

# **AUTOMAÇÃO DE REDES HIERÁRQUICAS CISCO (Sinapse):** Agilidade nos processos de detecção e correção de falhas em switches e roteadores utilizando *SysLog* e *NetFlow*.

**Caio Beraldo<sup>1</sup>**

<sup>1</sup>Fatec – Faculdade de Tecnologia de Garça  
Garça – SP

caio.beraldo@fatec.sp.gov.br

**Resumo.** *O objetivo deste projeto é desenvolver uma ferramenta que auxilie na detecção e correção de falhas em uma rede hierárquica de dispositivos Cisco. Serão utilizados os protocolos Syslog e Netflow que são protocolos de troca de mensagem utilizados para notificações de eventos e estatísticas de rede. O framework foi desenvolvido em Python, uma linguagem open source (de código fonte aberto) com grande portabilidade (para Windows, Unix/Linux, Mac OS, Android, iOS etc.) além de ser uma linguagem interativa (que não precisa ser pré-compilada). É utilizada principalmente para desenvolvimento de aplicativos de rede, scripting, teste de software e aplicações científicas. O intuito do software é trazer agilidade nos processos corretivos e preventivos para as empresas provedoras de serviços de internet e um melhor gerenciamento da QoS.*

**Palavras-Chave:** *Redes de computadores. Automação de redes. Acesso remoto. Agilidade. Resiliência. Provedores de internet.*

**Abstract.** *This project objective is to develop a tool that assists on the detection and correction of failures on hierarchical networks that uses Cisco devices. The application will use the Syslog and Netflow protocols, that are message exchange protocols used to send events notification and network statistics. The framework were developed using Python, a wide portability open source language (for Windows, Unix/Linux, Mac OS, Android, iOS etc.). Python is mainly used to develop network applications, scripting, software testing and scientific applications. The software intention is to fetch agility in the corrective and preventive processes and to provide a better management of QoS (Quality of Service) to companies that provide services and internet.*

**Key Words:** *Computer networks. Network automation. Remote access. Agility. Resilience. Network providers.*

## **1. INTRODUÇÃO**

Se tratando de provedores de internet, um dos grandes problemas que se pode observar é o tempo de detecção e correção de falhas na estrutura da rede. Em muitos casos, vê-se que os problemas são detectados apenas quando relatados pelos clientes ou após um tempo razoável

desde que ocorreu. Para auxiliar nesse e outros problemas vários projetos de rede foram desenvolvidos, entre eles o Projeto de Rede Hierárquica (PRH).

O PRH se baseia na divisão da rede em 3 camadas distintas sendo que cada uma delas irá fornecer funções específicas. Com isso o engenheiro de rede consegue separar e selecionar hardware, software e recursos específicos para dispositivos de cada camada. As três camadas do PRH são:

- **Camada de Acesso (CA)** – É responsável por fornecer acesso dos hosts (dispositivos finais como computadores, *smartphones*, *tablets*, *notebooks* etc.) à rede.
- **Camada de Distribuição (CD)** - Controla o limite entre CA e CN e faz restrição de acesso baseada em políticas para reduzir o tráfego na rede (ACLs, VLANs etc.).
- **Camada de Núcleo (CN)** - Fornece rápido acesso à CD.

Esse modelo de projeto foi desenvolvido que seja possível isolar os erros com a divisão da rede em camadas e trabalhar apenas nestas, facilitando o gerenciamento da rede. Para que o PRH seja implementado de maneira bem-sucedida há quatro princípios da engenharia estruturada devem ser seguidos:

- **Hierarquia** – Estrutura de alto nível que divide a rede em áreas menores e mais gerenciáveis com intuito de facilitar a solução dos problemas, isolando-os nestas áreas.
- **Modularidade** – Separação de funções existentes em uma rede (como bloco de serviços, data center, borda da internet etc.), facilitando o projeto de rede (PR) e sua manutenção.
- **Resiliência** – A rede deve se manter disponível para uso em qualquer condição, seja normal (tráfego esperado, eventos programados entre outros) ou anormal (falhas tanto de hardware como de software, tráfego incomum etc.).
- **Flexibilidade** – Flexibilidade de modificar a rede, adicionando novos serviços e recursos ou aumentando a capacidade de tráfego sem que haja grande interferência na rede (como troca de dispositivos de hardware importantes ou a necessidade de parar uma grande parte da rede para realizar a manutenção).

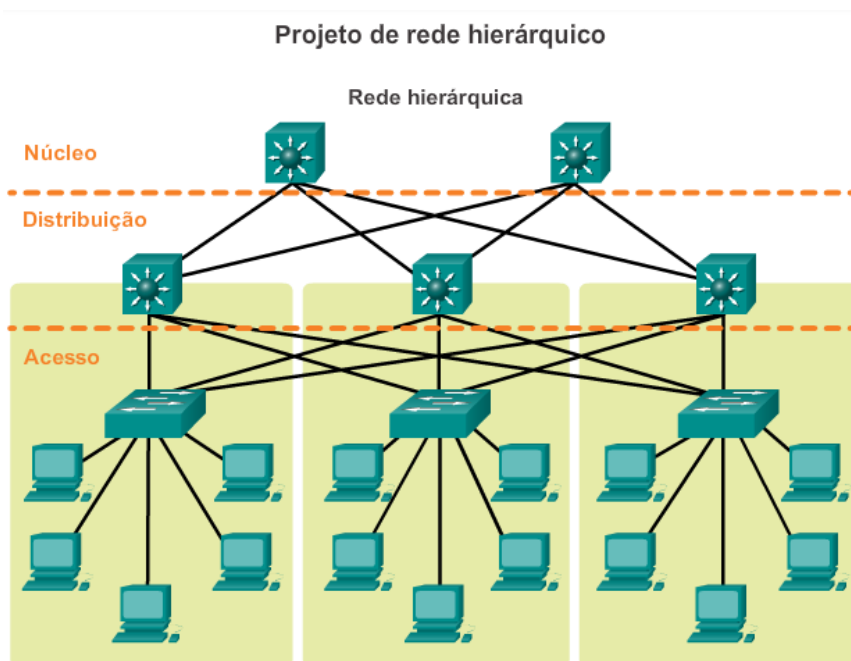


Figura 1 Exemplo e definição de um projeto de rede hierárquica.

Contudo, tratando-se do terceiro princípio da engenharia estruturada em uma rede hierárquica, a Resiliência, utiliza todos seus recursos para contornar as falhas que possam

acontecer tornando-as imperceptíveis. Mas estes recursos são limitados e quando acabam, as falhas vêm à tona. O administrador da rede deve passar pelo mesmo processo de verificação e correção para todos os erros que foram contornados, para tornar a rede funcional novamente. E isso pode causar consequências pois o administrador da rede deve começar a verificação dos erros do local mais provável que estes tenham ocorrido, podendo deixar de corrigir os outros locais onde ocorreram as falhas.

Por esse motivo é de extrema importância que o administrador da rede esteja sempre observando a rede e procurando falhas e problemas que ocorreram ou que possam ocorrer e estar sempre preparado para corrigi-los o mais rápido possível.

## **2. OBJETIVOS**

### **2.1. GERAIS**

Com isso é esperado que os processos de manutenção e análise se tornem mais ágeis e menos desgastantes para os administradores de rede removendo a necessidade de uma constante observação desta e dos eventos que ocorrem. E também trazer um melhor controle sobre a rede tanto física quanto logicamente. Principalmente, evitar o acúmulo de falhas na rede causados pela resiliência desta.

### **2.2. ESPECÍFICOS**

Os objetivos específicos para este projeto são:

- Analisar e buscar soluções para falhas que ocorram na rede para auxiliar nos processos de correção.
- Automatizar os processos de correção e detecção de falhas em uma rede hierárquica ocorridas cujas soluções forem encontradas pelo software.
- Gerar relatórios de utilização de links para auxiliar na melhoria de desempenho da rede.
- Notificar o administrador de rede dos eventos que ocorrem na rede como falhas, utilização de links e correções realizadas.
- Este projeto visa redes cujos protocolos *SysLog* e *NetFlow* estejam implementados.

## **3. PROCEDIMENTOS METODOLÓGICOS**

Para a realização deste projeto será necessária a pesquisa processos de engenharia de uma rede. Para isso serão analisados softwares que auxiliam nesses processos e empresas que atuam na área. Isto será feito juntamente à pesquisa bibliográfica que envolverá pesquisas aprofundadas sobre logics de rede (configurações, padrões, teorias, vulnerabilidades etc.) em livros, e-books e cursos online.

### **3.1. PYTHON**

O software será desenvolvido em Python, uma linguagem de programação com uma vasta documentação e comunidade. Ele possui uma sintaxe de programação limpa com um código fácil de ser lido. Além de ser *Open Source* (código fonte aberto) e de grande portabilidade (para Windows, Unix/Linux, Mac OS, Android, iOS etc.). Python também é uma linguagem interativa (que não precisa ser pré-compilada) e permite que o código seja agrupado em módulos e pacotes, facilitando e agilizando o processo de desenvolvimento. Ele é utilizado para

desenvolvimento Web, aplicativos de rede, *scripting*, teste de *software*, interfaces gráficas e aplicações científicas.

### **3.2. MYSQL**

Para a persistência de dados, seguindo a linha de portabilidade, será utilizada a base de dados relacional MySQL, trabalhando com a ferramenta MySQL Workbench Community (versão *free* do MySQL Workbench) que pode ser executado em Windows, Linux e OS X.

### **3.3. KIVY**

Visando manter a portabilidade do Python, o software foi desenvolvido com o Kivy, um pacote (módulo) Python de interface gráfica (como Tkinter e outros). O Kivy é um módulo multiplataforma e totalmente *free* (sobre a licença do MIT) que utiliza um mecanismo de interface construído sobre o OpenGL ES 2, usando linhas de interface rápidas e modernas.

### **3.4. VISUAL STUDIO**

O framework utilizado para o desenvolvimento do software foi o Visual Studio 2015 Community (VS2015), uma ferramenta de desenvolvimento da Microsoft que vem crescendo muito e agregando cada vez mais novas linguagens de programação e plataformas (sendo Python uma delas, assim como F#, JavaScript – Apache Cordova/Universal Apps –, Angular JavaScript, CoffeeScript, MySQL, Xamarin, ASP.Net, etc.).

### **3.5. PYCHARM**

Em conjunto com o VS2015 foi utilizado a PyCharm Community, uma versão do framework da JetBrains de menor peso para desenvolvimento de aplicações científicas em comparação com a versão Professional (para desenvolvimento Web).

### **3.6. ANACONDA 3**

Como Python é uma linguagem modular e de versões paralelas, foi utilizado o gerenciador de pacotes e ambientes virtuais Anaconda 3. Este é um gerenciador que cria ambientes virtuais Python para gerenciar os pacotes e versões por ambiente e não por instalação na máquina real.

### **3.7. GNS3**

Para garantir o funcionamento do software é necessário realizar teste em ambientes reais. Pelo fato dos dispositivos Cisco possuírem um valor muito alto foi utilizado o GNS3, que cria redes virtuais emulando dispositivos reais de rede.

### **3.8. SINAPSE**

#### **3.8.1. DIAGRAMA DE CASO DE USO**

O diagrama de caso de uso para o aplicativo é apresentado na figura 2. O usuário (gerente da rede) deve acessar o sistema e realizar o *login* para executar qualquer função do sistema. Quando realizar o *login* o usuário conseguirá configurar os parâmetros do sistema. O

sistema possui 3 operações (fluxos de trabalho) básicas: Conexão remota (via *telnet*), configuração do servidor Syslog e configuração do servidor Netflow.

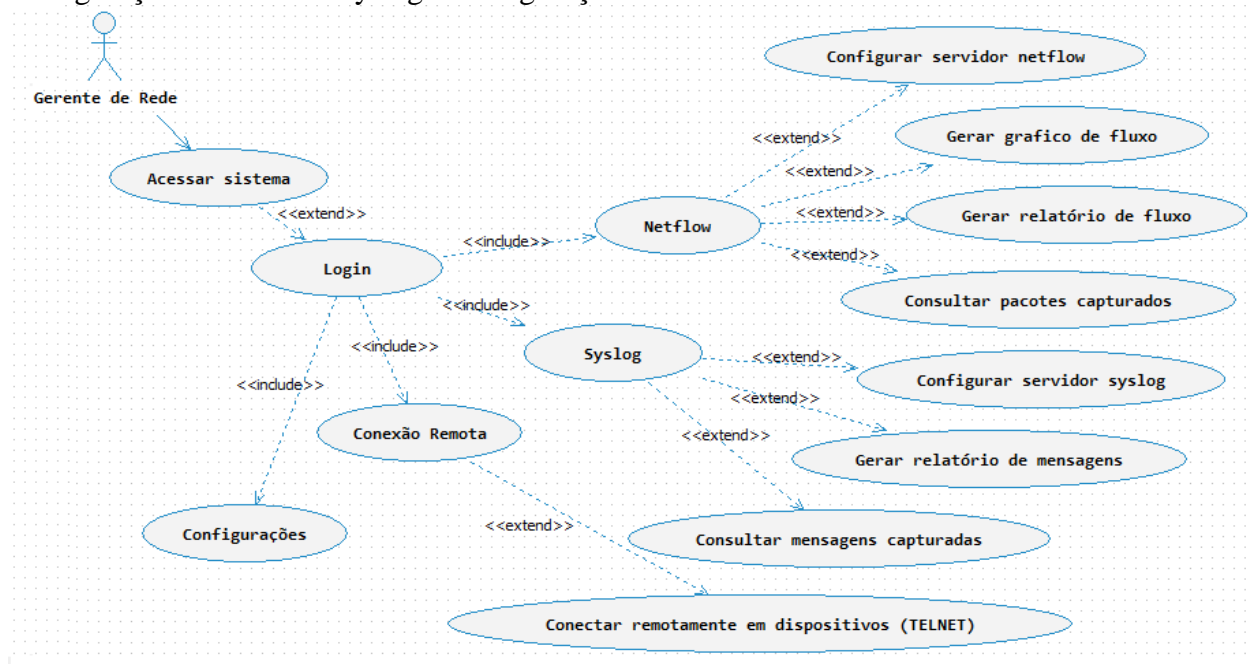


Figura 2 Diagrama de caso de uso do software.

### 3.8.2. FLUXOGRAMA

O fluxo principal do sistema (figura 3) inicia no dispositivo (roteador ou switch). Cada operação (Syslog e Netflow) gera um fluxo. Quando o dispositivo estiver configurado corretamente ele irá disparar as mensagens para o servidor (Sinapse). Quando o servidor recebe uma mensagem ele realiza uma análise para obter as informações do pacote enviado e persiste as informações na base de dados. Após isso ele atualiza as informações do aplicativo para serem exibidas ao usuário. O sistema toma dois rumos a partir desse ponto. Para o Syslog, ele gera uma notificação na tela com algumas opções (corrigir, ignorar, enviar por e-mail etc.). Já para o Netflow o *software* atualiza as informações do gráfico.



A base de dados do software é baseada no modelo relacional (figura 5). A tabela de usuários (*users*) armazena apenas os usuários que possuem permissão de acesso (gerentes da rede). As duas tabelas principais do sistema são: Netflow e Syslog. A tabela Netflow **armazenas** todos os dados estatísticos gerados pelo protocolo Netflow, enviados ao servidor. A tabela de Syslog armazena as mensagens recebidas pelo protocolo Syslog com data, hora, especificações de onde o evento ocorreu e a descrição da mensagem. Ela é vinculada as tabelas Accuracy, Severity e Commands, que são as tabelas que armazenam informações necessárias para detecção e correção dos erros ocorridos.

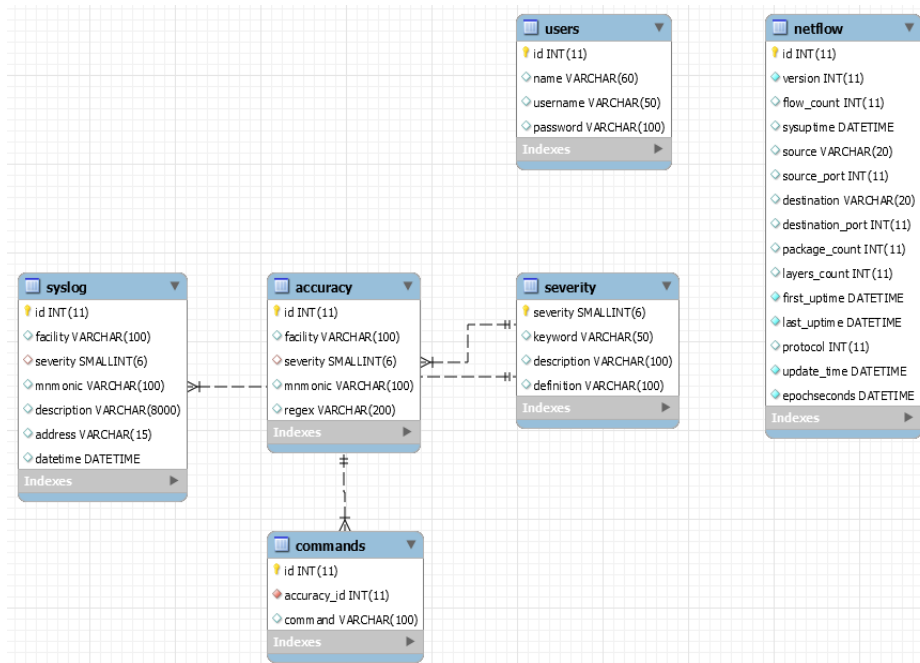


Figura 5 Diagrama de entidade relacionamento

### 3.8.5. TELAS DO SOFTWARE

É na tela exibida na figura 6 que os usuários realizam o *login*. São informados o nome de usuário e a senha, caso existam, o usuário obtém acesso ao sistema.

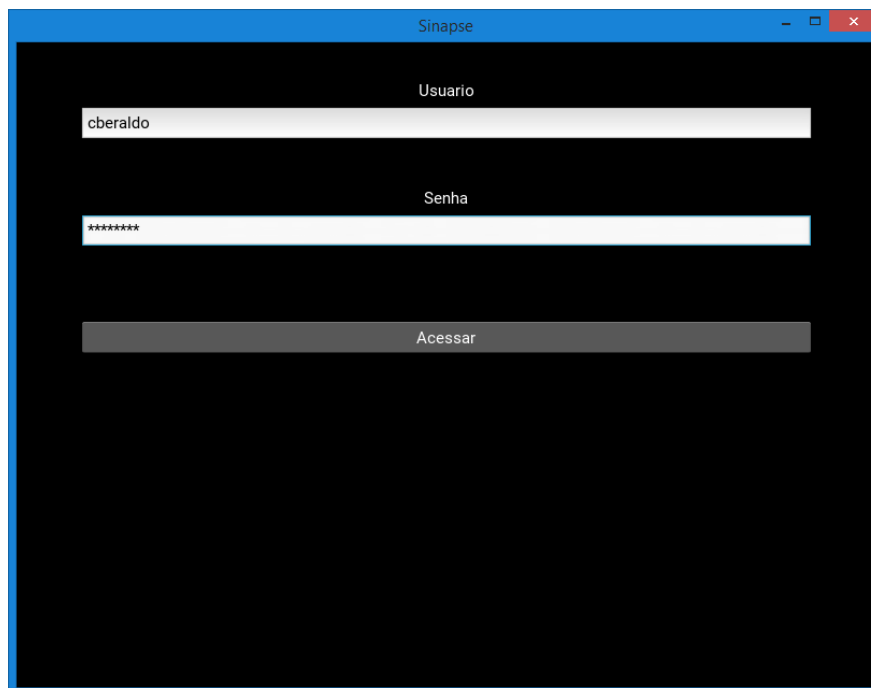


Figura 6 Tela de login do software.

O menu principal (figura 7) dá acesso a todas as outras telas do sistema. O usuário só pode **acessa-lo** caso realize o *login*.

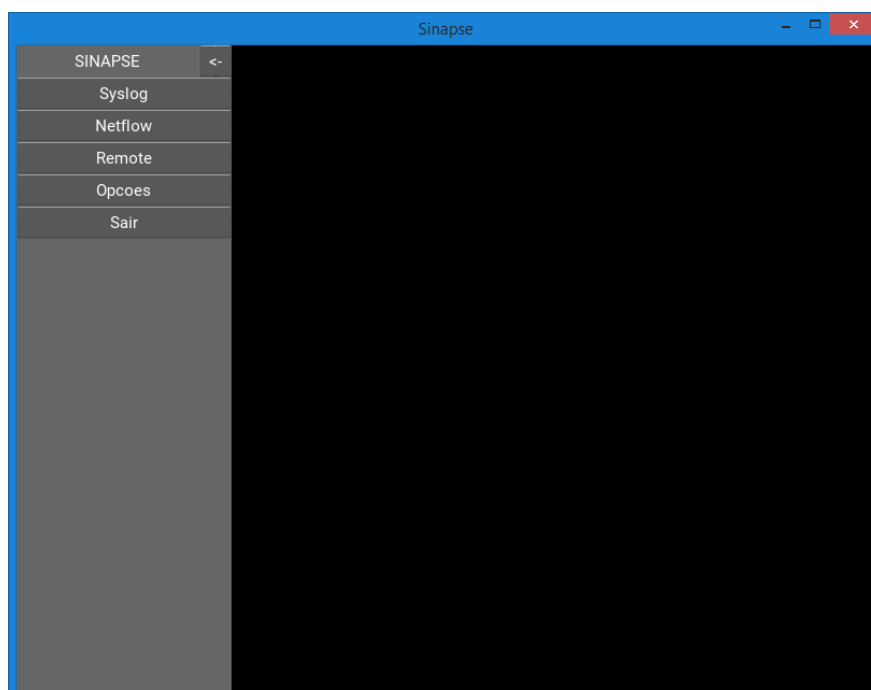


Figura 7 Menu principal.

A tela de configurações, aba Telnet (figura 8), possui informações e parametrizações sobre as conexões Telnet.

“User Name”: usuário padrão de conexão nos dispositivos.

“Password”: senha do usuário padrão.

“Port”: porta de conexão.

“Connection Timeout”: tempo limite para tentativa de conexões, em segundos.



“Reading Timeout”: tempo limite para leitura de retorno em conexões.

“Callback Timeout”: tempo aguardado antes de executar cada comando em uma conexão.

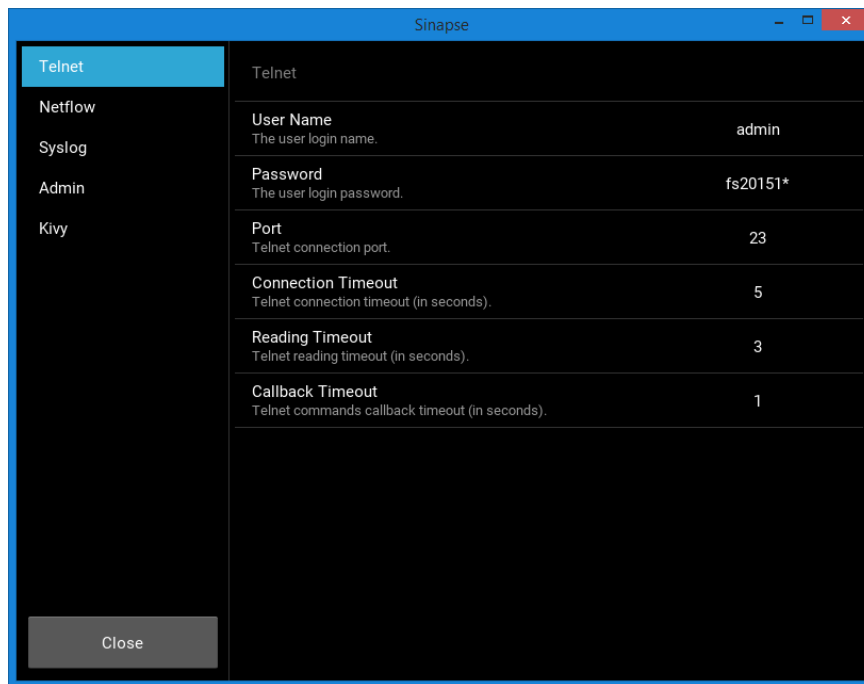


Figura 8 Tela de configurações do software (telnet).

A tela de configurações, aba Netflow (figura 9), possui parametrizações da tela de Netflow.

“Max package count”: número máximo de pacotes em um fluxo considerado normal (aceitável).

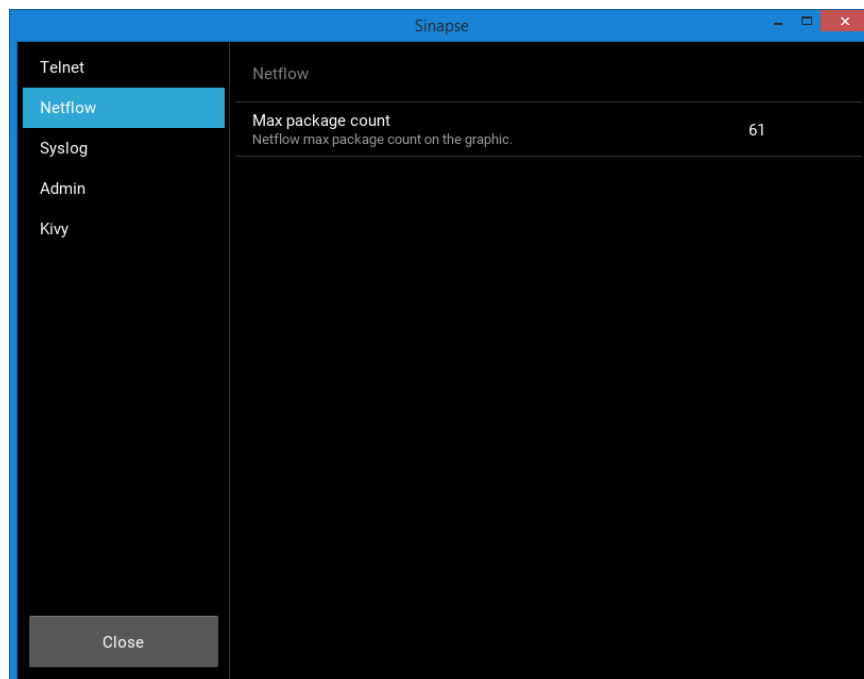


Figura 9 Tela de configurações do software (netflow).

Na figura 10 é exibida a tela de configurações, aba Syslog, possui parametrizações da tela de Syslog.  
“Show Syslog Message”: indica se uma notificação com opções será exibida ao receber uma mensagem Syslog.

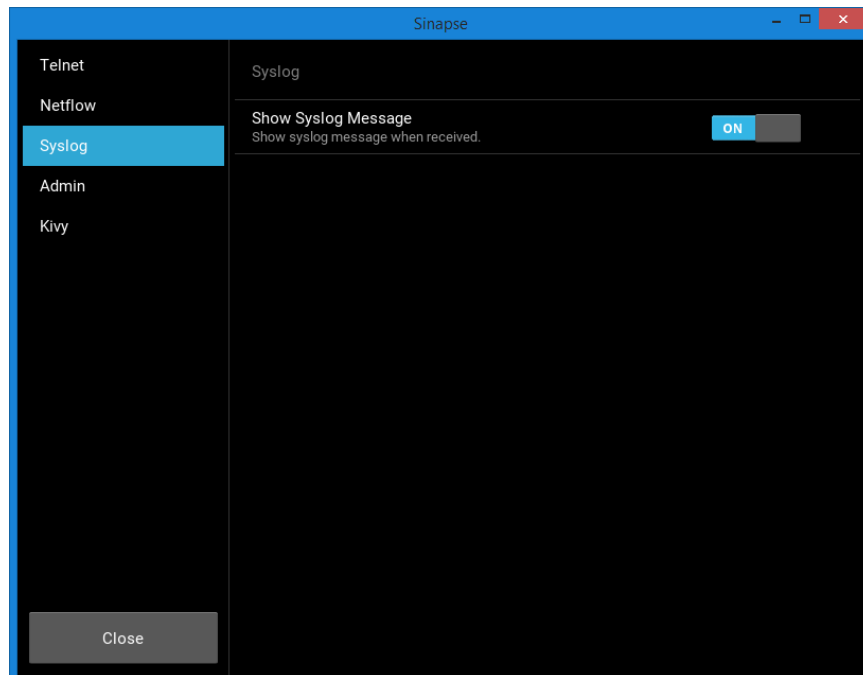


Figura 10 Tela de configurações do software (syslog).

Tela de configurações, aba Admin (figura 11), possui parametrizações internas do software.  
“E-mail”: e-mail para qual as mensagens Syslog/Netflow são enviadas na opção “Enviar por e-mail” nos detalhes de mensagens e também o e-mail utilizado para envio.  
“Password”: senha do e-mail utilizado para envio de mensagens.

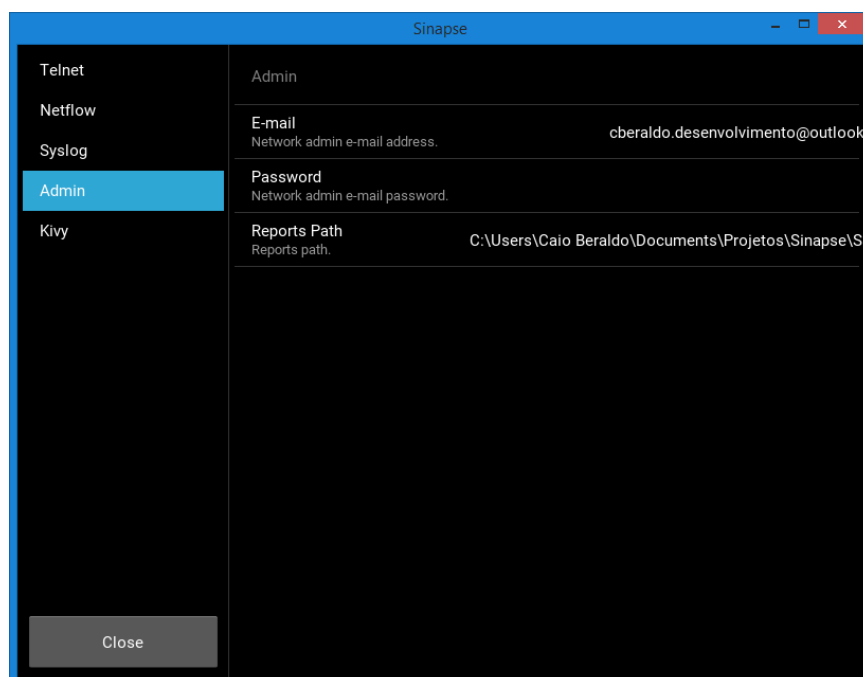


Figura 11 Tela de configurações do software (administrador).

A tela de gerenciamento de Syslog é mostrada na figura 12. Contém uma listagem de todas as mensagens capturadas pelo software. Ao clicar no índice da mensagem (primeira coluna, sequencialmente numerada) exibe detalhes da mensagem. O primeiro campo indica o endereço IP que será utilizado como servidor, para qual os dispositivos enviarão as mensagens, possui as opções de iniciar ou parar o servidor. O botão imprimir gera uma listagem totalizada das mensagens por endereço origem, dia, *facility* (local), *severity* (gravidade), e *minemonic* (minemônico – breve descrição).

The screenshot shows a window titled "Sinapse" with a search bar containing "172.16.1.10" and buttons for "Parar", "Imprimir", and "Voltar". Below is a table with the following data:

| # | Datetime            | Facility | Severity | Minemonic | Description   | Address    |
|---|---------------------|----------|----------|-----------|---|------------|
| 1 | 2016-10-23 01:02:04 | LINK     | 3        | UPDOWN    | Interface Loopback1, changed state to up                    | 172.16.1.1 |
| 2 | 2016-10-23 01:02:04 | LINK     | 5        | CHANGED   | Interface Loopback1, changed state to administratively down | 172.16.1.1 |
| 3 | 2016-10-23 01:02:04 | LINK     | 5        | CHANGED   | Interface Loopback1, changed state to administratively down | 172.16.1.1 |
| 4 | 2016-10-23 01:02:04 | LINK     | 5        | CHANGED   | Interface Loopback1, changed state to administratively down | 172.16.1.1 |
| 5 | 2016-10-23          | LINK     | 5        | CHANGED   | Interface Loopback1, changed                                | 172.16.1.1 |

Figura 12 Mensagens syslog capturadas.

Exemplo de notificação Syslog recebida, ou visualização de detalhes, pode ser observado na figura 13. Contêm opções de “Enviar por e-mail” (para o e-mail definido nos parâmetros) e “Corrigir”, que procura soluções para os problemas na base de dados e aplica no dispositivo origem.

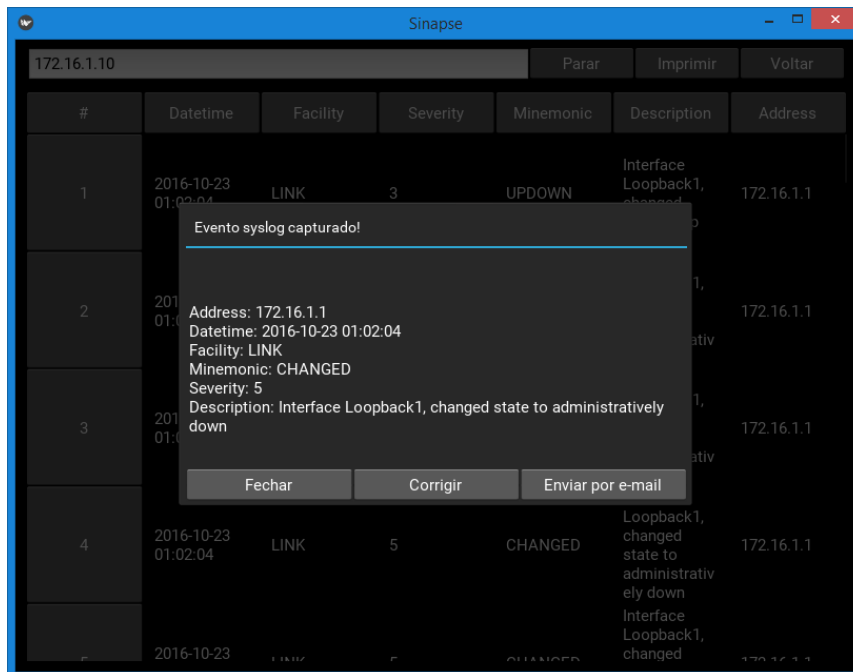


Figura 13 Detalhes da mensagem syslog capturada.

Na figura 14 é exibido um exemplo de correção automática de uma mensagem Syslog recebida. Ao clicar em “Corrigir” o sistema prepara a mensagem Syslog, consulta as análises existentes na base de dados e as executa. Caso encontre uma solução para o problema inicia uma conexão Telnet com o dispositivo onde o problema ocorreu e executa as correções, exibindo elas na tela, uma a uma.

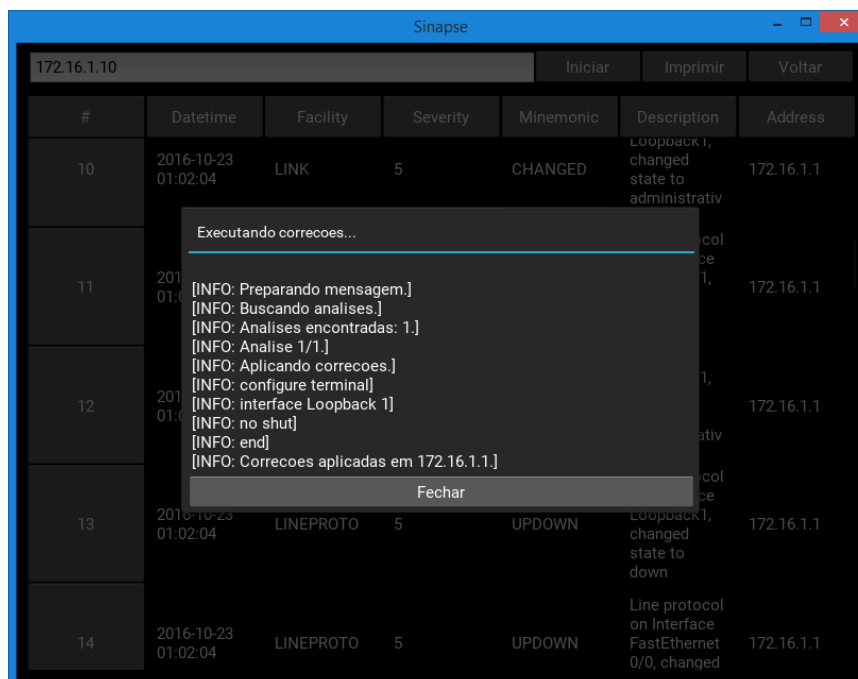


Figura 14 Exemplo de correção automática de falhas.

Uma tela de gerenciamento de Netflow é exibida na figura 15. Contém uma listagem de todas as mensagens capturadas pelo software. Ao clicar no índice da mensagem (primeira coluna, sequencialmente numerada) exibe detalhes da mensagem. O primeiro campo indica o endereço IP que será utilizado como servidor, para qual os dispositivos enviarão as mensagens, possui as opções de iniciar ou parar o servidor. O botão imprimir gera uma listagem totalizada das mensagens por endereço mês, endereço IP, *protol* (protocolo – TCP, UDP etc.). O botão “Grafico” exibe uma tela contendo o gráfico de fluxo de mensagens.

The screenshot shows a window titled "Sinapse" with a search bar containing "172.16.1.10" and buttons for "Iniciar", "Grafico", "Imprimir", and "Voltar". Below is a table with the following data:

| # | Version | Flow | Source       | Port  | Destination  | Port  | Packages | Protocol | Time                |
|---|---------|------|--------------|-------|--------------|-------|----------|----------|---------------------|
| 1 | 5       | 1    | 172.16.1.10  | 57621 | 172.16.1.255 | 57621 | 1        | 17       | 2016-11-11 02:42:15 |
| 2 | 5       | 2    | 172.16.1.100 |       | 172.16.1.1   | 2048  | 61       | 1        | 2016-11-11 02:41:45 |
| 3 | 5       | 2    | 172.16.1.10  | 57621 | 172.16.1.255 | 57621 | 1        | 17       | 2016-11-11 02:41:45 |
| 4 | 5       | 1    | 172.16.1.10  | 57621 | 172.16.1.255 | 57621 | 1        | 17       | 2016-11-11 02:41:15 |
| 5 | 5       | 2    | 172.16.1.10  | 57621 | 172.16.1.255 | 57621 | 1        | 17       | 2016-11-11 02:32:45 |
| 6 | 5       | 2    | 172.16.1.100 |       | 172.16.1.1   | 2048  | 61       | 1        | 2016-11-11 02:32:45 |
|   |         |      |              |       |              |       |          |          | 2016-11-11          |

Figura 15 Fluxos de pacotes netflow capturados.

Na figura 16 há o exemplo de visualização de detalhes de mensagem Netflow. Contém opções de “Enviar por e-mail” (para o e-mail definido nos parâmetros).

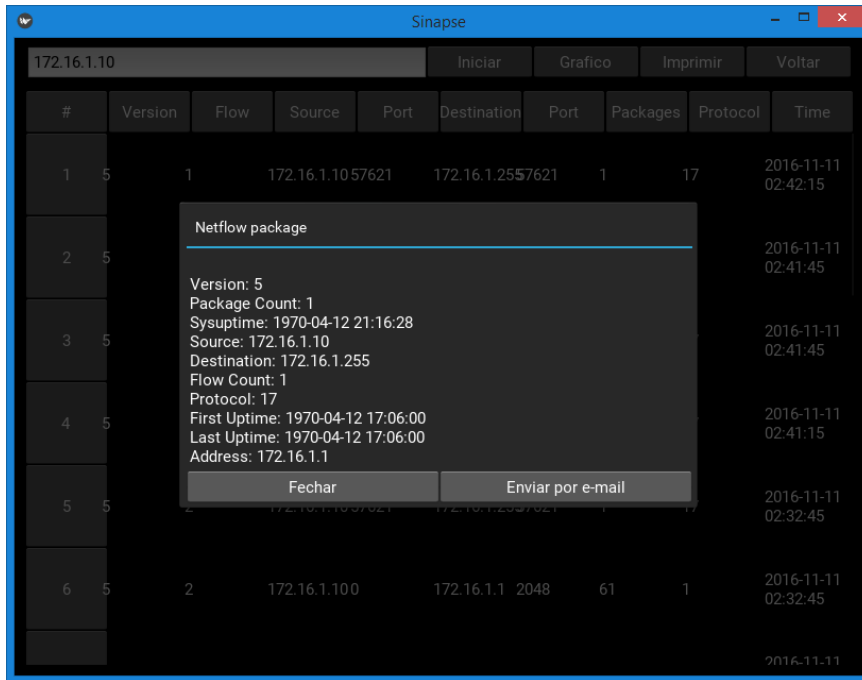


Figura 16 Detalhes do fluxo de pacotes netflow capturados.

Um exemplo de gráfico gerado pelo *software* pode ser visto na figura 17. Ele Exibe detalhes da mensagem ao clicar nos pontos do gráfico. São exibidas linhas azuis para fluxos dentro do limite permitido e vermelhas para fluxo que ultrapassem o limite definido nos parâmetros.

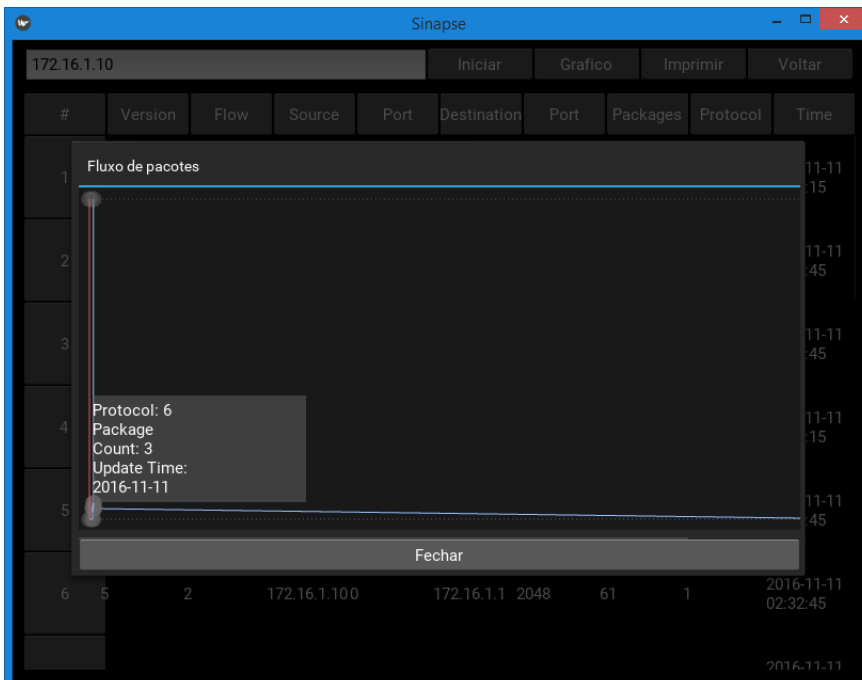


Figura 17 Gráfico de fluxo de pacotes netflow.

A tela de conexões remotas utilizando o protocolo Telnet pode ser vista na figura 18. O primeiro campo indica o endereço IP o qual o *software* irá se conectar. O campo na parte inferior da tela fica desabilitado até que será realizada uma conexão, nela são digitados os comandos a enviar. No centro da tela são exibidas informações da conexão, os comandos enviados e o retorno do dispositivo.

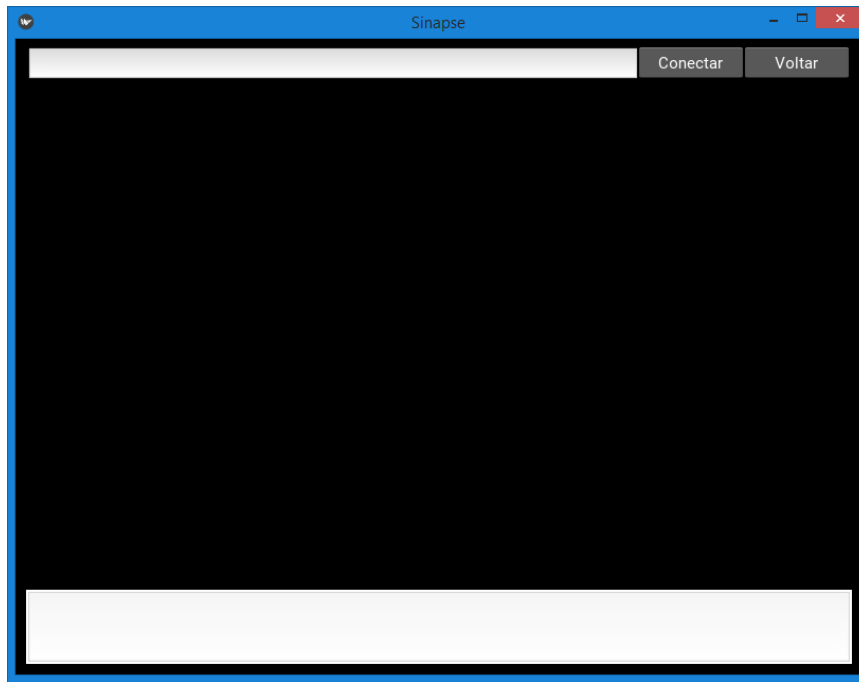


Figura 18 Tela de conexão remota (telnet).

### 3.8.6. RELATÓRIOS DO SOFTWARE

Exemplos de relatórios gerados pelo software, de Syslog e Netflow respectivamente, podem ser observados nas figuras 19 e 20.

#### Sinapse - Syslog

11/11/2016 01:06

| Address    | Facility  | Severity | Mnmonic  | Datetime            | Events |
|------------|-----------|----------|----------|---------------------|--------|
| 172.16.1.1 | LINEPROTO | 5        | UPDOWN   | 2016-10-23 01:02:04 | 18     |
| 172.16.1.1 | LINK      | 5        | CHANGED  | 2016-10-23 01:02:04 | 17     |
| 172.16.1.1 | LINK      | 3        | UPDOWN   | 2016-10-23 01:02:04 | 9      |
| 172.16.1.1 | SYS       | 5        | CONFIG_I | 2016-10-23 01:02:04 | 5      |

Figura 19 Exemplo de relatório Syslog.

## Sinapse - Netflow

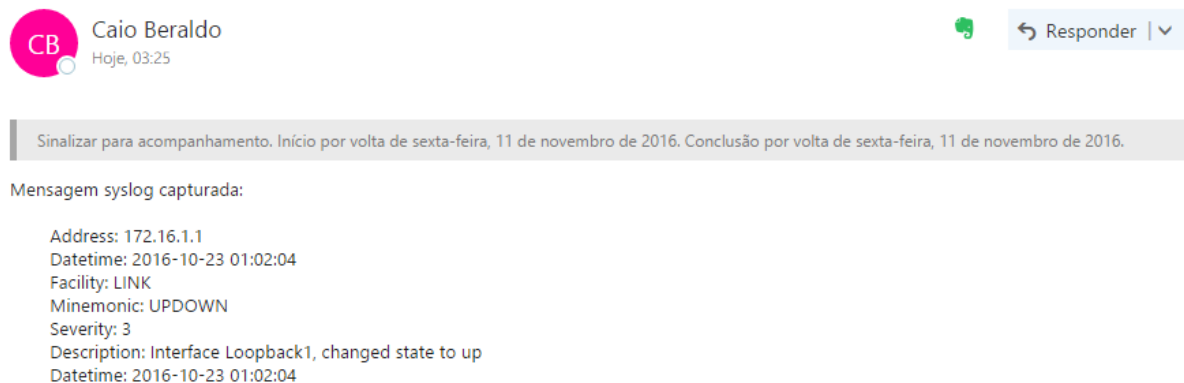
11/11/2016 02:15

| Version | Year | Month | Address    | Protocol | Total Flow | Total Packages |
|---------|------|-------|------------|----------|------------|----------------|
| 5       | 2016 | 11    | 172.16.1.1 | 1        | 8          | 122            |
| 5       | 2016 | 11    | 172.16.1.1 | 6        | 4          | 3              |
| 5       | 2016 | 11    | 172.16.1.1 | 17       | 48         | 30             |

Figura 20 Exemplo de relatório Netflow.

### 3.8.7. UTILITÁRIOS DO SOFTWARE

Exemplos de e-mails enviados pelo software, de Syslog e Netflow respectivamente, podem ser vistos nas figuras 21 e 22.



The screenshot shows an email interface. At the top left, there is a profile card for 'Caio Beraldo' with a pink circular icon containing 'CB' and the text 'Hoje, 03:25'. To the right of the profile card is a 'Responder' button with a dropdown arrow. Below the header is a grey bar containing the text: 'Sinalizar para acompanhamento. Início por volta de sexta-feira, 11 de novembro de 2016. Conclusão por volta de sexta-feira, 11 de novembro de 2016.' The main body of the email contains the text: 'Mensagem syslog capturada: Address: 172.16.1.1 Datetime: 2016-10-23 01:02:04 Facility: LINK Minemonic: UPDOWN Severity: 3 Description: Interface Loopback1, changed state to up Datetime: 2016-10-23 01:02:04'.

Figura 21 Exemplo de mensagem Syslog enviada por e-mail.





Sinalizar para acompanhamento. Início por volta de sexta-feira, 11 de novembro de 2016. Conclusão por volta de sexta-feira, 11 de novembro de 2016.

Fluxo de dados em 172.16.1.1:

```
Version: 5
Package Count: 3
Sysuptime: 1970-01-13 03:33:08
Source: 172.16.1.10
Destination: 192.168.0.1
Flow Count: 4
Protocol: 6
First Uptime: 1970-01-12 23:34:52
Last Uptime: 1970-01-13 02:04:52
```

Figura 22 Exemplo de mensagem Netflow enviada por e-mail.

#### 4. CONSIDERAÇÕES FINAIS

O *software* foi desenvolvido em uma versão beta, e deve ser liberada para utilização a partir de janeiro para acompanhamento das rotinas e necessidades dos usuários e aprimoramento das funções assim como desenvolvimento de novos módulos e aperfeiçoamento das correções.

Poderá ser implantado nas plataformas Windows, Mac OS e Unix/Linux. Mesmo que permita ser instalado em sistemas móveis (Android, iOS etc.) não é recomendado por questões de permissões de acesso em rede e capacidade de processamento do dispositivo.

Apesar da portabilidade do *software*, ainda há a possibilidade de se implementar a instalação para quais quer Sistemas Operacionais mesmo com baixa capacidade de processamento. Possuir uma interface amigável aos usuários e principalmente atender a todos os tipos de redes e dispositivos, não somente Cisco.

#### REFERÊNCIAS

Adel S. Elmaghraby; Yehia H. Khalil. **Computer networks resilience challenges: Routing protocols**. The 10th IEEE International Symposium on Signal Processing and Information Technology, Dez. 2010. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5711719>>. Acesso em: 01 out. 2016.

CISCO. **CCNA Routing & Switching: Connecting Networks**. Disponível em: <<https://1363048.netacad.com/courses/289081>>. Acesso em: 20 jan. 2015.

FILHO, Cláudio Rogério Carvalho. **Desenvolvimento de Aplicações Comerciais com Python e Kivy para Android, iOS, Windows, Linux e MacOS**. Disponível em: <<https://www.udemy.com/python-para-android-ios-windows-linux-mac/learn/v4/overview>>. Acesso em: 01 nov. 2016.

FILHO-SCHAEFFER, Alberto, et al. **PRESET: A toolset for the evaluation of network resilience strategies**. International Symposium on Integrated Network Management (IM 2013), Maio 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6572987/>>. Acesso em: 01 out. 2016.

GNS3 Technologies Inc. **The Official GNS3 Documentation**. Disponível em: <<https://www.gns3.com/support/docs/quick-start-guide-for-windows-us>>. Acesso em: 01 jun. 2016.

Oracle Corporation and/or its affiliates. **MySQL Documentation**. Disponível em: <<http://dev.mysql.com/doc/>>. Acesso em: 01 set. 2016.

Python Software Foundation. **Python 2.7.12 documentation**. Disponível em: <<https://docs.python.org/2.7/>>. Acesso em: 01 jun. 2016.

SCHÖLLER, Marcus; SMITH, Paul; HUTCHISON, David. **Assessing risk for network resilience**. Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Out. 2011. Disponível em: <<http://ieeexplore.ieee.org/document/6078858>>. Acesso em: 01 out. 2016.

TEODOSIU, Mihai Catalin. **Python and Networking Automation: Build 7 Python Apps**. Disponível em: <<http://academy.gns3.com>>. Acesso em: 01 jun. 2016.

Kivy Organization. **Kivy framework**. Disponível em: <<https://kivy.org/docs/api-kivy.html>>. Acesso em: 01 jun. 2016.