转发器，请参阅*《通用转发器》*手册中的“安装通用转发器软件”。

重型和轻型转发器

当通用转发器是转发数据的首选方式时，如果您想要在转发前分析或更改数据，或根据其内容控制数据的方向，则可能需要使用重型或轻型转发器。与通用转发器不同，重型和轻型转发器是禁用某些功能的完整 Splunk Enterprise 实例。重型和轻型转发器在功能和相应的资源空间大小方面不同。

重型转发器（有时被称为“常规转发器”）具有较索引器更小的空间，但是保留了大部分功能，无法执行分布式搜索除外。如果必要，可以禁用某些默认功能以减少空间大小，如 Splunk Web。重型转发器在转发数据之前分析数据，并可基于数据来源或事件类型等条件路由数据。

重型转发器的一个主要优势在于：在本地为数据建立索引，并把数据转发给另一个 Splunk 实例。您必须激活此功能。有关详细信息，请参阅本手册中的“使用 outputs.conf 配置转发器”。

轻型转发器具有更小的空间，功能更加有限。它仅转发未分析的数据。提供非常类似功能的通用转发器将取代它。轻型转发器已弃用，但仍然可大体上满足旧需求。

当安装通用转发器时，可以迁移同一主机上的任何（4.0 或更高版本）轻型转发器的检查点设置。有关更加详细的通用和轻型转发器比较，请参阅*《通用转发器》*手册中的“关于通用转发器”。

有关重型和轻型转发器功能的详细信息，请参阅本手册的“重型和轻型转发器功能”。

转发器比较

本表概述了三种转发器类型的相似性和差异性：

功能和能力	通用转发器	轻型转发器	重型转发器
Splunk Enterprise 实例类型	专用且可执行	完整 Splunk Enterprise，禁用大部分功能	完整 Splunk Enterprise，禁用部分功能
空间（内存，CPU 负载）	最小	小	中至高（取决于启用的功能）
捆绑 Python？	否	是	是
处理数据输入？	所有类型（但是，脚本式输入可能需要 Python 安装）	所有类型	所有类型
转发给 Splunk Enterprise？	是	是	是
转发给第三方系统？	是	是	是

作为中间转发器？	是	是	是
索引器确认（保证交付）？	可选	可选（4.2 版及更高版本）	可选（4.2 版及更高版本）
负载均衡？	是	是	是
数据复制？	是	是	是
按事件筛选？	否	否	是
事件路由？	否	否	是
事件分析？	有时	否	是
本地索引？	否	否	也可以选择设置 `outputs.conf` 中的 `indexAndForward` 属性
搜索/告警？	否	否	可选
Splunk Web？	否	否	可选

有关特定功能的详细信息，请参阅本主题剩余部分，以及手册中的其他转发主题。

转发器数据类型

转发器可以传输三种类型的数据：

- 原始
- 未分析
- 已分析

转发器可以发送的数据类型取决于转发器类型以及配置方式。通用转发器和轻型转发器可以发送原始或未分析的数据。重型转发器可以发送原始或分析的数据。

对于原始数据，转发器将未更改的数据通过纯 TCP 流发送。它不会将数据转换为 Splunk 通信格式。转发器收集并发送数据。这对于发送数据到非 Splunk 系统尤为有用。

对于未分析的数据，通用转发器执行最少处理。它不会检查数据流，但是会使用元数据标记流，以确定数据来源、来源类型和主机。它还会将数据流分割为 64KB 数据块，并在流上执行一些基本时间戳，以便事件本身不带可识别时间戳时接收索引器可以使用。通用转发器不会识别、检查或标记单独事件，除非将其配置以通过结构数据（如逗号分隔值文件）分析文件。

对于已分析的数据，重型转发器会把数据分割为单个事件，然后再标记这些事件并转发给 Splunk 索引器。它还会检查事件。由于数据已分析，因此转发器可以基于事件数据执行条件式路由，如字段值。

已分析和未分析格式同时被称为**处理过的**数据，以与原始数据区分开来。默认情况下，转发器将发送处理过的数据（通用转发器发送未分析的数据，同时重型转发器为已分析的数据。）要发送原始数据，在 outputs.conf 中设置 `sendCookedData=false` 属性/值对。

转发器和索引

转发器基于每个索引转发和路由数据。默认情况下，它们将转发所有外部数据，以及用于 `_audit` 内部索引的数据。在一些情况下，它们还为 `_internal` 内部索引转发数据。必要时，您可更改这一行为。有关详细信息，请参阅“按目标索引过滤数据”。

计划您的部署

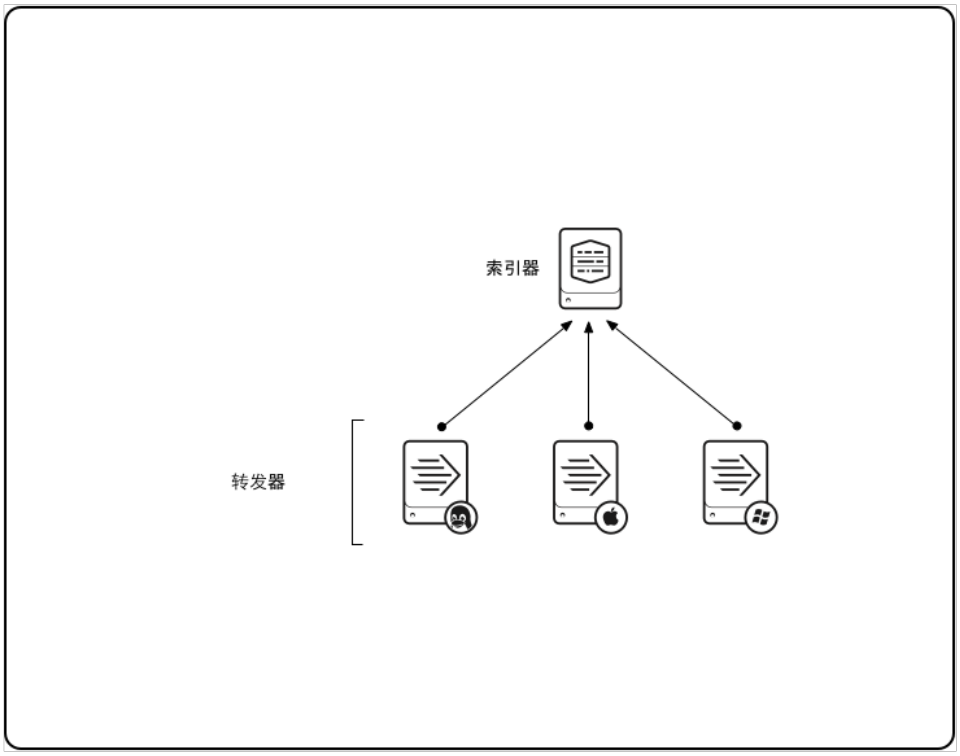
转发器部署拓扑

您可在范围广泛的方案中部署转发器。本主题提供了您可使用转发器新建的一些最有用拓扑的概述。有关如何配置各种部署拓扑的详细信息，请参阅“整合来自多个主机的数据”。

数据整合

数据整合是最常用的拓扑之一，其利用多个转发器发送数据给单个 Splunk 实例。方案通常涉及从工作站或生产服务器转发未分析数据到中央 Splunk Enterprise 实例整合和索引的通用转发器。在其他方案中，重型转发器可以把分析后的数据发送给中央 Splunk 索引器。

这里，三个通用转发器将发送数据到单个索引器：



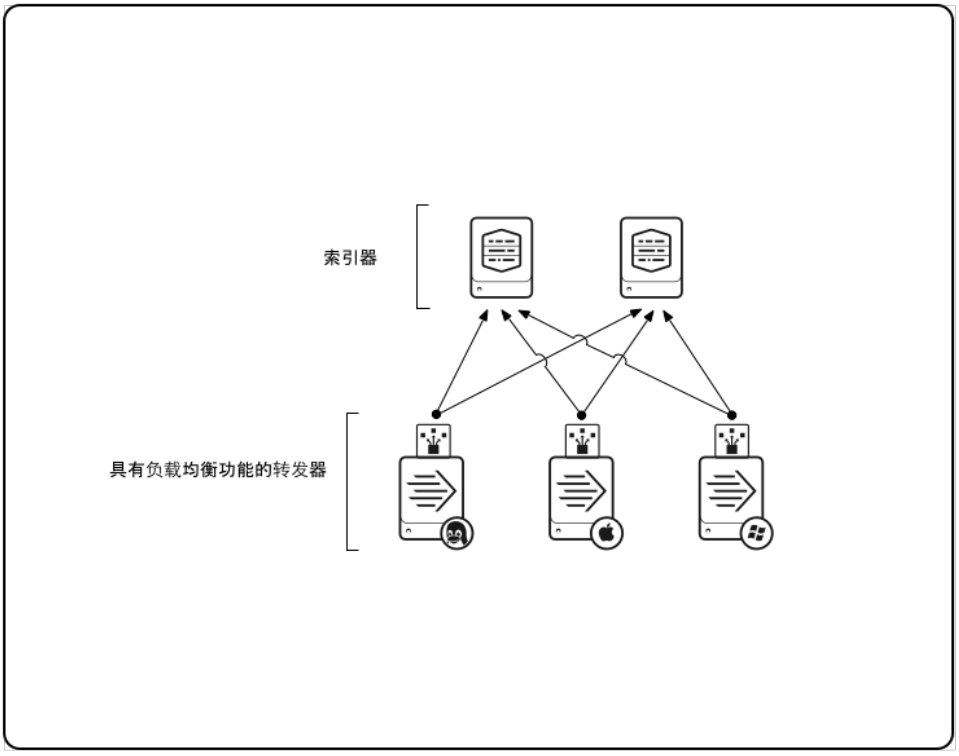
更多有关数据整合的信息，请参阅“整合来自多个主机的数据”。

负载均衡

负载均衡简化了跨几个索引器分发数据的流程，以处理高数据量，改进搜索性能的横向扩展和故障容错等整合。在负载均衡中，转发器将以指定间隔按顺序路由数据到不同索引器。

转发器将执行自动负载均衡，其中转发器将在设定时间间隔切换接收器。如果打开分析（对于重型转发器），切换将在事件边界出现。

本图有三个通用转发器，它们执行两个索引器之间的负载均衡：



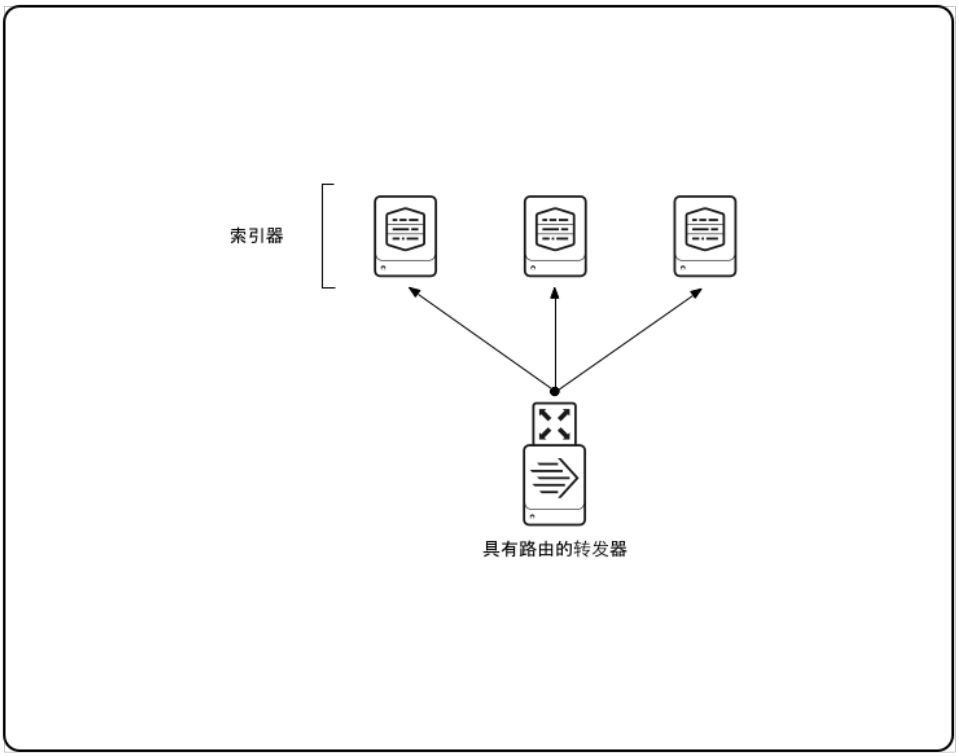
有关负载均衡的更多信息，请参阅“设置负载均衡”。

路由和筛选

在**数据路由**中，转发器会根据事件本身的数据来源、来源类型或模式等标准，将事件路由到特定的主机。事件级别的路由需要重型转发器。

转发器还会过滤并路由事件到特定队列，或通过路由到空队列以完全丢弃它们。

这里，重型转发器将基于事件模式路由数据到三个索引器：

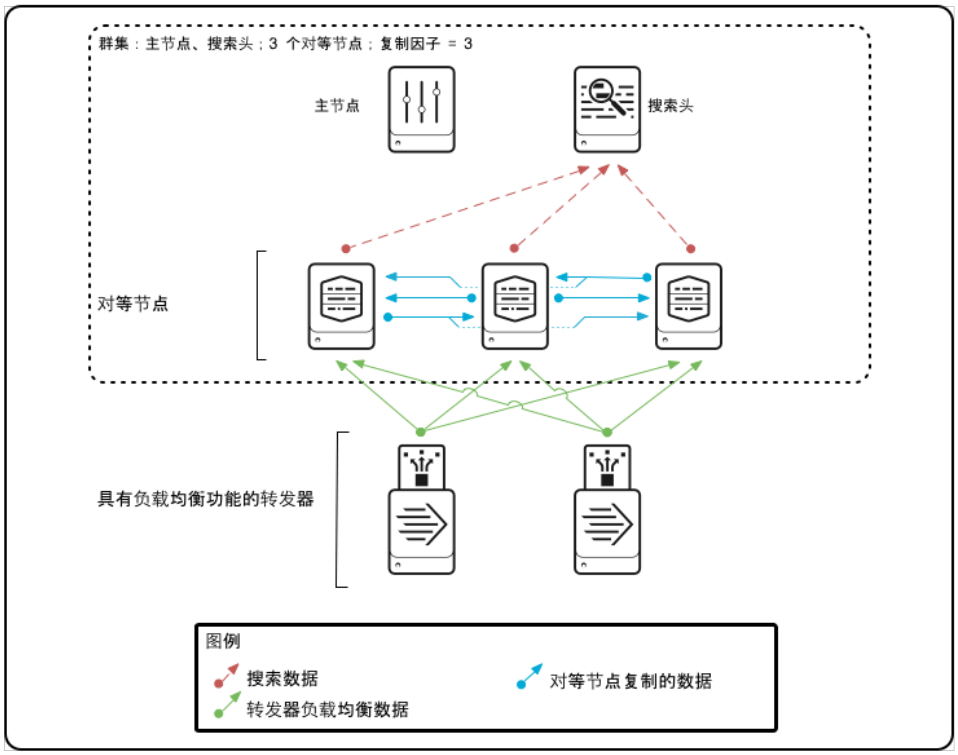


有关路由和筛选的更多信息，请参阅本手册中的“路由和筛选数据”。

转发器和索引器群集

您可以使用转发器发送数据到索引器群集的对等节点。Splunk 的最佳实践为此使用负载均衡转发器。

本图显示了发送数据给群集的两个负载均衡转发器：

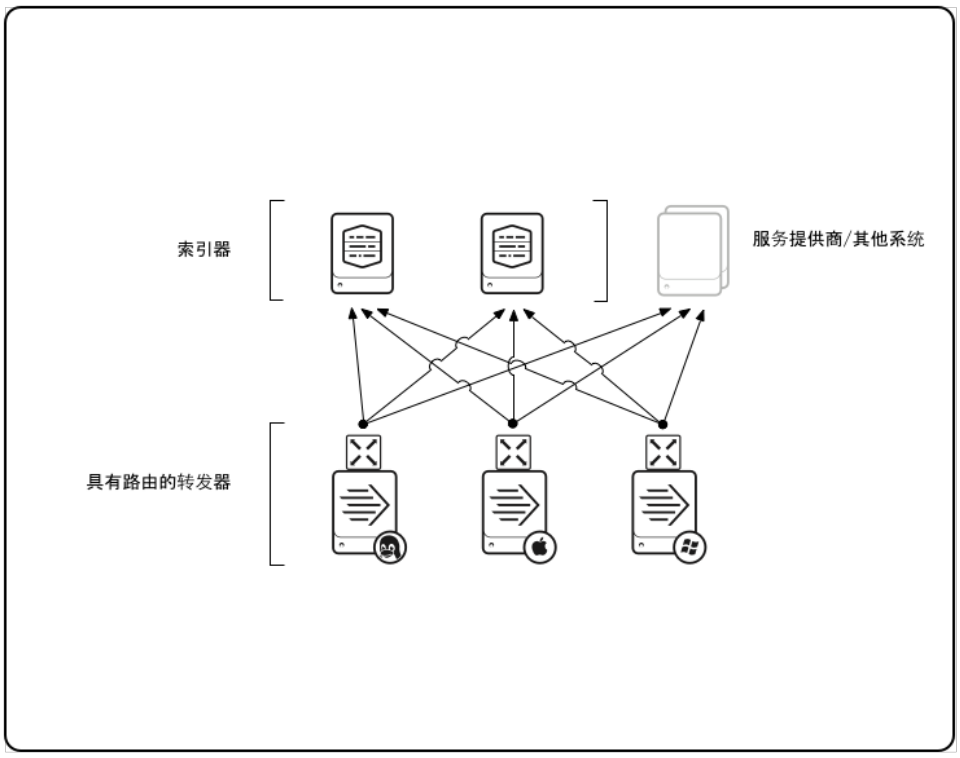


有关转发器和索引器群集的更多信息，请参阅*《管理索引器和索引器群集》*手册中的“使用转发器获取数据”。有关索引器群集的更多信息，请参阅“关于索引器群集和索引复制”。

转发到非 Splunk 系统

您可以通过重型转发器发送原始数据给第三方系统，如 syslog 聚合器。您可将它与数据路由组合，发送一些数据到非 Splunk 系统，其他数据则发送到一个或多个 Splunk 实例。

在此图中，三个转发器把数据路由到两个 Splunk 实例和一个非 Splunk 系统：



有关转发到非 Splunk 系统的更多信息，请参阅“转发数据到第三方系统”。

中间转发

要处理一些高级使用案例，您可能希望在一组转发器和索引器之间插入中间转发器。对于这种方案类型，原始转发器将发送数据到整合转发器，然后将数据转发到索引器。在某些情况下，中间转发器也会索引数据。

典型使用案例是需要中间索引的情况，无论是“存储并转发”要求还是启用本地搜索。（在这种情况下，您需要使用重型转发器。）如果需要限制对索引器计算机的访问权限，您还可以使用中间转发器；例如，出于安全原因。

要启用中间转发，请参阅“配置中间转发器”。

转发器和索引器之间的兼容性

*《通用转发器》*手册中包含转发器和索引器兼容性最新信息。请参阅该手册中“转发器和 Splunk Enterprise 索引器之间的兼容性”部分中的表格。

指标转发兼容性

只有当索引器和转发器都为 7.0.0 或以上版本时，才支持**指标**索引。不支持将指标数据从 6.6 或之前版本的转发器转发到 7.0 索引器。

转发与其他功能的兼容性

只有当索引器和转发器都为 6.x 或以上版本时，以下功能才可用：

- 动态文件标题。
- 结构化数据转发。
- 转发器时区传输。此外，当中间转发层仅由轻型或通用转发器组成时，转发器不会维持时区传输功能。

部署重型和轻型转发器

在 Splunk Enterprise 实例上启用转发

您可以在完整 Splunk Enterprise 实例上设置重型和轻型转发器。要了解如何配置通用转发器发送数据，请参阅本主题后续的“获取通用转发器手册中有关通用转发器的其他信息”。通用转发器的程序是不同的。

设置转发和接收：重型或轻型转发器

有关在 Splunk Enterprise 实例上启用接收的方法说明，请参阅“启用接收器”。如果您启用转发和接收之后，索引器上没有数据，请参阅“转发器/接收器连接故障排除”。

1. 指定将作为转发器和接受器的主机。
2. 将 Splunk Enterprise 安装在所有这些主机上。
3. 在每个接收器中，使用 Splunk Web 或 CLI 以启用接收。
4. 在每个转发器中，使用 Splunk Web 或 CLI 以启用转发。请参阅“部署重型转发器”或“部署轻型转发器”。
5. 在每个转发器中，使用 Splunk Web 或 CLI，或编辑 `inputs.conf` 以指定数据输入。
6. 在每个转发器中，使用 Splunk Web 或 CLI，或编辑 `outputs.conf` 以指定转发器应当发送数据的位置。
7. 在每个转发器中，重启 Splunk Enterprise 以提交配置更改并开始转发。
8. 在接收器中，搜索数据以确认转发按预期进行。例如：

```
host=<forwarder host name>
```

设置通用转发器

通用转发器是一个独立的产品，有独立的安装包和文档。有关通用转发器软件的详细信息，请参阅*《通用转发器》*手册。

1. 指定将作为转发器和接受器的主机。
2. 将 Splunk Enterprise 安装到接收器主机上。
3. 在每个接收器中，使用 Splunk Web 或 CLI 以启用接收。
4. 下载适用于转发器主机运行的操作系统的通用转发器软件。例如，如果转发器主机运行 Windows，下载 Windows 通用转发器。
5. 在转发器主机上安装通用转发器软件。如果主机运行 Windows，安装期间您可以配置部分通用转发器。
6. 安装通用转发器之后，配置通用转发器以发送数据到 Splunk Enterprise、Splunk Light 或 Splunk Cloud 索引器。
7. 配置您想要转发的数据输入。
8. 启动通用转发器。
9. 在接收器中，搜索数据以确认转发按预期进行。例如：

```
host=<forwarder host name>
```

获取通用转发器手册中有关通用转发器的其他信息

*《通用转发器》*手册中有关于如何安装、配置通用转发器软件以及排除相关问题的信息。有关安装通用转发器方法说明的详细信息，请参阅*《通用转发器》*手册中的“安装通用转发器软件”。有关如何使用通用转发器发送数据的更多信息，请参阅*《通用转发器》*手册中的以下主题之一：

- 如何转发数据至 Splunk Light
- 如何转发数据至 Splunk Cloud
- 如何转发数据至 Splunk Enterprise

重型和轻型转发器功能

本主题介绍附带重型和轻型转发器的功能，以及默认禁用的功能。

重型转发器详细信息

重型转发器拥有默认启用的所有 Splunk Enterprise 功能和模块，分布式搜索模块除外。文件 `$SPLUNK_HOME/etc/apps/SplunkForwarder/default/default-mode.conf` 包括本段落：

```
[pipeline:distributedSearch]
disabled = true
```

有关精确配置的详细视图，请参阅 `$SPLUNK_HOME/etc/apps/SplunkForwarder/default` 中的用于 SplunkForwarder 应用程序的配置文件。

轻型转发器详细信息

弃用的轻型转发器禁用 Splunk Enterprise 大部分功能。特别是，轻型转发器：

- 禁用事件签名和检查磁盘是否已满 (`$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/default-mode.conf`)。
- 仅限制内部数据输入为 `splunkd` 和指标日志 (`$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/inputs.conf`)。
- 禁用所有索引 (`$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/indexes.conf`)。
- 不使用 `transforms.conf`，同时不完全分析传入数据，但是使用来自 `props.conf` 的 `CHARSET`, `CHECK_FOR_HEADER`, `NO_BINARY_CHECK`, `PREFIX_SOURCETYPE`, 和 `sourcetype` 属性。
- 禁用 Splunk Web 界面 (`$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/web.conf`)。
- 限制吞吐量为 256KBps (`$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/limits.conf`)。
- 禁用 `$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/default-mode.conf` 中的以下模块：

```
[pipeline:indexerPipe]
disabled_processors= indexandforward, diskusage, signing,tcp-output-generic-processor, syslog-output-generic-processor, http-output-generic-processor, stream-output-processor

[pipeline:distributedDeployment]
disabled = true

[pipeline:distributedSearch]
disabled = true

[pipeline:fifo]
disabled = true

[pipeline:merging]
disabled = true

[pipeline:typing]
disabled = true

[pipeline:udp]
disabled = true

[pipeline:tcp]
disabled = true

[pipeline:syslogfifo]
disabled = true

[pipeline:syslogudp]
disabled = true

[pipeline:parsing]
disabled_processors=utf8, linebreaker, header, sendOut

[pipeline:scheduler]
disabled_processors = LiveSplunks
```

这些模块包括部署服务器（但不是部署客户端）、分布式搜索、命名管道/FIFO、来自网络端口的直接输入和计划程序。

通过按个案覆盖 `$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/default-mode.conf` 中的设置，可以启用轻型转发器的默认值以满足您的需求。

清理旧索引

当将一个索引器实例转换为轻型转发器时，除了其他之外，您需要禁用索引。此外，您无权限访问之前在实例中索引的任何数据。然而，该数据仍然存在。

如果希望清理系统的数据，您必须首先禁用 SplunkLightForwarder 应用，然后运行 CLI `clean` 命令并重新启用该应用。有关 `clean` 命令的信息，请参阅《*管理索引器和索引器群集*》手册中的"从 Splunk 删除索引的数据"。

转发结构化数据的注意事项

当您转发结构化数据（具有使用 `INDEXED_EXTRACTIONS` 功能的来源类型的数据）时，您所执行的任何分析、提取或筛选更改的操作必须在转发器上完成，而不是索引器。请参阅《*数据导入手册*》中的"转发从标头文件提取的数据"。

启用接收器

如果启用了接收，Splunk 实例会从转发器接收数据。

要启用转发和接收，您同时配置**接收器**和**转发器**。接收器是接收数据的 Splunk 实例；转发器发送数据给接收器。

有许多情况下，接收器是 Splunk **索引器**或索引器群集。也可以是另一个转发器，称为中间转发器。要了解中间转发器的工作原理，请参阅“中间转发器”。

转发器可以发送数据给多个接收器。相反地，一个接收索引器可以接受来自多个转发器的数据。设置转发器和接收器的方式取决于您数据所在的位置和数据应该前往的位置。

Splunk 最佳实践是先设置接收器，然后设置转发器以发送数据到那些接收器。

设置接收

将 Splunk 实例（索引器或转发器）启用为接收器之前，您必须先安装该实例。然后，您可以通过 Splunk Web、CLI 或 inputs.conf 配置文件启用接收实例。

设置使用 Splunk Web 接收

使用 Splunk Web 设置接收器：

1. 以管理员或等同于管理员的身份登录接收器。
2. 单击**设置 > 转发和接收**。
3. 在**配置接收**处，单击**新增**。
4. 指定您希望接收器侦听的 TCP 端口（**侦听端口**，也被称为**接收端口**）。例如，如果你输入 “9997”，接收器将从 9997 端口侦听连接。您可以指定任何未使用端口。您可以使用 `netstat` 等工具确定系统上可用的端口。确保您选择的端口未被 splunkweb 或 splunkd 使用。
5. 单击**保存**。Splunk 软件开始在您指定的端口处侦听传入的数据。

设置使用 Splunk CLI 接收

1. 从 shell 或命令提示，更改至 `$SPLUNK_HOME/bin` 目录：

```
cd $SPLUNK_HOME/bin
```

2. 运行 CLI 命令以启用接收：

```
splunk enable listen <port> -auth <username>:<password>
```

对于 `<port>`，替换您希望接收器侦听的端口（接收端口）。例如，如果你输入 “9997”，接收器将在 9997 端口接收数据。您可以指定任何未使用端口。您可以使用 `netstat` 等工具确定系统上可用的端口。确保您选择的端口未被 splunkweb 或 splunkd 使用。

`splunk enable listen` 命令会新建 `[splunktcp]` 段落（位于 `inputs.conf`）。例如，如果设置端口为 "9997"，它将新建段落 `[splunktcp://9997]`。

使用配置文件设置接收

您可以通过配置 `inputs.conf`（在 `$SPLUNK_HOME/etc/system/local` 中）允许 Splunk Enterprise 实例上接收。如果不存在，您可能需要新建此文件。

1. 使用文本编辑器，打开 `inputs.conf`（位于 `$SPLUNK_HOME/etc/system/local`）。
2. 要启用接收，添加指定接收端口的 `[splunktcp]` 段落。在本例中，接收端口是 9997：

```
[splunktcp://9997]
disabled = 0
```

3. 重新启动 Splunk 软件使更改生效。

表单 `[splunktcp://9997]` 和 `[splunktcp://:9997]`（一个或两个冒号）语义上等效。请使用其中之一。

部署重型转发器

您可以在完整的 Splunk Enterprise 实例中启用重型转发器

要启用转发和接收，您必须同时配置**接收器**和**转发器**。接收器是接收数据的 Splunk 实例；转发器发送数据给接收器。

Splunk 最佳做法是首先设置接收器，如“启用接收器”所述。然后，您可以设置转发器以发送数据到该接收器。

设置**重型**转发器是两步骤流程：

1.安装完整 Splunk Enterprise 实例。

2.在实例上启用转发。

设置转发

您可以使用 Splunk Web 或 CLI 来启用 Splunk 实例的转发。

您可以通过在 Splunk 实例上新建 `outputs.conf` 文件来启用和配置转发。尽管使用 `outputs.conf` 设置转发器需要一些更多初始知识，但在单个位置执行所有转发器配置方面存在明显的优势。大部分高级配置选项仅通过 `outputs.conf` 可用。此外，如果将要启用和配置一些转发器，则可通过编辑单个 `outputs.conf` 文件并制作每个转发器的副本，轻松实现这点。有关更多信息，请参阅“使用 outputs.conf 配置转发器”。

使用 Splunk Web 设置重型转发器

1.以管理员身份登录 Splunk Web，在此实例中将转发数据。

2.单击**设置 > 转发和接收**。

3.在**配置转发**处，单击**新增**。

4.输入接收 Splunk 实例的主机名称或 IP 地址，以及配置接收器时指定的**接收端口**。例如，您可以输入：`receivingserver.com:9997.`。为执行负载平衡的转发，您可以输入多个主机为逗号分隔的列表。

5.单击**保存**。

配置重型转发器以索引和转发数据

较轻型和通用转发器相比，重型转发器具有一个关键优势，即它可本地索引您的数据，以及转发数据到另一个索引。然而，默认关闭本地索引。如果希望在转发器上存储数据，您必须启用该功能 - 通过以上描述的方式或编辑 `outputs.conf`。

1.以管理员身份登录 Splunk Web，在此实例中将转发数据。

2.单击**设置 > 转发和接收**。

3.选择**转发默认。**

4.选择**是**存储并保留已索引数据的本地副本到转发器。

必须在 `outputs.conf.` 完成所有其他配置：

使用 CLI 设置重型转发

使用 CLI 按如下步骤在 Splunk Enterprise 实例上启用转发，然后配置转发到指定的接收器。

1.从命令或 shell 提示符，转到 `$SPLUNK_HOME/bin/`。

2.键入以下内容以启用转发：

```
splunk enable app SplunkForwarder -auth <username>:<password>
```

3.重新启动 Splunk Enterprise。

从 CLI 启动转发活动

以下过程发送数据到您指定的接收索引器。您可以发送数据到接收器之前，

1.从 shell 或命令提示符转到 `$SPLUNK_HOME/bin` 目录。

2.要启动转发活动，使用 `splunk add forward-server` 命令指定接收器：

```
splunk add forward-server <host>:<port> -auth <username>:<password>
```

3.在调用这些命令后，重新启动转发器。

从 CLI 停止转发活动

要结束转发活动，输入：

```
splunk remove forward-server <host>:<port> -auth <username>:<password>
```

禁用来自 CLI 的转发

即使结束转发活动，但是实例仍被配置为转发器。为将转发器转换为完整 Splunk Enterprise 实例，使用 `disable` 命令，如本主题中前面部分所述。

1. 从命令或 shell 提示符转到 `$SPLUNK_HOME/bin` 目录。

2. 键入以下内容以禁用转发：

```
splunk disable app SplunkForwarder -auth <username>:<password>
```

禁用转发后，本命令将转发器转换为完整 Splunk Enterprise 实例。

部署轻型转发器

重要提示：轻型转发器已在 Splunk Enterprise 6.0 版本中弃用。有关所有弃用功能的列表，请参阅《发行说明》中的"弃用功能"。

您可以在完整的 Splunk Enterprise 实例中安装轻型转发器。关于如何安装**通用转发器**以替换轻型转发器（建议），请参阅*《通用转发器》*手册中的"安装通用转发器软件"。

要启用转发和接收，请同时配置**接收器**和**转发器**。接收器接收数据，转发器发送数据至接收器。

Splunk 最佳实践是首先设置接收器。然后，您可以设置转发器以发送数据到该接收器。

设置**轻型**转发器的过程包括两个步骤：

1. 安装完整 Splunk Enterprise 实例。
2. 在实例上启用转发。

注意：把 Splunk 实例配置为轻型转发器时，请选择转发器许可证。有关更多信息，请参阅"Splunk 许可证类型"。

设置转发

您可以使用 CLI 作为启用转发的快速方式。

您可以通过在 Splunk 实例上新建 `outputs.conf` 文件来启用和配置转发。尽管使用 `outputs.conf` 设置转发器需要一些更多初始知识，但在单个位置执行所有转发器配置方面存在明显的优势。大部分高级配置选项仅通过 `outputs.conf` 可用。此外，如果将要启用和配置一些转发器，则可通过编辑单个 `outputs.conf` 文件并制作每个转发器的副本，轻松实现这点。有关更多信息，请参阅"使用 outputs.conf 配置转发器"主题。

使用 CLI 设置轻型转发

要设置轻型转发，请执行以下步骤：

1. 从 shell 或命令提示符导航到 `$SPLUNK_HOME/bin/` 目录，并运行以下命令：`splunk enable app SplunkLightForwarder -auth <username>:<password>`
2. 重新启动转发器。

要禁用轻型转发器模式， 请运行以下命令：

```
splunk disable app SplunkLightForwarder -auth <username>:<password>
```

本命令将转发器转换为完整的 Splunk Enterprise 实例。

从 CLI 启动转发活动

1. 从 shell 或命令提示，转到 `$SPLUNK_HOME/bin/` 目录：
2. 要启动转发活动，使用 `splunk add forward-server` 命令：`splunk add forward-server <host>:<port> -auth <username>:<password>` 指定接收器

要结束转发活动，输入：

```
splunk remove forward-server <host>:<port> -auth <username>:<password>
```

注意：尽管本命令将结束转发活动，但是实例仍被配置为转发器。为将实例转换为完整 Splunk Enterprise 实例，使用 `disable` 命令，如本主题中前面部分所述。

在调用这些命令后，重新启动转发器。

配置转发器

使用 inputs.conf 在转发器上配置数据集合

您可以通过编辑 inputs.conf 配置文件，以配置转发器中的数据传入。

几乎所有情况下，您都必须编辑 inputs.conf（$SPLUNK_HOME/etc/system/local 目录下）。如果您有一个已安装的应用并想对它的输入配置进行更改，可编辑 $SPLUNK_HOME/etc/apps/<appname>/local/inputs.conf。例如，如果您有已安装的适用于 Unix 和 Linux 的 Splunk 加载项，您可以在 $SPLUNK_HOME/etc/apps/Splunk_TA_nix/local/inputs.conf 中进行编辑。

切勿对 $SPLUNK_HOME/etc/system/default 中的 inputs.conf 进行更改。升级时，安装会覆盖此文件，删除您所做的任何更改。

关于转发器可收集的数据信息，请参阅“我可对哪些数据建立索引？”。

在更改配置文件的情况下，您必须重新启动转发器以使更改生效。

编辑 inputs.conf

1. 使用操作系统文件管理工具或 shell 或命令提示符，导航到 $SPLUNK_HOME/etc/system/local。
2. 打开 inputs.conf 进行编辑。如果不存在，您可能需要新建此文件。
3. 添加数据导入。
4. 一旦您添加了您的输入，保存文件并关闭它。
5. 重新启动转发器。
6. 在接收索引器上，登录并加载“搜索和报表”应用。
7. 运行搜索，并确认您在设置了数据导入的转发器上看到了结果：

```
host=<forwarder host name or ip address> source=<data source> earliest=1h
```

如果您没有看到任何结果，可访问故障排除页面查看可能的问题解决方案。

使用 outputs.conf 配置转发器

outputs.conf 文件定义转发器发送数据到接收器的方式。

您可以指定一些输出配置或通过 Splunk Web（仅重型/轻型转发器）或 CLI，但是大部分高级配置设置要求您编辑 outputs.conf。该主题介绍了各种拓扑，如负载均衡和数据路由，提供配置 outputs.conf 以支持这些拓扑的详细示例。

尽管 outputs.conf 是配置转发器的关键文件，它只是针对转发器应当发送数据的位置。如要指定转发器应当收集哪些数据，您必须单独配置这些输入。有关配置输入的详细信息，请参阅*《数据导入手册》*中的“添加数据和配置输入”。

outputs.conf 文件类型

单个转发器可以拥有多个 outputs.conf 文件（例如，一个位于应用目录，另一个在 /system/local 中）。无论拥有多少 outputs.conf 文件及其驻留位置，转发器都可组合所有设置，使用*《管理员手册》*中“配置文件程序”介绍的位置优先规则。

默认版本

Splunk Enterprise 随附单独的默认 outputs.conf 文件，位于 $SPLUNK_HOME/etc/system/default 中。

通用文件夹具有两个默认 outputs.conf 文件。请参阅*《通用转发器》*手册中的“使用 outputs.conf 配置转发”。

不要触摸任何配置文件的默认版本，原因在“关于配置文件”中介绍过。

自定义版本

当配置转发行为时，这些更改将保存到 outputs.conf 的自定义版本中。这是几种指定转发行为的方式：

- 在安装转发器时（仅 Windows 通用转发器）
- 通过运行 CLI 命令
- 通过使用 Splunk Web（仅重型/轻型转发器）
- 通过直接编辑 outputs.conf 文件

对于前三种方法，转发器将自动新建或编辑 outputs.conf 的自定义版本。这些版本的位置不同，取决于转发器类型和其他因素。

当通过 Splunk Web 或 CLI 启用重型/轻型转发器时，将在当前正在运行应用的目录中新建 outputs.conf 文件。例如，如果正使用搜索应用，Splunk Enterprise 将文件新建在 $SPLUNK_HOME/etc/apps/search/local/ 中。然后，您可

在此编辑它。

除了间接新建和编辑的任何 outputs.conf 文件之外（例如，通过 CLI），您还可以直接新建或编辑 `outputs.conf` 文件。Splunk 最佳实践是仅处理位于 `$SPLUNK_HOME/etc/system/local/` 文件的单个副本。（如果通过 CLI 进行配置更改使得文件副本已存在于该目录，则仅编辑该副本。）为保证分发和管理的简单性，您可以将所有非默认版本的设置合并为单个自定义的 `outputs.conf` 文件。

在对 `outputs.conf` 进行更改后，您必须重启转发器以便更改生效。

有关 `outputs.conf` 更多详细信息，请参阅 outputs.conf 规范文件。

配置级别

索引字段有两种输出处理器：`tcpout` 和 `syslog`。您可在三个段落级别配置它们：

- **全局。**（可选）在全局级别，您指定希望全局应用的任何属性，以及仅可为输出处理器在系统范围级别配置的某些属性。
- **目标组。**目标组定义一个或多个接收索引器的设置。每个输出处理器可以有多个目标组。可在目标组级别指定大部分配置设置。
- **单个服务器。**（可选）您可为目标组内的单个主机（接收器）指定配置值。.

优先处理更加特定级别的配置。例如，如果为目标组指定 `compressed=true`，则转发器将发送服务器到该目标组压缩数据，即使为全局级别设置 `compressed` 为 "false"。

注意：本讨论专注于 `tcpout` 处理器，这将使用 `[tcpout]` 标题。有关 syslog 输出处理器的信息，请参阅“转发数据到第三方系统”。

全局段落

这里，您将设置任何希望全局应用的属性。本段落可选。然而，这里有几个仅可在全局级别设置的属性，包括 `defaultGroup` 和 `indexAndForward`。

指定 `tcpout` 处理器的全局段落（使用 `[tcpout]` 标题）。

以下是全局 tcpout 段落的示例：

```
[tcpout]
defaultGroup=indexer1
indexAndForward=true
```

本全局段落包括两个属性/值对：

- **defaultGroup=indexer1** 这将告诉转发器发送所有数据到 "indexer1" 目标组。有关更多信息，请参阅“默认目标组”。
- **indexAndForward=true** 这将告诉转发器本地索引数据，以及转发数据到目标组的接收索引器。如果设置为 "false"（默认），转发器会转发而不索引数据。本属性仅可用于重型转发器，通用和轻型转发器无法索引数据。

默认目标组

要设置自动转发的默认组，在全局级别包括 `defaultGroup` 属性，在您的 `[tcpout]` 段落中：

```
[tcpout]
defaultGroup= <target_group1>, <target_group2>, ...
```

defaultGroup 指定一个或多个目标组，这将稍后在 `tcpout:<target_group>` 段落中定义。转发器会发送所有事件到指定组。

如果不希望自动转发数据，则不要设置 `defaultGroup` 属性。

有关使用 `defaultGroup` 属性的一些示例，请参阅“路由和过滤数据”。

目标组段落

目标组确定一组接收器。它还指定转发器如何发送数据到这些接收器。您可以定义多个目标组。

这里是目标组段落的基本模式：

```
[tcpout:<target_group>]
server=<receiving_server1>, <receiving_server2>, ...
<attribute1> = <val1>
```

```
<attribute2> = <val2>
...
```

要在目标组指定接收主机，请使用格式 `<hostname_or_ip_address>:<port>`，其中 `<port>` 是接收服务器的**接收端口**。例如，`myhost.Splunk.com:9997`。您可以指定多个接收器，同时转发器将在它们之间负载均衡。

有关如何使用目标组段落定义几个部署拓扑的信息，请参阅本主题后续的“定义典型部署拓扑”。

单服务器段落

您可以为单个接收索引器定义特定配置。然而，接收器还必须是目标组的成员。

当定义单服务器级别的属性时，它将优先于目标组或全局级别的任何定义。

这里是定义单服务器段落的语法：

```
[tcpout-server://<ipaddress_or_servername>:<port>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

示例

以下 `outputs.conf` 示例包含三个段落以发送 tcpout 到 Splunk Enterprise 接收器：

- 全局设置。在本例中，这里有一个设置，可以指定 `defaultGroup`。
- 单个目标组的设置由两个接收器组成。本例指定由两个接收器组成的负载均衡目标组。
- 设置目标组内的一个接收器。在本段落中，您可以指定特定于 `mysplunk_indexer1` 接收器的任何设置。

```
[tcpout]
defaultGroup=my_indexers

[tcpout:my_indexers]
server=mysplunk_indexer1:9997, mysplunk_indexer2:9996

[tcpout-server://mysplunk_indexer1:9997]
```

定义典型部署拓扑

本部分显示了如何配置转发器以支持几个典型部署拓扑。

负载均衡

要执行**负载均衡**，使用多个接收器指定一个目标组。在本例中，目标组由三个接收器组成：

```
[tcpout:my_LB_indexers]
server=10.10.10.1:9997,10.10.10.2:9996,10.10.10.3:9995
```

转发器会在指定的三个接收器之间平衡负载。如果一个接收器出现故障，转发器将自动切换到另一个可用接收器。

数据复制

要执行**数据复制**，指定多个目标组，每个在自己的段落中。在数据复制中，转发器将发送所有事件的副本到两个或多个目标组的接收器。数据复制通常在接收索引器上产生类似，但不完全相同的数据副本。这里是您设置数据复制的示例：

```
[tcpout]
defaultGroup=indexer1,indexer2

[tcpout:indexer1]
server=10.1.1.197:9997

[tcpout:indexer2]
server=10.1.1.200:9997
```

转发器将发送重复数据流到 `indexer1` 和 `indexer2` 目标组中指定的服务器。

使用负载均衡进行数据复制

您可以组合负载均衡与数据复制。例如：

```
[tcpout]
defaultGroup=cloned_group1,cloned_group2

[tcpout:cloned_group1]
server=10.10.10.1:9997, 10.10.10.2:9997, 10.10.10.3:9997

[tcpout:cloned_group2]
server=10.1.1.197:9997, 10.1.1.198:9997, 10.1.1.199:9997, 10.1.1.200:9997
```

转发器将发送完整数据流给 `cloned_group1` 和 `cloned_group2`.。该数据将在每个组内负载均衡，每 30 秒在接收器之间旋转（默认频率）。

对于 syslog 和其他输出类型，您必须明确指定路由。查看本手册中的“路由和过滤数据”。

常用属性

`outputs.conf` 文件提供了大量配置选项，这可提供理想的转发控制和灵活性。对于可用属性，其中几个尤为相关：

属性	默认	已配置	值
defaultGroup	n/a	全局段落	逗号分隔的一个或多个目标组列表。转发器将发送所有事件到所有指定目标组。如果不希望事件自动转发到目标组，不要设置本属性。
indexAndForward	false	全局段落	如果设置为 "true"，除了转发数据给接收索引器之外，转发器将本地索引所有数据。 **注意：**本属性仅可用于重型转发器。无法本地索引通用转发器。
server	n/a	目标组段落	必填。为转发器指定服务器，使其具有接收器的功能。这必须使用格式 `<hostname_or_ip_address>:<port>` 设置为值，其中 `<port>` 是接收服务器的接收端口。
disabled	false	任何段落级别	指定是否禁用段落。如果设置为 "true"，这相当于段落不在那里。
sendCookedData	true	全局或目标组段落	指定转发器是否在转发前处理数据。
compressed	false	全局或目标组段落	指定转发器是否发送压缩的数据。
ssl....	n/a	任何段落级别	用于配置 SSL 的一组属性。有关如何使用这些属性的信息，请参阅*《确保 Splunk Enterprise 安全》*手册中的“关于确保来自转发器的数据安全”。
useACK	false	全局或目标组段落	指定转发器是否等待索引器确认，确认数据已写入文件系统。请参阅“防止传输中的数据丢失”。
dnsResolutionInterval	300	全局或目标组段落	指定索引器 DNS 名称将分析为 IP 地址的基本时间间隔（秒）。请参阅本手册的“DNS 分辨率间隔”。

您可在此找到的 `outputs.conf.spec` 文件以及几个示例提供了这些和所有其他配置选项的详细信息。此外，其中的大部分设置将在处理特定转发方案的主题中讨论。

DNS 解析间隔

`dnsResolutionInterval` 属性指定接收器 DNS 名称将解析为 IP 地址的基本时间间隔（秒）。该值用于计算运行时间隔为如下：

```
run-time interval = dnsResolutionInterval + (number of receivers in 'server' attribute - 1) * 30
```

对于 `server` 属性中指定的每个额外接收器，运行时间隔延长 30 秒；即，对于每个转发器负载均衡的额外接收器。`dnsResolutionInterval` 属性默认为 300 秒。

例如，如果保留 300 秒的默认设置属性，同时转发器在 20 个索引器负载均衡，DNS 分辨率将每 14 分钟出现：

```
(300 + ((20 - 1) * 30)) = 870 seconds = 14 minutes
```

如果您将 `dnsResolutionInterval` 更改为 600 秒，并将负载均衡索引器的数量保留为 20，则 DNS 分辨率将每 19.5 分钟出现：

```
(600 + ((20 - 1) * 30)) = 1170 seconds = 19.5 minutes
```

升级转发器

升级重型和轻型转发器

要升级重型或轻型转发器，升级支持该转发器的 Splunk 实例。当在重型或轻型转发器上执行升级时，除了在升级后确认转发继续工作之外，不需要特定说明或其他说明。

关于安装说明和提示，请参阅*《安装手册》*中的“如何升级 Splunk Enterprise”。

执行高级配置

设置负载均衡

通过**负载均衡**，转发器将跨几个接收实例分发数据。每个接收器都会获得其中一部分数据，所有接收器将保留全部数据。要访问完整的转发数据，您需要跨所有接收器设置分布式搜索。请参阅*分布式搜索*中的“关于分布式搜索”。

负载均衡允许横向扩展以改进性能。此外，其自动切换能力确保计算机中断时的复原设置。如果主机出现故障，转发器会发送数据给下一个可用接收器。

当从路由器等网络设备获取数据时，还可使用负载均衡。要处理 syslog 和其他跨 TCP 端口 514 生成的数据，单个通用转发器可监控端口 514 并跨几个索引器分发传入数据。

注意：不应使用外部负载均衡器来实现转发器与接收器之间的负载均衡。这种做法不会产生您预期的结果。使用转发器附带的负载均衡操作。

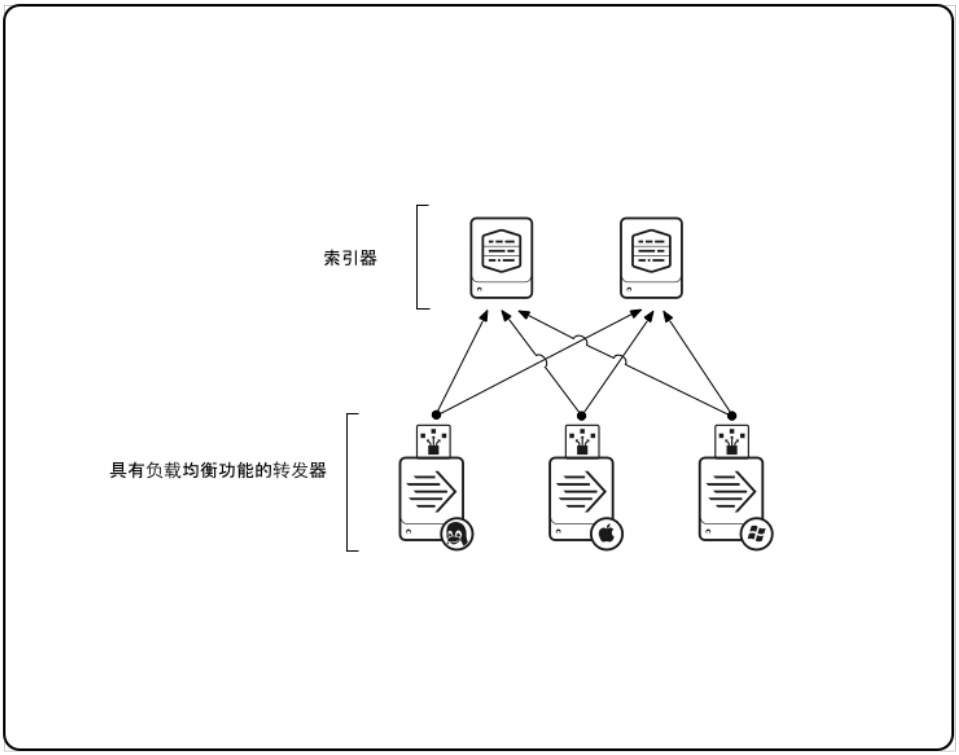
负载均衡如何工作

转发器会执行自动负载均衡。转发器根据您可以指定的指定时间或数量间隔将数据路由到不同索引器。例如，如果有一个负载均衡组由索引器 A、B 和 C 组成，则转发器会按指定时间间隔将数据流随机切换至组中的另一个索引器。转发器可能从索引器 B 切换到索引器 A，再切换到索引器 C，依此类推。如果某一索引器故障，转发器会立即切换到另一个索引器。

转发器监视的每个输入都有一个数据流。转发器会确定某一数据流是否可以安全切换到另一个索引器。然后，它会按指定时间间隔将此数据流切换到新选择的索引器。如果转发器无法将此数据流安全切换到新索引器，它会使与前一个索引器的连接保持打开状态，并继续发送此数据流，直到该数据流被安全发送出去。

当通用转发器监视 TCP 网络数据流时无法切换索引器，除非转发器出现 End-of-file (EOF) 或接收索引器故障。如果出现这种情况，转发器会切换至列表中的下一个可用索引器。在转发数据到索引器之前，由于通用转发器不会分析数据并确定事件边界（与重型转发器不同），因此不知道何时可安全切换到下一个索引器，除非接收 EOF。

下图给出的是典型负载均衡方案，其中三个转发器在一组双接收索引器之间发送负载均衡数据：



配置接收目标以进行负载均衡的选项

为接收索引器指定静态或 DNS 列表

当为负载均衡配置一组目标接收器时，您可以选择 DNS 或静态列表。

选用 DNS 列表可具备更大的灵活性，且能够简化扩展，尤其对于大型部署。通过 DNS，您可以更改一组接收器，

而无需重新编辑每个转发器的 `outputs.conf` 文件。

选用静态列表可以为各接收器指定不同端口。如果您需要在单个主机上运行的多个接收器之间执行负载均衡，这很有用，因为每个接收器可以侦听单独网络端口。

选择负载均衡方法

您可以选择转发器在负载均衡列表中的索引器间负载均衡的方式。

- **按时间**。默认的负载均衡方法是负载均衡列表中转发器更改索引器的频率。Outputs.conf 中的 `autoLBFrequency` 设置控制转发器在索引器之间的切换频率。默认频率为每 30 秒，但是您可以将频率设置得更高或更低。
- **按数量**。另一个选项是在转发器在负载均衡列表中的索引器之间切换之前，设置转发器发送到索引器的数据量。`outputs.conf` 中的 `autoLBVolume` 设置控制转发器切换另一个索引器之前，转发器发送到接收索引器的数据量。默认情况下，不启用此设置（0 字节）。如果您设置为其他非零字节，则转发器将根据其已发送的数据量更改索引器。

如果您启用两种设置，则转发器将根据以下逻辑选择索引器：

- 如果转发器已发送超过 `autoLBVolume` 字节的数据到索引器，则转发器会更改索引器，无论自上次更改接收索引器之后是否超过 `autoLBFrequency`。
- 如果转发器在 `autoLBFrequency` 秒过去之前未发送超过 `autoLBVolume` 字节的数据，则转发器会在该时间过去之后更改索引器。

指定静态列表目标

1. 在您想设置静态列表目标的转发器上，编辑 `$SPLUNK_HOME/etc/system/local/outputs.conf`。您可能必须预先新建该文件。
2. 在 `outputs.conf` 文件中，在目标组 `[tcpout]` 段落中指定每个接收器。
3. 保存 `outputs.conf` 文件。
4. 重新启动转发器。转发器将数据发送到静态列表目标。

静态负载均衡配置文件示例

在以下示例中，目标组由三个接收器组成，通过 IP 地址和接收端口号指定。通用转发器在三个接收器之间均衡负载。如果一个接收器出现故障，转发器将切换到列表上的另一个接收器。

```
[tcpout: my_LB_indexers]
server=10.10.10.1:9997,10.10.10.2:9996,10.10.10.3:9995
```

在以下示例中，目标组由四个接收器组成，通过 IP 地址和接收端口号指定：通用转发器已配置为在其切换至列表中其他接收器之前，发送特定数量的数据（在此情况中为 1 MB）到接收器。

```
[tcpout: my_LB_indexers]
server=10.10.10.1:9997,10.10.10.2:9996,10.10.10.3:9995,10.10.10.4:9994
autoLBVolume=1048576
```

在以下示例中，目标组由三个接收器组成，一个通过 DNS 主机名指定，另外两个通过 IP 地址指定。通用转发器已配置为在切换至另一个接收器之前，发送数据到一个索引器 3 分钟（180 秒）。

```
[tcpout:My_LB_Indexers]
server=192.168.1.15:9997,192.168.1.179:9997,server1.mktg.example.com:9997
autoLBFrequency=180
```

指定 DNS 列表目标

1. 在您想设置 DNS 列表目标的转发器上，编辑 `$SPLUNK_HOME/etc/system/local/outputs.conf`。您可能必须预先新建该文件。
2. 在 `outputs.conf` 文件中，在目标组 `[tcpout]` 段落中指定单个主机。
3. 保存 `outputs.conf` 文件。
4. 重新启动转发器。转发器将数据发送到 DNS 列表目标。
5. 在 DNS 服务器中，为每个主机的 IP 地址新建 DNS A 记录，引用 `outputs.conf.` 中指定的服务器名称。

   ```
   splunkreceiver.mycompany.com   A   10.10.10.1
   splunkreceiver.mycompany.com   A   10.10.10.2
   splunkreceiver.mycompany.com   A   10.10.10.3
   ```
6. 在您的 DNS 服务器上重新加载更新的配置。根据网络拓扑大小，DNS 更改生效可能需要一点时间。

基于 DNS 的负载均衡配置文件示例

在以下示例中，转发器已配置为发送数据到看起来是单个主机的设备。您在 DNS 中所作更改使得此主机名引用了三个不同的 IP 地址。

```
[tcpout:my_LB_indexers]
server=splunkreceiver.mycompany.com:9997
```

转发器使用 DNS 列表以负载均衡，按间隔发送数据，在指定的接收器之间切换。如果接收器不可用，转发器将跳过并发送数据到列表上的另一个接收器。

如果您的拓扑具有许多转发器，则 DNS 列表方法可让您在 DNS 服务器上进行更改以更新一组接收器，而无需编辑 `outputs.conf`。

为横向扩展配置通用转发器负载均衡

在为横向扩展配置负载均衡时，您应首先确定需求，尤其是横向扩展，无论您是否有故障转移需求。这有助于您基于这些需求开发拓扑，可能包括多个转发器以及接收器和一个搜索头，以跨接收器搜索。

设置基于 DNS 的负载均衡

此程序假定三个通用转发器和三个接收器的拓扑，并使用 DNS 列表指定接收器。要让基于 DNS 的负载均衡发挥作用，您必须配置接收器，使他们必须全部侦听同一网络端口。

有关如何安装通用转发器的更多信息，请参阅*《通用转发器》*手册中的"安装通用转发器软件"。

1. 配置三台计算机在网络上进行通信。此示例使用了以下 IP 地址：`10.10.10.1`、`10.10.10.2` 和 `10.10.10.3`，但您可以使用您网络中的有效地址。
2. 安装一组三个 Splunk Enterprise 实例作为接收器。
3. 在接收器上配置接收。指定同一接收端口。例如：
   ```
   ./splunk enable listen 9997 -auth <username>:<password>
   ```
4. 安装通用转发器组。
5. 在 DNS 服务器上，为每个接收器 IP 地址设置含 A 记录的 DNS 列表。
   ```
   splunkreceiver.mycompany.com   A   10.10.10.1
   splunkreceiver.mycompany.com   A   10.10.10.2
   splunkreceiver.mycompany.com   A   10.10.10.3
   ```
6. 在您的 DNS 服务器上重新加载更新的配置。根据网络拓扑大小，DNS 更改生效可能需要一点时间。
7. 为所有要使用的转发器配置 `outputs.conf` 文件。此示例会指定用于 DNS 列表的 DNS 服务器名称以及接收器正在侦听的端口：
   ```
   [tcpout]
   defaultGroup=my_LB_indexers

   [tcpout:my_LB_indexers]
   disabled=false
   autoLBFrequency=40
   server=splunkreceiver.mycompany.com:9997
   ```
   本 `outputs.conf` 文件使用 `autoLBFrequency` 属性设置 40 秒的负载均衡频率。每 40 秒，转发器将切换至另一个接收器。默认频率为 30 秒。
8. 分发 `outputs.conf` 文件到所有转发器。您可以使用**部署服务器**处理本分发。

从 CLI 指定负载均衡

您还可以使用 CLI 指定负载均衡。您可使用本语法，在开始一组接收器的转发活动时这么做：

```
./splunk add forward-server <host>:<port> -method autobalance
```

其中，`<host>:<port>` 是接收器的主机和接收器端口。

本例将新建四个接收器的负载均衡组：

```
./splunk add forward-server indexer1:9997 -method autobalance
./splunk add forward-server indexer2:9997 -method autobalance
./splunk add forward-server indexer3:9997 -method autobalance
./splunk add forward-server indexer4:9997 -method autobalance
```

配置转发器使用 SOCKS 代理

本主题介绍如何配置带有套接字安全版本 5 (SOCKS5) 代理服务器的转发器作为意图向代理服务器之外的索引器转发数据的目标。

默认情况下，Splunk 转发器需要有直接网络连接到任意接收索引器。如果防火墙阻止了转发器和索引器之间的连接，转发器则不能向索引器发送数据。

从 Splunk Enterprise 6.3 版本开始，您可以配置转发器使用 SOCKS5 代理主机来向索引器发送数据。您可以通过指定转发器上的 `outputs.conf` 配置文件中一个段落中的属性来进行此操作。在您配置并重新启动转发器之后，它会连接到 SOCKS5 代理主机，如果您提供了凭据则可选择按需求对服务器进行验证。代理主机建立与索引器的连接，并且转发器通过代理连接开始发送数据。

任意类型的 Splunk 转发器都可以通过 SOCKS5 代理主机发送数据。

SOCKS5 客户端的该实现符合互联网工程任务组 (IETF) 请求注解 (RFC) 备忘录 #1928。有关该备忘录的信息，请参阅“网络工作组：请求注解：1928”在 IETF 网站上的 (http://www.ietf.org/rfc/rfc1928.txt)。

要配置 SOCKS5 代理连接，可编辑 `outputs.conf` 中的段落并指定某些属性来启用代理。有关有效的代理属性列表，请参阅“代理配置值”。您不能在 Splunk Web 上配置代理服务器。

使用配置文件配置 SOCKS5 代理连接

1. 复制一份 `$SPLUNK_HOME/etc/system/default/outputs.conf` 并将其放入 $SPLUNK_HOME/etc/system/local</code>。

2. 打开 `$SPLUNK_HOME/etc/system/local/outputs.conf` 进行编辑。

3. 在 `outputs.conf` 中定义转发服务器或输出组，具体方式为新建 `[tcpout]` 或 `[tcpout-server]` 段落。请参阅“使用 outputs.conf 配置转发器”。

4. 在应该具备 SOCKS5 代理支持的连接的段落中，为符合您的代理配置的 SOCKS 添加属性。您至少必须指定 `socksServer` 属性以启用代理支持。

5. 保存文件并将其关闭。

6. 重新启动转发器。

7. 在接收索引器上，使用“搜索和报表”应用确认索引器收到数据。

代理配置值

使用下面的属性来在转发器上配置 SOCKS5：

属性	描述	默认
`socksServer`	向转发器告知主机名或 IP 地址以及为转发数据应该连接到的 SOCKS5 代理的端口。 您可以指定 `host:port` 或 `IP address:port` 之一。您必须指定主机名或 IP 地址以及端口。您必须指定该属性以启用 SOCKS5 支持。	N/A
`socksUsername`	（可选）如果在连接阶段要求验证，告知转发器使用该用户名来验证 SOCKS5 代理主机。	N/A
`socksPassword`	（可选）告知转发器在验证 SOCKS5 代理主机（在连接阶段要求验证）时提供该密码。 当加载与段落关联的配置时，转发器会混淆该密码。不过，还有一些安全方面的注意事项。请参阅“安全注意事项”。	N/A
`socksResolveDNS`	（可选）在向 SOCKS5 代理主机传递信息之前，告知转发器是否应该使用 DNS 来解析输出组里索引器的主机名称。 当您设置该属性为 `true` 时，转发器照样将索引器名称发送给 SOCKS5 代理主机，然后 SOCKS5 代理主机必须通过 DNS 解析索引器主机名称。例如，如果转发器和代理服务器位于由不同的 DNS 服务器服务的不同网络中，则设置为 `true`。 当设置为 `false` 时，转发器尝试通过自己的 DNS 来解析索引器主机名称，如果成功，则将解析出的索引器 IP 地址发送给 SOCKS5 代理主机。 此属性仅适用于您在 `[tcpout]` 或 `[tcpout-server]` 段落中为索引器指定了主机名称的情况。如果您指定了 IP 地址，则不会进行 DNS 解析。	`false`

SOCKS5 支持的示例

以下是启用 SOCKS5 代理支持的 `outputs.conf` 段落的一些示例：

该示例建立与除了主机以外将数据转发至索引器的 SOCKS5 代理主机的连接。

```
[tcpout]
defaultGroup = proxy_indexers

[tcpout:proxy_indexers]
server = indexer1.slapstick.com:9997, indexer2.slapstick.com:9997
socksServer = prx.slapstick.com:1080
```

该示例在尝试发送数据前使用凭据来验证代理主机，并告知代理主机解析 DNS 以确定发送数据要连接的索引器：

```
[tcpout]
defaultGroup = socksCredentials

[tcpout:socksCredentials]
server = indexer3.slapstick.com:9997
socksServer = prx.slapstick.com:1081
socksUsername = proxysrv
socksPassword = letmein
socksResolveDNS = true
```

安全注意事项

当使用该功能时，请注意以下警告：

- SOCKS5 代理支持仅存在于转发器和所包含的索引器之间。不支持使用带有任何其他 Splunk 功能、应用或加载项的 SOCKS。
- SOCKS5 协议用明文形式发送验证凭据。由于实现此结果，这些凭据对于中间人攻击者来说相当脆弱。这意味着攻击可以秘密传递，并可能更改 SOCKS 客户端和 SOCKS 代理主机之间的通信。这是 SOCKS 协议的一个注意事项，而不是在 Splunk 软件中此功能的实现。
- 为了获得最安全的结果，仅在 SOCKS 代理主机保护的网络内部的转发器上使用 SOCKS 属性。在未受保护的环境中部署转发器会导致通过第三方甚至通过启用了 SOCKS 代理支持的转发器拦截 SOCKS 凭据。

配置中间转发器

本主题提供了关于如何设置中间转发器层的说明。

中间转发是转发器从一个或多个转发器接收数据然后向另一个索引器发送该数据的位置。例如，此类设置适用于以下场景：有许多主机分布在不同的地理区域，您想在该区域设置一个中央主机来接收转发器的数据，然后再将这些数据转发给索引器。所有转发器类型都可用作中间转发器。

关于如何在通用转发器中设置中间转发的说明，请参阅*《通用转发器》*手册中的“配置中间转发器”。

使用 Splunk Web 设置中间转发器

1. 在 Splunk Web 登录至您想要将其配置为中间转发器的 Splunk 实例。

2. 在系统栏中，选择**设置 > 转发和接收**。

3. 在“接收数据”下面单击**新增**。随后加载“接收数据 > 新增”页面。

4. 在**在此端口侦听**字段中，输入实例应当为传入转发器连接侦听的端口号。

5. 单击**保存**。转发器开始在指定的端口侦听；Splunk Web 显示“接收数据”页面。

6. 在“接收数据”下面单击**转发和接收**。Splunk Web 再次显示“转发和接收”页面。

7. 在“配置转发”行的“转发数据”下面单击**新增**。随后加载“转发数据 > 新增”页面。

8. 在“主机”字段，输入主机名或 IP 地址以及应当接收转发数据的索引器端口。

注意：请勿使用之前为此实例指定的端口，除非您已在接收器中配置相同的端口号。

9. 单击**保存**。Splunk Web 保存配置，转发器尝试连接至指定的主机和端口。

10. 重新启动转发器。请单击系统栏上的**设置 > 服务器控件。**

11. 单击**重新启动 Splunk**。

对于其他主机重复这些说明，以设置中间转发器层。

使用配置文件设置中间转发

1.在您想要用作中间转发器的主机上打开命令或 shell 提示。

2.编辑 `inputs.conf` 以配置转发器来接收数据，如“使用 inputs.conf 在转发器上配置数据集合”中所述。

3.配置转发器向接收索引器发送数据，如“使用 outputs.conf 配置转发器”中所述。

4.（可选）在中间转发器中编辑 `inputs.conf` 以配置任何本地数据输入。

5.重新启动转发器。

重复这些步骤来向该层添加更多的转发器。

配置转发器使用中间转发层

要设置其他转发器向中间转发层发送数据：

1.如果您还没有通用转发器，则先安装。

2.将转发器配置为将数据发送到中间转发器。

3.（可选）在转发器上配置本地数据导入。

4.重新启动转发器。

测试配置

要确认中间层运行正常:

1.使用 Splunk Web，登录至接收索引器。

2.打开“搜索和报表”应用。

3.运行搜索，其包含到主机（您配置其向中间转发器发送数据）之一的引用：例如：

```
host=<name or ip address of forwarder> index=_internal
```

如果您未看到事件发生，那么主机未被正确配置。有关可能的解决方案信息，请参阅“转发器/接收器连接故障排除”。

防止传输中的数据丢失

如果有 Splunk Enterprise，您可以通过启用**索引器确认**功能，防止**转发**数据至**索引器**过程中丢失数据。通过索引器确认，**转发器**将重新发送未被索引器确认为“已收到”的数据。

您开启 `outputs.conf` 中转发器上的索引器确认。默认禁用该功能。

注意：要使索引器确认正常运行，转发器和索引器都必须是 4.2 或更高版本。否则，传输将在未确认的情况下进行。

索引器确认和索引器群集

当使用转发器发送数据到索引器群集中的对等节点时，您应正常启用索引器确认。有关转发器和群集的更多信息，请阅读《*管理索引器和索引器群集*》手册中的“使用转发器获取数据”。

当一切正常时，索引器确认如何运行

转发器以大约 64kB 块的方式不断向索引器发送数据。在其等待队列中，转发器会在内存中为每个块保留一份副本，直到它收到索引器的确认为止。等待期间，转发器还会继续发送更多的数据块。

如果一切顺利，索引器会：

1.接收数据块。

2.分析数据。

3.将数据作为事件写入文件系统（原始数据和索引数据）。

4.将确认发送到转发器。

确认将告诉转发器索引器已收到数据，并成功写入文件系统。收到确认后，转发器便会从内存中释放数据块。

如果等待队列的大小足够，它不会在等待确认抵达时填充。但是，请参阅本部分以了解可能的问题以及解决这些问题

的方法，包括如何增加等待队列大小。

当出现故障时，索引器确认如何运行

当循环流程出现故障时，转发器就不会收到确认。然后，它将尝试重新发送数据块。

为何没有确认？

这里是转发器可能无法收到确认的原因：

- 索引器在接收数据后发生故障 - 例如，由于计算机故障。
- 索引器无法写入文件系统 - 例如，由于磁盘已满。
- 在确认前往转发器途中，网络中断。

转发器如何处理故障

在发送数据块后，转发器将在其等待队列中保留数据副本，直到收到确认。与此同时，它将继续正常发送额外数据块。如果转发器未在 300 秒（默认）内收到确认，它将关闭连接。您可以设置 `readTimeout` 属性（位于 `outputs.conf` 中），以更改等待时间。

如果转发器设置用于**自动负载均衡**，它将打开至组中的另一个索引器的连接（如果可用）并发送数据给它。如果转发器未设置自动负载均衡，则它会尝试和以前一样打开至同一索引器的连接并重新发送数据。

转发器会在等待队列中保留数据，直到收到确认。一旦等待队列填满，转发器将停止发送额外块，直到收到其中一个块的确认，此时将释放队列中的空间。

转发器可能关闭连接的其他原因

实际上有三种条件会导致转发器关闭网络连接：

- 读取超时。转发器未在 300（默认）秒内收到确认。这是以上描述的条件。
- 写入超时。转发器无法在 300（默认）秒内完成网络写入。该值可以在 `outputs.conf` 中配置，即设置 `writeTimeout`。
- 读取/写入失败。通常的原因包括索引器的计算机崩溃或网络中断。

在所有这些情况下，转发器将尝试打开至负载均衡组下一个索引器的连接，或在未启用负载均衡的情况下至同一索引器。

重复可能性

索引器可能索引同一数据块两次。如果网络问题阻止确认抵达转发器，将会出现这种情况。例如，假定索引器收到一个数据块，分析并写入文件系统。然后，它将生成确认。然而，在返回转发器时，网络中断使得转发器不会收到确认。当网络恢复时，转发器会重新发送数据块，索引器将分析并作为新数据写入。

要处理此类可能性，转发器每次重新发送数据块时，它将写入事件到其 `splunkd.log`，请注意，这可能是一个重复进程。管理员负责使用日志信息以跟踪索引器上的重复数据。

下面是重复警告的示例：

```
10-18-2010 17:32:36.941 WARN  TcpOutputProc - Possible duplication of events with
channel=source::/home/jkerai/splunk/current-install/etc/apps/sample_app
/logs/maillog.1|host::MrT|sendmail|, streamId=5941229245963076846, offset=131072
subOffset=219 on host=10.1.42.2:9992
```

启用索引器确认

您配置转发器上的索引器确认。在 `outputs.conf` 中设置 `useACK` 属性为 `true`：

```
[tcpout:<target_group>]
server=<server1>, <server2>, ...
useACK=true
```

默认情况下，`useACK` 将设置为 `false`。

注意：您可以全局或按目标组设置 `useACK`，在 `[tcpout]` 或 `[tcpout:<target_group>]` 段落级别。您无法为 `[tcpout-server: ...]` 段落级别的单个接收索引器设置它。

有关更多信息，请参阅 outputs.conf 规范文件。

索引器确认和转发的数据吞吐量

转发器使用等待队列管理索引器确认流程。本队列的默认最大大小为 21MB，这通常已经足够。然而，在少数情况下，您可能需要手动调整等待队列大小。

如果希望获得有关等待队列的更多信息，请阅读本部分。其中介绍了如何配置等待队列大小。它还提供了有关等待队列如何运行的详细信息。

如何配置等待队列大小

您不直接设置等待队列大小。与此相反，您设置内存输出队列的大小，同时等待队列大小自动设置为输出队列大小的三倍。要配置输出队列大小，使用 `maxQueueSize` 属性（位于 `outputs.conf`）。

`maxQueueSize` 属性的默认值是 `auto`。Splunk 建议您保留本设置。它将基于是否启用索引器确认优化队列大小：

- 当 `useACK=true` 时，输出队列大小是 7MB，等待队列大小是 21MB。
- 当 `useACK=false` 时，输出队列大小是 500KB。

如果需要，可以将 `maxQueueSize` 设置为特定值。有关 `maxQueueSize` 的进一步详细信息，请参阅 outputs.conf 规范文件。

请注意以下有关 `maxQueueSize=auto` 建议的点：

- 当打开索引器确认时，增加队列大小仅在重新启动转发器后生效。
- `auto` 设置仅对 5.0.4 和更高版本的转发器可用。对于运行索引器确认的更早转发器版本，您需要明确设置 `maxQueueSize` 属性为 7MB。

等待队列为何重要

如果您启用索引器确认，转发器将使用等待队列管理确认流程。由于转发器将连续发送数据块，同时不会在发送下个块之前等待确认，因此其等待队列通常会保留许多数据块，每个块都在等待自己的确认。转发器将继续发送块，直到填满等待队列，此时将停止转发。然后，转发器将等待收到确认，从而从队列释放块以恢复转发。

当网络或索引器出现故障时，等待队列将填满；然而，它仍会保持填满，即使索引器恢复正常运行。这是因为索引器仅在写入数据到文件系统后发送确认。写入文件系统的任何延迟将减慢确认节奏，导致等待队列填满。

这里有正常运行的索引器可能延迟写入数据到文件系统（同时延迟其发送确认）的若干原因：

- 索引器非常繁忙。例如，在数据抵达时，索引器可能正在处理多个搜索请求，或大量转发器正在发送数据。
- 索引器正在接收过*少*数据。为提高效率，索引器仅定期写入文件系统 -- 无论在写入队列填满或超时几秒后。如果写入队列很慢填满，则索引器将等待超时写入。如果数据仅来自几个转发器，索引器会以超时条件结束，即使每个转发器正在发送正常数量的数据。由于写入队列以热数据桶存在，因此当某些数据桶正收到少量数据时，将出现这种情况。通常，这意味着某个索引正获得少量数据。

为确保吞吐量不因转发器等待索引器确认而降低性能，您通常应保留 `maxQueueSize=auto` 的默认设置。在少数情况下，您可能需要增加等待队列大小，以便转发器拥有足够空间在内存中保留所有块，同时等待确认抵达。另一方面，如果您有许多转发器正在提供数据给单个索引器，同时每个转发器具有一些数据源，则可以使用更少大小保留几 MB 的内存。

当接收器是转发器而不是索引器时

您还可在通过中间转发器传输数据时使用索引器确认；即，原始转发器发送数据到中间转发器，然后转发到索引器。对于本方案，如果希望使用索引器确认，则建议您沿数据传输的所有段启用它。这样，您可确保沿整个路径从原始转发器传输到索引器。

如果您拥有发送数据到中间转发器的原始转发器，则将反过来会转发该数据到索引器。要启用沿整个传输路径的索引器确认，您必须启用它两次：首先启用原始转发器和中间转发器之间的段，然后再次启用中间转发器和索引器之间的段。

如果同时启用这两个传输段，中间转发器将等待收到索引器的确认，然后发送确认回原始转发器。

然而，如果仅启用其中一个段，则仅会收到该部分传输的索引器确认。例如，假定为原始转发器至中间转发器的段，而不是中间转发器至索引器的段启用索引器确认。在这种情况下，只要它发送数据到索引器，中间转发器就会发送确认回原始转发器。然后，它依赖于 TCP 安全传输数据到索引器。由于未为本第二段启用索引器确认，因此中间转发器将无法验证至索引器的数据传输。第二种情况带有限的值，同时不建议使用。

路由和过滤数据

您可以使用重型转发器过滤和路由事件数据到 Splunk 实例。您还可以执行选择性索引和转发，这样您可以本地索引一些数据，并转发您未索引到单独的索引器的数据。

有关路由数据到非 Splunk 系统的信息，请参阅“转发数据到第三方系统”。

有关执行选择性索引和转发的信息，请参阅本主题中的“执行选择性索引和转发”。

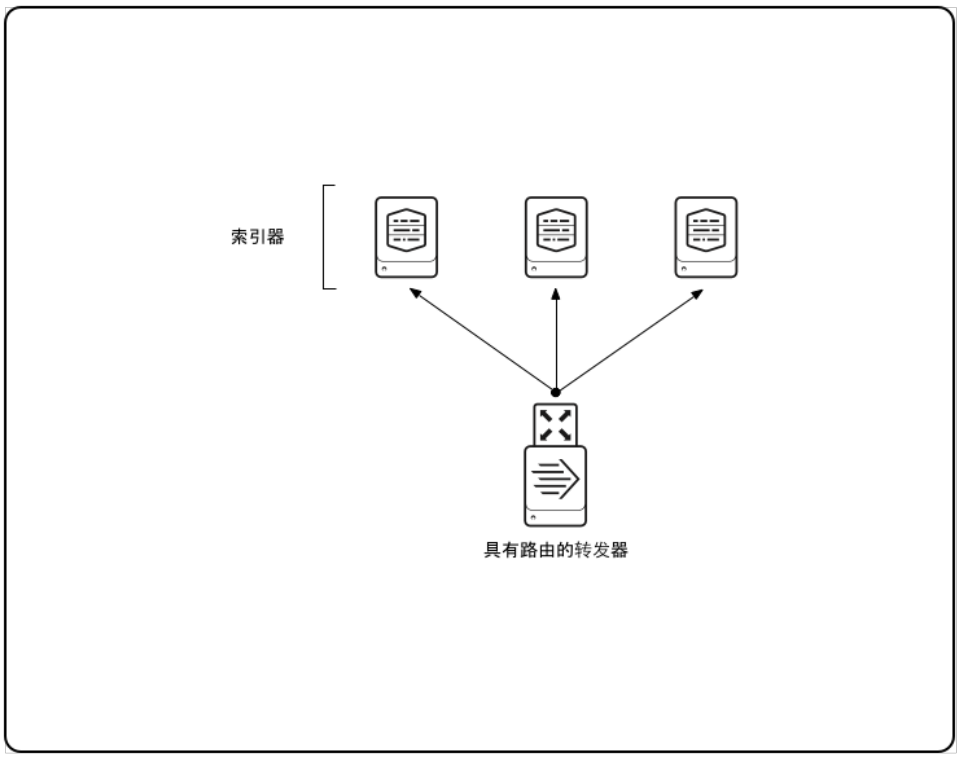
路由并筛选转发器功能

重型转发器可以基于各种标准过滤和路由数据到特定接收器，如数据来源、来源类型或事件本身的模式。例如，您可以发送来自一个计算机组的所有数据到一个索引器，发送第二个计算机组的所有数据到第二个索引器。重型转发器还可查看事件并相应过滤或路由。例如，您可以使用重型转发器检查 WMI 事件代码，从而过滤或路由 Windows 事件。本主题介绍了一些典型路由方案。

除了路由到接收器之外，重型转发器还会过滤并路由事件到特定队列，或通过发送到空队列丢弃数据。

仅重型转发器可根据事件路由或过滤所有数据。通用转发器和轻型转发器无法检查单个事件，除非提取带有结构化数据的字段。这些转发器仍可以根据主机、来源或来源类型转发数据。还可基于数据的输入段落路由，如以下“基于数据的输入发送输入到特定索引器”子主题所述。某些输入类型在获取数据类型时可以过滤数据类型。

如下为转发器路由数据到三个索引器的简单说明：



配置路由

您仅可在重型转发器上配置路由。

1. 通过回答如下问题确定路由的使用条件：
   - 您将如何确定事件类型？
   - 您将在哪里路由事件？
2. 在进行路由的 Splunk 实例中打开 shell 或命令提示。
3. 编辑 `$SPLUNK_HOME/etc/system/local/props.conf` 以添加 `TRANSFORMS-routing` 设置，确定基于事件元数据的路由。例如：

   ```
   [<spec>]
   TRANSFORMS-routing=<transforms_stanza_name>
   ```

   在此 `props.conf` 段落中：

   - `<spec>` 可以为：
     - `<sourcetype>`，即事件的来源类型
     - `host::<host>`，其中 <host> 是事件的主机
     - `source::<source>`，其中 <source> 是事件的数据来源
   - 如果您有多个 TRANSFORMS 属性，每个属性使用唯一的名称。例如："TRANSFORMS-routing1"、"TRANSFORMS-routing2" 等等。
   - `<transforms_stanza_name>` 必须是唯一。

   在 `transforms.conf` 中新建条目时使用此处指定的 `<transforms_stanza_name>`

   本主题以下示例显示了如何使用本语法。

4. 保存文件。
5. 编辑 `$SPLUNK_HOME/etc/system/local/transforms.conf` 以指定目标组，并基于事件模式设置其他路由条件。例如：

```
[<transforms_stanza_name>]
REGEX=<routing_criteria>
DEST_KEY=_TCP_ROUTING
FORMAT=<target_group>,<target_group>,....
```

   在此 `transforms.conf` 段落中：

   - `<transforms_stanza_name>` 必须与您在 `props.conf` 中指定的名称匹配。
   - 在 `<routing_criteria>` 中，输入决定要路由哪个事件的正则表达式规则。需要本行。如果不希望额外筛选超出 `props.conf` 指定的元数据，使用 "`REGEX = .`"。
   - `DEST_KEY` 应设置为 `_TCP_ROUTING` 以通过 TCP 发送事件。还可为其他输出处理器设置为 `_SYSLOG_ROUTING` 或 `_HTTPOUT_ROUTING`。
   - 设置 `FORMAT` 为匹配您在 `outputs.conf` 中定义的目标组名称的 `<target_group>`。如果您指定一个以上的目标组，请使用逗号将它们隔开。以逗号分隔的列表将复制事件到多个目标组。

   本主题以下示例显示了如何使用本语法。

6. 编辑 `$SPLUNK_HOME/etc/system/local/outputs.conf` 以定义路由数据的目标组。例如：

```
[tcpout:<target_group>]
server=<ip>:<port>
```

   在此 `outputs.conf` 段落中：

   - 您可以设置 `<target_group>` 以匹配 `transforms.conf` 中指定的名称。
   - 您可以设置 IP 地址和端口以匹配接收服务器。

   本主题介绍的使用案例遵照这一模式。

过滤和路由事件数据到目标组

在本例中，重型转发器将过滤三种类型的事件，将它们路由到不同目标组。转发器将依照这些标准过滤和路由：

- 具有 "syslog" 的来源类型的事件至负载均衡目标组
- 包含单词 "error" 的事件至第二个目标组
- 所有其他事件至默认目标组

1. 在将进行路由的实例中打开命令或 shell 提示。
2. 编辑 `$SPLUNK_HOME/etc/system/local/props.conf`（位于 `$SPLUNK_HOME/etc/system/local`）以设置两个 `TRANSFORMS-routing` 属性：一个用于 syslog 数据，一个默认用于所有其他数据。

```
[default]
TRANSFORMS-routing=errorRouting

[syslog]
TRANSFORMS-routing=syslogRouting
```

3. 编辑 `$SPLUNK_HOME/etc/system/local/transforms.conf` 以设置每个路由转换的路由规则。

```
[errorRouting]
REGEX=error
DEST_KEY=_TCP_ROUTING
FORMAT=errorGroup

[syslogRouting]
REGEX=.
DEST_KEY=_TCP_ROUTING
FORMAT=syslogGroup
```

   在本例中，如果 syslog 事件包含单词 "error"，它将路由到 `syslogGroup`，而不是 `errorGroup`。这是因为之前在 `props.conf` 指定的设置。这些设置指示通过 `syslogRouting` 转换过滤所有 syslog 事件，而所有非 syslog（默认）事件通过 `errorRouting` 转换过滤。因此，仅检查非 syslog 事件的错误。

4. 编辑 `$SPLUNK_HOME/etc/system/local/outputs.conf` 以定义目标组。

```
[tcpout]
defaultGroup=everythingElseGroup

[tcpout:syslogGroup]
server=10.1.1.197:9996, 10.1.1.198:9997

[tcpout:errorGroup]
server=10.1.1.200:9999

[tcpout:everythingElseGroup]
server=10.1.1.250:6666
```

syslogGroup 和 errorGroup 将依照 transforms.conf 中指定的规则接收事件。所有其他事件将路由到默认组 everythingElseGroup。

复制数据子集到第三方系统

本例使用数据过滤以路由两个数据流。它将转发：

- 以处理过的形式转发所有数据到 Splunk Enterprise 索引器 (10.1.12.1:9997)
- 以原始形式复制的数据子集到第三方计算机 (10.1.12.2:1234)

该示例将作为 TCP 发送这两个流。要将第二个流作为 syslog 数据发送，首先通过索引器路由数据。

1. 在将进行路由的 Splunk 实例中打开命令或 shell 提示。
2. 编辑 $SPLUNK_HOME/etc/system/local/props.conf。
```
[syslog]
TRANSFORMS-routing = routeAll, routeSubset
```
3. 编辑 $SPLUNK_HOME/etc/system/local/transforms.conf 并添加以下文本：
```
[routeAll]
REGEX=(.)
DEST_KEY=_TCP_ROUTING
FORMAT=Everything

[routeSubset]
REGEX=(SYSTEM|CONFIG|THREAT)
DEST_KEY=_TCP_ROUTING
FORMAT=Subsidiary,Everything
```
4. 编辑 $SPLUNK_HOME/etc/system/local/outputs.conf 并添加以下文本：
```
[tcpout]
defaultGroup=nothing

[tcpout:Everything]
disabled=false
server=10.1.12.1:9997

[tcpout:Subsidiary]
disabled=false
sendCookedData=false
server=10.1.12.2:1234
```
5. 重新启动 Splunk 软件。

过滤事件数据并发送到队列

尽管类似于基于转发器的路由，但是可由索引器以及重型转发器执行队列路由。它不会使用 outputs.conf 文件，而是使用 props.conf 和 transforms.conf。

可以通过把不需要的数据路由至 nullQueue 将这些数据删除，Unix /dev/null 设备的 Splunk 等效部分。以这种方式筛掉数据时，Splunk 不会转发该数据，同时也不计入您的索引量中。

请参阅本主题后续的“路由和筛选结构化数据的注意事项”。

丢弃特定事件并保留其余事件

本示例将发送它们到 nullQueue，丢弃 /var/log/messages 中的所有 sshd 事件：

1. 在 props.conf 中，设置 TRANSFORMS-null 属性：
```
[source::/var/log/messages]
TRANSFORMS-null= setnull
```
2. 在 transforms.conf 中新建相应段落。设置 DEST_KEY 为“队列”，同时设置 FORMAT 为“空队列”：
```
[setnull]
REGEX = \[sshd\]
DEST_KEY = queue
FORMAT = nullQueue
```
3. 重新启动 Splunk Enterprise。

保留特定事件并丢弃其余事件

只保留一些事件并丢弃其余事件需要两个转换。在此方案中，与之前不同，setnull 转换将所有事件路由到 nullQueue，而 setparsing 转换将选择 sshd 事件并将它们发送到 indexQueue

与其他索引时间字段提取相同，转换处理按您指定的顺序从左到右进行。关键区别在于您指定段落的顺序。在此示例中，列表中必须首先显示 setnull 段落这是因为如果您将其设置在最后，它将匹配所有事件并将所有事件发送到 nullQueue，因为是最后一个转换，它将丢弃所有这些事件，即使是之前匹配 setparsing 段落的事件也将被丢弃。

将 setnull 转换设置在首位时，它将匹配所有事件并在标记后发送到 nullQueue。然后进行 setparsing 转换，并标记匹配 [sshd] 的事件以转到 indexQueue.结果就是包含 [sshd] 的事件得以通过，所有其他事件将被丢弃。

1. 编辑 props.conf 并添加以下：
```
[source::/var/log/messages]
TRANSFORMS-set= setnull,setparsing
```
2. 编辑 transforms.conf 并添加以下：
```
[setnull]
REGEX = .
DEST_KEY = queue
FORMAT = nullQueue

[setparsing]
REGEX = \[sshd\]
DEST_KEY = queue
FORMAT = indexQueue
```
3. 重新启动 Splunk Enterprise。

"筛选 WMI 和事件"日志事件

您可以在转发器级别上直接过滤 WinEventLog 事件。

否则要过滤 WMI 事件，使用 props.conf 中的 [WMI:WinEventLog:Security] 来源类型段落。以下示例使用正则表达式以过滤掉两个 Windows 事件代码 592 和 593：

1. 编辑 props.conf 并添加以下：
```
[WinEventLog:Security]
TRANSFORMS-wmi=wminull
```
2. 编辑 transforms.conf 并添加以下：
```
[wminull]
REGEX=(?m)^EventCode=(592|593)
DEST_KEY=queue
FORMAT=nullQueue
```
3. 重新启动 Splunk Enterprise。

按目标索引筛选数据

转发器具有 forwardedindex 筛选器，您可以通过该筛选器指定是否基于目标索引转发数据。例如，如果您有一个数据导入且您已将其中包含的事件指定到 index1 索引，还有一个您希望其事件归到 index2 索引的数据导入，您可以使用过滤器仅转发指定到 index1 索引的数据，同时忽略指定到 index2 索引的数据。

forwardedindex 过滤器使用 **whitelists** 和 **blacklists** 指定筛选。有关设置多个索引的信息，请参阅*《管理索引器和索引器群集手册》*中的"新建自定义索引"。

您仅可以使用全局 [tcpout] 段落中的 forwardedindex 过滤器。如果在任何其他 outputs.conf 段落下指定筛选器，则筛选器不会运行。

您可以使用 outputs.conf 中的 forwardedindex.<n>.whitelist|blacklist 设置，以指定应转发的数据。通过使用过滤目标索引的正则表达式来配置设置完成此操作。

默认行为

默认情况下，转发器将发送用于所有外部索引的数据，包括默认索引和任何您已新建的索引。对于内部索引，默认行为因正在执行转发的组件而异：

- **通用转发器**。仅发送用于 _audit 内部索引的数据。它不会为其他内部索引转发数据。其默认 outputs.conf 文件（位于 $SPLUNK_HOME/etc/apps/SplunkUniversalForwarder/default）通过这些属性指定该行为：

```
[tcpout]
forwardedindex.0.whitelist = .*
forwardedindex.1.blacklist = _.*
forwardedindex.2.whitelist = _audit
```

- **重型转发器**。发送用于 _audit 和 _internal 内部索引的数据。其默认 outputs.conf 文件（位于 $SPLUNK_HOME/etc/system/default）通过这些属性指定该行为：

```
[tcpout]
forwardedindex.0.whitelist = .*
forwardedindex.1.blacklist = _.*
forwardedindex.2.whitelist = (_audit|_internal)
```

在大部分部署中，您无需覆盖默认设置。

有关如何将索引列入白名单和黑名单的更多信息，请参阅 outputs.conf 规范文件。有关默认和自定义 `outputs.conf` 文件及其位置的更多信息，请参阅 outputs.conf 文件类型。

转发所有外部和内部索引数据

如果希望转发所有内部索引数据，以及所有外部数据，则可覆盖默认 forwardedindex 索引器属性：

```
#Forward everything
[tcpout]
forwardedindex.0.whitelist = .*
# disable these
forwardedindex.1.blacklist =
forwardedindex.2.whitelist =
```

仅为单个索引转发数据

如果希望仅转发用于单个索引的数据（例如，如同 `inputs.conf` 中指定），并丢弃未用于该索引的任何数据，请以这种方式配置 `outputs.conf`：

```
[tcpout]
#Disable the current filters from the defaults outputs.conf
forwardedindex.0.whitelist =
forwardedindex.1.blacklist =
forwardedindex.2.whitelist =

#Forward data for the "myindex" index
forwardedindex.0.whitelist = myindex
```

首先，这会禁用来自默认 `outputs.conf` 文件的所有筛选器。然后，它将为您自己的索引设置筛选器。务必以 0 开始筛选器编号：`forwardedindex.0`。

注意：如果在系统上的 `outputs.conf` 另一个副本中设置了其他筛选器，则同时禁用这些。

您可以使用 CLI `btools` 命令确保没有任何其他筛选器位于系统上的其他 `outputs.conf` 文件：

```
splunk cmd btool outputs list tcpout
```

在组合所有配置文件版本后，本命令将返回 `tcpout` 段落的内容。

使用带本地索引的 forwardedindex 属性

在重型转发器上，您可以本地索引。为此，您必须设置 `indexAndForward` 属性为 "true"。否则，转发器仅转发您的数据，而不会保存到转发器上。另一方面，`forwardedindex` 属性仅筛选转发的数据；它们不会筛选任何保存到本地索引的数据。

在 nutshell 中，本地索引和转发器筛选是完全分离的操作，这彼此不协调。在您执行黑名单筛选时，这会有意外含义：

- 如果您设置 `indexAndForward` 为 "true"，然后通过 `forwardedindex` 黑名单属性筛选出一些数据，则转发器不会转发列入黑名单的数据，但仍会索引该数据。
- 如果您设置 `indexAndForward` 为 "false"（无本地索引），然后筛选出一些数据，则转发器将丢弃整个已筛选数据（因为它无法转发或索引已筛选的数据。）

基于数据导入路由输入到特定索引器

在本方案中，您将基于数据导入，使用 `inputs.conf` 和 `outputs.conf` 路由数据到特定索引器。通用转发器和轻型转发器可执行此类路由。

这里是显示如何执行此类操作的示例。

1. 在 `outputs.conf` 中，为每个接收索引器新建段落：
   ```
   [tcpout:systemGroup]
   server=server1:9997

   [tcpout:applicationGroup]
   server=server2:9997
   ```
2. 在 `inputs.conf` 中，指定 `_TCP_ROUTING` 以在 `outputs.conf` 中设置每个输入将用于路由的段落：

```
[monitor://.../file1.log]
_TCP_ROUTING = systemGroup

[monitor://.../file2.log]
_TCP_ROUTING = applicationGroup
```

转发器会将 file1.log 数据路由到 server1，将 file2.log 数据路由到 server2。

执行选择性索引和转发

您可以本地索引和存储数据，以及向上转发数据到接收索引器。可使用两种方法来执行本操作：

- **在转发之前索引所有数据。**为此，仅需启用 outputs.conf 中的 indexAndForward 属性。
- **在转发它或其他数据之前，索引数据子集。**这被称为**选择性索引**。通过选择性索引，您可以本地索引几个数据，然后转发到接收索引器。或者，您可以选择仅转发不希望本地索引的数据。

如果还启用了选择性索引，不要启用 [tcpout] 段落的 indexAndForward 属性。

配置选择性索引

要使用选择性索引，您必须修改 inputs.conf 和 outputs.conf。

在本示例中，[IndexAndForward] 段落的存在，包括 index 和 selectiveIndexing 设置将打开转发器的选择性索引。它将启用具有 _INDEX_AND_FORWARD_ROUTING 设置的任何输入的本地索引（在 inputs.conf 中指定）。

inputs.conf 中存在的 _INDEX_AND_FORWARD_ROUTING 设置将指示重型转发器本地索引该输入。您可将设置配置为希望的任何字符串值。转发器自行查找设置。

使用与这里显示完全相同的 [indexAndForward] 段落。

1. 在 outputs.conf 中添加 [indexAndForward] 段落：
   ```
   [indexAndForward]
   index=true
   selectiveIndexing=true
   ```
   注意：这是全局段落，这仅需要在 outputs.conf 中显示一次。
2. 为接收索引器的每个集包含目标组段落：
   ```
   [tcpout:<target_group>]
   server = <ip address>:<port>, <ip address>:<port>, ...
   ...
   ```
   转发器使用 inputs.conf 中命名的 <target_group> 路由输入。
3. 在 inputs.conf 中添加 _INDEX_AND_FORWARD_ROUTING 设置到希望本地索引的每个输入段落：
   ```
   [input_stanza]
   _INDEX_AND_FORWARD_ROUTING=<any_string>
   ...
   ```
4. 添加 _TCP_ROUTING 设置到希望转发的每个输入段落：
   ```
   [input_stanza]
   _TCP_ROUTING=<target_group>
   ...
   ```
   <target_group> 是用于 outputs.conf 的名称，以指定接收索引器的目标组。

以下几个部分显示了如何在各种方案中使用选择性索引。

本地索引一个输入，然后转发剩余输入

在本例中，转发器将本地索引来自一个输入的数据，但是不会将其转发。它还从两个其他输入转发数据，但是不会本地索引这些输入。

1. 在 outputs.conf 中，新建以下段落：
   ```
   [tcpout]
   defaultGroup=noforward
   disabled=false

   [indexAndForward]
   index=true
   selectiveIndexing=true

   [tcpout:indexerB_9997]
   server = indexerB:9997
   ```

```
[tcpout:indexerC_9997]
server = indexerC:9997
```

由于 `defaultGroup` 被设置为非现有组 "noforward"（意味着没有 `defaultGroup`），则转发器仅转发路由到 `inputs.conf` 的明确目标组的数据。它丢弃所有其他数据。

2. 在 `inputs.conf` 中，新建以下段落：

```
[monitor:///mydata/source1.log]
_INDEX_AND_FORWARD_ROUTING=local

[monitor:///mydata/source2.log]
_TCP_ROUTING=indexerB_9997

[monitor:///mydata/source3.log]
_TCP_ROUTING=indexerC_9997
```

结果是，转发器：

- 索引本地 `source1.log` 数据，但是不会将其转发（因为其输入段落中没有任何明确路由，同时 `outputs.conf` 中没有默认组）。
- 转发 `source2.log` 数据到 indexerB，但是不会本地索引。
- 转发 `source3.log` 数据到 indexerC，但是不会本地索引。

本地索引一个输入，然后转发所有输入

本示例与之前的示例几乎相同。差别在于，您在此仅本地索引一个输入，但是之后示例会转发所有输入，包括本地索引的输入。

1. 在 `outputs.conf` 中，新建以下段落：

```
[tcpout]
defaultGroup=noforward
disabled=false

[indexAndForward]
index=true
selectiveIndexing=true

[tcpout:indexerB_9997]
server = indexerB:9997

[tcpout:indexerC_9997]
server = indexerC:9997
```

2. 在 `inputs.conf` 中，新建以下段落：

```
[monitor:///mydata/source1.log]
_INDEX_AND_FORWARD_ROUTING=local
_TCP_ROUTING=indexerB_9997

[monitor:///mydata/source2.log]
_TCP_ROUTING=indexerB_9997

[monitor:///mydata/source3.log]
_TCP_ROUTING=indexerC_9997
```

与上一示例的唯一差别在于，您为本地索引的输入指定了 `_TCP_ROUTING` 属性。转发器将同时路由 `source1.log` 和 `source2.log` 到 `indexerB_9997` 目标组，但是仅本地索引来自 `source1.log` 的数据。

另一种方式是本地索引一个输入，然后转发所有输入

您可以设置 `defaultGroup` 为实际目标组，实现与上一示例完全相同的结果。

1. 在 `outputs.conf` 中，新建以下段落：

```
[tcpout]
defaultGroup=indexerB_9997
disabled=false

[indexAndForward]
index=true
selectiveIndexing=true

[tcpout:indexerB_9997]
server = indexerB:9997
```

```
[tcpout:indexerC_9997]
server = indexerC:9997
```

本 outputs.conf 将设置 defaultGroup 为 indexerB_9997。

2. 在 inputs.conf 中，新建以下段落：

```
[monitor:///mydata/source1.log]
_INDEX_AND_FORWARD_ROUTING=local

[monitor:///mydata/source2.log]
_TCP_ROUTING=indexerB_9997

[monitor:///mydata/source3.log]
_TCP_ROUTING=indexerC_9997
```

即使您未为 source1.log 设置明确路由，转发器会仍将其发送到 indexerB_9997 目标组，因为 outputs.conf 将该组指定为 defaultGroup。

选择性索引和内部日志

在 outputs.conf 中启用选择性索引之后，转发器将仅本地索引带有 _INDEX_AND_FORWARD_ROUTING 设置的输入。这适用于 /var/log/splunk 目录的内部日志（在默认 etc/system/default/inputs.conf 中指定）。默认情况下，转发器不会索引那些日志。如果希望索引它们，您必须添加输入段落到您的本地 inputs.conf 文件（这将优先于默认文件）并包括 _INDEX_AND_FORWARD_ROUTING 属性：

```
[monitor://$SPLUNK_HOME/var/log/splunk]
index = _internal
_INDEX_AND_FORWARD_ROUTING=local
```

路由和筛选结构化数据的注意事项

Splunk 软件不会分析已转发到索引器上的结构化数据

当您向索引器转发结构化数据时，该数据到达索引器之后 Splunk 软件不对其进行分析，即使已通过 INDEXED_EXTRACTIONS 及其相关属性在该索引器上配置了 props.conf。转发数据跳过索引器上的下列队列，这就排除了索引器上对于该数据的任何分析：

- parsing
- aggregation
- typing

转发数据在到达索引器时必须已经过分析。要实现该目标，您还必须在发送数据的转发器上设置 props.conf。这包括 INDEXED_EXTRACTIONS 的配置，以及其他任何分析、筛选、匿名及路由规则。通用转发器能够单独地为结构化数据执行这些任务。请参阅“转发从标头文件提取的数据”。

转发数据到第三方系统

Splunk 转发器可以通过纯 TCP 套接字或打包为标准 syslog 的方式把原始数据转发到非 Splunk 系统。由于转发到非 Splunk 系统，因此它们仅可以发送原始数据。

通过编辑 outputs.conf、props.conf 和 transforms.conf，，您可以配置重型转发器，利用其将数据有条件地路由至第三方系统，与有条件地路由数据到其他 Splunk 实例方法一样。您可以按主机、数据来源或数据来源类型筛选数据。您还可以使用正则表达式进一步量化数据。

向第三方系统转发数据是 Splunk 软件提供的多种搜索结果导出方法之一。有关您可用的其他导出方法的信息，请参阅*《搜索手册》*中的“导出搜索结果”。

TCP 数据

您可以使用任何类型的转发器，如通用转发器，以转发 TCP 数据至第三方系统：

1.配置接收主机的第三方以预期在 TCP 端口传入数据。

2.编辑 outputs.conf 以指定接收主机和端口。

要路由数据，您必须使用重型转发器，这将有能力分析数据。

3.编辑 props.conf 以确定要路由的数据。

4.编辑 transforms.conf 以根据在 props.conf 中配置的内容确定数据将路由到的位置。

编辑配置文件

要转发数据，编辑 outputs.conf：

- 指定接收服务器的目标组。
- 指定每个接收服务器的 IP 地址和 TCP 端口。
- 设置 sendCookedData 为 false，以便转发器发送原始数据。

要只是在重型转发器中路由和筛选数据，请同时编辑 props.conf 和 transforms.conf：

- 在 props.conf 中，指定数据流的主机、数据来源或来源类型。指定要对输入执行的转换。
- 在 transforms.conf 中，定义转换并指定 _TCP_ROUTING。您还可以使用正则表达式进一步筛选数据。

转发所有数据

本例显示了如何从转发器发送所有数据到第三方系统。由于正在发送所有数据，因此您仅需要编辑 outputs.conf：

```
[tcpout]

[tcpout:fastlane]
server = 10.1.1.35:6996
sendCookedData = false
```

转发数据子集

本例显示了如何使用重型转发器筛选数据子集，并发送子集到第三方系统：轻型和通用转发器无法路由或筛选数据。

1.编辑 props.conf 和 transforms.conf 以指定筛选条件。

在 props.conf 中，应用 bigmoney 转换到所有以 nyc 开头的主机名称：

```
[host::nyc*]
TRANSFORMS-nyc = bigmoney
```

在 transforms.conf 中，配置 bigmoney 转换以指定 TCP_ROUTING 为 DEST_KEY，以及 bigmoneyreader 目标组为 FORMAT：

```
[bigmoney]
REGEX = .
DEST_KEY=_TCP_ROUTING
FORMAT=bigmoneyreader
```

2.在 outputs.conf 中，为非 Splunk 服务器和默认目标组同时定义 bigmoneyreader 目标组，以接收任何其他数据：

```
[tcpout]
defaultGroup = default-clone-group-192_168_1_104_9997

[tcpout:default-clone-group-192_168_1_104_9997]
server = 192.168.1.104:9997

[tcpout:bigmoneyreader]
server=10.1.1.197:7999
sendCookedData=false
```

转发器将从所有以 nyc 开头的主机名称发送所有数据到 bigmoneyreader 目标组指定的非 Splunk 服务器。它将从所有其他主机发送数据到 default-clone-group-192_168_1_104_9997 目标组指定的服务器。

注意：如果希望仅转发在 props.conf 和 transforms.conf 中专门确定的数据，设置 defaultGroup=nothing。

Syslog 数据

您可以配置重型转发器以标准 syslog 格式发送数据。转发器将通过单独的输出处理器发送数据。syslog 输出处理器不对通用或轻型转发器可用。

syslog 输出处理器将发送 RFC 3164 兼容事件到基于 TCP/UDP 的服务器和端口，使得任何不兼容数据的负载兼容 RFC 3164。

默认情况下，Splunk 软件不会更改事件的内容，以使其字符集与第三方服务器兼容。您可以在 props.conf 中指定 SEDCMD 配置，以使数据包含第三方服务器无法处理的某些字符。要从“Windows 事件日志”事件中删除换行符时，该选项很实用。请参阅*《数据导入手册》*中“通过 sed 脚本使数据匿名”。

您还可以使用 props.conf 和 transforms.conf 筛选数据。当您进行此操作时，您需要指定 _SYSLOG_ROUTING 为

DEST_KEY。

转发 syslog 数据到第三方主机

1.确定第三方接收主机。

2.在要发送数据至第三方主机的转发器上打开 `$SPLUNK_HOME/etc/system/local/outputs.conf` 进行编辑。

3.在 `outputs.conf` 文件中添加一个可在 `syslog` 目标组中指定接收主机的段落。

```
[syslog]
defaultGroup=syslogGroup

[syslog:syslogGroup]
server = 10.1.1.197:514
```

如果要为 syslog 数据定义多个事件类型，您必须在所有事件类型名称中包含字符串 "syslog"。

转发 syslog 数据

在 `outputs.conf` 中，指定 `syslog` 目标组：

```
[syslog:<target_group>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

目标组段落需要本属性：

所需属性	默认	值
server	n/a	这必须使用格式：`<hostname_or_ ipaddress>:<port>`。这是 syslog 服务器的 IP 地址或服务器名称和 syslog 服务器正在侦听的端口的组合。请注意，syslog 服务器默认使用端口 514。

这些属性可选：

可选属性	默认	值
type	udp	传输协议。必须被设置为 "tcp" 或 "udp"。
priority	<13> - 这代表 1 的实用工具（“用户”）和 5 的严重性（“通知”）。	Syslog 优先级。这必须是长度为 1 至 3 位的整数，以尖括号括起；例如：<34>.本值将显示在 syslog 标题。 通过 syslog 界面调用模拟通过的数字；有关更多信息，请参阅 outputs.conf。 计算优先级值为 (<facility> * 8) + <severity>。如果实用工具是 4（严重性/授权消息），严重性是 2（关键条件），则优先级值将是：(4 * 8) + 2 = 34，您在配置文件中指定为 <34>。
syslogSourceType	n/a	这必须使用格式 `sourcetype::syslog`，syslog 消息的来源类型。
timestampformat	“”	当添加时间戳到标题时使用的格式。这必须使用格式：<%b %e %H:%M:%S>。有关详细信息，请参阅《数据导入手册》中的“配置时间戳”。

发送数据子集到 syslog 服务器

本例显示如何配置重型转发器，通过端口 514 从名称以 "nyc" 开头的主机发送数据到名为 "loghost.example.com" 的 syslog 服务器：

1.编辑 `props.conf` 和 `transforms.conf` 以指定筛选条件。

在 `props.conf` 中，应用 `send_to_syslog` 转换到所有以 `nyc` 开头的主机名称：

```
[host::nyc*]
```

```
TRANSFORMS-nyc = send_to_syslog
```

在 `transforms.conf` 中，配置 `send_to_syslog` 转换以指定 `_SYSLOG_ROUTING` 为 `DEST_KEY`，以及 `my_syslog_group` 目标组为 `FORMAT`：

```
[send_to_syslog]
REGEX = .
DEST_KEY = _SYSLOG_ROUTING
FORMAT = my_syslog_group
```

2.在 `outputs.conf` 中，定义非 Splunk 服务器的 `my_syslog_group` 目标组：

```
[syslog:my_syslog_group]
server = loghost.example.com:514
```

转发故障排除

转发器/接收器连接故障排除

如果无法进行转发，或者转发过程异常，可以根据这些故障排除步骤来确定原因。

接收器不接受其接收端口上新的连接

如果接收索引器上的内部队列被阻止，在指定时间间隔无法向队列插入数据后，索引器将关闭接收/侦听 (`splunktcp`) 端口。一旦队列可再次接收数据，索引器将重新打开端口。

但是，一旦队列被阻止，有时（仅在 Windows 计算机上）索引器无法重新打开端口。为进行修复，您必须重新启动索引器。

如果存在此问题，可以将 inputs.conf 中接收器的 `stopAcceptorAfterQBlock` 属性设置为高值，因此它就不会快速关闭端口。这一属性决定了关闭端口前索引器等待的时间。默认值为 300 秒（5 分钟）。

如果您使用负载均衡的转发器，它们将基于超时时间间隔使数据流切换至负载均衡组中另一个索引器，设置为 `writeTimeout` 属性放在 outputs.conf 中。这将在接收索引器已阻止队列时引发自动故障。

混淆接收和管理端口

作为设置转发器的一部分，指定接收器的 `hostname/IP_address` 和 `port`。转发器使用这些发送数据给接收器。在配置接收器时，请务必指定作为接收端口的端口。如果错误指定接收器的管理端口，接收器将生成类似以下的错误：

```
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - SSL Error = error:140760FC:SSL
routines:SSL23_GET_CLIENT_HELLO:unknown protocol
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - ACCEPT_RESULT=-1 VERIFY_RESULT=0
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - SSL Error for fd from HOST:localhost.localdomain,
IP:127.0.0.1, PORT:53075
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - SSL Error = error:140760FC:SSL
routines:SSL23_GET_CLIENT_HELLO:unknown protocol
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - ACCEPT_RESULT=-1 VERIFY_RESULT=0
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - SSL Error for fd from HOST:localhost.localdomain,
IP:127.0.0.1, PORT:53076
splunkd.log:03-01-2010 13:35:28.653 ERROR TcpInputFd - SSL Error = error:140760FC:SSL
routines:SSL23_GET_CLIENT_HELLO:unknown protocol
splunkd.log:03-01-2010 13:35:28.654 ERROR TcpInputFd - ACCEPT_RESULT=-1 VERIFY_RESULT=0
splunkd.log:03-01-2010 13:35:28.654 ERROR TcpInputFd - SSL Error for fd from HOST:localhost.localdomain,
IP:127.0.0.1, PORT:53077
splunkd.log:03-01-2010 13:35:28.654 ERROR TcpInputFd - SSL Error = error:140760FC:SSL
routines:SSL23_GET_CLIENT_HELLO:unknown protocol
splunkd.log:03-01-2010 13:35:28.654 ERROR TcpInputFd - ACCEPT_RESULT=-1 VERIFY_RESULT=0
```

已关闭接收器套接字

如果接收索引器的队列已满，它会关闭接收器套接字，以防止其他转发器与它连接。如果启用负载均衡的转发器不再转发给该接收器，它会发送数据到列表上的另一个索引器。如果转发器未清空负载均衡，它会保留数据直到您解决了问题。

当队列畅通后，接收器套接字会自动重新打开。

通常，接收器会在数据流之后，那是因为磁盘已满而导致无法继续写入数据，或者接收器本身尝试转发数据到不接收数据的另一个 Splunk Enterprise 实例。

如果套接字被阻止，`splunkd.log` 中会显示以下警告消息：

```
Stopping all listening ports. Queues blocked for more than N seconds.
```

此消息将在重新打开套接字时显示：

```
Started listening on tcp ports. Queues unblocked.
```

禁用接收

要通过 CLI 禁用接收，运行 `splunk disable listen` 命令：

```
splunk disable listen -port <port> -auth <username>:<password>
```

您还可以通过删除 `[splunktcp]` 段落（从 `inputs.conf`）禁用接收。

问答

有什么问题吗？请访问 Splunk Answers，查看 Splunk 社区有哪些与配置转发相关的问题和答案。