

# Centralização de logs: da geração do evento à investigação

journald · rsyslog · NTP · LogAnalyzer

## DISCIPLINA

*Segurança em Sistemas Operacionais e Redes de Computadores II*

## CONTEÚDO

Geração de eventos · Coleta local · Encaminhamento · Análise centralizada

### Geração

journald captura eventos do kernel e serviços

### Coleta

rsyslog filtra e processa registros locais

### Transporte

Encaminhamento via rede para servidor central

### Análise

LogAnalyzer consolida e apresenta os dados

# O que é um log?

**Definição:** registro de um evento produzido pelo sistema operacional, serviço ou dispositivo de rede — contendo data, hora e contexto suficiente para reconstruir o que ocorreu.

## Login SSH bem-sucedido

```
Accepted publickey for alice from 10.0.0.5 port 42310
```

Identifica usuário, método de autenticação, IP de origem e porta efêmera.

## Uso de sudo

```
alice : TTY=pts/0 ; COMMAND=/usr/bin/apt upgrade
```

Registra quem executou, em qual terminal e qual comando privilegiado foi invocado.

## Falha de autenticação

```
Failed password for root from 203.0.113.9 port 55214
```

Sinaliza tentativa de acesso não autorizado — IP externo tentando força bruta na conta root.

## Conexão VPN

```
peer 10.8.0.2 connected, cipher AES-256-GCM
```

Confirma estabelecimento de túnel seguro com o cipher negociado registrado explicitamente.

📌 Cada linha de log conta uma história: quem, quando, onde, o quê e o resultado. A ausência de qualquer um desses campos compromete a investigação forense.

# Facilities

Uma **facility** identifica a origem de uma mensagem de log — qual componente do sistema gerou o evento — permitindo filtrar e rotear logs com precisão.

## Facilities do Sistema

|                      |                                  |
|----------------------|----------------------------------|
| kern (0)             | Mensagens do kernel do sistema.  |
| user (1)             | Processos gerados por usuários.  |
| mail (2)             | Sistema de e-mail e componentes. |
| daemon (3)           | Daemons do sistema em execução.  |
| auth/authpriv (4/10) | Autenticação e segurança.        |

## Demais Facilities

|                       |                                  |
|-----------------------|----------------------------------|
| syslog (5)            | Mensagens internas do syslogd.   |
| lpr (6)               | Sistema de impressão.            |
| cron (9)              | Tarefas agendadas pelo cron.     |
| local0–local7 (16–23) | Reservadas para uso customizado. |

# Severity Levels e Filtragem

## Níveis de Severidade

Os níveis de severidade indicam a urgência do evento, do mais crítico ao menos crítico.

| Nível | Nome    | Descrição                         |
|-------|---------|-----------------------------------|
| 0     | emerg   | Sistema inutilizável              |
| 1     | alert   | Ação imediata necessária          |
| 2     | crit    | Condição crítica                  |
| 3     | err     | Erro                              |
| 4     | warning | Aviso                             |
| 5     | notice  | Condição normal mas significativa |
| 6     | info    | Informação                        |
| 7     | debug   | Depuração                         |

## Filtragem no rsyslog

A sintaxe **facility.severity** combina os dois eixos para rotear mensagens.

```
auth.*      /var/log/auth.log
*.err       @@172.17.0.2:514
kern.crit   /var/log/kernel_crit.log
```

auth.\* captura todos os níveis de eventos da facility de autenticação; \*.err captura erros de qualquer facility; e kern.crit captura apenas mensagens críticas do kernel.

## MOTIVAÇÃO

# Por que centralizar logs?

## ✗ Sem centralização

Cada máquina guarda seus próprios registros de forma isolada. Correlacionar eventos entre hosts exige acesso individual a cada sistema — lento, propenso a erros e inviável em escala.

- Investigação fragmentada por máquina
- Linha do tempo inconsistente entre hosts
- Logs perdidos se a máquina for comprometida ou desligada

## ✓ Com servidor central

Visão unificada de todos os eventos. A correlação entre máquinas e redes distintas torna-se possível e eficiente.

- Correlação de eventos entre múltiplos hosts
- Linha do tempo consistente para investigação forense
- Preservação garantida mesmo após desligamento do host comprometido

# Requisitos de um registro de segurança

## ISO/IEC 27001:2022 – Controles relevantes

| 1                                                                                                                                                | 2                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>A.8.15 Logging</b><br><br>Registros devem permitir reconstruir o evento. <b>A norma define o objetivo, não o formato exato de cada linha.</b> | <b>A.8.17 Clock Synchronization</b><br><br>Relógios sincronizados via NTP são necessários para ordenar corretamente eventos originados em máquinas diferentes. |

 ISO/IEC 27001 e 27002 orientam objetivos e controles, mas não substituem uma política detalhada de retenção, acesso e proteção dos logs.

## Campos recomendados para auditoria e investigação

| Campo     | Descrição                                 |
|-----------|-------------------------------------------|
| O quê     | Ação executada                            |
| Quando    | Data, hora e fuso horário                 |
| Onde      | Máquina ou serviço                        |
| Quem      | Usuário (quando disponível)               |
| De onde   | IP, terminal ou aplicação                 |
| Resultado | Sucesso, falha, bloqueio ou erro          |
| Recurso   | Recurso afetado e identificador de sessão |

LABORATÓRIO

# Arquitetura do laboratório

Papéis de cada componente

## journald

Registra eventos localmente no formato binário do systemd

## NTP

Mantém horários consistentes entre todas as máquinas

## rsyslog

Coleta, filtra, armazena e encaminha logs para o servidor central

## LogAnalyzer

Apenas consulta e apresenta — não gera nem processa logs

# journald → rsyslog: integração local

## Módulo imjournal

Permite ao rsyslog ler eventos do `systemd-journald` diretamente, sem duplicar o pipeline de coleta.

```
module(load="imjournal"  
        StateFile="imjournal.state")
```

⚠ Use **somente uma** regra ativa para gravar em `/var/log/syslog`. Os módulos `imuxsock` e `imjournal` podem duplicar registros se ambos estiverem ativos simultaneamente.

## Validar e testar a configuração

Execute os comandos abaixo em sequência para confirmar que a integração está funcional:

```
# 1. Verificar sintaxe da configuração  
rsyslogd -N1  
  
# 2. Reiniciar o serviço  
systemctl restart rsyslog  
  
# 3. Gerar mensagem de teste  
logger "TESTE-JOURNAL-RSYSLOG"  
  
# 4. Confirmar presença no syslog  
grep "TESTE-JOURNAL-RSYSLOG" /var/log/syslog
```

Se a string aparecer em `/var/log/syslog`, o pipeline `journald → rsyslog` está operacional.

# Encaminhar logs para outro servidor

## UDP vs TCP: escolha consciente

| Protocolo               | Vantagem                         | Limitação                               |
|-------------------------|----------------------------------|-----------------------------------------|
| @172.17.0.2:514<br>UDP  | Menor sobrecarga de rede         | Sem confirmação — pode perder mensagens |
| @@172.17.0.2:514<br>TCP | Confirma entrega, mais confiável | Maior overhead, conexão persistente     |

- ⊗ UDP e TCP simples **não criptografam** o conteúdo. Em produção, utilize TLS, RELP ou encaminhe através de um túnel VPN.

## Servidor central — escutar UDP/514

```
module(load="imudp")  
input(type="imudp" port="514")
```

# Atividade prática e conclusão

## Roteiro do laboratório

### 1 Gerar eventos de autenticação

Realize um login SSH bem-sucedido e uma tentativa malsucedida para o mesmo host.

### 2 Executar ação privilegiada

Execute um comando com `sudo` e envie uma mensagem manual com `logger`.

### 3 Verificar no servidor central

Localize os eventos no SSORI-SERVER-LOCAL e compare: horário, origem, usuário, ação e resultado.

## Pergunta para reflexão

*"Se o servidor AWS estiver em outra rede privada, como os logs chegarão ao servidor central em 172.17.0.2?"*

→ Esta pergunta introduz a necessidade de uma **VPN** e de rotas entre as redes — tema do próximo módulo.

## Conceitos revisados nesta aula

- `journald` como registrador local do `systemd`
- `rsyslog` como coletor, filtro e encaminhador
- NTP como requisito para correlação temporal
- LogAnalyzer como interface de consulta centralizada
- UDP vs TCP e a ausência de criptografia nativa