PRÁTICA — Análise de Logs no Linux

Professor: Marcos Brandão

Versão: 2026

---

1. Objetivos

Ao final desta prática, o aluno deverá ser capaz de:

- compreender o que é um arquivo de log;
- identificar informações importantes em registros de SSH;
- utilizar `cat`, `head`, `tail`, `grep` e `wc`;
- utilizar pipes (`|`) para combinar comandos;
- acompanhar um arquivo de log em tempo real;
- identificar falhas e sucessos de autenticação;
- localizar usuários e endereços IP em um log;
- realizar uma investigação básica de eventos de segurança.

PARTE 1 — ENTENDENDO OS LOGS

2. O que é um log?

Um **log** é um registro de acontecimentos de um sistema.

Um servidor Linux pode registrar, por exemplo:

- tentativas de login;
- logins realizados com sucesso;
- falhas de autenticação;
- conexões SSH;
- usuários inexistentes;
- endereços IP de origem;
- inicialização de serviços;
- erros do sistema.

Neste laboratório utilizaremos dois arquivos:

/home/lab_ssh.log
/home/ssh_tempo_real.log

`lab_ssh.log`

Arquivo fictício preparado pelo professor. Permite realizar os exercícios sem depender de acontecimentos reais.

`ssh_tempo_real.log`

Arquivo alimentado pelo servidor com registros reais do serviço SSH. Será utilizado para observar novos eventos.

PARTE 2 — INTERPRETANDO UM REGISTRO

Um registro pode apresentar:

Sep 09 10:45:35 vps67062 sshd-session[1101167]: Failed password for invalid user ubuntu from 104.234.186.154 port 38526 ssh2

Podemos interpretá-lo da seguinte maneira:

Informação	Significado
`Sep 09`	Data
`10:45:35`	Horário
`vps67062`	Nome do servidor
`sshd-session`	Processo relacionado ao SSH
`Failed password`	Falha de autenticação

`invalid user ubuntu`	Usuário utilizado não existe
`104.234.186.154`	Endereço IP de origem
`38526`	Porta de origem
`ssh2`	Protocolo SSH

Importante: uma falha de senha não significa necessariamente que o servidor foi invadido. Ela mostra uma tentativa de autenticação que não teve sucesso.

PARTE 3 — PREPARAÇÃO DO LABORATÓRIO

3. Arquivo fictício

O professor disponibilizará:

/home/lab_ssh.log

Verifique:

ls -l /home/lab_ssh.log

---

4. Criando o arquivo de acompanhamento em tempo real

Esta etapa deve ser realizada pelo **professor/administrador**, pois exige privilégios administrativos.

Crie o arquivo:

sudo touch /home/ssh_tempo_real.log

Defina `root` como proprietário e a turma como grupo:

sudo chown root:turma_informatica_2026 /home/ssh_tempo_real.log

Defina as permissões:

sudo chmod 640 /home/ssh_tempo_real.log

Confira:

ls -l /home/ssh_tempo_real.log

O resultado deverá ser semelhante a:

-rw-r----- 1 root turma_informatica_2026 0 Sep 09 16:00 /home/ssh_tempo_real.log

As permissões `640` significam:

Usuário	Permissão

`root`	leitura e escrita
`turma_informatica_2026`	somente leitura
demais usuários	nenhuma

Assim, os alunos podem analisar o arquivo, mas não devem conseguir modificá-lo.

PARTE 4 — COPIANDO O JOURNAL SSH EM TEMPO REAL

5. Iniciando a captura

No Debian utilizado neste laboratório, os registros do SSH podem ser consultados através do `journalctl`.

Para acompanhar o serviço SSH:

sudo journalctl -u ssh -f

A opção:

-f

significa **follow**, ou seja, acompanhar novos eventos.

Para enviar esses novos registros para o arquivo de laboratório:

sudo sh -c 'journalctl -u ssh -f -n 0 >> /home/ssh_tempo_real.log'

Entendendo o comando

`journalctl` consulta o journal do sistema.

journalctl

`-u ssh` seleciona somente registros relacionados à unidade SSH:

-u ssh

`-f` acompanha novos registros:

-f

`-n 0` determina que não queremos inicialmente as últimas linhas já existentes. Queremos acompanhar somente **novos eventos**:

-n 0

O operador:

>>

adiciona informações ao final do arquivo sem apagar seu conteúdo anterior.

Finalmente:

sudo sh -c '...'

faz com que o comando e o redirecionamento sejam executados com privilégios administrativos.

Isso é necessário porque simplesmente executar:

sudo journalctl -u ssh -f >> /home/ssh_tempo_real.log

pode resultar em:

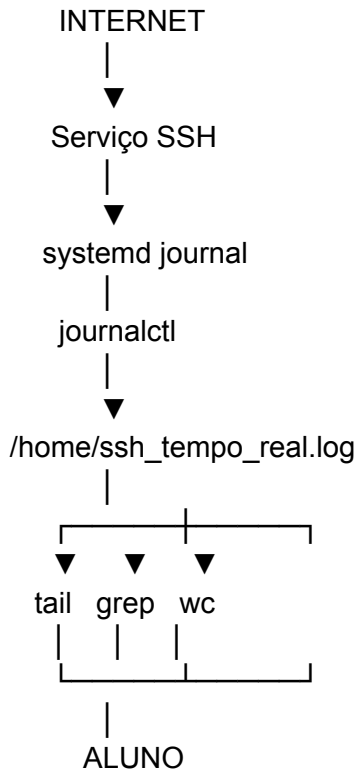
Permission denied

O motivo é que o `sudo` seria aplicado ao `journalctl`, mas o redirecionamento `>>` ainda seria realizado pelo shell do usuário atual.

---

6. Fluxo do laboratório

O funcionamento pode ser representado assim:



PARTE 5 — COMANDOS PARA ANÁLISE

7. `cat` — Visualizar todo o arquivo

Execute:

cat /home/lab_ssh.log

O comando `cat` exibe todo o conteúdo do arquivo.

É indicado principalmente para arquivos pequenos.

---

8. `head` — Visualizar o início

head /home/lab_ssh.log

Por padrão, mostra as primeiras 10 linhas.

Para visualizar somente as primeiras 5:

head -n 5 /home/lab_ssh.log

Para as primeiras 20:

head -n 20 /home/lab_ssh.log

---

9. `tail` — Visualizar o final

tail /home/lab_ssh.log

Mostra as últimas 10 linhas.

Para mostrar as últimas 5:

tail -n 5 /home/lab_ssh.log

Para mostrar as últimas 20:

tail -n 20 /home/lab_ssh.log

Em logs, o `tail` é especialmente importante porque os eventos mais recentes normalmente são adicionados ao final do arquivo.

---

10. `tail -f` — Acompanhar em tempo real

Agora utilizaremos o arquivo real:

tail -f /home/ssh_tempo_real.log

O terminal permanecerá aguardando novos registros.

Quando o servidor receber um novo evento SSH, ele deverá aparecer na tela.

Para interromper:

CTRL + C

Essa é uma forma simples de realizar monitoramento em tempo real.

---

11. `grep` — Pesquisar informações

O `grep` procura linhas contendo determinado texto.

Para localizar falhas de autenticação:

grep "Failed password" /home/lab_ssh.log

O resultado apresentará somente registros contendo:

Failed password

---

12. Localizando usuários inexistentes

Execute:

grep "Invalid user" /home/lab_ssh.log

Poderão aparecer registros relacionados a usuários como:

admin
oracle
ubuntu
postgres

Isso indica tentativas de autenticação utilizando nomes de usuários que não existem no servidor.

---

13. Tentativas contra `root`

Execute:

grep "root" /home/lab_ssh.log

Para ser mais específico:

grep "Failed password for root" /home/lab_ssh.log

Agora estamos procurando especificamente tentativas de autenticação malsucedidas para `root`.

---

14. Logins bem-sucedidos

Execute:

grep "Accepted password" /home/lab_ssh.log

`Accepted password` indica que uma autenticação foi aceita.

Por exemplo:

Accepted password for professor...

Em uma análise real, devemos verificar:

- qual usuário entrou;
- de qual IP;
- em qual horário;
- se aquele acesso era esperado.

---

15. `grep -c` — Contando ocorrências

Podemos contar as falhas:

grep -c "Failed password" /home/lab_ssh.log

Contar usuários inválidos:

grep -c "Invalid user" /home/lab_ssh.log

E contar autenticações aceitas:

grep -c "Accepted password" /home/lab_ssh.log

A opção `-c` significa **count**, ou contagem.

---

16. Pesquisando um endereço IP

Podemos investigar um endereço específico:

grep "104.234.186.154" /home/lab_ssh.log

Para contar quantas linhas mencionam esse IP:

grep -c "104.234.186.154" /home/lab_ssh.log

Isso ajuda a acompanhar diferentes ações relacionadas ao mesmo endereço.

---

17. `wc -l` — Contando linhas

Para descobrir quantas linhas existem:

wc -l /home/lab_ssh.log

A opção `-l` significa **lines**.

---

18. Utilizando o pipe `|`

O pipe permite utilizar a saída de um comando como entrada de outro.

Exemplo:

grep "Failed password" /home/lab_ssh.log | wc -l

Primeiro:

grep "Failed password"

localiza as falhas.

Depois:

|

envia o resultado para:

wc -l

que conta as linhas.

Fluxo:

---

19. Combinando `tail` e `grep`

Podemos analisar somente os registros mais recentes:

tail -n 20 /home/lab_ssh.log | grep "Failed password"

O comando:

1. seleciona as últimas 20 linhas;
2. envia essas linhas através do pipe;
3. procura `Failed password`;
4. apresenta somente as falhas encontradas.

---

20. `grep -i` — Ignorando maiúsculas e minúsculas

Execute:

grep -i "ROOT" /home/lab_ssh.log

A opção:

-i

faz com que `grep` ignore diferenças entre letras maiúsculas e minúsculas.

Assim, a pesquisa poderá encontrar:

root
ROOT
Root

PARTE 6 — MONITORAMENTO REAL

21. Observando o servidor

Com a captura sendo executada pelo professor, o aluno deverá abrir:

tail -f /home/ssh_tempo_real.log

Observe os novos eventos.

Se aparecer algo como:

Invalid user ahmed from 104.234.186.154

significa que houve uma tentativa de autenticação utilizando um usuário inexistente.

Se aparecer:

Failed password for root from 104.234.186.154

houve uma tentativa malsucedida de autenticação como `root`.

Se aparecer:

Accepted password for usuario...

houve uma autenticação aceita.

ATIVIDADES PRÁTICAS

Atividade 1 — Conhecendo o arquivo

Utilizando `/home/lab_ssh.log`, descubra:

1. Quantas linhas existem no arquivo?
2. Quais são as 5 primeiras linhas?
3. Quais são as 5 últimas linhas?
4. Qual é o primeiro horário registrado?
5. Qual é o último horário registrado?

Atividade 2 — Investigando falhas

Descubra:

1. Quantos registros possuem `Failed password`?
2. Quantos registros possuem `Invalid user`?
3. Quantas linhas mencionam `root`?
4. Quais usuários inexistentes foram utilizados?

Anote os comandos utilizados.

Atividade 3 — Investigando acessos válidos

Pesquise:

Accepted password

Responda:

1. Quais usuários conseguiram autenticar?
2. Quantas autenticações aceitas existem?
3. O usuário `professor` aparece?
4. Quantas autenticações aceitas existem para `professor`?

Atividade 4 — Investigação de um IP

Investigue:

104.234.186.154

Responda:

1. Quantas vezes esse IP aparece?
2. Quais usuários foram utilizados por ele?
3. Houve tentativa contra `root`?
4. Houve tentativa com usuário inexistente?
5. Existe algum `Accepted password` associado a esse IP?
6. Existe evidência no arquivo de que esse IP conseguiu autenticar?

Atividade 5 — Monitoramento em tempo real

Execute:

tail -f /home/ssh_tempo_real.log

Observe o arquivo durante alguns minutos.

Quando surgir um novo evento, anote:

Horário:
IP:
Usuário:
Tipo de evento:
Resultado:

Depois classifique o evento como:

[ ] Login aceito
[ ] Senha incorreta
[ ] Usuário inexistente
[ ] Conexão encerrada
[ ] Outro

DESAFIO — ANALISTA DE SEGURANÇA

Você foi designado como responsável por analisar possíveis tentativas de acesso ao servidor.

Descubra:

> **Qual endereço IP apresenta o maior número de tentativas malsucedidas de autenticação no arquivo `lab_ssh.log`?**

Comece com:

grep "Failed password" /home/lab_ssh.log

Escolha um IP suspeito e filtre:

grep "Failed password" /home/lab_ssh.log | grep "45.148.10.92"

Depois conte:

grep "Failed password" /home/lab_ssh.log | grep -c "45.148.10.92"

Repita a investigação com outros IPs e compare.

RELATÓRIO FINAL

Ao concluir a investigação, preencha:

RELATÓRIO DE ANÁLISE DE LOG

Aluno:
Data:

IP investigado:

Quantidade de falhas:

Usuários utilizados:

Tentou autenticar como root?
( ) Sim
( ) Não

Utilizou usuários inexistentes?
( ) Sim
( ) Não

Foi encontrado login aceito?
( ) Sim
( ) Não

Comandos utilizados:

Conclusão da investigação:

RESUMO DOS COMANDOS

Comando	Função
`cat arquivo`	Exibe todo o arquivo
`head arquivo`	Mostra as primeiras linhas
`head -n 5 arquivo`	Mostra as primeiras 5 linhas
`tail arquivo`	Mostra as últimas linhas
`tail -n 5 arquivo`	Mostra as últimas 5 linhas
`tail -f arquivo`	Acompanha novas linhas

`grep "texto" arquivo`	Pesquisa um texto
`grep -i "texto" arquivo`	Ignora maiúsculas/minúsculas
`grep -c "texto" arquivo`	Conta linhas encontradas
`wc -l arquivo`	Conta linhas
`|`	Conecta dois comandos

Conclusão

A análise de logs é uma atividade importante na administração de servidores Linux.

Com comandos relativamente simples, como:

tail
grep
wc

é possível localizar falhas de autenticação, usuários inexistentes, endereços IP e outros eventos importantes.

Neste laboratório trabalhamos com dois cenários:

Arquivo controlado:

/home/lab_ssh.log

utilizado para exercícios.

Arquivo em tempo real:

/home/ssh_tempo_real.log

alimentado a partir dos registros reais do SSH.

Dessa forma, o aluno começa investigando um cenário preparado pelo professor e depois aplica os mesmos conhecimentos na observação de eventos reais de um servidor Debian.

Essa versão já separa bem **preparação do professor**, **prática do aluno**, **arquivo fictício** e **monitoramento real**, o que facilita bastante aplicar o módulo em laboratório.