Red Hat Enterprise Linux 10

风险缩减和恢复操作

备份数据、日志监控和管理安全更新

Last Updated: 2025-09-25

Red Hat Enterprise Linux 10 风险缩减和恢复操作

备份数据、日志监控和管理安全更新

Legal Notice

Abstract

使用 Relax-and-Recover (ReaR)灾难恢复工具来最大程度减少灾难事件的影响，并有结构化计划在失败后恢复系统和数据。了解如何管理和监控您的安全更新，以提高系统稳定性、安全性和性能。使用日志文件检查记录的系统事件，以识别、故障排除和避免问题和监控系统功能。

Table of Contents

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 恢复系统

Red Hat Enterprise Linux 提供 Relax-and-Recover (ReaR)工具来使用现有备份恢复系统。您可以使用这个工具作为灾难恢复解决方案，也用于系统迁移。

该工具可让您执行以下任务：

- 生成可引导镜像，并使用镜像从现有备份中恢复系统。
- 复制原始存储布局。
- 恢复用户和系统文件。
- 将系统还原到不同的硬件中。

另外，对于灾难恢复，您还可以将某些备份软件与 ReaR 集成。

1.1. 设置 REAR 并手动创建备份

使用以下步骤，使用 Relax-and-Recover(ReaR)工具安装软件包，来创建一个救援系统，配置并生成一个备份。

先决条件

- 根据备份恢复计划完成必要的配置。
  请注意： 您可以使用 **NETFS** 备份方法，该方法是 ReaR 完全整合的、内置的方法。

流程

1. 安装 ReaR 工具：

   ```
   # dnf install rear
   ```

2. 在您选择的编辑器中修改 ReaR 配置文件，例如：

   ```
   # vi /etc/rear/local.conf
   ```

3. 在 **/etc/rear/local.conf** 中添加备份设置详情。例如，在使用 **NETFS** 备份方法时添加以下行：

   ```
   BACKUP=NETFS
   BACKUP_URL=backup.location
   ```

   使用备份位置的 URL 替换 *backup.location*。

4. 要将 ReaR 配置为在创建新归档时保留之前的备份归档，还需要将以下行添加到配置文件中：

   ```
   NETFS_KEEP_OLD_BACKUP_COPY=y
   ```

5. 要让递增形式进行备份，在每个运行中只备份修改了的文件，添加以下行：

   ```
   BACKUP_TYPE=incremental
   ```

6. 创建一个救援系统：

```
# rear mkrescue
```

7. 根据恢复计划创建备份。例如，在使用 **NETFS** 备份方法时请输入：

```
# rear mkbackuponly
```

另外，您可以通过运行以下命令来在一个步骤中创建救援系统和备份：

```
# rear mkbackup
```

该命令结合了 **rear mkrescue** 和 **rear mkbackuponly** 命令的功能。

1.2. 在 64 位 IBM Z 构架上使用 REAR 救援镜像

64 位 IBM Z 构架中提供了基本的 Relax 和 Recover (ReaR)功能，并被完全支持。您只能在 z/VM 环境中在 IBM Z 上创建 ReaR 救援镜像。备份和恢复逻辑分区(LPAR)还没有测试。

当前唯一可用的输出方法是 Initial Program Load(IPL)。IPL 生成一个内核和一个初始 RAM 磁盘(initrd)，可与 **zIPL** 引导装载程序一起使用。

先决条件

- 已安装 **rear** 软件包。

流程

1. 将以下变量添加到 **/etc/rear/local.conf** 中来配置 ReaR，以便在 64 位 IBM Z 构架上生成救援镜像：

   a. 要配置 **IPL** 输出方法，请添加 **OUTPUT=IPL**。

   b. 要配置备份方法和目的地，请添加 **BACKUP** 和 **BACKUP_URL** 变量。例如：

```
BACKUP=NETFS

BACKUP_URL=nfs://<nfsserver_name>/<share_path>
```

重要

目前在 64 位 IBM Z 构架上不支持本地备份存储。

   c. 可选： 您还可以配置 **OUTPUT_URL** 变量来保存内核和 **initrd** 文件。默认情况下，**OUTPUT_URL** 与 **BACKUP_URL** 保持一致。

2. 要执行备份和救援镜像创建：

```
# rear mkbackup
```

3. 这会在 **BACKUP_URL** 或 **OUTPUT_URL** （如果设置）变量指定的位置创建内核和 initrd 文件，并使用指定的备份方法进行备份。

警告

救援过程会重新格式化连接到系统的所有 DASD （直接附加存储设备）。如果系统存储设备中存在任何有价值的数据，则不要尝试系统恢复。这还包括用 zipl 引导装载程序、ReaR 内核和 initrd 准备的设备，用来引导到救援环境。确保保留一份副本。

4. 要恢复系统，请使用之前创建的 ReaR 内核和 initrd 文件，并从直接附加存储设备(DASD)或光纤通道协议(FCP)附加的 SCSI 设备使用 **zipl** 引导装载程序、内核和 **initrd** 进行引导。

5. 当救援内核和 **initrd** 引导时，它会启动 ReaR 救援环境。继续系统恢复。

其他资源

- 在 z/VM 中安装
- 使用一个准备的 DASD

1.3. REAR 排除

ReaR 工具会根据恢复过程中在 **/var/lib/rear/layout/disklayout.conf** 布局文件中的描述重新创建原始系统的存储布局。存储布局包括分区、卷组、逻辑卷、文件系统和其他存储组件。

ReaR 在创建救援镜像时创建布局文件，并在镜像中嵌入这个文件。您还可以使用 **rear savelayout** 命令创建布局文件。这可让您快速创建布局文件并检查它，而无需创建整个救援镜像。

布局文件描述了原始系统的整个存储布局，但有一些例外，因为 ReaR 从布局文件中排除一些存储组件，并在恢复过程中重新创建。从布局排除存储组件由以下配置变量控制：

- **AUTOEXCLUDE_DISKS**
- **AUTOEXCLUDE_MULTIPATH**
- **AUTOEXCLUDE_PATH**
- **EXCLUDE_RECREATE**

在配置变量从布局文件中排除某些文件系统也会从备份中排除它们的内容。您还可以使用 **BACKUP_PROG_EXCLUDE** 配置变量，从备份中排除文件或者目录树，而无需将文件系统从布局文件中排除。

当以这种方式排除文件系统中的所有文件和目录时，文件系统会在恢复过程中重新创建，但会为空，因为备份不包含要恢复到其中的任何数据。这对包含临时数据且不需要保留的文件系统很有用，或者用于使用独立于 ReaR 的方法备份的数据。

BACKUP_PROG_EXCLUDE 变量是传递至 tar 或 rsync 的 glob 样式通配符模式的数组。请注意，需要引用模式以防止 shell 在读取配置文件时被 shell 扩展。此变量的默认值在 **/usr/share/rear/conf/default.conf** 文件中设置。默认值包含 **/tmp** configured 模式，它排除 /tmp 目录下的所有文件和目录，但不排除 **/tmp** 目录本身。

如果您需要排除其他文件和目录，请将带有 **+** 字符的进一步模式附加到变量中，而不是覆盖它来保留默认值。例如，除了默认值外，要排除 **/data/temp** 目录下的所有文件和目录，请使用：

```
BACKUP_PROG_EXCLUDE+=( '/data/temp/*' )
```

您可以查看 **/usr/share/rear/conf/default.conf** 文件中的配置变量的默认值，并可以更改本地 **/etc/rear/local.conf** 配置文件中的这些值。

您还可以配置由内部 **NETFS** 和 **RSYNC** 备份方法备份哪些文件。默认情况下，如果布局文件中包括文件系统，则所有已挂载的基于磁盘的文件系统中的文件由 **rear mkbackup** 或 **rear mkbackuponly** 备份。

rear mkbackup 命令在日志中列出备份排除模式。您可以在 **/var/log/rear** 目录中找到日志文件。这可用于在执行完整系统恢复前验证排除的规则。例如，日志可以包含以下条目：

```
2025-04-29 10:17:41.312431050 Making backup (using backup method NETFS)
2025-04-29 10:17:41.314369109 Backup include list (backup-include.txt contents):
2025-04-29 10:17:41.316197323   /
2025-04-29 10:17:41.318052001 Backup exclude list (backup-exclude.txt contents):
2025-04-29 10:17:41.319857125   /tmp/*
2025-04-29 10:17:41.321644442   /dev/shm/*
2025-04-29 10:17:41.323436363   /var/lib/rear/output/*
```

在前面的输出中，整个 root 文件系统包含在备份中，但 **/tmp**、**/dev/shm** 和 **/var/lib/rear/output** 目录下的所有文件和目录除外。

其他资源

- ReaR **用户指南中的布局配置** 章节，安装 **rear** 软件包位于 **/usr/share/doc/rear/relax-and-recover-user-guide.html**
- 系统中 **glob (3)** 手册页

第 2 章 使用日志文件对问题进行故障排除

您可以使用日志文件中的信息来故障排除和监控系统功能。日志文件包含有关系统的消息，包括内核以及其上运行的服务和应用程序。Red Hat Enterprise Linux 中的日志记录系统基于内置的 **syslog** 协议。然后，各种程序使用 **syslog** 记录事件并将其整理到日志文件中。

2.1. 处理 SYSLOG 信息的服务

以下服务处理 **syslog** 信息：

systemd-journald 守护进程

从以下源收集信息并将其转发到 **Rsyslog** 以进行进一步处理：

- 内核
- 引导过程的早期阶段
- 启动并运行守护进程的标准和错误输出
- **syslog**

Rsyslog 服务

根据类型和优先级对 **syslog** 消息进行排序，并将其写入 **/var/log** 目录中的文件。**/var/log** 目录会永久保存日志信息。

2.2. 存储 SYSLOG 信息的子目录

/var/log 目录中的以下子目录存储 **syslog** 信息：

/var/log/messages

除以下以外的所有 **syslog** 信息

/var/log/secure

安全和验证相关的信息和错误

/var/log/maillog

与邮件服务器相关的消息和错误

/var/log/cron

与定期执行任务相关的日志文件

/var/log/boot.log

与系统启动相关的日志文件

2.3. 查看日志的命令

您可以使用 Journal 查看和管理日志文件，这是 **systemd** 的一个组件。它解决了与传统日志记录相关的问题，与系统的其余部分紧密集成，并支持各种日志记录技术以及日志文件的访问管理。

您可以使用 **journalctl** 命令查看系统日志中的信息，例如：

```
$ journalctl -b | grep kvm
May 15 11:31:41 localhost.localdomain kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
May 15 11:31:41 localhost.localdomain kernel: kvm-clock: cpu 0, msr 76401001, primary cpu clock
```

查看系统信息

journalctl

显示所有收集的日志条目。

journalctl FILEPATH

显示与特定文件相关的日志。例如： **journalctl /dev/sda** 命令显示与 **/dev/sda** 文件系统相关的日志。

journalctl -b

显示当前引导的日志。

journalctl -k -b -1

显示当前引导的内核日志。

查看有关特定服务的信息

journalctl -b _SYSTEMD_UNIT=*<name.service>*

过滤日志以显示与 **systemd** 服务匹配的条目。

journalctl -b _SYSTEMD_UNIT=*<name.service>* _PID=*<number>*

合并匹配。例如，这个命令显示与***<name.service>*** 和 PID ***<number>*** 匹配的 **systemd-units** 的日志。

journalctl -b _SYSTEMD_UNIT=*<name.service>* _PID=*<number>* + _SYSTEMD_UNIT=*<name2.service>*

加号(+)分隔符将两个表达式按逻辑 OR 组合在一起。例如，这个命令显示来自带有 **PID** 的 ***<name.service>*** 服务进程的所有消息，加上来自 ***<name2.service>*** 服务（来自其任何进程）的所有消息。

journalctl -b _SYSTEMD_UNIT=*<name.service>* _SYSTEMD_UNIT=*<name2.service>*

此命令显示与引用同一字段的任一表达式匹配的所有条目。在这里，这个命令会显示与 systemd-unit ***<name.service>*** 或 systemd-unit ***<name2.service>*** 匹配的日志。

查看与特定引导相关的日志

journalctl --list-boots

显示引导号、其 ID 以及与引导相关的第一条和最后一个消息的时间戳列表。您可以在下一个命令中使用 ID 来查看详细信息。

journalctl --boot=ID _SYSTEMD_UNIT=*<name.service>*

显示有关指定的引导 ID 的信息。

2.4. 其他资源

- 您系统上的 **journalctl (1)** 手册页
- 配置远程日志记录解决方案

第 3 章 在 WEB 控制台中查看和过滤日志

Red Hat Enterprise Linux web 控制台提供了一个图形界面来访问、查看和过滤日志。您可以使用最常用的功能，而无需记住对应的命令和选项。

3.1. 查看 WEB 控制台中的日志

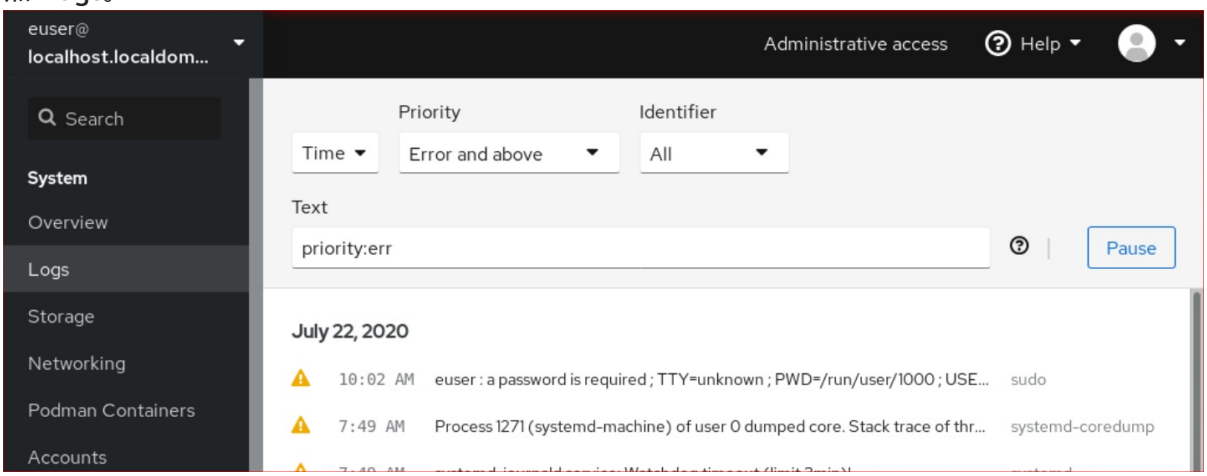
RHEL web 控制台日志部分是 **journalctl** 实用程序的 UI。您可以在 web 控制台界面中访问系统日志。

先决条件

- 已安装 RHEL 10 web 控制台。
  具体步骤请参阅安装并启用 Web 控制台。

流程

1. 登录到 RHEL 10 web 控制台。
2. 点 **Logs**。



3. 点击列表中的选定日志条目条目，打开日志条目详情。

注意

您可以使用 **暂停** 按钮在显示时暂停新日志条目。恢复新日志条目后，Web 控制台将加载您使用 **Pause** 按钮后报告的所有日志条目。

您可以根据时间、优先级或标识符过滤日志。如需更多信息，请参阅 web 控制台中的检查和过滤日志。

3.2. 在 WEB 控制台中过滤日志

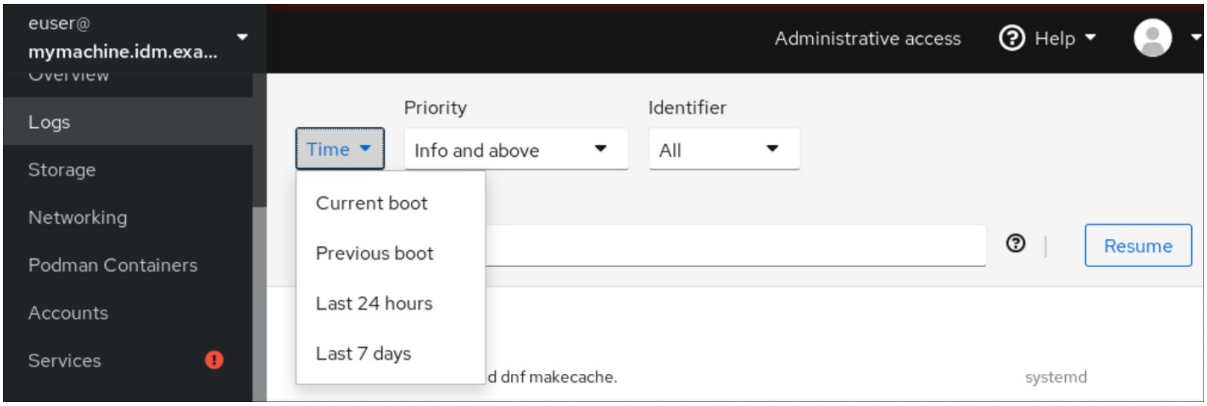
您可以在 web 控制台中过滤日志条目。

先决条件

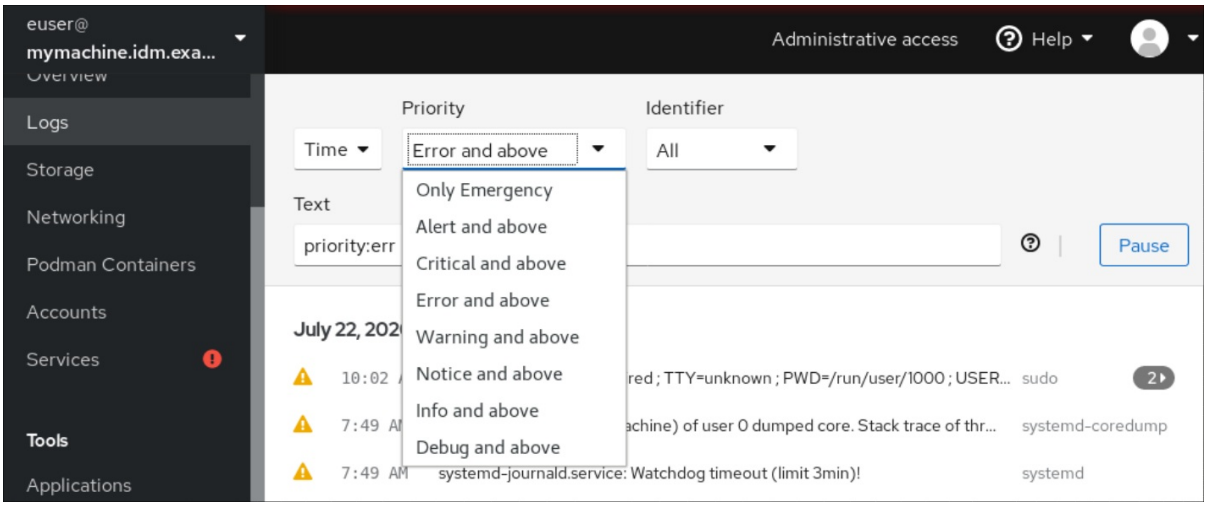
- 已安装 RHEL 10 web 控制台。
  具体步骤请参阅安装并启用 Web 控制台。

流程

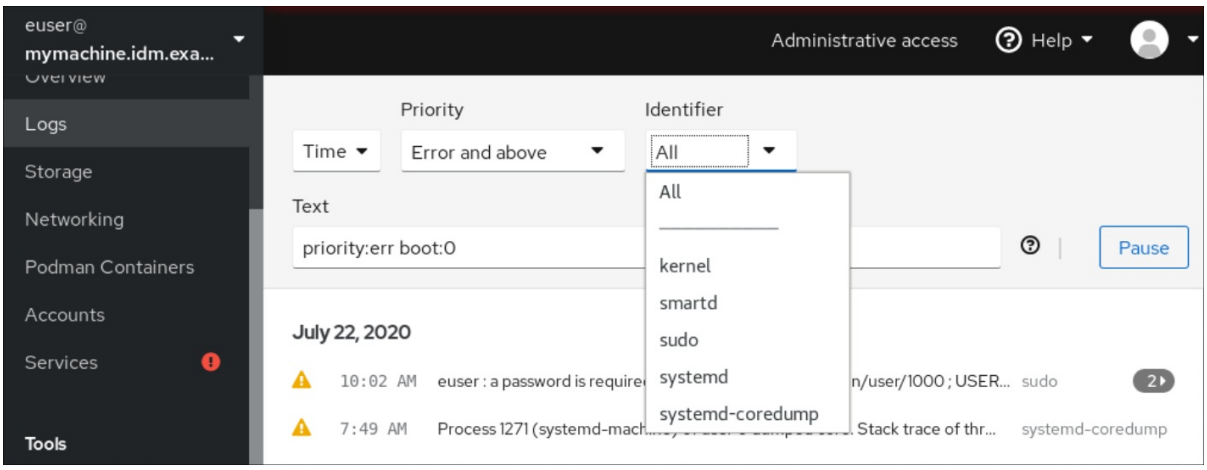
1. 登录到 RHEL 10 web 控制台。
2. 点 **Logs**。
3. 默认情况下，Web 控制台显示最新的日志条目。要根据具体时间范围过滤，请点 **Time** 下拉菜单并选择一个首选的选项。



4. 默认情况下会显示 **Error 及更高级别**的日志列表。要根据不同的优先级过滤，请点击 **Error 及更高**下拉菜单并选择一个首选的优先级。



5. 默认情况下，Web 控制台会显示所有标识符的日志。要过滤特定标识符的日志，请点 **All** 下拉菜单并选择标识符。



6. 要打开日志条目，请点所选日志。

3.3. 在 WEB 控制台中过滤日志的文本搜索选项

文本搜索选项功能为过滤日志提供了大量选项。如果您决定使用文本搜索过滤日志，您可以使用三个下拉菜单中定义的预定义选项，或者您可以自己键入整个搜索。

下拉菜单

您可以使用三个下拉菜单来指定搜索的主参数：

- **时间** ：此下拉菜单包含搜索的不同时间范围的预定义搜索。
- **Priority**：此下拉菜单提供了不同优先级级别的选项。它对应于 **journalctl --priority** 选项。默认优先级值为 **Error 及以上**。每次在不指定其它优先级时，会设置它。
- **Identifier**：在这个下拉菜单中，您可以选择要过滤的标识符。对应于 **journalctl --identifier** 选项。

限定符

您可以使用六个限定符来指定搜索。它们包含在用于过滤日志表的 Options 中。

日志字段

如果要搜索特定日志字段，可以用其内容指定字段。

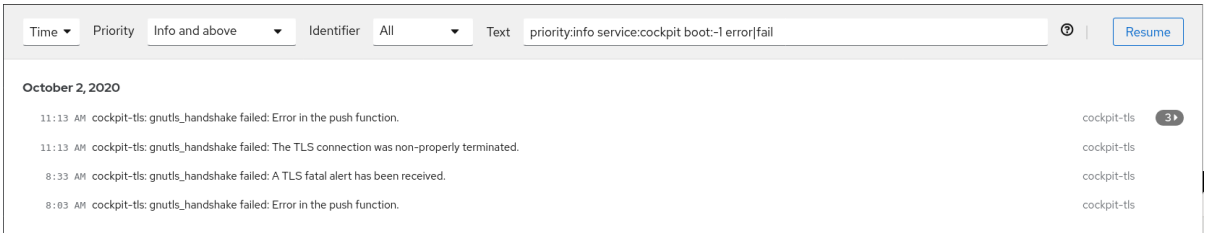
在日志信息中进行自由文本搜索

您可以在日志消息中过滤您选择的任何文本字符串。字符串也可以采用正则表达式的形式。

高级日志过滤 I

过滤 2020 年 10 月 22 日之后带有 'systemd' 识别的、日志字段 'JOB_TYPE' 是 'start' 或 'restart 的所有日志信息。

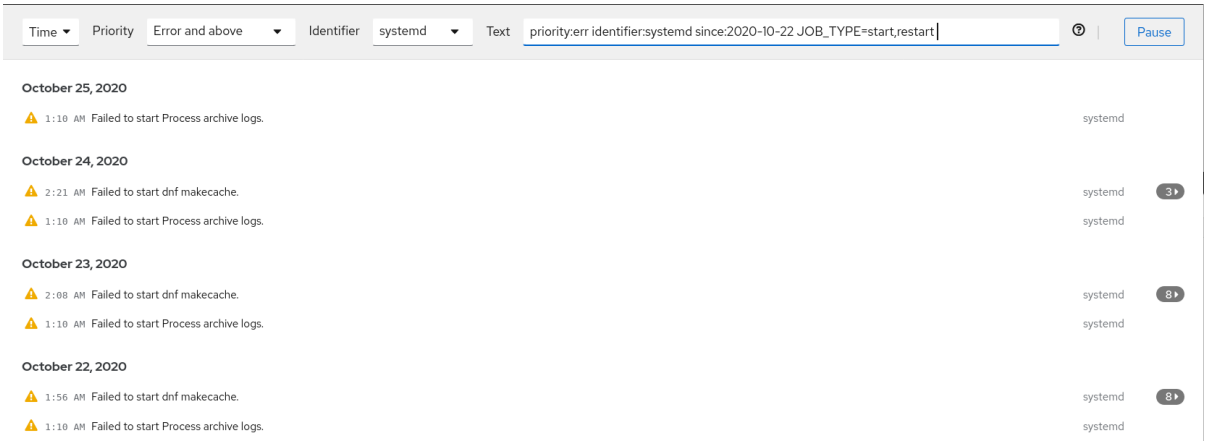
1. 在搜索字段中输入 **identifier:systemd since:2020-10-22 JOB_TYPE=start,restart**。
2. 检查结果。



高级日志过滤 II

过滤上一次启动前出现的所有来自"cockpit.service' systemd 单元且邮件正文包含"error"或"fail"的所有日志消息。

1. 在搜索字段中输入 **service:cockpit boot:-1 error|fail**。
2. 检查结果。



3.4. 使用文本搜索框过滤 WEB 控制台中的日志

您可以在 web 控制台中使用文本搜索框，根据不同的参数过滤日志。搜索结合了过滤下拉菜单、限定符、日志字段和自由格式字符串搜索的使用。

先决条件

- 已安装 RHEL 10 web 控制台。
  具体步骤请参阅安装并启用 Web 控制台。

流程

1. 登录到 RHEL 10 web 控制台。
2. 点 **Logs**。
3. 使用下拉菜单指定您想要过滤的三个主要的限定符 - 时间范围、优先级和标识符。
   优先级（Priority） 限定符总需要有一个值。如果没有指定，它会自动过滤 **Error 及以上** 优先级。请注意，您设置的选项反映了在文本搜索框中。
4. 指定您要过滤的日志字段。
   您可以添加几个日志字段。
5. 您可以使用自由格式的字符串搜索任何其他内容。搜索框也接受正则表达式。

3.5. 日志过滤选项

您可以使用 **journalctl** 选项来过滤 web 控制台中的日志。其中一些选项作为 web 控制台界面的下拉菜单的一部分提供。

表 3.1. 表

选项名称	使用	注
priority	按消息优先级过滤输出。取单个数字或文本日志级别。日志级别是常见的 syslog 日志级别。如果指定了单一日志级别，则会显示具有此日志级别的所有消息或低（更重要）日志级别。	包括在**优先级**下拉菜单中。

选项名称	使用	注
identifier	显示被 syslog 标识为 SYSLOG_IDENTIFIER 的信息。可多次指定。	包括在 Identifier 下拉菜单中。
follow	仅显示最新的日志条目，并在新条目附加到日志中时持续打印新条目。	没有包含在下拉菜单中。
service	显示指定 **systemd** 单元的消息。可多次指定。	没有包含在下拉菜单中。对应于 **journalctl --unit** 参数。
boot	显示来自特定启动的消息。 正整数会在日志的开头查找启动，等号为零的整数会从日志末尾查找启动。因此, 1 表示日志中的第一个引导（按时间顺序排列），2 为第 2 个，以此类推 ; -0 是最后一次引导，-1 是最后一次引导的前一个，以此类推。	在时间下拉菜单中作为 Current boot 或 Previous boot。其他选项需要手动编写。
since	开始显示指定日期更新或分别位于指定日期或比指定日期旧的条目。日期规格应为 "2012-10-30 18:17:16"。如果省略了时间部分，使用 "00:00:00"。如果只省略了秒的组件，使用 ":00"。如果省略了日期的部分，使用当前日期。另外，还可以使用 "yesterday"、"today"、"tomorrow"（分别代表前一天、当天和明天的 00:00:00），以及 "now"（代表当前时间）。最后，可以指定相对时间，前缀为 "-" 或 "+"，分别引用当前时间前或之后的时间。	没有包含在下拉菜单中。

第 4 章 使用 RHEL 系统角色配置 SYSTEMD 日志

使用 **journald** RHEL 系统角色，您可以自动化 **systemd** 日志，并使用 Red Hat Ansible Automation Platform 配置持久性日志记录。

4.1. 使用 JOURNALD RHEL 系统角色配置持久性日志记录

默认情况下，**systemd** 日志只将日志存储在 **/run/log/journal** 中的小环缓冲区中，它不是持久的。重启系统也会移除日志数据库日志。您可以使用 **journald** RHEL 系统角色在多个系统中配置持久性日志记录。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Configure journald
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure persistent logging
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.journald
      vars:
        journald_persistent: true
        journald_max_disk_size: <size>
        journald_per_user: true
        journald_sync_interval: <interval>
```

示例 playbook 中指定的设置包括如下：

journald_persistent: true

启用持久性日志记录。

journald_max_disk_size: *<size>*

指定日志文件的最大磁盘空间大小（以 MB 为单位），例如 **2048**。

journald_per_user: true

配置 **journald** 来为每个用户单独保留日志数据。

journald_sync_interval: *<interval>*

设置同步间隔（以分钟为单位），例如 **1**。

有关 playbook 中使用的所有变量的详情，请查看控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.journald/README.md** 文件。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

第 5 章 配置远程日志记录解决方案

要确保在日志记录服务器中集中记录来自环境中各种机器的日志，您可以将 **Rsyslog** 应用程序配置为将适合客户端系统中特定条件的日志记录到服务器。

5.1. RSYSLOG 日志记录服务

Rsyslog 应用程序与 **systemd-journald** 服务相结合，在 Red Hat Enterprise Linux 中提供了本地和远程记录支持。**rsyslogd** 守护进程会持续读取 **systemd-journald** 服务从 Journal 接收的 **syslog** 消息。然后 **rsyslogd** 会过滤并处理这些 **syslog** 事件，并将它们记录到 **rsyslog** 日志文件，或者根据自己的配置将它们转发到其他服务。

rsyslogd 守护进程还提供扩展的过滤、加密受保护的转发消息、输入和输出模块，并支持使用 TCP 和 UDP 协议进行传输。

在 **/etc/rsyslog.conf** 中，是 **rsyslog** 的主要配置文件，您可以根据 **rsyslogd** 处理消息来指定规则。通常，您可以通过其来源和主题（设施）和紧急情况（优先级）对消息进行分类，然后分配在消息适合这些条件时应执行的操作。

在 **/etc/rsyslog.conf** 中，您还可以看到 **rsyslogd** 维护的日志文件列表。大多数日志文件位于 **/var/log/** 目录中。**httpd** 和 **samba** 等一些应用将其日志文件存储在 **/var/log/** 中的子目录中。

其他资源

- 您系统上的 **rsyslogd (8)** 和 **rsyslog.conf (5)** 手册页
- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档

5.2. 安装 RSYSLOG 文档

Rsyslog 应用程序在 https://www.rsyslog.com/doc/ 上提供了广泛的在线文档，但您也可以在本地安装 **rsyslog-doc** 文档软件包。

先决条件

- 您已在系统中激活了 **AppStream** 软件仓库。
- 您有权使用 **sudo** 安装新软件包。

流程

- 安装 **rsyslog-doc** 软件包：

```
# dnf install rsyslog-doc
```

验证

- 在您选择的浏览器中打开 **/usr/share/doc/rsyslog/html/index.html** 文件，例如：

```
$ firefox /usr/share/doc/rsyslog/html/index.html &
```

5.3. 通过 TCP 配置服务器进行远程记录

Rsyslog 应用程序可让您运行日志服务器并配置各个系统将其日志文件发送到日志记录服务器。要通过 TCP 使用远程日志，请同时配置服务器和客户端。服务器收集和分析由一个或多个客户端系统发送的日志。

使用 Rsyslog 应用程序，您可以维护一个集中的日志系统，该系统可通过网络将日志消息转发到服务器。为了避免服务器不可用时消息丢失，您可以为转发操作配置操作队列。这样，无法发送的消息将存储在本地，直到服务器再次可访问为止。请注意，此类队列无法针对使用 UDP 协议的连接配置。

omfwd 插件通过 UDP 或 TCP 提供转发。默认协议是 UDP。由于插件内置在内，因此不必加载它。

默认情况下，**rsyslog** 使用端口 **514** 上的 TCP。

先决条件

- rsyslog 已安装在服务器系统上。
- 您以 **root** 身份登录到服务器中。
- 使用 **semanage** 命令，为可选步骤安装 **policycoreutils-python-utils** 软件包。
- **firewalld** 服务在运行。

流程

1. 可选： 要为 **rsyslog** 流量使用不同的端口，请将 **syslogd_port_t** SELinux 类型添加到端口。例如，启用端口 **30514** ：

```
# semanage port -a -t syslogd_port_t -p tcp 30514
```

2. 可选： 要为 **rsyslog** 流量使用不同的端口，请将 **firewalld** 配置为允许该端口上传入的 **rsyslog** 流量。例如，允许端口 **30514** 上的 TCP 流量：

```
# firewall-cmd --zone=<zone_name> --permanent --add-port=30514/tcp
success
# firewall-cmd --reload
```

3. 在 **/etc/rsyslog.d/** 目录中创建一个新文件（例如，**remotelog.conf**），并插入以下内容：

```
# Define templates before the rules that use them
# Per-Host templates for remote systems
template(name="TmplAuthpriv" type="list") {
    constant(value="/var/log/remote/auth/")
    property(name="hostname")
    constant(value="/")
    property(name="programname" SecurePath="replace")
    constant(value=".log")
    }

template(name="TmplMsg" type="list") {
    constant(value="/var/log/remote/msg/")
    property(name="hostname")
    constant(value="/")
```

```
    property(name="programname" SecurePath="replace")
    constant(value=".log")
    }

# Provides TCP syslog reception
module(load="imtcp")

# Adding this ruleset to process remote messages
ruleset(name="remote1"){
    authpriv.*   action(type="omfile" DynaFile="TmplAuthpriv")
     *.info;mail.none;authpriv.none;cron.none
action(type="omfile" DynaFile="TmplMsg")
}

input(type="imtcp" port="30514" ruleset="remote1")
```

4. 将更改保存到 **/etc/rsyslog.d/remotelog.conf** 文件。

5. 测试 **/etc/rsyslog.conf** 文件的语法：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-2.el8, config validation run...
rsyslogd: End of config validation run. Bye.
```

6. 确保 **rsyslog** 服务在日志记录服务器中运行并启用：

```
# systemctl status rsyslog
```

7. 重新启动 **rsyslog** 服务。

```
# systemctl restart rsyslog
```

8. 可选： 如果没有启用 **rsyslog**，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

您的日志服务器现在已配置为从您环境中的其他系统接收和存储日志文件。

其他资源

- **rsyslogd (8)**, **rsyslog.conf (5)**, **semanage (8)**, 和 **firewall-cmd (1)** man page
- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档

5.4. 通过 TCP 配置远程日志记录到服务器

您可以配置系统，以通过 TCP 协议将日志消息转发到服务器。**omfwd** 插件通过 UDP 或 TCP 提供转发。默认协议是 UDP。因为插件内置在内，所以不必加载它。

先决条件

- **rsyslog** 软件包安装在应该向服务器报告的客户端系统上。

- 您已为远程日志记录配置了服务器。
- 在 SELinux 中允许指定的端口并在防火墙中打开。
- 系统包含 **policycoreutils-python-utils** 软件包，它为 SELinux 配置中添加非标准端口提供 **semanage** 命令。

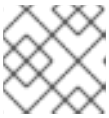
流程

1. 在 **/etc/rsyslog.d/** 目录中创建一个名为 的新文件，如 **10-remotelog.conf**，并插入以下内容：

```
*.* action(type="omfwd"
     queue.type="linkedlist"
     queue.filename="example_fwd"
     action.resumeRetryCount="-1"
     queue.saveOnShutdown="on"
     target="example.com" port="30514" protocol="tcp"
    )
```

其中：

- **queue.type="linkedlist"** 设置启用 LinkedList 内存中队列，
- **queue.filename** 设置定义磁盘存储。备份文件是使用 **example_fwd** 前缀，在之前全局 **workDirectory** 指令指定的工作目录中创建的。
- **action.resumeRetryCount -1** 设置防止 **rsyslog** 在服务器没有响应而重试连接时丢弃消息，
- 如果 **rsyslog** 关闭，**queue.saveOnShutdown="on"** 设置会保存内存中的数据。
- 最后一行将所有收到的消息转发到日志记录服务器。端口规格是可选的。
  使用这个配置，**rsyslog** 会向服务器发送消息，但如果远程服务器无法访问，则会将消息保留在内存中。只有 **rsyslog** 耗尽配置的内存队列空间或需要关闭时，才能创建磁盘上的文件，从而让系统性能受益。



注意

rsyslog 按照一般顺序处理配置文件 **/etc/rsyslog.d/**。

2. 重新启动 **rsyslog** 服务。

```
# systemctl restart rsyslog
```

验证

验证客户端系统向服务器发送信息：

1. 在客户端系统中发送测试信息：

```
# logger test
```

2. 在服务器系统上，查看 **/var/log/messages** 日志，例如：

```
# cat /var/log/remote/msg/hostname/root.log
Feb 25 03:53:17 hostname root[6064]: test
```

其中 *hostname* 是客户端系统的主机名。请注意，日志包含输入 **logger** 命令的用户的用户名，本例中为 **root**。

其他资源

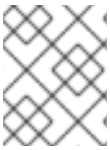
- 您系统上的 **rsyslogd (8)** 和 **rsyslog.conf (5)** 手册页
- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档

5.5. 配置 TLS 加密的远程日志记录

默认情况下，Rsyslog 以纯文本格式发送 remote-logging 通信。如果您的场景需要保护这个通信频道，您可以使用 TLS 加密它。

要通过 TLS 使用加密传输，请同时配置服务器和客户端。服务器收集和分析由一个或多个客户端系统发送的日志。

您可以使用 **ossl** 网络流驱动程序(OpenSSL)或 **gtls** 流驱动程序(GnuTLS)。



注意

如果您的系统具有更高的安全性，例如，没有连接到任何网络或有严格授权的系统，请使用独立的系统作为认证授权(CA)。

您可以在服务器端上，在 **global**、**module** 和 **input** 级别上使用流驱动程序自定义连接设置，在客户端上，在 **global** 和 **action** 级别上使用流驱动程序自定义连接设置。更为具体的配置会覆盖更常规的配置。例如，您可以在全局设置中对大多数连接使用 **ossl**，而在 input 和 action 设置只对特定连接使用 **gtls**。

先决条件

- 有对客户端和服务器系统的 **root** 访问权限。
- 以下软件包安装在服务器和客户端系统上：
  - **rsyslog** 软件包。
  - 对于 **ossl** 网络流驱动程序，是 **rsyslog-openssl** 软件包。
  - 对于 **gtls** 网络流驱动程序，是 **rsyslog-gnutls** 软件包。
  - 对于使用 **certtool** 命令生成证书，是 **gnutls-utils** 软件包。
- 在您的日志服务器中，以下证书位于 **/etc/pki/ca-trust/source/anchors/** 目录中，并使用 **update-ca-trust** 命令更新您的系统配置：
  - **ca-cert.pem** - 一个 CA 证书，它可以在日志记录服务器和客户端上验证密钥和证书。
  - **server-cert.pem** - 日志记录服务器的公钥。
  - **server-key.pem** - 日志记录服务器的私钥。

- 在您的日志记录客户端中，以下证书位于 **/etc/pki/ca-trust/source/anchors/** 目录中，并使用 **update-ca-trust** 来更新您的系统配置：
  - **ca-cert.pem** - 一个 CA 证书，它可以在日志记录服务器和客户端上验证密钥和证书。
  - **client-cert.pem** - 客户端的公钥。
  - **client-key.pem** - 客户端的私钥。
  - 如果服务器运行 RHEL 9.2 或更高版本，且启用了 FIPS 模式，客户端必须支持 Extended Master Secret(EMS)扩展或使用 TLS 1.3。没有 EMS 的 TLS 1.2 连接会失败。如需更多信息，请参阅 TLS 扩展"扩展主 Secret"强制 文章（红帽知识库）。

流程

1. 配置服务器以从您的客户端系统接收加密日志：
   a. 在 **/etc/rsyslog.d/** 目录中创建一个新文件，例如 **securelogser.conf**。
   b. 要加密通信，配置文件必须包含指向服务器的证书文件的路径、所选身份验证方法，以及支持 TLS 加密的流驱动程序。在 **/etc/rsyslog.d/securelogser.conf** 文件中添加以下行：

```
# Set certificate files
global(
  DefaultNetstreamDriverCAFile="/etc/pki/ca-trust/source/anchors/ca-cert.pem"
  DefaultNetstreamDriverCertFile="/etc/pki/ca-trust/source/anchors/server-cert.pem"
  DefaultNetstreamDriverKeyFile="/etc/pki/ca-trust/source/anchors/server-key.pem"
)

# TCP listener
module(
  load="imtcp"
  PermittedPeer=["client1.example.com", "client2.example.com"]
  StreamDriver.AuthMode="x509/name"
  StreamDriver.Mode="1"
  StreamDriver.Name="ossl"
)

# Start up listener at port 514
input(
  type="imtcp"
  port="514"
)
```

注意

如果您更喜欢 GnuTLS 驱动程序，请使用 **StreamDriver.Name="gtls"** 配置选项。有关比 **x509/name** 严格性低的验证模式的更多信息，请参阅使用 **rsyslog-doc** 软件包安装的文档。

   c. 可选：要自定义连接配置，请使用以下内容替换 **input** 部分：

```
input(
  type="imtcp"
  Port="50515"
```

```
  StreamDriver.Name="<driver>"
  streamdriver.CAFile="/etc/rsyslog.d/<ca1>.pem"
  streamdriver.CertFile="/etc/rsyslog.d/<server1_cert>.pem"
  streamdriver.KeyFile="/etc/rsyslog.d/<server1_key>.pem"
)
```

- 根据您要使用的驱动程序，将 ***<driver>*** 替换为 **ossl** 或 **gtls**。
- 将 ***<ca1&*** gt; 替换为 CA 证书， ***<server1_cert*** > 替换为证书，将 < ***server1_key*** > 替换为自定义连接的密钥。

d. 将更改保存到 **/etc/rsyslog.d/securelogser.conf** 文件。

e. 验证 **/etc/rsyslog.conf** 文件的语法以及 **/etc/rsyslog.d/** 目录中的任何文件：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-2.el8, config validation run (level 1)...
rsyslogd: End of config validation run. Bye.
```

f. 确保 **rsyslog** 服务在日志记录服务器中运行并启用：

```
# systemctl status rsyslog
```

g. 重启 **rsyslog** 服务：

```
# systemctl restart rsyslog
```

h. 可选： 如果没有启用 Rsyslog，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

2. 配置客户端以将加密日志发送到服务器：

a. 在客户端系统上，在 **/etc/rsyslog.d/** 目录中创建一个名为 的新文件，如 **securelogcli.conf**。

b. 在 **/etc/rsyslog.d/securelogcli.conf** 文件中添加以下行：

```
# Set certificate files
global(
  DefaultNetstreamDriverCAFile="/etc/pki/ca-trust/source/anchors/ca-cert.pem"
  DefaultNetstreamDriverCertFile="/etc/pki/ca-trust/source/anchors/client-cert.pem"
  DefaultNetstreamDriverKeyFile="/etc/pki/ca-trust/source/anchors/client-key.pem"
)


# Set up the action for all messages
*.* action(
  type="omfwd"
  StreamDriver="ossl"
  StreamDriverMode="1"
  StreamDriverPermittedPeers="server.example.com"
```

```
  StreamDriverAuthMode="x509/name"
  target="server.example.com" port="514" protocol="tcp"
)
```

注意

如果您更喜欢 GnuTLS 驱动程序，请使用 **StreamDriver.Name="gtls"** 配置选项。

c. 可选： 要自定义连接配置，将 **action** 部分替换为以下内容：

```
local1.* action(
  type="omfwd"
  StreamDriver="<driver>"
  StreamDriverMode="1"
  StreamDriverAuthMode="x509/certvalid"
  streamDriver.CAFile="/etc/rsyslog.d/<ca1>.pem"
  streamDriver.CertFile="/etc/rsyslog.d/<client1_cert>.pem"
  streamDriver.KeyFile="/etc/rsyslog.d/<client1_key>.pem"
  target="server.example.com" port="514" protocol="tcp"
  )
```

- 根据您要使用的驱动程序，将 ***<driver>*** 替换为 **ossl** 或 **gtls**。
- 将 ***<ca1&*** gt; 替换为 CA 证书， ***<client1_cert*** > 替换为证书，**<client1_key** > 替换为自定义连接的密钥。

d. 将更改保存到 **/etc/rsyslog.d/securelogcli.conf** 文件中。

e. 验证 **/etc/rsyslog.conf** 文件的语法以及 **/etc/rsyslog.d/** 目录中的其他文件：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-2.el8, config validation run (level 1)...
rsyslogd: End of config validation run. Bye.
```

f. 确保 **rsyslog** 服务在日志记录服务器中运行并启用：

```
# systemctl status rsyslog
```

g. 重启 **rsyslog** 服务：

```
# systemctl restart rsyslog
```

h. 可选： 如果没有启用 Rsyslog，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

验证

要验证客户端系统向服务器发送信息，请按照以下步骤执行：

1. 在客户端系统中发送测试信息：

```
# logger test
```

2. 在服务器系统上，查看 **/var/log/messages** 日志，例如：

```
# cat /var/log/remote/msg/<hostname>/root.log
Feb 25 03:53:17 <hostname> root[6064]: test
```

其中 ***<hostname>*** 是客户端系统的主机名。请注意，该日志包含输入 logger 命令的用户的用户名，本例中为 **root**。

其他资源

- **certtool (1)**, **openssl (1)**, **update-ca-trust (8)**, **rsyslogd (8)**, 和 **rsyslog.conf (5)** man page
- 安装了位于 **/usr/share/doc/rsyslog/html/index.html**的 **rsyslog-doc** 软件包的文档
- 使用带有 TLS 的日志记录系统角色

5.6. 配置服务器以通过 UDP 接收远程日志信息

Rsyslog 应用程序可让您将系统配置为从远程系统接收日志信息。要通过 UDP 使用远程日志记录，请同时配置服务器和客户端。接收服务器收集并分析一个或多个客户端系统发送的日志。默认情况下，**rsyslog** 使用端口 **514** 上的 UDP 从远程系统接收日志信息。

按照以下步骤配置服务器，以通过 UDP 协议收集和分析一个或多个客户端系统发送的日志。

先决条件

- rsyslog 已安装在服务器系统上。
- 您以 **root** 身份登录到服务器中。
- 为使用 **semanage** 命令的可选步骤安装了 **policycoreutils-python-utils** 软件包。
- **firewalld** 服务在运行。

流程

1. 可选：要将不同的端口用于 **rsyslog** 流量，而不是默认端口 **514** ：
   a. 将 **syslogd_port_t** SELinux 类型添加到 SELinux 策略配置中，使用您要 **rsyslog** 的端口号替换 ***portno*** ：

```
# semanage port -a -t syslogd_port_t -p udp portno
```

   b. 配置 **firewalld** 以允许传入的 **rsyslog** 流量，使用您要 **rsyslog** 使用的端口替换 ***portno***，区替换 **zone**：

```
# firewall-cmd --zone=zone --permanent --add-port=portno/udp
success
# firewall-cmd --reload
```

   c. 重新载入防火墙规则：

```
# firewall-cmd --reload
```

2. 在 **/etc/rsyslog.d/** 目录中创建一个新的 **.conf** 文件，如 **remotelogserv.conf**，并插入以下内容：

```
# Define templates before the rules that use them
# Per-Host templates for remote systems
template(name="TmplAuthpriv" type="list") {
    constant(value="/var/log/remote/auth/")
    property(name="hostname")
    constant(value="/")
    property(name="programname" SecurePath="replace")
    constant(value=".log")
    }

template(name="TmplMsg" type="list") {
    constant(value="/var/log/remote/msg/")
    property(name="hostname")
    constant(value="/")
    property(name="programname" SecurePath="replace")
    constant(value=".log")
    }

# Provides UDP syslog reception
module(load="imudp")

# This ruleset processes remote messages
ruleset(name="remote1"){
     authpriv.*   action(type="omfile" DynaFile="TmplAuthpriv")
      *.info;mail.none;authpriv.none;cron.none
action(type="omfile" DynaFile="TmplMsg")
}

input(type="imudp" port="514" ruleset="remote1")
```

其中 **514** 是 **rsyslog** 默认使用的端口号。您可以指定不同的端口。

3. 验证 **/etc/rsyslog.conf** 文件以及 **/etc/rsyslog.d/** 目录中的所有 **.conf** 文件的语法：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-2.el8, config validation run...
```

4. 重新启动 **rsyslog** 服务。

```
# systemctl restart rsyslog
```

5. 可选： 如果没有启用 **rsyslog**，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

其他资源

- **rsyslogd (8)** , **rsyslog.conf (5)**, **semanage (8)**, 和 **firewall-cmd (1)** man page

- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档

5.7. 通过 UDP 配置远程日志记录到服务器

您可以配置系统，以通过 UDP 协议将日志消息转发到服务器。**omfwd** 插件通过 UDP 或 TCP 提供转发。默认协议是 UDP。因为插件内置在内，所以不必加载它。

先决条件

- **rsyslog** 软件包安装在应该向服务器报告的客户端系统上。
- 您已为远程日志记录配置了服务器，如配置服务器 以通过 UDP 接收远程日志信息。

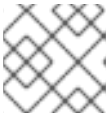
流程

1. 在 **/etc/rsyslog.d/** 目录中创建一个新的 **.conf** 文件，如 **10-remotelogcli.conf**，并插入以下内容：

```
*.* action(type="omfwd"
     queue.type="linkedlist"
     queue.filename="example_fwd"
     action.resumeRetryCount="-1"
     queue.saveOnShutdown="on"
     target="example.com" port="portno" protocol="udp"
    )
```

其中：

- **queue.type="linkedlist"** 设置启用一个 LinkedList 内存中队列。
- **queue.filename** 设置定义磁盘存储。备份文件使用之前全局 **workDirectory** 指令指定的工作目录中的 ***example_fwd*** 前缀创建。
- **action.resumeRetryCount -1** 设置可防止 **rsyslog** 在重试时丢弃消息（如果服务器没有响应）。
- 如果 **rsyslog** 关闭，启用的 **queue.saveOnShutdown="on"** 设置会保存内存中的数据。
- ***portno*** 值是您要 **rsyslog** 使用的端口号。默认值为 **514**。
- 最后一行将所有收到的消息转发到日志记录服务器，端口规格是可选的。
  使用这个配置，**rsyslog** 会向服务器发送消息，但如果远程服务器无法访问，则会将消息保留在内存中。只有 **rsyslog** 耗尽配置的内存队列空间或需要关闭时，才能创建磁盘上的文件，从而让系统性能受益。



注意

rsyslog 按照一般顺序处理配置文件 **/etc/rsyslog.d/**。

2. 重新启动 **rsyslog** 服务。

```
# systemctl restart rsyslog
```

3. 可选： 如果没有启用 **rsyslog**，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

验证

要验证客户端系统向服务器发送信息，请按照以下步骤执行：

1. 在客户端系统中发送测试信息：

```
# logger test
```

2. 在服务器系统中，查看 **/var/log/remote/msg/*hostname*/root.log** 日志，例如：

```
# cat /var/log/remote/msg/hostname/root.log
Feb 25 03:53:17 hostname root[6064]: test
```

其中 **hostname** 是客户端系统的主机名。请注意，该日志包含输入 logger 命令的用户的用户名，本例中为 **root**。

其他资源

- **rsyslogd(8)** 和 **rsyslog.conf(5)** 手册页。
- 在 **/usr/share/doc/rsyslog/html/index.html** 上安装了 **rsyslog-doc** 软件包的文档。

5.8. RSYSLOG 中的负载均衡帮助程序

当在集群中使用时，您可以通过修改 **RebindInterval** 设置来提高 Rsyslog 负载均衡。

RebindInterval 指定当前连接中断的时间间隔，并被重新建立。此设置适用于 TCP、UDP 和 RELP 流量。负载平衡器将信息作为新连接，并将消息转发到另一个物理目标系统。

当目标系统更改了其 IP 地址时，**RebindInterval** 会很有用。Rsyslog 应用程序在建立连接时缓存 IP 地址，因此信息会发送到同一服务器。如果 IP 地址更改，UDP 数据包会丢失，直到 Rsyslog 服务重启为止。重新建立连接可确保 DNS 再次解析 IP。

TCP、UDP 和 RELP 流量使用 RebindInterval 示例

```
action(type="omfwd" protocol="tcp" RebindInterval="250" target="example.com" port="514" …)

action(type="omfwd" protocol="udp" RebindInterval="250" target="example.com" port="514" …)

action(type="omrelp" RebindInterval="250" target="example.com" port="6514" …)
```

5.9. 配置可靠的远程日志记录

通过可靠的事件日志记录协议(RELP)，您可以降低消息丢失的风险通过 TCP 发送和接收 **syslog** 消息。RELP 提供可靠的事件消息交付，这对于无法接受消息丢失的环境中非常有用。要使用 RELP，请配置服务器上运行的 **imrelp** 输入模块并接收日志，以及在客户端上运行的 **omrelp** 输出模块，并将日志发送到日志记录服务器。

先决条件

- 您已在服务器和客户端系统中安装了 **rsyslog**、**librelp** 和 **rsyslog-relp** 软件包。
- 在 SELinux 中允许指定的端口并在防火墙中打开。

流程

1. 配置客户端系统以可靠远程记录：
   a. 在客户端系统上，在 **/etc/rsyslog.d/** 目录中创建一个新的 **.conf** 文件，例如 **relpclient.conf**，并插入以下内容：

   ```
   module(load="omrelp")
   *.* action(type="omrelp" target="_target_IP_" port="_target_port_")
   ```

   其中：

   - ***target_IP*** 是日志记录服务器的 IP 地址。
   - ***target_port*** 是日志记录服务器的端口。

   b. 保存对 **/etc/rsyslog.d/relpclient.conf** 文件的更改。
   c. 重新启动 **rsyslog** 服务。

   ```
   # systemctl restart rsyslog
   ```

   d. 可选：如果没有启用 **rsyslog**，请确保 **rsyslog** 服务在重启后自动启动：

   ```
   # systemctl enable rsyslog
   ```

2. 配置服务器系统以可靠远程记录：
   a. 在服务器系统中，在 **/etc/rsyslog.d/** 目录中创建一个新的 **.conf** 文件，例如 **reserv.conf**，并插入以下内容：

   ```
   ruleset(name="relp"){
   *.* action(type="omfile" file="_log_path_")
   }


   module(load="imrelp")
   input(type="imrelp" port="_target_port_" ruleset="relp")
   ```

   其中：

   - ***log_path*** 指定存储消息的路径。
   - ***target_port*** 是日志记录服务器的端口。使用与客户端配置文件中相同的值。

   b. 保存对 **/etc/rsyslog.d/relpserv.conf** 文件的更改。
   c. 重新启动 **rsyslog** 服务。

```
# systemctl restart rsyslog
```

d. 可选：如果没有启用 **rsyslog**，请确保 **rsyslog** 服务在重启后自动启动：

```
# systemctl enable rsyslog
```

验证

验证客户端系统向服务器发送信息：

1. 在客户端系统中发送测试信息：

```
# logger test
```

2. 在服务器系统中，查看指定 ***log_path*** 的日志，例如：

```
# cat /var/log/remote/msg/hostname/root.log
Feb 25 03:53:17 hostname root[6064]: test
```

其中 **hostname** 是客户端系统的主机名。请注意，该日志包含输入 logger 命令的用户的用户名，本例中为 **root**。

其他资源

- 您系统上的 **rsyslogd (8)** 和 **rsyslog.conf (5)** 手册页
- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档

5.10. 支持的 RSYSLOG 模块

要扩展 Rsyslog 应用程序的功能，您可以使用特定的模块。模块提供额外的输入(Input 模块)、输出（输出模块）和其他功能。模块也可以提供在加载模块后可用的其他配置指令。

您可以输入以下命令列出您系统中安装的输入和输出模块：

```
# ls /usr/lib64/rsyslog/{i,o}m*
```

在安装了 **rsyslog-doc** 软件包后，您可以在 **/usr/share/doc/rsyslog/html/configuration/modules/idx_output.html** 文件中查看所有可用的 **rsyslog** 模块。

5.11. 配置 NETCONSOLE 服务为将内核信息记录到远程主机

当无法登录到磁盘或使用串行控制台时，您可以使用 **netconsole** 内核模块和同名的服务将内核消息通过网络记录到远程 **rsyslog** 服务中。

先决条件

- 远程主机上已安装系统日志服务，如 **rsyslog**。
- 远程系统日志服务被配置为接收来自此主机的日志条目。

流程

流程

1. 安装 **netconsole-service** 软件包：

```
# dnf install netconsole-service
```

2. 编辑 **/etc/sysconfig/netconsole** 文件，并将 **SYSLOGADDR** 参数设为远程主机的 IP 地址：

```
# SYSLOGADDR=192.0.2.1
```

3. 启用并启动 **netconsole** 服务：

```
# systemctl enable --now netconsole
```

验证

- 在远程系统日志服务器上显示 **/var/log/messages** 文件。

5.12. 其他资源

- 在 **/usr/share/doc/rsyslog/html/index.html** 文件中安装有 **rsyslog-doc** 软件包的文档
- 您系统上的 **rsyslog.conf (5)** 和 **rsyslogd (8)** 手册页
- 在不使用 journald 或最小化 journald 的情况下配置系统日志记录 （红帽知识库）
- RHEL 默认日志设置对性能的负面影响及其缓解措施 （红帽知识库）

第 6 章 使用 RHEL 系统角色配置日志记录

您可以使用 **logging** RHEL 系统角色将本地和远程主机配置为自动记录服务器，以便从多个客户端系统收集日志。

日志记录解决方案提供多种读取日志和多个日志记录输出的方法。

例如，日志记录系统可接受以下输入：

- 本地文件
- **systemd/journal**
- 网络上的另一个日志记录系统

另外，日志记录系统还可有以下输出：

- 日志存储在 **/var/log/** 目录中的本地文件中
- 日志发送到 Elasticsearch 引擎
- 日志被转发到另一个日志系统

使用 **logging** RHEL 系统角色，您可以组合输入和输出以适合您的场景。例如，您可以配置一个日志解决方案，将来自 **日志** 的输入存储在本地文件中，而从文件读取的输入则被转发到另一个日志系统，并存储在本地日志文件中。

6.1. 使用 LOGGING RHEL 系统角色过滤本地日志消息

您可以使用 **logging** RHEL 系统角色的基于属性的过滤器，根据各种情况过滤本地日志消息。因此，您可以实现，例如：

- 日志清晰度：在高流量环境中，日志可能会快速增长。专注于特定消息（如错误）有助于更快地识别问题。
- 优化的系统性能：大量日志通常与系统性能下降有关。仅针对重要事件选择记录日志可以防止资源耗尽，从而使您的系统更有效地运行。
- 增强了安全性：通过安全消息，如系统错误和失败的登录进行高效过滤，有助于仅捕获相关日志。这对于检测漏洞和满足合规性标准非常重要。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 **~/playbook.yml** ：

```
---
- name: Deploy the logging solution
```

```
hosts: managed-node-01.example.com
tasks:
  - name: Filter logs based on a specific value they contain
    ansible.builtin.include_role:
      name: redhat.rhel_system_roles.logging
    vars:
      logging_inputs:
        - name: files_input
          type: basics
      logging_outputs:
        - name: files_output0
          type: files
          property: msg
          property_op: contains
          property_value: error
          path: /var/log/errors.log
        - name: files_output1
          type: files
          property: msg
          property_op: "!contains"
          property_value: error
          path: /var/log/others.log
      logging_flows:
        - name: flow0
          inputs: [files_input]
          outputs: [files_output0, files_output1]
```

示例 playbook 中指定的设置包括如下：

logging_inputs

定义一个记录输入字典的列表。**type: basics** 选项涵盖了来自 **systemd** 日志或 Unix 套接字的输入。

logging_outputs

定义一个记录输出字典的列表。**type: files** 选项支持将日志存储在本地文件中，通常存储在 **/var/log/** 目录中。**property: msg**; **property: contains** 和 **property_value: error** 选项指定所有包含 **error** 字符串的日志都存储在 **/var/log/errors.log** 文件中。**property: msg**; **property: !contains**; 和 **property_value: error** 选项指定所有其他日志都放在 **/var/log/others.log** 文件中。您可以将 **error** 值替换为您要过滤的字符串。

logging_flows

定义一个记录流字典的列表，以指定 **logging_inputs** 和 **logging_outputs** 之间的关系。**inputs: [files_input]** 选项指定一个从其开始处理日志的输入的列表。**outputs: [files_output0, files_output1]** 选项指定一个日志发送到的输出的列表。

有关 playbook 中使用的所有变量的详情，以及 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件和 **rsyslog.conf (5)** 及 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

验证

1. 在受管节点上，测试 **/etc/rsyslog.conf** 文件的语法：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-6.el8, config validation run...
rsyslogd: End of config validation run. Bye.
```

2. 在受管节点上，验证系统是否向日志发送包含 **error** 字符串的消息：

   a. 发送测试信息：

```
# logger error
```

   b. 查看 **/var/log/errors.log** 日志，例如：

```
# cat /var/log/errors.log
Aug  5 13:48:31 hostname root[6778]: error
```

   其中 ***hostname*** 是客户端系统的主机名。请注意，该日志包含输入 logger 命令的用户的用户名，本例中为 **root**。

6.2. 使用 LOGGING RHEL 系统角色应用远程日志解决方案

您可以使用 **logging** RHEL 系统角色配置远程日志记录解决方案，其中一个或多个客户端可以从 **systemd-journal** 服务获取日志，并将其转发到远程服务器。服务器从 **remote_rsyslog** 和 **remote_files** 配置接收远程输入，并将日志输出到以远程主机名命名的目录中的本地文件中。

因此，您可以涵盖您需要的用例，例如：

- 集中式日志管理：从单一存储点收集、访问和管理多台机器的日志消息简化了日常监控和故障排除任务。此外，此用例还减少了登录到单个机器来检查日志消息的需要。
- 增强了安全性：在一个中央位置存储日志消息可增加它们处于安全且防止篡改的环境中的几率。这样的环境可以更轻松地检测和更有效地响应安全事件，并满足审计要求。
- 提高了日志分析的效率：协调来自多个系统的日志消息对于快速故障排除跨多个机器或服务的复杂问题非常重要。这样，您可以快速分析并交叉引用来自不同源的事件。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。

- 在服务器或客户端系统的 SELinux 策略中定义端口，并为这些端口打开防火墙。默认 SELinux 策略包括端口 601、514、6514、10514 和 20514。要使用其他端口，请参阅 修改客户端和服务器系统上的 SELinux 策略。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Deploy the logging solution
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure the server to receive remote input
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_inputs:
          - name: remote_udp_input
            type: remote
            udp_ports: [ 601 ]
          - name: remote_tcp_input
            type: remote
            tcp_ports: [ 601 ]
        logging_outputs:
          - name: remote_files_output
            type: remote_files
        logging_flows:
          - name: flow_0
            inputs: [remote_udp_input, remote_tcp_input]
            outputs: [remote_files_output]

- name: Deploy the logging solution
  hosts: managed-node-02.example.com
  tasks:
    - name: Configure the server to output the logs to local files in directories named by
remote host names
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_inputs:
          - name: basic_input
            type: basics
        logging_outputs:
          - name: forward_output0
            type: forwards
            severity: info
            target: <host1.example.com>
            udp_port: 601
          - name: forward_output1
            type: forwards
            facility: mail
            target: <host1.example.com>
            tcp_port: 601
        logging_flows:
```

```
    - name: flows0
      inputs: [basic_input]
      outputs: [forward_output0, forward_output1]
```

示例 playbook 的第一个 play 中指定的设置包括以下：

logging_inputs

定义一个记录输入字典的列表。**type: remote** 选项涵盖通过网络来自其他日志记录系统的远程输入。**udp_ports: [601]** 选项定义要监控的 UDP 端口号的列表。**tcp_ports: [601]** 选项定义要监控的 TCP 端口号的列表。如果 **udp_ports** 和 **tcp_ports** 都设置了，则 **udp_ports** 被使用，**tcp_ports** 被丢弃。

logging_outputs

定义一个记录输出字典的列表。**type: remote_files** 选项使输出将日志存储到每个远程主机的本地文件和程序名称产生的日志中。

logging_flows

定义一个记录流字典的列表，以指定 **logging_inputs** 和 **logging_outputs** 之间的关系。**inputs: [remote_udp_input, remote_tcp_input]** 选项指定输入的列表，日志的处理从该列表开始。**outputs: [remote_files_output]** 选项指定输出的列表，日志被发送到此输出中。

示例 playbook 的第二个 play 中指定的设置包括以下：

logging_inputs

定义一个记录输入字典的列表。**type: basics** 选项涵盖了来自 **systemd** 日志或 Unix 套接字的输入。

logging_outputs

定义一个记录输出字典的列表。**type: forwards** 选项支持通过网络将日志发送到远程日志服务器。**severity: info** 选项是指具有信息重要性的日志消息。**facility: mail** 选项是指正在生成日志消息的系统程序的类型。**target: *<host1.example.com>*** 选项指定远程日志记录服务器的主机名。**udp_port: 601**/**tcp_port: 601** 选项定义远程日志记录服务器侦听的 UDP/TCP 端口。

logging_flows

定义一个记录流字典的列表，以指定 **logging_inputs** 和 **logging_outputs** 之间的关系。**inputs: [basic_input]** 选项指定输入的列表，日志的处理从该列表开始。**outputs: [forward_output0, forward_output1]** 选项指定输出的列表，日志被发送到此输出中。

有关角色变量的详情和有关 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件以及 **rsyslog.conf (5)** 和 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

验证

1. 在客户端和服务器系统上测试 **/etc/rsyslog.conf** 文件的语法：

```
# rsyslogd -N 1
rsyslogd: version 8.1911.0-6.el8, config validation run (level 1), master config
/etc/rsyslog.conf
rsyslogd: End of config validation run. Bye.
```

2. 验证客户端系统向服务器发送信息：

   a. 在客户端系统中发送测试信息：

```
# logger test
```

   b. 在服务器系统上，查看 **/var/log/*<host2.example.com>*/messages** 日志，例如：

```
# cat /var/log/<host2.example.com>/messages
Aug  5 13:48:31 <host2.example.com> root[6778]: test
```

   其中 ***<host2.example.com>*** 是客户端系统的主机名。请注意，该日志包含输入 logger 命令的用户的用户名，本例中为 **root**。

6.3. 使用带有 TLS 的 LOGGING RHEL 系统角色

传输层安全性(TLS)是一种加密协议，旨在允许通过计算机网络的安全通信。

您可以使用 **logging** RHEL 系统角色配置日志消息的安全传输，其中一个或多个客户端可以从 **systemd-journal** 服务获取日志，并使用 TLS 将它们传送到远程服务器。

当通过不太可信的或公共网络（如互联网）发送敏感数据时，通常使用远程日志记录解决方案中的 TLS 来传输日志。另外，通过在 TLS 中使用证书，您可以确保客户端将日志转发到正确的可信服务器。这可以防止诸如"中间人"的攻击。

6.3.1. 配置带有 TLS 的客户端日志

您可以使用 **logging** RHEL 系统角色在 RHEL 客户端上配置日志，并使用 TLS 加密将日志传送到远程日志系统。

此流程创建一个私钥和证书。接下来，它对 Ansible 清单中的 clients 组中的所有主机配置 TLS。TLS 对信息的传输进行加密，确保日志在网络安全传输。

注意

您不必在 playbook 中调用 证书 RHEL 系统角色来创建证书。当设置了 **logging_certificates** 变量时，logging RHEL 系统角色会自动调用它。

要让 CA 能够为创建的证书签名，受管节点必须在 IdM 域中注册。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。

- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。
- 受管节点已在 IdM 域中注册。
- 如果要在管理节点上配置的日志服务器运行 RHEL 9.2 或更高版本，且启用了 FIPS 模式，客户端必须支持扩展 Master Secret (EMS)扩展或使用 TLS 1.3。没有 EMS 的 TLS 1.2 连接会失败。如需更多信息，请参阅红帽知识库解决方案 强制执行 TLS 扩展"Extended Master Secret" 。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Configure remote logging solution by using TLS for secure transfer of logs
  hosts: managed-node-01.example.com
  tasks:
    - name: Deploying files input and forwards output with certs
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_certificates:
          - name: logging_cert
            dns: ['www.example.com']
            ca: ipa
            principal: "logging/{{ inventory_hostname }}@IDM.EXAMPLE.COM"
        logging_pki_files:
          - ca_cert: /local/path/to/ca_cert.pem
            cert: /local/path/to/logging_cert.pem
            private_key: /local/path/to/logging_cert.pem
        logging_inputs:
          - name: input_name
            type: files
            input_log_path: /var/log/containers/*.log
        logging_outputs:
          - name: output_name
            type: forwards
            target: your_target_host
            tcp_port: 514
            tls: true
            pki_authmode: x509/name
            permitted_server: 'server.example.com'
        logging_flows:
          - name: flow_name
            inputs: [input_name]
            outputs: [output_name]
```

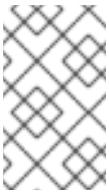
示例 playbook 中指定的设置包括如下：

logging_certificates

此参数的值传递给 **certificate** RHEL 系统角色中的 **certificate_requests**，用于创建私钥和证书。

logging_pki_files

使用这个参数，您可以配置日志记录用来查找用于 TLS 的 CA、证书和密钥文件的路径和其他设置，使用以下一个或多个子参数指定：**ca_cert**、**ca_cert_src**、**cert**、**cert_src**、**private_key**、**private_key_src** 和 **tls**。



注意

如果您使用 **logging_certificates** 在受管节点上创建文件，请不要使用 **ca_cert_src**,**cert_src** 和 **private_key_src**，它们用于复制不是由 **logging_certificates** 创建的文件。

ca_cert

表示受管节点上 CA 证书文件的路径。默认路径为 **/etc/pki/tls/certs/ca.pem**，文件名由用户设置。

cert

表示受管节点上证书文件的路径。默认路径为 **/etc/pki/tls/certs/server-cert.pem**，文件名由用户设置。

private_key

表示受管节点上私钥文件的路径。默认路径为 **/etc/pki/tls/private/server-key.pem**，文件名由用户设置。

ca_cert_src

代表控制节点上 CA 证书文件的路径，该路径将复制到目标主机上 **ca_cert** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

cert_src

表示控制节点上证书文件的路径，其将被复制到目标主机上 **cert** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

private_key_src

表示控制节点上私钥文件的路径，其将被复制到目标主机上 **private_key** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

tls

将此参数设置为 **true** 可确保通过网络安全地传输日志。如果您不想要安全封装器，您可以设置 **tls: false**。

有关角色变量的详情和有关 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件以及 **rsyslog.conf (5)** 和 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

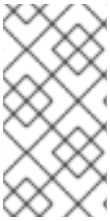
其他资源

- 从 CA 请求证书，并使用 RHEL 系统角色创建自签名证书

6.3.2. 配置带有 TLS 的服务器日志

您可以使用 **logging** RHEL 系统角色在 RHEL 服务器上配置日志，并使用 TLS 加密从远程日志系统接收日志。

此流程创建一个私钥和证书。接下来，它对 Ansible 清单中 server 组中的所有主机配置 TLS。



注意

您不必在 playbook 中调用 **证书** RHEL 系统角色来创建证书。**日志记录** RHEL 系统角色自动调用它。

要让 CA 能够为创建的证书签名，受管节点必须在 IdM 域中注册。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。
- 受管节点已在 IdM 域中注册。
- 如果要在管理节点上配置的日志服务器运行 RHEL 9.2 或更高版本，且启用了 FIPS 模式，客户端必须支持扩展主 Secret (EMS)扩展或使用 TLS 1.3。没有 EMS 的 TLS 1.2 连接会失败。如需更多信息，请参阅红帽知识库解决方案 强制执行 TLS 扩展"Extended Master Secret" 。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Configure remote logging solution by using TLS for secure transfer of logs
  hosts: managed-node-01.example.com
  tasks:
    - name: Deploying remote input and remote_files output with certs
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_certificates:
          - name: logging_cert
            dns: ['www.example.com']
            ca: ipa
            principal: "logging/{{ inventory_hostname }}@IDM.EXAMPLE.COM"
        logging_pki_files:
          - ca_cert: /local/path/to/ca_cert.pem
            cert: /local/path/to/logging_cert.pem
            private_key: /local/path/to/logging_cert.pem
        logging_inputs:
          - name: input_name
            type: remote
            tcp_ports: [514]
```

```
          tls: true
          permitted_clients: ['clients.example.com']
      logging_outputs:
        - name: output_name
          type: remote_files
          remote_log_path: /var/log/remote/%FROMHOST%/%PROGRAMNAME:::secpath-
replace%.log
          async_writing: true
          client_count: 20
          io_buffer_size: 8192
      logging_flows:
        - name: flow_name
          inputs: [input_name]
          outputs: [output_name]
```

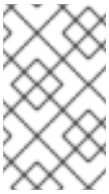
示例 playbook 中指定的设置包括如下：

logging_certificates

此参数的值传递给 **certificate** RHEL 系统角色中的 **certificate_requests**，用于创建私钥和证书。

logging_pki_files

使用这个参数，您可以配置日志记录用来查找用于 TLS 的 CA、证书和密钥文件的路径和其他设置，使用以下一个或多个子参数指定：**ca_cert**、**ca_cert_src**、**cert**、**cert_src**、**private_key**、**private_key_src** 和 **tls**。



注意

如果您使用 **logging_certificates** 在受管节点上创建文件，请不要使用 **ca_cert_src**,**cert_src** 和 **private_key_src**，它们用于复制不是由 **logging_certificates** 创建的文件。

ca_cert

表示受管节点上 CA 证书文件的路径。默认路径为 **/etc/pki/tls/certs/ca.pem**，文件名由用户设置。

cert

表示受管节点上证书文件的路径。默认路径为 **/etc/pki/tls/certs/server-cert.pem**，文件名由用户设置。

private_key

表示受管节点上私钥文件的路径。默认路径为 **/etc/pki/tls/private/server-key.pem**，文件名由用户设置。

ca_cert_src

代表控制节点上 CA 证书文件的路径，该路径将复制到目标主机上 **ca_cert** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

cert_src

表示控制节点上证书文件的路径，其将被复制到目标主机上 **cert** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

private_key_src

表示控制节点上私钥文件的路径，其将被复制到目标主机上 **private_key** 指定的位置。如果使用 **logging_certificates**，请不要使用它。

tls

将此参数设置为 **true** 可确保通过网络安全地传输日志。如果您不想要安全封装器，您可以设置 **tls: false**。

有关角色变量的详情和有关 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件以及 **rsyslog.conf (5)** 和 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

其他资源

- 从 CA 请求证书，并使用 RHEL 系统角色创建自签名证书

6.4. 使用带有 RELP 的 LOGGING RHEL 系统角色

可靠的事件日志协议(RELP)是一种通过 TCP 网络记录数据和消息的网络协议。它确保了事件消息的可靠传递，您可以在不容许任何消息丢失的环境中使用它。

RELP 发送者以命令的形式传输日志条目，接收器会在处理后确认它们。为确保一致性，RELP 将事务数保存到传输的命令中，以便进行任何类型的消息恢复。

您可以考虑在 RELP 客户端和 RELP Server 间的远程日志系统。RELP 客户端将日志传送给远程日志系统，RELP 服务器接收由远程日志系统发送的所有日志。要实现这种用例，您可以使用 **logging** RHEL 系统角色将日志记录系统配置为可靠地发送和接收日志条目。

6.4.1. 使用 RELP 配置客户端日志

您可以使用 **logging** RHEL 系统角色配置保存在带有 RELP 的远程日志系统的日志消息传输。

此流程对 Ansible 清单中 **客户端** 组中的所有主机配置 RELP。RELP 配置使用传输层安全(TLS)来加密消息传输，保证日志在网络上安全传输。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Configure client-side of the remote logging solution by using RELP
  hosts: managed-node-01.example.com
  tasks:
    - name: Deploy basic input and RELP output
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_inputs:
          - name: basic_input
            type: basics
        logging_outputs:
          - name: relp_client
            type: relp
            target: logging.server.com
            port: 20514
            tls: true
            ca_cert: /etc/pki/tls/certs/ca.pem
            cert: /etc/pki/tls/certs/client-cert.pem
            private_key: /etc/pki/tls/private/client-key.pem
            pki_authmode: name
            permitted_servers:
              - '*.server.example.com'
        logging_flows:
          - name: example_flow
            inputs: [basic_input]
            outputs: [relp_client]
```

示例 playbook 中指定的设置包括如下：

target

这是一个必需的参数，用于指定运行远程日志系统的主机名。

port

远程日志记录系统正在监听的端口号。

tls

确保日志在网络上安全地传输。如果您不想要安全打包程序，可以将 **tls** 变量设置为 **false**。在与 RELP 工作时，默认的 **tls** 参数被设置为 true，且需要密钥/证书和 triplets {**ca_cert**、**cert**、**private_key**} 和/或 {**ca_cert_src**,**cert_src**,**private_key_src**}。

- 如果设置了 {**ca_cert_src**,**cert_src**,**private_key_src**} 三元组，则默认位置 **/etc/pki/tls/certs** 和 **/etc/pki/tls/private** 被用作受管节点上的目的地，以便从控制节点传输文件。在这种情况下，文件名与 triplet 中的原始名称相同
- 如果设置了 {**ca_cert**,**cert**,**private_key**} 三元组，则文件在日志配置前应位于默认路径上。
- 如果两个三元组都设置了，则文件将从控制节点的本地路径传输到受管节点的特定路径。

ca_cert

表示 CA 证书的路径。默认路径为 **/etc/pki/tls/certs/ca.pem**，文件名由用户设置。

cert

表示证书的路径。默认路径为 **/etc/pki/tls/certs/server-cert.pem**，文件名由用户设置。

private_key

表示私钥的路径。默认路径为 **/etc/pki/tls/private/server-key.pem**，文件名由用户设置。

ca_cert_src

代表被复制到受管节点的本地 CA 证书文件路径。如果指定了 **ca_cert**，则其被复制到该位置。

cert_src

代表被复制到受管节点的本地证书文件路径。如果指定了 **cert**，则会将其复制到该位置。

private_key_src

代表被复制到受管节点的本地密钥文件路径。如果指定了 **private_key**，则会将其复制到该位置。

pki_authmode

接受身份验证模式为 **name** 或 **fingerprint**。

permitted_servers

日志客户端允许通过 TLS 连接和发送日志的服务器列表。

输入

日志输入字典列表。

输出

日志输出字典列表。

有关角色变量的详情和有关 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件以及 **rsyslog.conf (5)** 和 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

6.4.2. 配置带有 RELP 的服务器日志

您可以使用 **logging** RHEL 系统角色配置服务器，以从带有 RELP 的远程日志记录系统接收日志信息。

此流程对 Ansible 清单中 **服务器** 组中的所有主机配置 RELP。RELP 配置使用 TLS 加密消息传输，以保证在网络上安全地传输日志。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 您用于连接到受管节点的帐户对它们具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 ~/**playbook.yml** ：

```
---
- name: Configure server-side of the remote logging solution by using RELP
  hosts: managed-node-01.example.com
  tasks:
    - name: Deploying remote input and remote_files output
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.logging
      vars:
        logging_inputs:
          - name: relp_server
            type: relp
            port: 20514
            tls: true
            ca_cert: /etc/pki/tls/certs/ca.pem
            cert: /etc/pki/tls/certs/server-cert.pem
            private_key: /etc/pki/tls/private/server-key.pem
            pki_authmode: name
            permitted_clients:
              - '*client.example.com'
        logging_outputs:
          - name: remote_files_output
            type: remote_files
        logging_flows:
          - name: example_flow
            inputs: [relp_server]
            outputs: [remote_files_output]
```

示例 playbook 中指定的设置包括如下：

port

远程日志记录系统正在监听的端口号。

tls

确保日志在网络上安全地传输。如果您不想要安全打包程序，可以将 **tls** 变量设置为 **false**。在与 RELP 工作时，默认的 **tls** 参数被设置为 true，且需要密钥/证书和 triplets {**ca_cert**、**cert**、**private_key**} 和/或 {**ca_cert_src**,**cert_src**,**private_key_src**}。

- 如果设置了 {**ca_cert_src**,**cert_src**,**private_key_src**} 三元组，则默认位置 **/etc/pki/tls/certs** 和 **/etc/pki/tls/private** 被用作受管节点上的目的地，以便从控制节点传输文件。在这种情况下，文件名与 triplet 中的原始名称相同
- 如果设置了 {**ca_cert**,**cert**,**private_key**} 三元组，则文件在日志配置前应位于默认路径上。
- 如果两个三元组都设置了，则文件将从控制节点的本地路径传输到受管节点的特定路径。

ca_cert

表示 CA 证书的路径。默认路径为 **/etc/pki/tls/certs/ca.pem**，文件名由用户设置。

cert

表示证书的路径。默认路径为 **/etc/pki/tls/certs/server-cert.pem**，文件名由用户设置。

private_key

表示私钥的路径。默认路径为 **/etc/pki/tls/private/server-key.pem**，文件名由用户设置。

ca_cert_src

代表被复制到受管节点的本地 CA 证书文件路径。如果指定了 **ca_cert**，则其被复制到该位置。

cert_src

代表被复制到受管节点的本地证书文件路径。如果指定了 **cert**，则会将其复制到该位置。

private_key_src

代表被复制到受管节点的本地密钥文件路径。如果指定了 **private_key**，则会将其复制到该位置。

pki_authmode

接受身份验证模式为 **name** 或 **fingerprint**。

permitted_clients

日志记录服务器允许通过 TLS 连接和发送日志的客户端列表。

输入

日志输入字典列表。

输出

日志输出字典列表。

有关角色变量的详情和有关 **rsyslog** 的更多信息，请参阅控制节点上的 **/usr/share/ansible/roles/rhel-system-roles.logging/README.md** 文件以及 **rsyslog.conf (5)** 和 **syslog (3)** 手册页。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不能防止错误的、但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

第 7 章 审计系统

审计不会为您的系统提供额外的安全，而是用于发现系统上使用的安全策略的违规。可以通过其他安全措施(如 SELinux)进一步防止这些违规。

7.1. LINUX 审计

借助 Linux 审计，您可以跟踪系统的相关信息。通过遵循预配置的规则，审计会生成日志条目，以尽可能多地记录系统上发生的事件的信息。对于关键任务环境而言至关重要，可用来确定安全策略的违反者及其所执行的操作。

例如，审计可以在其日志文件中记录以下信息：

- 事件的日期、时间、类型和结果
- 主题和对象的敏感度标签
- 事件与触发事件的用户身份的关联
- 所有对审计配置的修改，并尝试访问审计日志文件
- 所有身份验证机制的使用，如 SSH 和 Kerberos 等
- 对任何可信数据库的更改，如 **/etc/passwd**
- 尝试向或从系统导入或导出信息
- 根据用户身份、主题和对象标签和其他属性包含或排除事件

审计系统的使用也是许多安全相关认证的一项要求。审计旨在满足或超出以下认证或合规指南的要求：

- 受控访问保护配置文件(CAPP)
- 标记的安全保护配置文件(LSPP)
- 规则集基本访问控制(RSBAC)
- 国家工业安全计划操作手册(NISPOM)
- 联邦信息安全管理法案(FISMA)
- 支付卡行业 - 数据安全标准(PCI-DSS)
- 安全技术实施指南(STIG)

审计还由国家信息保障合作伙伴(NIAP)和最佳安全行业(BSI)评估。

使用案例

监视文件访问

审计可以跟踪文件或目录是否已被访问、修改、执行或文件属性是否已被改变。例如，这有助于检测对重要文件的访问，并在其中一个文件损坏时提供审计跟踪。

监控系统调用

可将审计配置为在每次使用特定系统调用时生成日志条目。例如，这可用于通过监控 **settimeofday**、**clock_adjtime** 和其他与时间相关的系统调用来跟踪对系统时间的修改。

记录用户运行的命令

审计可以跟踪文件是否已被执行，因此可以定义一个规则以记录每次特定命令的执行。例如，可以对 **/bin** 目录中的每个可执行文件定义一个规则。然后，可以按用户 ID 搜索生成的日志条目，以生成每个用户所执行的命令的审计跟踪。

记录系统路径名称的执行

除了观察在规则调用时将路径转换为 inode 的文件访问之外，审计现在还可以观察路径的执行，即使路径在规则调用中不存在，或者在规则调用后文件被替换了。这允许规则在升级程序可执行文件后或甚至在其安装之前继续运行。

记录安全事件

pam_faillock 认证模块能够记录失败的登录尝试。也可以将审计设置为记录失败的登录尝试，并提供试图登录的用户的额外信息。

搜索事件

审计提供了 **ausearch** 工具，可用于过滤日志条目，并根据多个条件提供完整的审计跟踪。

运行总结报告

aureport 实用程序可用于生成记录事件的日常报告等。然后，系统管理员可以分析这些报告，并进一步调查可疑的活动。

监控网络访问

nftables、**iptables** 和 **ebtables** 工具可以配置为触发审计事件，使系统管理员能够监控网络访问。

注意

审计可能会影响系统性能，具体取决于收集的信息量。

7.2. 审计系统架构

审计系统由两个主要部分组成：用户空间应用程序和工具，以及内核端系统调用处理。内核组件接收用户空间应用程序的系统调用，并通过以下过滤器对其进行过滤：**user**、**task**、**fstype** 或 **exit**。

系统调用通过 **exclude** 过滤器后，它将通过上述其中一个过滤器发送，根据审计规则配置，将其发送到审计守护进程以进行进一步处理。

用户空间审计守护进程从内核收集信息，并在日志文件中创建条目。其他审计用户空间工具与审计守护进程、内核审计组件或审计日志文件进行交互：

- **auditctl** Audit 控制工具与内核审计组件进行交互，来管理规则并控制事件生成进程的许多设置和参数。
- 其余的审计工具将审计日志文件的内容作为输入，并根据用户的要求生成输出。例如，**aureport** 工具生成所有记录的事件的报告。

在 RHEL 10 中，审计分配程序守护进程(**audisp**)功能集成到审计守护进程(**auditd**)中。用于实时分析程序与审计事件交互的插件配置文件默认位于 **/etc/audit/plugins.d/** 目录中。

7.3. 安全环境的审计设置

默认的 **auditd** 配置应该适合于大多数环境。但是，如果您的环境必须满足严格的安全策略，您可以在 **/etc/audit/auditd.conf** 文件中更改审计守护进程配置的以下设置：

log_file

包含审计日志文件的目录（通常为 **/var/log/audit/**）应位于单独的挂载点上。这可以防止其他进程消耗此目录的空间，并为审计守护进程提供准确的剩余空间检测。

max_log_file

指定单个审计日志文件的最大大小，必须设置为充分利用保存审计日志文件的分区上的可用空间。**max_log_file** 参数指定最大文件大小（以 MB 为单位）。给出的值必须是数字。

max_log_file_action

一旦达到 **max_log_file** 中设置的限制，决定要采取什么行动，应将其设置为 **keep_logs**，以防止审计日志文件被覆盖。

space_left

指定磁盘上剩余的可用空间量，其是**space_left_action**参数中设置的触发时所采取的操作。必须设置一个数字，让管理员有足够的时间来响应，并释放磁盘空间。**space_left** 的值取决于审计日志文件的生成速度。如果 space_left 的值被指定为整数，它将被解释为绝对大小(MiB)。如果值被指定为 1 到 99 之间的数字，后跟一个百分比符号（例如 5%），则审计守护进程会根据包含 **log_file** 的文件系统的大小来计算绝对大小（以 MB 为单位）。

space_left_action

建议将 **space_left_action** 参数设置为 **email** 或 使用适当通知方法的 **exec**。

admin_space_left

指定绝对最小可用空间量，其是 **admin_space_left_action** 参数中设置的触发时所采取的操作，必须设置一个值，为记录管理员所执行的操作保留足够的空间。此参数的数字值应小于 space_left 的数。您还可以在数字后面附加一个百分比符号（例如 1%），以便审计守护进程根据磁盘分区计算数值。

admin_space_left_action

应设置为 **single** 来将系统置于单用户模式，并允许管理员释放一些磁盘空间。

disk_full_action

指定当保存审计日志文件的分区上没有可用空间时触发的操作，必须设置为 **halt** 或 **single**。当审计无法记录事件时，这可确保系统关闭或以单用户模式运行。

disk_error_action

指定当在包含审计日志文件的分区上检测到错误时触发的操作，必须设置为 **syslog**、**single** 或**halt**，具体取决于您处理硬件故障的本地安全策略。

flush

应设置为 **incremental_async**。它与 **freq** 参数相结合，该参数决定了在强制与硬盘进行硬盘同步前可以将多少条记录发送到磁盘。**freq** 参数应设置为**100**。这些参数可确保审计事件数据与磁盘上的日志文件同步，同时保持良好的活动性能。

其余配置选项应根据您的本地安全策略来设置。

7.4. 启动和控制 AUDITD

配置了 **auditd** 后，启动服务以收集 审计信息，并将它存储在日志文件中。以 root 用户身份运行以下命令来启动 **auditd** ：

```
# service auditd start
```

将 **auditd** 配置为在引导时启动：

```
# systemctl enable auditd
```

您可以使用 **# auditctl -e 0** 命令临时禁用 **auditd**，并使用 **# auditctl -e 1** 重新启用它。

您可以使用 **service auditd *<action>*** 命令对 **auditd** 执行其他操作，其中 *<action>* 可以是以下之一：

stop

停止 **auditd**。

restart

重新启动 **auditd**。

reload 或 force-reload

重新加载 **/etc/audit/auditd.conf** 文件中 **auditd** 的配置。

rotate

轮转 **/var/log/audit/** 目录中的日志文件。

resume

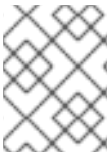
在其之前被暂停后重新恢复审计事件记录，例如，当保存审计日志文件的磁盘分区中没有足够的可用空间时。

condrestart 或 try-restart

只有当 **auditd** 运行时才重新启动它。

status

显示 **auditd** 的运行状态。



注意

service命令是与 **auditd** 守护进程正确交互的唯一方法。您需要使用 **service** 命令，以便正确记录 **auid** 值。您只将 **systemctl** 命令用于两个操作： **enable** 和 **status**。

7.5. 审计日志条目

默认情况下，审计系统将日志条目存储在 **/var/log/audit/audit.log** 文件中；如果启用了日志轮转，则轮转的 **audit.log** 文件也在存储同一个目录中。

添加以下审计规则，来记录读取或修改 **/etc/ssh/sshd_config** 文件的每次尝试：

```
# auditctl -w /etc/ssh/sshd_config -p warx -k sshd_config
```

如果 **auditd** 守护进程正在运行，使用以下命令在审计日志文件中创建新事件，例如：

```
$ cat /etc/ssh/sshd_config
```

audit.log 文件中的该事件如下。

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13
a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=1000 uid=1000
gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=1
comm="cat" exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="sshd_config"
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config" inode=409248
dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
```

```
nametype=NORMAL cap_fp=none cap_fi=none cap_fe=0 cap_fver=0
type=PROCTITLE msg=audit(1364481363.243:24287) :
proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

以上事件由四个记录组成，它们共享相同的时间戳和序列号。记录始终以 **type=** 关键字开头。每个记录由多个 ***name=value*** 对组成，它们之间由空格或逗号分开。对上述事件的详细分析如下：

第一条记录

type=SYSCALL

type 字段包含记录的类型。在本例中，**SYSCALL** 值指定此记录是由对内核的系统调用触发的。

msg=audit(1364481363.243:24287):

msg 字段记录：

- 记录的时间戳和唯一 ID 的格式为 **audit(*time_stamp*:*ID*)**。如果多个记录是作为同一审计事件的一部分而产生的，则它们共享相同的时间戳和 ID。时间戳使用 Unix 时间格式 - 自 1970 年 1 月 1 日 00:00:00 UTC 以来的秒数。
- 内核或用户空间应用程序提供的各种特定于事件的 ***name=value*** 对。

arch=c000003e

arch 字段包含系统的 CPU 架构信息。该值 **c000003e** 以十六进制表示法编码。当使用 **ausearch** 命令搜索审计记录时，请使用 **-i** 或 **--interpret** 选项来自动将十六进制值转换成人类可读的等效值。**c000003e** 值被解释为 **x86_64**。

syscall=2

syscall字段记录了发送到内核的系统调用的类型。值 **2** 可以与 **/usr/include/asm/unistd_64.h** 文件中人类可读的等效值匹配。在本例中，**2** 是 **打开** 系统调用。请注意，**ausyscall** 工具允许您将系统调用号转换为人类可读的等效值。使用 **ausyscall --dump** 命令显示所有系统调用及其编号的列表。如需更多信息，请参阅 **ausyscall**(8)手册页。

success=no

success 字段记录了该特定事件中记录的系统调用是成功还是失败。在这种情况下，调用不成功。

exit=-13

exit 字段包含一个值，指定系统调用返回的退出码。此值因不同的系统调用而不同。您可以使用以下命令将值解释成人类可读的等效值：

```
# ausearch --interpret --exit -13
```

请注意，上例假定您的审计日志包含一个失败的事件，其退出码为 **-13** 。

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

a0至**a3**字段记录了该事件中系统调用的前四个参数，用十六进制符号编码。这些参数取决于使用的系统调用，可以通过 **ausearch** 工具来解释它们。

items=1

items 字段包含系统调用记录后面的 PATH 辅助记录的数量。

ppid=2686

ppid 字段记录了父进程ID（PPID）。在这种情况下，**2686** 是父进程（如 **bash**）的 PPID 。

pid=3538

pid 字段记录了进程 ID（PID）。在本例中，**3538** 是 **cat** 进程的 PID。

auid=1000

auid字段记录了审计用户 ID，即loginuid。此 ID 在登录时分配给用户，并被每个进程继承，即使用户的身份改变了，例如使用 **su - john** 命令切换用户帐户。

uid=1000

uid 字段记录了启动分析过程的用户的用户 ID。使用以下命令可以将用户 ID 解释成用户名：**ausearch -i --uid *UID***。

gid=1000

gid 字段记录了启动分析过程的用户的组 ID。

euid=1000

euid 字段记录了启动分析过程的用户的有效用户 ID。

suid=1000

suid 字段记录了启动分析过程的用户的设置用户 ID。

fsuid=1000

fsuid 字段记录了启动分析进程的用户的文件系统用户 ID。

egid=1000

egid 字段记录了启动分析过程的用户的有效组 ID。

sgid=1000

sgid 字段记录了启动分析过程的用户的组 ID。

fsgid=1000

fsgid 字段记录了启动分析进程的用户的文件系统组 ID。

tty=pts0

tty 字段记录了分析过程被调用的终端。

ses=1

ses 字段记录了分析过程被调用的会话的会话 ID。

comm="cat"

comm 字段记录了用于调用分析过程的命令行名称。在本例中，cat **命令用于触发此审计事件**。

exe="/bin/cat"

exe 字段记录了用于调用分析过程的可执行文件的路径。

subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

subj 字段记录了被分析的进程在执行时被标记的 SELinux 上下文。

key="sshd_config"

key 记录了与在审计日志中生成该事件的规则相关联的管理员定义的字符串。

第二条记录

type=CWD

在第二条记录中，**type** 字段值为 **CWD** - 当前工作目录。此类型用于记录从中调用第一条记录中指定的系统调用的进程的工作目录。
此记录的目的是记录当前进程的位置，以防在相关 PATH 记录中捕获到相对路径。这样，就可以重建绝对路径。

msg=audit(1364481363.243:24287)

msg 字段持有与第一条记录中的值相同的时间戳和 ID 值。时间戳使用 Unix 时间格式 - 自 1970 年 1 月 1 日 00:00:00 UTC 以来的秒数。

cwd="/home/*user_name*"

cwd 字段包含系统调用所在目录的路径。

第三条记录

type=PATH

在第三条记录中，**type** 字段值为 **PATH**。审计事件包含作为参数传递给系统调用的每个路径的 **PATH** 类型记录。在这个审计事件中，只有一个路径(**/etc/ssh/sshd_config**) 被用作参数。

msg=audit(1364481363.243:24287):

msg 字段拥有与第一和第二条记录中的值相同的时间戳和 ID 值。

item=0

item 字段表示在 **SYSCALL** 类型记录所引用的项目总数中，当前记录是哪个项目。这个数是以零为基础的；值为 **0** 表示它是第一项。

name="/etc/ssh/sshd_config"

name 字段记录了作为参数传递给系统调用的文件或目录的路径。在本例中，它是 **/etc/ssh/sshd_config** 文件。

inode=409248

inode 字段包含与该事件中记录的文件或目录相关联的 inode 号。以下命令显示与 **409248** inode 号相关联的文件或目录：

```
# find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00

dev 字段指定了包含该事件中记录的文件或目录的设备的次要和主要 ID。在本例中，值表示 **/dev/fd/0** 设备。

mode=0100600

mode 字段记录文件或目录权限，由数字标记。它是 **st_mode** 字段中的 **stat** 命令返回。如需更多信息，请参阅 **stat(2)** 手册页。在这种情况下，**0100600** 可以解释为 **-rw-------**，这意味着只有 root 用户对 **/etc/ssh/sshd_config** 文件具有读和写的权限。

ouid=0

ouid 字段记录了对象所有者的用户 ID。

ogid=0

ogid 字段记录了对象所有者的组 ID。

rdev=00:00

rdev 字段包含一个记录的设备标识符，仅用于特殊文件。在这种情况下，不会使用它，因为记录的文件是一个常规文件。

obj=system_u:object_r:etc_t:s0

obj 字段记录了 SELinux 上下文，在执行时，记录的文件或目录被贴上了标签。

nametype=NORMAL

nametype 字段记录了每个路径记录在给定系统调用的上下文中的操作意图。

cap_fp=none

cap_fp 字段记录了与设置文件或目录对象的基于文件系统的允许能力有关的数据。

cap_fi=none

cap_fi 字段记录了与文件或目录对象的基于继承文件系统的能力设置有关的数据。

cap_fe=0

cap_fe 字段记录了文件或目录对象基于文件系统能力的有效位的设置。

cap_fver=0

cap_fver 字段记录了文件或目录对象基于文件系统能力的版本。

第四条记录

type=PROCTITLE

type 字段包含记录的类型。在本例中，**PROCTITLE** 值指定此记录提供触发此审计事件的完整命令行，该事件是由对内核的系统调用触发的。

proctitle=636174002F6574632F7373682F737368645F636F6E666967

proctitle 字段记录了用于调用分析过程的命令的完整命令行。该字段采用十六进制表示法编码，不允许用户影响审计日志解析器。对触发此审计事件的命令进行文本解码。当使用 **ausearch** 命令搜索审计记录时，请使用 **-i** 或 **--interpret** 选项来自动将十六进制值转换成人类可读的等效值。**636174002F6574632F7373682F737368645F636F6E666967** 值解释为 **cat /etc/ssh/sshd_config**。

7.6. 使用 AUDITCTL 定义的审计规则示例

审计系统根据一组规则进行操作，这些规则定义日志文件中所捕获的内容。使用 **auditctl** 工具，可以在命令行或 **/etc/audit/rules.d/** 目录中设置审计规则。

auditctl 命令使您能够控制审计系统的基本功能，并定义决定记录哪些审计事件的规则。

文件系统规则示例

1. 要定义一条规则，记录对 **/etc/passwd** 文件的所有写访问和每个属性的修改：

```
# auditctl -w /etc/passwd -p wa -k passwd_changes
```

2. 要定义一条规则，记录对 **/etc/selinux/** 目录中所有文件的写访问和每个属性的修改：

```
# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

系统调用规则示例

1. 要定义一条规则，当程序每次使用 **adjtimex** 或 **settimeofday** 系统调用时就创建一条日志，系统使用 64 位构架：

```
# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
```

2. 定义一条规则，在 ID 为 1000 或以上的系统用户每次删除或重命名文件时创建一条日志：

```
# auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F
auid!=4294967295 -k delete
```

请注意，**-F auid!=4294967295** 选项用于排除未设置登录 UID 的用户。

可执行文件规则

要定义一条规则，记录所有 **/bin/id** 程序的执行，请执行以下命令：

```
# auditctl -a always,exit -F exe=/bin/id -F arch=b64 -S execve -k execution_bin_id
```

其他资源

- 您系统上的 **auditctl (8)** 手册页

7.7. 审计持久性规则

要定义在重启过程中保持不变的审计规则，必须直接将其包含在 **/etc/audit/rules.d/audit.rules** 文件中，或者使用 **augenrules** 程序读取位于**/etc/audit/rules.d/** 目录中的规则。

请注意，每次 **auditd** 服务启动时都会生成 **/etc/audit/audit.rules** 文件。**/etc/audit/rules.d/** 中的文件使用相同的 **auditctl** 命令行语法来指定规则。哈希符号(#)后面的空行和文本将被忽略。

另外，您可以使用 **auditctl** 命令来从用 **-R** 选项指定的文件中读取规则，例如：

```
# auditctl -R /usr/share/audit/sample-rules/30-stig.rules
```

augenrules脚本读取位于**/etc/audit/rules.d/**目录下的规则，并将它们编译成**audit.rures**文件。这个脚本会根据文件的自然排列顺序，按特定顺序处理以 **.rules** 结尾的所有文件。这个目录中的文件被组织到具有如下含义的组中：

10
: 内核和 auditctl 配置

20
: 可以匹配常规规则但您想要不同匹配的规则

30
: 主规则

40
: 可选规则

50
: 特定于服务器的规则

70
: 系统本地规则

90
: 完成（不可变）

规则并非是一次全部使用。它们是策略的一部分，应仔细考虑，并将单个文件复制到 **/etc/audit/rules.d/**。例如，要在 STIG 配置中设置系统，请复制规则 **10-base-config**、**30-stig**、**31-privileged** 和 **99-finalize**。

在 **/etc/audit/rules.d/** 目录中有了规则之后，运行带有 **--load** 参数的 **augenrules** 脚本来加载它们：

```
# augenrules --load
/sbin/augenrules: No change
No rules
enabled 1
failure 1
pid 742
rate_limit 0
...
```

其他资源

- 系统中的 **audit.rules (8)** 和 **augenrules (8)** 手册页

7.8. 预先配置的审计规则文件符合标准

要配置 Audit 以符合特定的认证标准（如 OSPP、PCI DSS 或 STIG），您可以使用 **audit** 软件包安装的一组预配置的规则文件来作为起点。示例规则位于 **/usr/share/audit/sample-rules** 目录中。

警告

sample-rules 目录中的 Audit 示例规则并不详尽，也不是最新的，因为安全标准是动态的，并可能会发生变化。提供这些规则仅为了演示如何构建和编写审计规则。它们不能确保立即符合最新的安全标准。要根据特定安全准则使您的系统符合最新的安全标准，请使用 基于 SCAP 的安全合规性工具。

30-nispom.rules

满足国家工业安全计划操作手册"信息系统安全"一章中指定的要求的审计规则配置.

30-ospp-v42*.rules

满足 OSPP（通用目的操作系统保护配置文件）配置文件版本 4.2 中定义的要求的审计规则配置。

30-pci-dss-v31.rules

满足支付卡行业数据安全标准(PCI DSS)v3.1 要求的审计规则配置。

30-stig.rules

满足安全技术实施指南(STIG)要求的审计规则配置。

要使用这些配置文件，将其复制到 **/etc/audit/rules.d/** 目录中，并使用 **augenrules --load** 命令，例如：

```
# cd /usr/share/audit/sample-rules/
# cp 10-base-config.rules 30-stig.rules 31-privileged.rules 99-finalize.rules /etc/audit/rules.d/
# augenrules --load
```

您可以使用编号方案对审核规则进行排序。如需更多信息，请参阅 **/usr/share/audit/sample-rules/README-rules** 文件。

其他资源

- 您系统上的 **audit.rules (7)** 手册页

7.9. 禁用 AUGENRULES

您可以禁用 **augenrules** 工具，它将审计切换为使用 **/etc/audit/audit.rules** 文件中定义的规则。

流程

1. 将 **/usr/lib/systemd/system/auditd.service** 文件复制到 **/etc/systemd/system/** 目录中：

```
# cp -f /usr/lib/systemd/system/auditd.service /etc/systemd/system/
```

2. 在您选择的文本编辑器中编辑 **/etc/systemd/system/auditd.service** 文件，例如：

```
# vi /etc/systemd/system/auditd.service
```

3. 注释掉包含 **augenrules** 的行，将包含 **auditctl -R** 命令的行取消注释：

```
#ExecStartPost=-/sbin/augenrules --load
ExecStartPost=-/sbin/auditctl -R /etc/audit/audit.rules
```

4. 重新载入 **systemd** 守护进程以获取 **auditd.service** 文件中的修改：

```
# systemctl daemon-reload
```

5. 重启 **auditd** 服务：

```
# service auditd restart
```

其他资源

- 系统中的 **augenrules (8)** 和 **audit.rules (8)** 手册页
- auditd 服务重启覆盖对 /etc/audit/audit.rules （红帽知识库）所做的更改

7.10. 设置审计来监控软件更新

您可以使用预先配置的规则 **44-installers.rules** 将 Audit 配置为监控以下安装软件的工具：

- **dnf** [1]
- **yum**
- **pip**
- **npm**
- **cpan**
- **gem**

- **luarocks**

要监控 **rpm** 实用程序，请安装 **rpm-plugin-audit** 软件包。然后，审计会在安装或升级软件包时生成 **SOFTWARE_UPDATE** 事件。您可以通过在命令行中输入 **ausearch -m SOFTWARE_UPDATE** 来列出这些事件。

注意

预配置的规则文件不能用于 **ppc64le** 和 **aarch64** 架构的系统。

先决条件

- **auditd** 根据 第 7.3 节 “安全环境的审计设置” 配置。

流程

1. 将预先配置的规则文件 **44-installers.rules** 从 **/usr/share/audit/sample-rules/** 目录复制到 **/etc/audit/rules.d/** 目录中：

```
# cp /usr/share/audit/sample-rules/44-installers.rules /etc/audit/rules.d/
```

2. 加载审计规则：

```
# augenrules --load
```

验证

1. 列出载入的规则：

```
# auditctl -l
-p x-w /usr/bin/dnf-3 -k software-installer
-p x-w /usr/bin/yum -k software-installer
-p x-w /usr/bin/pip -k software-installer
-p x-w /usr/bin/npm -k software-installer
-p x-w /usr/bin/cpan -k software-installer
-p x-w /usr/bin/gem -k software-installer
-p x-w /usr/bin/luarocks -k software-installer
```

2. 执行安装，例如：

```
# dnf reinstall -y vim-enhanced
```

3. 在审计日志中搜索最近的安装事件，例如：

```
# ausearch -ts recent -k software-installer
&amp;#8211;&amp;#8211;&amp;#8211;&amp;#8211;
time->Thu Dec 16 10:33:46 2021
type=PROCTITLE msg=audit(1639668826.074:298):
proctitle=2F7573722F6C6962657865632F706C6174666F726D2D707974686F6E002F75737
22F62696E2F646E66007265696E7374616C6C002D790076696D2D656E68616E636564
type=PATH msg=audit(1639668826.074:298): item=2 name="/lib64/ld-linux-x86-64.so.2"
inode=10092 dev=fd:01 mode=0100755 ouid=0 ogid=0 rdev=00:00
obj=system_u:object_r:ld_so_t:s0 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
```

```
cap_fver=0 cap_frootid=0
type=PATH msg=audit(1639668826.074:298): item=1 name="/usr/libexec/platform-python"
inode=4618433 dev=fd:01 mode=0100755 ouid=0 ogid=0 rdev=00:00
obj=system_u:object_r:bin_t:s0 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
cap_fver=0 cap_frootid=0
type=PATH msg=audit(1639668826.074:298): item=0 name="/usr/bin/dnf" inode=6886099
dev=fd:01 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:rpm_exec_t:s0
nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1639668826.074:298): cwd="/root"
type=EXECVE msg=audit(1639668826.074:298): argc=5 a0="/usr/libexec/platform-python"
a1="/usr/bin/dnf" a2="reinstall" a3="-y" a4="vim-enhanced"
type=SYSCALL msg=audit(1639668826.074:298): arch=c000003e syscall=59 success=yes
exit=0 a0=55c437f22b20 a1=55c437f2c9d0 a2=55c437f2aeb0 a3=8 items=3 ppid=5256
pid=5375 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=3
comm="dnf" exe="/usr/libexec/platform-python3.6"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="software-installer"
```

7.11. 使用审计监控用户登录时间

要监控特定时间登录的用户，您可以使用 **ausearch** 或 **aureport** 工具，它们提供了显示相同信息的不同方法。这不需要以任何特定的方式配置审计。

先决条件

- **auditd** 根据 第 7.3 节 “安全环境的审计设置” 配置。

流程

要显示用户登录时间，请使用以下命令之一：

- 在审计日志中搜索 **USER_LOGIN** 消息类型：

```
# ausearch -m USER_LOGIN -ts '12/02/2020' '18:00:00' -sv no
time->Mon Nov 22 07:33:22 2021
type=USER_LOGIN msg=audit(1637584402.416:92): pid=1939 uid=0 auid=4294967295
ses=4294967295 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=login acct="
(unknown)" exe="/usr/sbin/sshd" hostname=? addr=10.37.128.108 terminal=ssh res=failed'
```

  - 您可以使用 **-ts** 选项指定日期和时间。如果不使用这个选项，**ausearch** 将提供从当天开始的结果，如果您省略时间，**ausearch** 将提供从午夜开始的结果。
  - 您可以使用 **-sv yes** 选项来过滤成功的登录尝试，**-sv no** 用来过滤失败的登录尝试。

- 将 **ausearch** 命令的原始输出传送给 **aulast** 工具，它以类似于 **last** 命令的输出格式显示输出。例如：

```
# ausearch --raw | aulast --stdin
root     ssh          10.37.128.108    Mon Nov 22 07:33 - 07:33  (00:00)
root     ssh          10.37.128.108    Mon Nov 22 07:33 - 07:33  (00:00)
root     ssh          10.22.16.106     Mon Nov 22 07:40 - 07:40  (00:00)
reboot   system boot  4.18.0-348.6.el8 Mon Nov 22 07:33
```

- 使用 **aureport** 命令及 **--login -i** 选项来显示登录事件列表。

```
# aureport --login -i

Login Report
============================================
# date time auid host term exe success event
============================================
1. 11/16/2021 13:11:30 root 10.40.192.190 ssh /usr/sbin/sshd yes 6920
2. 11/16/2021 13:11:31 root 10.40.192.190 ssh /usr/sbin/sshd yes 6925
3. 11/16/2021 13:11:31 root 10.40.192.190 ssh /usr/sbin/sshd yes 6930
4. 11/16/2021 13:11:31 root 10.40.192.190 ssh /usr/sbin/sshd yes 6935
5. 11/16/2021 13:11:33 root 10.40.192.190 ssh /usr/sbin/sshd yes 6940
6. 11/16/2021 13:11:33 root 10.40.192.190 /dev/pts/0 /usr/sbin/sshd yes 6945
```

其他资源

- 您系统上的 **ausearch (8)**, **aulast (8)** 和 **aureport (8)** 手册页

7.12. 其他资源

- RHEL Audit 系统参考 （红帽知识库）
- 容器中的 auditd 执行选项 （红帽知识库）
- Linux Audit Documentation Project 页面（Github.com）
- 系统上的 **/usr/share/doc/audit/** 目录
- **auditd (8)**、 **auditctl (8)**, **ausearch (8)**, **audit.rules (7)**, **audispd.conf (5)**, **audispd (8)**, **auditd.conf (5)**, **ausearch-expression (5)**, **aulast (8)**, **aulastlog (8)**, **aureport (8)**, **ausyscall (8)**, ausyscall (8), **autrace (8)**, **auvirt (8)** man page

[1] 由于**dnf** 在 RHEL 中是符号链接，因此**dnf** 审计规则中的路径必须包含符号链接的目标。要接收正确的审计事件，请通过将 **path=/usr/bin/dnf** 路径改为 **/usr/bin/dnf-3** 来修改 **44-installers.rules** 文件。

第 8 章 管理及监控安全更新

了解如何安装安全更新,并显示更新的额外详情，以保护您的 Red Hat Enterprise Linux 系统免受新发现的威胁和漏洞的攻击。

8.1. 识别安全更新

为了确保企业系统不受当前和未来的安全威胁，系统需要定期进行安全更新。红帽产品安全团队为您提供了安心部署和维护企业解决方案所需的指导。

8.1.1. 什么是安全公告？

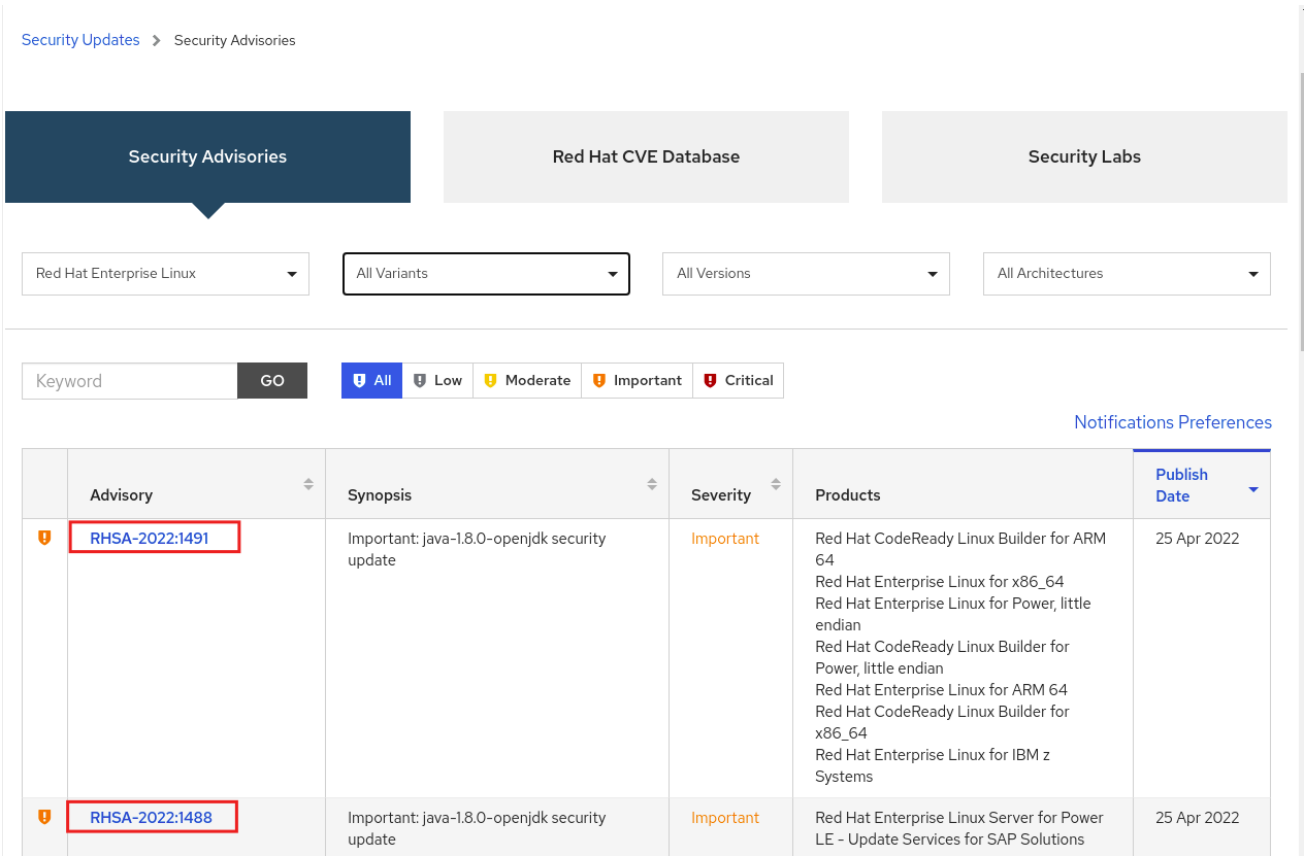
红帽安全公告（Red Hat Security Advisories，简称 RHSA）记录了有关红帽产品和服务中安全漏洞的信息。

每个 RHSA 包括以下信息：

- 重要性
- 类型和状态
- 受影响的产品
- 修复问题的摘要
- 问题相关的报告链接。请注意，不是所有的报告都是公开的。
- 公共漏洞和暴露（Common Vulnerabilities and Exposures，简称 CVE）编号以及更多详情（如攻击复杂性）的链接。

红帽客户门户（Red Hat Customer Portal）提供了红帽发布的红帽安全公告列表。您可以通过访问红帽安全公告列表中的公告 ID 来显示特定公告的详情。

图 8.1. 安全公告列表



此外，您还可以根据特定产品、变体、版本和架构过滤结果。例如，只显示 Red Hat Enterprise Linux 9 公告，您可以设置以下过滤器：

- 产品: Red Hat Enterprise Linux
- 变体: 所有变体
- Version: 9
- （可选）选择一个次版本。

其他资源

- 红帽安全公告列表
- 红帽安全公告分析

8.1.2. 显示主机上未安装的安全更新

您可以使用 **dnf** 实用程序列出系统的所有可用安全更新。

先决条件

- 红帽订阅已附加到主机。

流程

- 列出主机上尚未安装的所有可用安全更新：

```
# dnf updateinfo list updates security
…
RHSA-2019:0997 Important/Sec. platform-python-3.6.8-2.el8_0.x86_64
RHSA-2019:0997 Important/Sec. python3-libs-3.6.8-2.el8_0.x86_64
RHSA-2019:0990 Moderate/Sec.  systemd-239-13.el8_0.3.x86_64
…
```

8.1.3. 显示在主机上安装的安全更新

您可以使用 **dnf** 实用程序列出已安装系统的安全更新。

流程

- 列出主机上安装的所有安全更新：

```
# dnf updateinfo list security --installed
…
RHSA-2019:1234 Important/Sec. libssh2-1.8.0-7.module+el8+2833+c7d6d092
RHSA-2019:4567 Important/Sec. python3-libs-3.6.7.1.el8.x86_64
RHSA-2019:8901 Important/Sec. python3-libs-3.6.8-1.el8.x86_64
…
```

如果安装了多个软件包更新，**dnf** 将列出该软件包的所有公告。在上例中，自系统安装以来，已安装了 **python3-libs** 软件包的两个安全更新。

8.1.4. 使用 DNF 显示特定公告

您可以使用 **dnf** 实用程序显示可用于更新的特定公告信息。

先决条件

- 红帽订阅已附加到主机。
- 您知道安全公告的 ID。
- 公告提供的更新没有安装。

流程

- 显示特定的公告，例如：

```
# dnf updateinfo info RHSA-2019:0997
==============================================================================
  Important: python3 security update
==============================================================================
  Update ID: RHSA-2019:0997
       Type: security
    Updated: 2019-05-07 05:41:52
       Bugs: 1688543 - CVE-2019-9636 python: Information Disclosure due to urlsplit improper
NFKC normalization
       CVEs: CVE-2019-9636
Description: …
```

8.2. 安装安全更新

在 Red Hat Enterprise Linux 中，您可以安装特定的安全公告以及所有可用的安全更新。您还可以将系统配置为自动下载和安装安全更新。

8.2.1. 安装所有可用的安全更新

要保持系统的安全性，您可以使用 **dnf** 工具安装所有当前可用的安全更新。

先决条件

- 红帽订阅已附加到主机。

流程

1. 使用 **dnf** 工具安装安全更新：

```
# dnf update --security
```

   没有 **--security** 参数，**dnf update** 会安装所有更新，包括 bug 修复和增强。

2. 按 **y** 确认并启动安装：

```
…
Transaction Summary
==========================================
Upgrade  … Packages

Total download size: … M
Is this ok [y/d/N]: y
```

3. 可选：在安装更新的软件包后列出需要手动重启系统的进程：

```
# dnf needs-restarting
1107 : /usr/sbin/rsyslogd -n
1199 : -bash
```

   上一命令只列出需要重启的进程，而不是服务。也就是说，您无法使用 **systemctl** 工具重启列出的进程。例如，当拥有此进程的用户退出时，输出中的 **bash** 进程会被终止。

8.2.2. 安装特定公告提供的安全更新

在某些情况下，您可能只希望安装特定的更新。例如，某个特定的服务可以在不需要停机的情况下进行更新，您可以只为该服务安装安全更新，并在以后安装剩余的安全更新。

先决条件

- 红帽订阅已附加到主机。
- 您知道您要更新的安全公告的 ID。
  如需更多信息，请参阅 识别安全公告更新 部分。

流程

1. 安装特定的公告，例如：

```
# dnf update --advisory=RHSA-2019:0997
```

2. 或者，使用 **dnf upgrade-minimal** 命令进行更新，以应用具有最小版本更改的特定公告，例如：

```
# dnf upgrade-minimal --advisory=RHSA-2019:0997
```

3. 按 **y** 确认并启动安装：

```
…
Transaction Summary
==========================================
Upgrade  … Packages

Total download size: … M
Is this ok [y/d/N]: y
```

4. 可选：在安装更新的软件包后列出需要手动重启系统的进程：

```
# dnf needs-restarting
1107 : /usr/sbin/rsyslogd -n
1199 : -bash
```

上一命令只列出需要重启的进程，而不是服务。这意味着，您无法使用 **systemctl** 工具重启列出的所有进程。例如，当拥有此进程的用户退出时，输出中的 **bash** 进程会被终止。

8.2.3. 自动安装安全更新

您可以配置您的系统，以便其自动下载并安装所有安全更新。

先决条件

- 红帽订阅已附加到主机。
- **dnf-automatic** 软件包已安装。

流程

1. 在 **/etc/dnf/automatic.conf** 文件中，在 **[commands]** 部分，确保将 **upgrade_type** 选项被设置为 **default** 或 **security** ：

```
[commands]
#  What kind of upgrade to perform:
# default                            = all available upgrades
# security                           = only the security upgrades
upgrade_type = security
```

2. 启用并启动 **systemd** 计时器单元：

```
# systemctl enable --now dnf-automatic-install.timer
```

验证

验证

1. 验证计时器是否已启用：

```
# systemctl status dnf-automatic-install.timer
```

其他资源

- 您系统上的 **dnf-automatic (8)** 手册页