

Ferramenta de Gerenciamento de Redes e Computadores Baseado em Shell Script

Ed Carlos P. da Silva, Thiago Alefy A. Sousa, Thalia K. Sampaio Gurgel.

Departamento de Informática – FANAT - Universidade do Estado do Rio Grande do Norte - UERN

Rua Professor Antônio Campos, s/n, BR 110, km 48, Bairro Costa e Silva – Mossoró/RN - Brazil 59600-000

{edcarlospessoa, taas.sousa, thaliasampaio8}@gmail.com

Abstract. Managing a network and its assets using software allows the detection and correction of problems that are occurring or are likely to occur, isolating their cause Comer, (2001). Based on this principle the investment is justified to the extent that the administrator can monitor all the complexity and heterogeneity of the network in which he is responsible.

Thinking about this and considering the complexity of the existing tools, this work proposes a simple, easy to use and adaptive tool for Linux environment based on debian distributions and their derivatives.

Keywords: Network management, shell script, performance.

Resumo. Gerenciar uma rede e seus ativos usando software permite a detecção e correção de problemas que estejam ocorrendo ou que provavelmente vão ocorrer, isolando sua causa Comer, (2001). Partindo deste princípio o investimento se justifica na medida em que o administrador consegue monitorar toda a complexidade e heterogeneidade da rede na qual é responsável.

Pensando assim e levando em consideração a complexidade das ferramentas existentes hoje em dia, este trabalho propõe uma ferramenta simples, de fácil utilização e adaptação para ambiente Linux baseados em distribuições debian e seus derivados.

Palavras-chaves: Gerenciamento de redes, shell script, desempenho.

1. Introdução

Existem atualmente diversas ferramentas como o Dry Ear 1.0 para a administração de redes e seus mais diversos ativos, acessos remotos, controle de usuários, entre outros, mas é um fato raro encontrarmos uma ferramenta de manuseio e adaptação simples para as variações possíveis e infundáveis das redes.

“O gerenciamento de redes de computadores se baseia nos conceitos de gerentes e agentes, o aumento da diversidade dos componentes das redes, vem tornando o gerenciamento de redes essencial” COR, (1996). Para uma boa saúde na rede se faz

necessário um bom gerenciamento, um sistema de gerenciamento é importante para detectar e prever falhas, monitorar o desempenho, planejar futuras expansões, evitando dessa forma o baixo desempenho da rede, travamento de equipamentos ou a queda nos serviços de rede, além de garantir um melhor funcionamento de equipamentos que são gerenciados (roteadores, switches) que vão ser indispensáveis para que a rede possa funcionar, atendendo às necessidades dos usuários em geral.

Esse projeto tem como objetivo apresentar uma ferramenta de gerenciamento de redes usando shell script e SSH visando facilitar o trabalho do gerente de rede que necessita de dados importantes sobre a rede na qual é responsável de forma rápida e precisa.

A ferramenta elaborada na disciplina de Redes de Computadores, especificamente com foco em gerenciamento de redes, considerando melhorar a obtenção de informações de sobre os diversos serviços e hardwares dos sistemas.

O principal objetivo é, de uma forma rápida e transparente, pegar o status de um objeto gerenciado, sendo inserido seu respectivo IP da máquina ou ativo de rede, e assim que a informação estiver pronta automaticamente ela retorna por meio de um arquivo txt. Citando por exemplo algumas informações que o sistema pode retornar, elas podem ser sobre informações sobre vídeo, estado das conexões usb, update e upgrade disponíveis, unidades montadas, tipo e uso de memória, temperatura, informações sobre os processos do sistema, resolução de telas, relatório completo da máquina, dados da rede, placa de som, processador, hardware, arquitetura, etc.

Algumas das ferramentas de monitoração de informações de redes e máquinas que servirão como referência para esse trabalho foram a MONIT e a MUNIN. Uma referência para a administração de redes e computadores usando shell script é o livro Administração de Redes com scripts: Bash Script, Python e VBScript Costa (2010).

Este artigo está organizado de forma que expomos os problemas que motivaram a criação da ferramenta na introdução, descrevemos o gerenciamento, falamos sobre as tecnologias utilizadas e como funciona a solução desenvolvida.

2- Gerenciamentos de Redes

Segundo Saydam (1996), o gerenciamento de rede inclui a disponibilização, a integração e a coordenação de elementos de hardware, software e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer as exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável. fazendo-se necessário que as informações de rede sejam retornadas rapidamente para que não cheguem ultrapassadas.

A necessidade de um gerenciamento de rede deu-se devido ao crescimento exponencial do número de usuários e de aplicações. Com isso, as redes se tornaram mais complexas e estratégicas, fazendo com que seus recursos e aplicações sejam indispensáveis à operação das empresas e instituições.

3- Tecnologias Utilizadas

Para a construção do modelo de gerenciamento de redes, foram utilizadas algumas ferramentas, dentre elas podemos destacar o SSH, que é um protocolo de rede criptográfico para operação de serviços de rede de forma segura sobre uma rede insegura. A melhor aplicação de exemplo conhecida é para login remoto a sistemas de computadores pelos usuários. O SSH fornece um canal seguro sobre uma rede insegura em uma arquitetura cliente-servidor, conectando uma aplicação cliente SSH com um servidor SSH. Aplicações comuns incluem login em linha de comando remoto e execução remota de comandos, mas qualquer serviço de rede pode ser protegido com SSH. A aplicação mais visível do protocolo é para acesso a contas shell em sistemas operacionais do tipo Unix NWG, (2006).

Outra tecnologia utilizada foi o Shell Script, poderosa ferramenta de automação de instruções UNIX. Com um arquivo de texto executável o usuário ou sistema é capaz de executar uma sequência de operações, instruções e testes. Usos mais convencionais e comuns são em executáveis de instalação/configuração e para geração de relatórios e análise destes. E finalizando a ferramenta com um programa desenvolvido em linguagem de programação C para a escolha das funções.

4- Modelo Implementado

Nesta demonstração foram usadas duas máquinas virtuais feitas no programa Virtual Machine Manager, ambas tendo o sistema operacional linux distribuição debian 8.6, usando um núcleo no processador intel Xeon E312 e quinhentos e doze mega de memória, como mostrado na Figura 1.

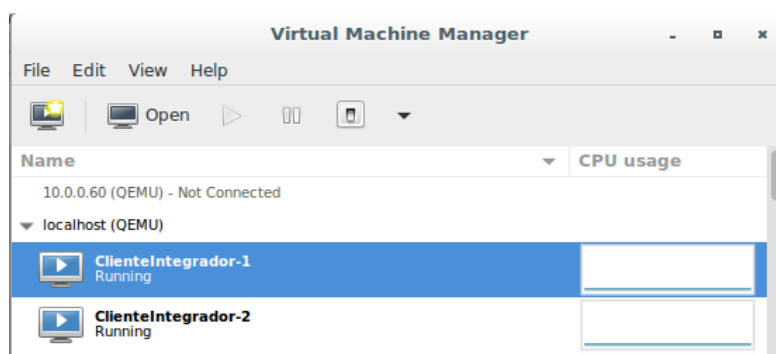


Figura 1. Máquinas virtuais criadas para a demonstração.

O programa é executado e são pedidas as informações relacionadas ao IP e o nome do usuário para termos acesso aos menus principais, conforme a Figura 2.

Essa comunicação é feita usando o protocolo SSH com chaves compartilhadas, sendo que o administrador ROOT do sistema tem acesso liberado a todas as máquinas.

```

luke@client2:~$ ./menu
Entre com o número do ip: 192.168.122.2
Entre com o usuário: root
1- Informações de maquina
2- Informações de rede
3- Menu ação
4- logs
5- Exit

```

Figura 2. O programa sendo executado.

Seguindo adiante foi selecionado o menu um que corresponde ao submenu de informações de máquina, onde existem diversas opções de escolha, conforme a Figura 3.

```

luke@client2:~$ ./menu
Entre com o número do ip: 192.168.122.2
Entre com o usuário: root
1- Informações de maquina
2- Informações de rede
3- Menu ação
4- logs
5- Exit
1
Entrou em informações de maquina!
1- Gerar relatório sistema
2- Gerar relatório data instalação
3- Gerar relatório processador
4- Gerar relatório memória RAM e SWAP
5- Gerar relatório de mídias
6- Gerar relatório de unidades montadas
7- Gerar relatório de HARDWARE
8- Gerar relatório de USB's
9- Gerar relatório de arquitetura e SO
10- Gerar relatório com Hostname
11- Gerar relatório driver vídeo
12- Gerar relatório placa de som
13- Gerar relatório dispositivos de gravação
14- Gerar relatório de Audio
15- Gerar relatório de idiomas do sistema
16- Gerar relatório completo
17- Gerar relatório temperatura
18- Voltar menu anterior

```

Figura 3. Menu de escolha.

Na Figura 4, supondo que preciso instalar uma versão correta de um programa, e para isso teria que saber detalhes em relação ao processador, foi escolhida a opção três, que se refere a geração de relatório sobre a arquitetura do processador. O programa avisa que a tarefa foi concluída e volta ao menu anterior.

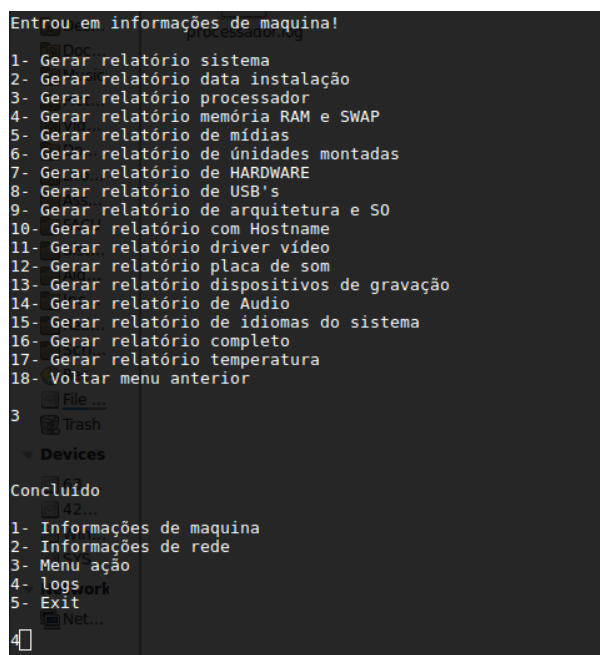


Figura 4. Escolha de função.

O passo seguinte demonstra que o programa envia o arquivo de log que contém as informações solicitadas para o centro de controle, deixando uma cópia do arquivo armazenada na própria máquina, conforme a Figura 5.

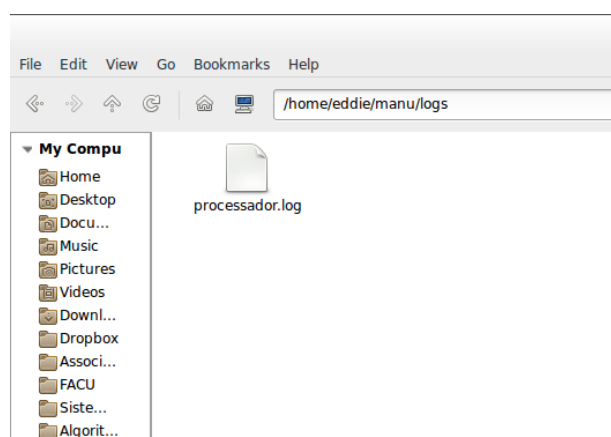


Figura 5. Arquivo de log.

O arquivo de log que é apresentado na Figura 6, contém as informações sobre a arquitetura do processador juntamente com a data e horário da consulta.

```

1 dryear 1.0
2
3 *** processador ***
4
5 processor      : 0
6 vendor_id     : GenuineIntel
7 cpu family    : 6
8 model        : 42
9 model name    : Intel Xeon E312xx (Sandy Bridge)
10 stepping     : 1
11 microcode    : 0x1
12 cpu MHz      : 2194.918
13 cache size   : 4096 KB
14 physical id  : 0
15 siblings     : 1
16 core id      : 0
17 cpu cores    : 1
18 apicid       : 0
19 initial apicid : 0
20 fpu          : yes
21 fpu_exception : yes
22 cpuid level  : 13
23 wp          : yes
24 flags        : fpu de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc
                rep_good nopl eagerfpu pni pclmulqdq sse3 cx16 sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx hypervisor lahf_lm xsaveopt
25 bogomips     : 4389.83
26 clflush size : 64
27 cache alignment : 64
28 address sizes : 40 bits physical, 48 bits virtual
29 power management:
30
31
32 =====
33 Relatório criado em:
34 Sáb Mar 4 21:33:33 BRT 2017
35 dryear 1.0
36
37
38
Saving file '/home/eddie/manu/logs/processador.log'... Plain Text Tab Width: 8 Ln 39, Col 1 INS

```

Figura 6. As informações do arquivo de log.

O código em Shell script usado na obtenção e envio das informações é demonstrado na Figura 7.

```

root@client2:~# cat processador.sh
echo "dryear 1.0" >> processador.log
echo >> processador.log
echo
echo "*** processador ***" >> processador.log
echo >> processador.log
head -n 29 /proc/cpuinfo >> processador.log
echo >> processador.log
echo ===== >> processador.log
echo Home
echo Desktop
echo "Relatório criado em:" >> processador.log
date >> processador.log
SSH_AUTH_SOCK=0 scp *.log root@192.168.122.1:/home/noc/manu/logs/
echo Concluído
root@client2:~#

```

Figura 7. Código desenvolvido em shell script.

Na Figura 8, supondo que estou tendo problemas de duplicação de ip em uma rede e preciso fixar o ip de alguns dispositivos, seleciono a opção 1 que lida com o cachê ARP do sistema, o que me possibilita na Figura 10 visualizar os dados de identificação física MAC (Media Access Control) das placas de rede.



Figura 8. Demonstrando uma captura de informações sobre a rede.

Conforme ocorreu na Figura 5, o comando envia o arquivo de log, usando o protocolo SSH com chave compartilhada, do objeto gerenciado para o centro de controle demonstrado na Figura 9.

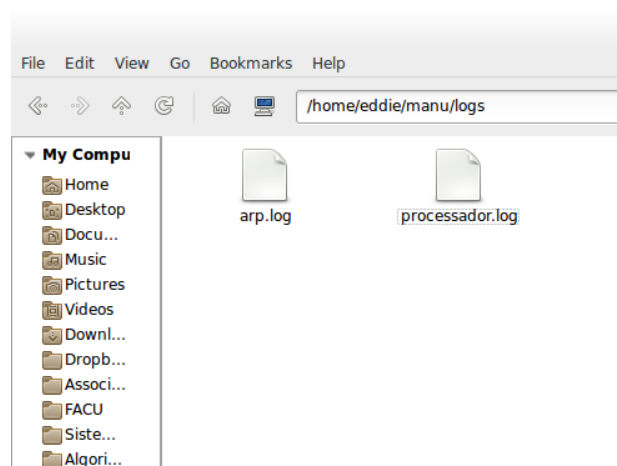


Figura 9. O arquivo de log é enviado para o centro de controle.

A Figura 10 revela os dados obtidos, juntamente com a data e hora.

```

1 dryear 1.0
2
3 *** Cache ARP ***
4
5 ? (192.168.122.3) em 52:54:00:3f:a0:92 [ether] em eth0
6 ? (192.168.122.1) em fe:54:00:3b:d7:66 [ether] em eth0
7
8 =====
9 Relatório criado em:
10 Dom Mar 5 09:42:12 BRT 2017

```

Figura 10. Relatório ARP

Parte do código do menu que é executado na máquina do gerente da rede é demonstrada na Figura 11. Foi usada a linguagem C desenvolver a escolha e conexão do sistema de gerenciamento, ela possui todos os menus e submenus das funções disponíveis.

```

#include <stdio.h>
#include <locale.h>
char ip[16];
char User[50];
char porta[6];
char var[10000];
char op[2];
main()
{
    setlocale(LC_ALL, "Portuguese");
    conexao();
    menu();
}
int conexao(){
    printf("Entre com o número do ip: ");
    scanf("%s",ip);
    printf("Entre com o usuário: ");
    scanf("%s",User);
}
int menu()
{
    int tipo1;
    printf("\n1- Informações de maquina\n2- Informações de rede\n3- Menu ação\n4- logs\n5- Exit\n");
    scanf("%d", &tipo1);
    switch(tipo1)
    {
        case 1: printf("Entrou em informacoes de maquina!\n"); submenu1(); break;
        case 2: printf("Entrou em Informações rede!\n"); submenu2(); break;
        case 3: printf("Entrou em menu ação!\n"); submenu3(); break;
        case 4: system("nemo /home/noc/manu/logs/"); break;
        case 5: exit (0);
        default: printf("Introduza o valor correcto respectivo ao menu que quer seleccionar!\n"); menu();
    }
}

```

Figura 11. Parte do código em linguagem C.

5- Resultados Obtidos

Os ganhos com a agilidade do processo de descoberta de falhas, instabilidades ou obtenção de informações diversas é perceptível. Pelo fato da ferramenta ser tão simples, pois usa linguagem C em relação ao menu de escolha e shell script nas estações (que agem como agentes do sistema) executam os comandos, enviam um arquivo .log para o centro de operações, guardando uma cópia nele mesmo.

Usando ferramentas open source estamos diversificando as funções para abranger os mais diversos tipo de ocorrências, seguindo a lógica aqui demonstrada.

Como projeto futuro pretendemos desenvolver uma interface gráfica mais amigável, para mostrar graficamente alguns resultados usando o software Grafana [GRAF]. e a integração do sistema com os protocolos do SNMP conforme as RFCs 3411 e 3418. [RFC].

6- Referências Bibliográficas

COMER, Douglas E. (2001), Redes de Computadores e Internet. 2a ed. EDITORA Bookman.

COR (1996) Corporate Communications, Network Management, Virtual Reality for System and Network Management.

[RFC] https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol.
acessado em 5/3/2017.

[GRAF] <http://grafana.org/> Acessado em 5/3/2017.

NWG (2006) Network Working Group of the IETF, January 2006, RFC 4251, The Secure Shell (SSH) Protocol Architecture.

MONIT <http://mmonit.com/monit/>. Acessado em 17/3/2017.

MUNIN <http://munin-monitoring.org/> Acessado em 17/3/2017.

COSTA, Daniel Gouveia (2010) Administração de Redes com scripts: Bash Script, Python e VBScript – 2º ed. Rio de Janeiro - Brasport