30 ferramentas que todo sysadmin Linux deve conhecer | Linux Descomplicado

by Ricardo Ferreira · 23 de setembro de 2015

12-16 minutos

Os administradores de sistema (*sysadmins*) são responsáveis, rotineiramente, por operações de sistemas e serviços de produção. Um dos papéis críticos é assegurar que os serviços de rede estejam disponíveis durante todos os dias e toda hora (24/7). Para isso, é preciso que haja planejamento, estratégias de gestão, manutenções programadas, auditorias de segurança, entre outras técnicas. Além disso, como qualquer outra profissão, os sysadmins possuem suas ferramentas de trabalho. Portanto, a fim de que esse profissional possa utilizar ferramentas adequadas no momento certo e que possa manter disponível todos os sistemas que administra, segue uma lista com 30 ferramentas categorizadas da seguinte forma: rede, segurança, armazenamento, registros (logs), rotinas de backup, produtividade e monitoramento do sistema.

Tenha agilidade e padronização na entrega dos serviços de TI. E, assim, ganhe tempo e seja eficiente na entrega desses serviços. **Saiba como**

RECOMENDO QUE LEIA:

10 comandos que todo usuário Linux deve saber
20 comandos Linux que você talvez não conheça

Ferramentas de Rede

1 – PING

Comando que verifica a conectividade fim-a-fim e outros parâmetros, como: jitter e perda de pacotes. Útil para verificar a disponibilidade de algum ativo de rede.

Exemplo:

$ ping [opções] [host/IP de destino]

2 – TRACEROUTE

Comando que mostra o caminho percorrido por um pacote para chegar ao seu destino. Este comando mostra na tela o caminho percorrido entre os Gateways da rede e o tempo gasto de retransmissão. É útil para encontrar ativos defeituosos na rede caso o pacote não esteja chegando ao seu destino.

Exemplo:

$ traceroute [opções] [host/IP de destino]

3 – MTR

O comando MTR combina a funcionalidade dos teste de ping e traceroute em uma única ferramenta de diagnóstico. A máquina de origem deverá possuir o recurso MTR instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

Exemplo:

$ mtr [opções] [host/IP de destino]

4 – NETCAT

O Netcat é uma ferramenta de rede que permite abrir portas TCP/UDP. Permite forçar conexões UDP/TCP. Util para realizar rastreamento de portas ou realizar transferências de arquivos bit a bit entre os equipamentos. Um verdadeiro canivete suíço de rede TCP/IP. Útil para solucionar problemas de políticas de firewall e disponibilidade do serviço.

Exemplo:

$ nc [opções] [host/IP de destino]

5 – SS (obsoleto NETSTAT)

Comando que mostra conexões de rede, tabela de roteamento, estatísticas de interfaces, conexões

masquerade, e mensagens. Em resumo, é um utilitário de estatísticas de rede que pode mostrar informações de status e estatísticas sobre conexões abertas de rede (TCP / UDP, endereços IP), tabelas de roteamento, o tráfego TX / RX e protocolos. Útil para o diagnóstico relacionado com a rede e ajuste de desempenho.

Exemplo:

$ ss [opções]

6 – TCPDUMP

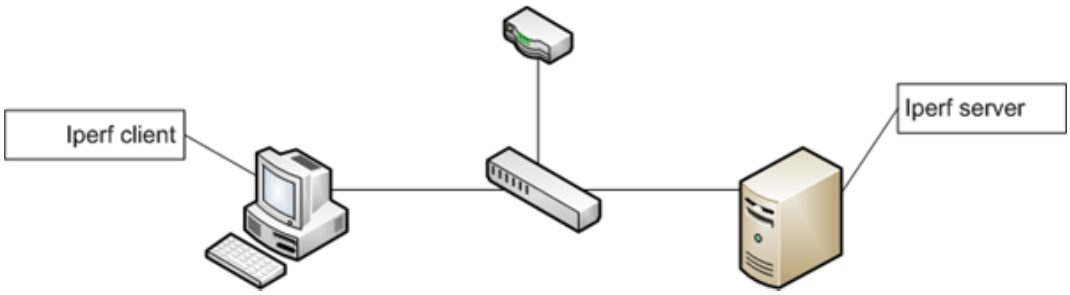
Utilitário que monitora a conexão TCP/IP. O monitoramento é feito especificando a interface desejada. A saída do comando é o trafego de pacotes enviados e recebido juntamente com endereços de origem e destino. Útil para monitorar todo tráfego que entra e sai da placa de rede (sniffer). A máquina de origem deverá possuir o recurso TCPDUMP instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

http://www.tcpdump.org/

7 – IPERF

O Iperf consiste em um software de análise de performance de banda e cálculo de perda de datagramas na rede, cria fluxo de dados sob TCP e UDP e permite medir e analisar o desempenho da rede. Ele trabalha no modo cliente/servidor. Útil para testar/medir o *throughput* da rede, pois serve como ferramenta de simulação de

conectividade, diagnóstico da situação do cabeamento, entre outros.



https://iperf.fr/

Exemplo:

Servidor

$ iperf -s

Cliente

$ iperf -c [host/IP servidor]

$ nc [opções] [host/IP de destino]

8 – IP (obsoelto IFCONFIG)

Com o próposito de aproveitar ao máximo o novo subsistema de rede criado, em detrimento de comandos obsoletos (“arp”, “ifconfig” e “route”), uma nova ferramenta foi desenvolvida, “o comando ip”. A principal diferença desse comando é que além de reunir todas as funções dos comandos obsoletos, ele ainda nos oferece muitas outras. Em resumo, a funcionalidade fornecida na nova suíte iproute2, simplificou e condensou estas ferramentas no novo “comando ip”.

Exemplo:

$ ip neigh show
$ ip addr show
$ ip route show

FERRAMENTAS de Segurança

9 – IPTABLES

O Iptables é uma ferramenta para criar e administrar regras e assim filtrar pacotes de redes (firewall). Pode funcionar baseado no endereço, porta de origem, destino do pacote, prioridade. Ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar. Em firewalls mais restritivos, o pacote é bloqueado e registrado para que o administrador do sistema tenha conhecimento sobre o que está acontecendo em seu sistema.

http://www.netfilter.org/projects/iptables/

Exemplo:

$ iptables [-t table] <opção> chain regra-especifica

10 – NMAP

O comando nmap é um scanner de portas, ou seja, ele mostra quais portas estão abertas no ativo de rede. A máquina de origem deverá possuir o recurso NMAP instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

Exemplo:

$ nmap [opções] [host/IP destino]

11 – LYNIS

Lynis é uma ferramenta de auditoria. Ela examina o sistema e os programas disponíveis para detectar problemas de segurança. Além da informação relacionada com segurança, também mostrará informações gerais sobre pacotes instalados e erros de configuração. A máquina de origem deverá possuir o recurso LYNIS instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

https://cisofy.com/lynis/

Exemplo:

$ lynis –check-all

12 – RKHUNTER

O RKhunter é uma excelente ferramenta para detectar trojans, rootkits e outros possíveis problemas de segurança em servidores linux. A máquina de origem deverá possuir o recurso RKHUNTER instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

http://rkhunter.sourceforge.net/

Exemplo:

$ rkhunter -c

13 – CRYPTSETUP

Usado para criar e gerenciar partições de disco criptografados.

Exemplo:

$ cryptsetup –verbose –verify-passphrase luksFormat /dev/sdbX

FERRAMENTAS de armazenamento

14 – DF

Mostra o espaço livre/ocupado de cada partição.

Exemplo:

$ df [opções]

15 – DU

Mostra o espaço ocupado por arquivos e sub-diretórios do diretório atual.

Exemplo:

$ du [opções]

RECOMENDO QUE LEIA:

Administrando múltiplos terminais virtuais usando a ferramenta screen
Saiba como aprender 12 comandos Linux em apenas alguns minutos

16 – MOUNT

Comando usado para montar unidades de armazenamento no sistema de arquivos.

Exemplo:

$ mount [opções] [dispositivo-ponto_de_montagem]

17 – LVM

Um conjunto de ferramentas via terminal para gerenciamento de grupos de volumes e volumes físicos/lógicos, que permite criar, redimensionar, dividir e mesclar volumes em cima de vários discos físicos com o tempo de inatividade mínimo.

18 – FDISK

É o utilitário que realiza particionamento de discos rígidos. É considerado uma das melhores ferramentas para gerenciar partições no HD.

Exemplo:

$ fdisk [opções]

19 – FSCK

É uma ferramenta usada para verificar a consistência de um sistema de arquivos do sistema.

Exemplo:

$ fsck [opções] [sistema-arquivos]

FERRAMENTAS de registros (logs)

20 – TAIL

Mostra as linhas finais de um arquivo texto. Usado para monitorar a evolução de um arquivo de log.

Exemplo:

$ tail [opções]

21 – LOGROTATE

A ferramenta *logrotate* tem como objetivo rotacionar automaticamente logs de aplicativos segundo a necessidade e a organização que o administrador de sistemas (SysAdmin) deseje. Ela pode dividir, comprimir grandes arquivos de log em um intervalo de tempo pré-definido. Útil para administração de serviços que podem produzir um grande volume de arquivos de log. A máquina de origem deverá possuir o recurso LOGROTATE instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

22 – GREP

Comando que serve para procurar por um texto dentro de um arquivo(s) ou no dispositivo de entrada padrão. Permite uso de expressões regulares.

Exemplo:

$ grep [expressão] [arquivo] [opções]

23 – AWK

AWK é uma linguagem utilizada para processamento de informações em texto, como o conteúdo de um arquivo – principalmente informações em colunas – ou a saída de outros comandos, como cat, grep etc.

Exemplo:

$ awk '/^UUID/' /etc/fstab

24 – SED

A ferramenta SED, junto ao AWK, são as duas principais linguagens para manipulação de arquivos e streams do Unix/Linux. Ambas possuem vasta abrangência e o que uma não pode fazer, a outra provavelmente o fará. Com o SED é possível substituir e "casar" padrões, sempre por meio de Expressões Regulares. O SED, assim como o AWK, lê um arquivo, linha por linha, e aplica a expressão do parâmetro a cada uma delas.

Exemplo:

$ sed [expressão] [arquivo]

FERRAMENTAS de backup

24 – RSYNC

Rsync é o comando utilizado para copiar e sincronizar arquivos e diretórios remotamente. Com a ajuda do comando rsync, você pode copiar e sincronizar seus arquivos remotamente e localmente através de diretórios, em discos de rede, realizar backups de dados e espelhamento entre dois computadores com Linux. A máquina de origem deverá possuir o recurso RSYNC instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

https://rsync.samba.org/

Exemplo:

$ rsync [opções] [origem] [destino]

25 – DUPLICTY

Duplicity é uma ferramenta que oferece o método de backup incremental encriptando os dados a serem armazenados. Ele usa algoritmos da ferramenta rsync no método de sincronismos de dados, como o librsync; e o GnuPG para encriptar os dados. A máquina de origem deverá possuir o recurso DUPLICITY instalado, caso não possua, basta instalá-lo conforme link.

Exemplo:

$ duplicity [pasta-origem] scp://user@remote_site.com /[pasta-destino]

FERRAMENTAS de produtividade

26 – SCREEN

O Screen é um multiplexador de terminais que permite ao usuário, em uma mesma sessão, abrir várias janelas e realizar atividades paralelas. Ou seja, as janelas que ele cria estão dentro de uma mesma sessão e isso é muito útil. Por exemplo, numa única sessão remota via SSH, várias sessões virtuais poderam ser iniciadas em conjunto. A máquina de origem deverá possuir o recurso SCREEN instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

Exemplo:

$ screen

27 – APROPOS

Procura por programas/comandos através da descrição. É útil quando precisamos fazer alguma coisa mas não sabemos qual comando usar. Ele faz sua pesquisa nas páginas de manual existentes no sistema e lista os comandos/programas que atendem a consulta.

Exemplo:

$ apropos [descrição]

FERRAMENTAS de monitoramento

28 – NETHOGS

Nethogs é uma ferramenta de linha de comando do tipo

“top” para medir o consumo de banda. É uma ferramenta que mostra a largura de banda utilizada por processos individualmente e os classifica listando os mais usados (tráfego maior de dados). No caso de um pico na largura de banda, o nethogs detecta o processo responsável e identifica o PID, o usuário e o caminho do programa. A máquina de origem deverá possuir o recurso NETHOGS instalado, caso não possua, basta instalá-lo conforme sua distro Linux.

Exemplo:

$ nethogs

29 – IOTOP

É um utilitário open source baseado na ferramenta “top” que tem por finalidade monitorar leitura/escrita de disco e traçar exatamente quais os processos ou usuários que estão consumindo recursos.

Exemplo:

$ iotop [opções]

30 – VMSTAT

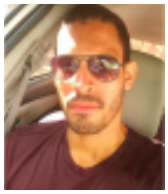
Ferramenta que mostra uma visualização em forma de tabela para processos, uso de RAM e paginação, I/O (entrada e saída de dados) e atividade da CPU

Exemplo:

$ vmstat [opções]

Mais Ferramentas aqui

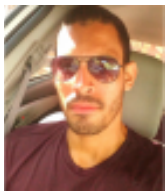
- Sobre
- Últimos Posts



Ricardo Ferreira

Fundador do Linux Descomplicado - LD.

Sempre em busca de novos conhecimentos, preza por conteúdo de qualidade e auto-explicativo. Por isso, persiste em criar um site com artigos relevantes para todos os leitores do Linux Descomplicado!



Últimos posts por Ricardo Ferreira (exibir todos)

Comentários

comentários