



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Segurança da Informação

Diego Bruno Martins

**Desenvolvimento de mecanismos de segurança para
monitoramento com Zabbix**

Americana, SP

2017



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Segurança da Informação

Diego Bruno Martins

**Desenvolvimento de mecanismos de segurança para
monitoramento com Zabbix**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do Prof. Me. Henri Alves Godoy

Área de concentração: Segurança com monitoramento Zabbix

Americana, SP.

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

M342d MARTINS, Diego Bruno

Desenvolvimento de mecanismos de segurança para monitoramento com Zabbix./ Diego Bruno Martins. – Americana: 2017.

60f.

Monografia (Curso de Tecnologia em Segurança da Informação) - -
Faculdade de Tecnologia de Americana – Centro Estadual de Educação
Tecnológica Paula Souza

Orientador: Prof. Ms.Henri Alves de Godoy

1. Segurança em sistemas de informação I. GODOY, Henri Alves de II.
Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de
Tecnologia de Americana

CDU: 681.518.5

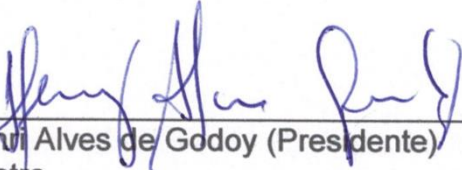
Diego Bruno Martins

**Desenvolvimento de mecanismos de segurança para
monitoramento com Zabbix**


Trabalho de graduação apresentado
como exigência parcial para obtenção do
título de Tecnólogo em Segurança da
Informação pelo CEETEPS/Faculdade de
Tecnologia – FATEC/ Americana.
Área de concentração: Segurança com
monitoramento Zabbix

Americana, 26 de junho de 2017.

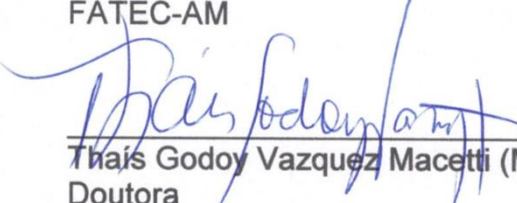
Banca Examinadora:



Henri Alves de Godoy (Presidente)
Mestre
FATEC-AM



Pedro Domingos Antonioli (Membro)
Doutor
FATEC-AM



Thais Godoy Vazquez Macetti (Membro)
Doutora
FATEC-AM

AGRADECIMENTOS

Em primeiro lugar a FATEC e aos professores, ao meu orientador, a minha família e aos meus amigos, pelo incentivo e por acreditarem no meu trabalho.

RESUMO

O presente texto aborda o desenvolvimento de métodos de monitoramento utilizando a ferramenta Zabbix para garantir segurança em uma rede de computadores. O estudo foi realizado na estrutura de um provedor de internet usando para monitoramento protocolo SNMP, agente Zabbix e *scripts* customizados. Foi desenvolvido um sistema de encaminhamento de mensagens dos eventos da rede, buscando auxiliar na tomada de ações e aumentar a disponibilidade dos ativos da rede. São apresentadas etapas de configuração do monitoramento pela interface web e no servidor por linha de comando, *scripts* com explicação passo a passo, gráficos de resultados, exemplos de alarmes de eventos e de encaminhamento de mensagens.

Palavras Chave: Zabbix; Segurança da Informação; Monitoramento.

ABSTRACT

The present text addresses the development of monitoring methods. Use Zabbix to ensure security in a computer network. The study was conducted on the structure of an internet provider using the SNMP protocol monitoring, Zabbix agent and custom scripts. A message forwarding system was developed for the network events, seeking to assist in taking actions and increasing the availability of network assets. It is a step of configuring web-based and server-side monitoring by command line, scripts with step-by-step explanation, result graphs, examples of event alarms, and message forwarding.

Keywords: *Zabbix, Information Security; Monitoring.*

SUMÁRIO

1	INTRODUÇÃO	1
2	SEGURANÇA DA INFORMAÇÃO	3
2.1	INTEGRIDADE	4
2.2	DISPONIBILIDADE	4
2.3	CONFIDENCIALIDADE	5
3	GERENCIAMENTO DE REDES.....	6
3.1	HISTÓRIA DAS REDES DE COMPUTADORES.....	6
3.2	TIPOS DE REDES DE COMPUTADORES	7
3.3	GERENCIAMENTO DE REDES.....	11
3.4	SOFTWARE DE GERÊNCIA DE REDE	12
3.5	MODELO DE GERENCIAMENTO SNMP	14
4	ZABBIX	17
4.1	HISTÓRIA DO ZABBIX.....	17
4.2	COMPOSIÇÃO E ARQUITETURA	17
4.4	MONITORAÇÃO PERSONALIZADA.....	20
5	DESENVOLVIMENTO.....	22
5.1	MONITORAMENTO DO ESTADO DA PORTA	23
5.2	SCRIPT DE INFORMAÇÃO DO TRANSCEIVER.....	28
5.3	CHECKSUM /ETC/PASSWD.....	37
5.4	SCRIPT NOTIFICAÇÃO SMS	39
6.	CONSIDERAÇÕES FINAIS	47
	REFERÊNCIAS BIBLIOGRÁFICAS.....	48

LISTA DE FIGURAS

Figura 1: Exemplo de uma rede local	7
Figura 2: Exemplo de rede metropolitana	8
Figura 3: Exemplo de rede WAN.....	9
Figura 4: Exemplo de uma rede ponto-a-ponto	10
Figura 5: Exemplo de uma rede cliente/servidor	11
Figura 6: Ciclo do gerenciamento de rede	12
Figura 7: Arquitetura de gerência SNMP	15
Figura 8: Protocolo SNMP.....	16
Figura 9: Arquitetura do Zabbix	18
Figura 10: Especificações de <i>hardware</i> e SO	22
Figura 11: Informativo de <i>performance</i>	23
Figura 12: Configuração (<i>Item prototypes</i>).....	24
Figura 13: <i>Item prototypes</i>	25
Figura 14: <i>Trigger</i> (Documentar porta).....	26
Figura 15: <i>Trigger</i> (Porta <i>Down</i>).....	27
Figura 16: Evento (Porta <i>Down</i>)	28
Figura 17: <i>Script</i> de acesso (Telnet)	29
Figura 18: Arquivo (lista de informações)	30
Figura 19: <i>Script external check</i> (<i>Transceiver information</i>)	31
Figura 20: Itens (<i>Transceiver information</i>).....	32
Figura 21: Criação de Item (<i>Transceiver information</i>)	33
Figura 22: <i>Trigger</i> (<i>Transceiver information RxPower Status</i>).....	34
Figura 23: Gráfico de informação de potência de sinal do <i>transceiver</i>	34
Figura 24: Gráfico de informação de temperatura do <i>transceiver</i>	35
Figura 25: Gráfico de informação de corrente do <i>transceiver</i>	35
Figura 26: Gráfico de informação de tensão do <i>transceiver</i>	36
Figura 27: Item (<i>Checksum</i>)	37
Figura 28: <i>Trigger</i> (<i>Checksum</i>).....	38
Figura 29: Evento (<i>Checksum</i>).....	39
Figura 30: <i>Script</i> que envia mensagens SMS	40
Figura 31: Administração de usuários (<i>mídia</i>)	40
Figura 32: Tipos de mídias	41

Figura 33: Criação da mídia SMS	42
Figura 34: Configuração da ação (SMS)	43
Figura 35: Condições da ação.....	44
Figura 36: Operações da ação	45
Figura 37: Mensagens SMS recebidas	46

LISTA DE TABELAS

Tabela 1 - Classificação das redes ordenadas por características	9
Tabela 2 - Resumo de modelo de gerenciamento ISO	13
Tabela 3 - Módulo Zabbix.....	17
Tabela 4 - Definições do Zabbix.....	19
Tabela 5 - Forma de inclusão de novas funcionalidades	21

1 INTRODUÇÃO

Atualmente torna-se cada vez mais frequente o uso e a necessidade de dispositivos computacionais, seja para utilização pessoal ou empresarial. Conseqüentemente as redes de computadores, que interligam esses dispositivos, também tiveram sua ampliação bastante significativa.

O crescimento das redes de computadores nas empresas pode causar problemas como: quedas de *links*, serviços indisponíveis e problemas de visibilidade da rede em razão do não monitoramento dos dispositivos que o constituem (BUENO, 2012).

Segundo Albuquerque (2001), as redes de computadores são responsáveis por realizar serviços necessários a grande parte das organizações empresariais, logo atividades são prejudicadas ou até mesmo tornam-se indisponíveis caso os serviços prestados pela rede tenham falhas.

O monitoramento de redes vem ganhando cada dia mais importância, pois se gerenciada de forma correta é capaz de manter a eficiência da rede de computadores e buscar o aperfeiçoamento do desempenho.

Na gerencia de redes, o monitoramento tem como objetivos, verificar a saúde da rede, a ocorrência de desvios ou acidentes, capacidade de fluxo de dados, informando qualquer anormalidade ao administrador, com o intuito de evitar a perda de informações (SANTOS, 2015).

A proposta deste trabalho é a utilização da ferramenta ZABBIX no monitoramento de redes de uma empresa de telecom, baseado nos pilares da segurança da informação que são: integridade, disponibilidade e confidencialidade.

O ZABBIX trata-se de uma plataforma integrada capaz de auxiliar um administrador a resolver a maioria dos problemas encontrados nas redes de computadores de forma mais eficiente e eficaz (BONOMO, 2006).

O ZABBIX é capaz de monitorar diversos parâmetros de vários ativos em uma determinada rede de computadores, sendo *open Source* (não necessita da aquisição de licenças), também é possível realizar modificações em seus códigos com a finalidade de atender necessidades específicas (LIMA, 2014).

O objetivo geral foi desenvolver métodos de monitoramento utilizando a ferramenta Zabbix para garantir segurança em uma rede de computadores. O estudo será realizado na estrutura de um provedor de *internet* usando para monitoramento protocolo SNMP, agente Zabbix e *scripts* customizados.

Como objetivos específicos:

- Monitorar e documentar as portas dos *switches*, com *triggers* e eventos caso ocorra mudança de estado (*Up/Down*).
- Desenvolver *script* python customizado que acessa o *switch* e coleta informação de potência de sinal, temperatura, e alarmes do *transceiver* para criação de gráficos e *triggers* para eventos, que auxiliam nas tomadas de decisões, e garantir a disponibilidade do serviço.
- Usar *template* Servidor Linux padrão do Zabbix para checar a integridade dos arquivos de usuários e senhas do sistema, a fim de garantir a confidencialidade das informações.
- Criar um *script* que envia mensagens SMS para o celular do analista informando os eventos mais críticos da rede, para que possa garantir maior disponibilidade dos serviços e um tempo de indisponibilidade menor.

O trabalho foi estruturado em seis capítulos, sendo que o primeiro é a introdução, o segundo aborda a segurança da informação, o terceiro apresenta rede de computadores, conceitua gerenciamento de redes, o quarto explica sobre a ferramenta Zabbix, o quinto é apresentado a metodologia, e o sexto se reserva às considerações finais.

2 SEGURANÇA DA INFORMAÇÃO

De acordo com o dicionário Ferreira (2010), informação é o conjunto de dados acerca de alguém ou de algo.

Para Dantas (2011), a segurança da informação protege a informação de diversos tipos de ameaças, para que possibilite a continuidade do negócio, aumentando o retorno e as oportunidades sobre o investimento e diminuindo o risco.

A NBR ISO/IEC 27002 (ABNT, 2005) define a segurança da informação como:

“Preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas”.

A NBR ISO/IEC 17799 (ABNT, 2005), afirma que a segurança da informação é:

“Especialmente importante no ambiente dos negócios, cada vez mais interconectado. Como um resultado deste incrível aumento da interconectividade, a informação está agora exposta a um crescente número e a uma grande variedade de ameaças e vulnerabilidades”.

De acordo com Sêmola (2003), os princípios básicos da segurança da informação são integridade, disponibilidade e confidencialidade, que será abordado nos próximos capítulos.

2.1 INTEGRIDADE

De acordo com NBR ISO/IEC 27002 (ABNT, 2005), “A integridade é a garantia da exatidão e completeza da informação e dos métodos de processamento”.

Segundo o Tribunal de contas da União (TCU, 2012), a integridade consiste na fidedignidade da informação. Indica a similaridade dos dados que são armazenados com relação às inserções, alterações e processamentos autorizados. Também garante a conformidade dos dados repassados pelo emissor e os que são recebidos pelo destinatário.

“A manutenção da integridade pressupõe a garantia de não violação dos dados com intuito de alteração, gravação ou exclusão, seja ela acidental ou proposital” (TCU, 2012).

Se a informação é corrompida, falsificada, ou quando ocorrem substituições, inserções ou exclusões de parte do conteúdo é caracterizado como quebra da integridade (DANTAS, 2011).

2.2 DISPONIBILIDADE

Segundo NBR ISO/IEC 27002 (ABNT, 2005), “A disponibilidade é a garantia de que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes sempre que necessário”.

Deve-se garantir que as informações estejam disponíveis, sem interrupções em seu fornecimento para quem é de direito e por período acordado (TCU, 2012).

A quebra de disponibilidade ocorre quando a informação não se apresenta disponível para os seus usuários e destinatários no período necessário. Ao garantir a disponibilidade da informação, propicia-se o alcance da leitura, do transporte e armazenamento da informação (DANTAS, 2011).

2.3 CONFIDENCIALIDADE

De acordo com NBR ISO/IEC 27002 (ABNT, 2005), “A confidencialidade é a garantia de que a informação é acessível somente por pessoas autorizadas a terem acesso”.

Ao manter a confidencialidade, assegura-se que pessoas não autorizadas não consigam acesso às informações, seja de forma acidental ou proposital (TCU, 2012).

Para Dantas (2011), “a perda da confidencialidade é a perda do segredo da informação. Garantir a confidencialidade é assegurar o valor da informação e evitar a divulgação indevida”.

3 GERENCIAMENTO DE REDES

Uma rede de computadores é um grupo de computadores autônomos, ou seja, independentes e interconectados por uma única tecnologia. Seu objetivo é o compartilhamento de recursos e a troca de informação dos computadores interconectados, serviços de correios eletrônicos (e-mail), acesso à *internet*, troca de arquivos, entre outros (TANENBAUM, 2003).

3.1 HISTÓRIA DAS REDES DE COMPUTADORES

As redes de computadores foram instruídas na década de 1960 e tinham como propósito realizar a troca de dados entre dois computadores. O armazenamento de dados ocorria através de cartões perfurados. Na década de 1970, foi criada a Arpanet, que possibilitou a criação de uma rede que interligava universidades, empresas e corporações militares. Nesta época também foram criados os serviços de e-mail, FTP e DNS. Nos anos 90 ocorreu a expansão do uso da internet, diferentes redes ganharam destaque no mercado. Na construção das redes locais computadores (LAN's) o uso de Ethernet ganhou espaço e se popularizou. O acesso à internet nas empresas era realizado através de linha discada, mas suas limitações e custos altos induziram a sua substituição por linhas de *frame relay* (conexão dedicada com velocidades de 64 kbits). Esse tipo de conexão à internet permitiu que a conexão fosse compartilhada entre os computadores da rede (FRANCISCATTO *et al*, 2014).

Atualmente as redes podem ser construídas por diferentes possibilidades: redes cabeadas, entre elas estão Ethernet e fibra óptica, sem-fio (Wireless) que são representadas por rádio, Bluetooth, Wi-Fi, infravermelho (HUNECKE, 2011).

A evolução das redes de computadores e sua expansão não modificou seu fundamento principal que é compartilhar recursos de *hardware* e também de *software* e proporcionar a troca de informações (MORIMOTO, 2007).

3.2 TIPOS DE REDES DE COMPUTADORES

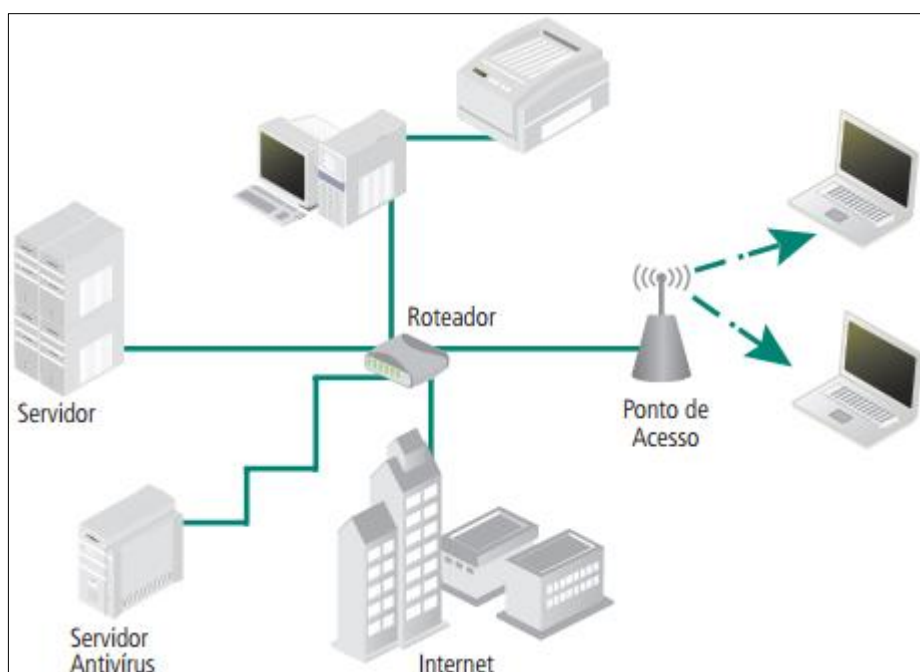
De acordo com Franciscatto *et al*, (2014), as redes de computadores são classificadas de acordo com sua disposição geográfica e hierarquia.

São classificadas de acordo com seu alcance, em LAN, MAN e WAN.

LAN (*Local Area Network* – Rede de Área Local) - as distâncias alcançadas são de algumas centenas de metros, tem como características as altas taxas de transmissão, (AMARAL, 2012), geralmente é composta por computadores conectados entre si por dispositivos como *switch*, *hub* e placas de rede, que permitem a troca de informações e recursos (FRANCISCATTO *et al*, 2014).

A Figura 1 exemplifica a estrutura de uma rede LAN com interligação a uma rede *wireless*.

Figura 1: Exemplo de uma rede local



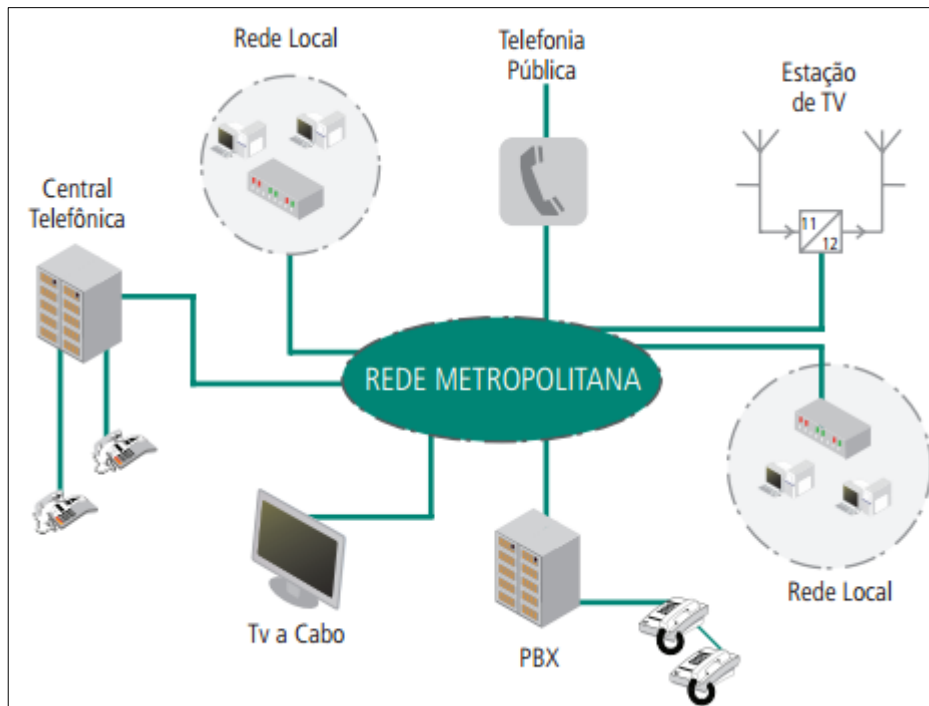
Fonte: Amaral (2012)

MAN (*Metropolitan Area Network* – Rede de Área Metropolitana) – possui abrangência maior que as redes LAN, compreendendo espaços como campus,

região ou até mesmo cidade, mas em contrapartida tem taxas de transmissão menores e erros mais elevados (AMARAL, 2012).

A Figura 2 exemplifica a estrutura de uma rede MAN.

Figura 2: Exemplo de rede metropolitana

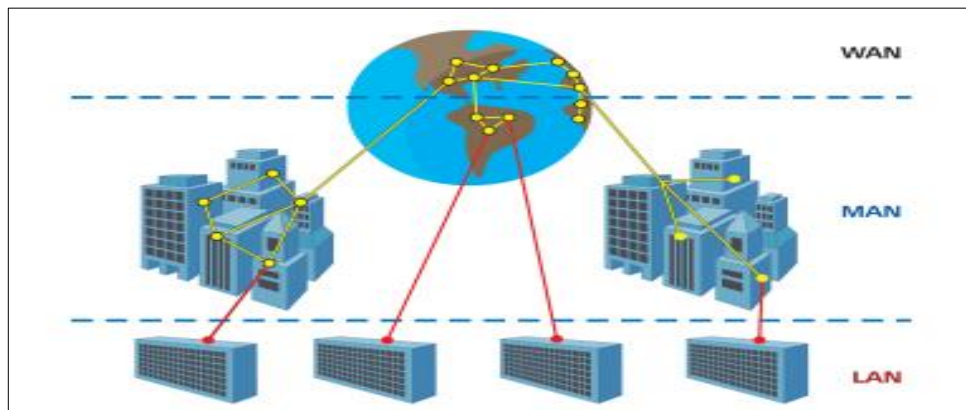


Fonte: Amaral (2012)

WAN (*Wide Area Network* – Rede de Área Extensa) é uma rede de longas distâncias, pode fazer a interligação de continentes utilizando-se de enlaces externos como satélites e cabos, sejam eles submersos ou terrestres, possui baixa taxa de transmissão e erros mais elevados que das redes MAN (AMARAL, 2012).

A Figura 3 exemplifica a estrutura de uma rede WAN.

Figura 3: Exemplo de rede WAN



Fonte: CTISM

A Tabela 1, a seguir sintetiza as características de cada tipo de rede de computadores, descritas acima:

Tabela 1 - Classificação das redes ordenadas por características

Classificação	Taxa de transmissão	Taxa de erros	Distâncias
WAN	Na ordem de 622 Mbps	Alta	Milhares de quilômetros
MAN	Na ordem de 2,5 Gbps	Média	Centenas de quilômetros
LAN	Na ordem de 10 Gbps	Baixa	Centenas de metros

Fonte: Amaral (2012)

De acordo Franciscatto *et al* (2014), com a abrangência das redes *wireless* (sem fio), novas classificações foram adotadas são elas:

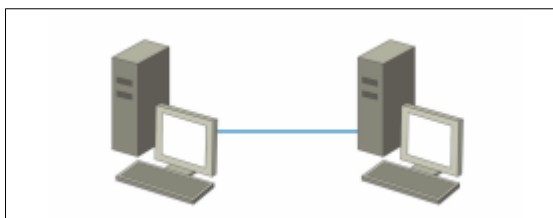
- WMAN – rede de área metropolitana sem-fio destina-se principalmente a operadores de telecomunicações;
- WWAN – rede de longa distância sem-fio são comumente utilizadas para criação de redes de transmissão celular;
- RAN – considerada uma subcategoria de uma MAN, uma RAN (*Regional Area Network*), corresponde a uma rede de computadores de uma região geográfica específica;
- CAN – uma CAN (*Campus Area Network*) corresponde a uma rede de computadores formada por computadores dispostos em edifícios, prédios, campus, entre outros” (MENDES, 2007)

A classificação por hierarquia refere-se a como os computadores de uma determinada rede se comunicam. Os principais tipos são rede ponto-a-ponto e as redes de cliente-servidor.

Rede Ponto-a-Ponto – Os computadores trocam informação entre si, podendo ser arquivos ou até mesmo recursos. As principais características deste tipo de rede são: utilização em pequenas redes; implementação de baixo custo e fácil, segurança reduzida e utilizam sistema de cabeamento simples (FRANCISCATTO *et al*, 2014).

A Figura 4 mostra um exemplo de rede ponto-a-ponto.

Figura 4: Exemplo de uma rede ponto-a-ponto

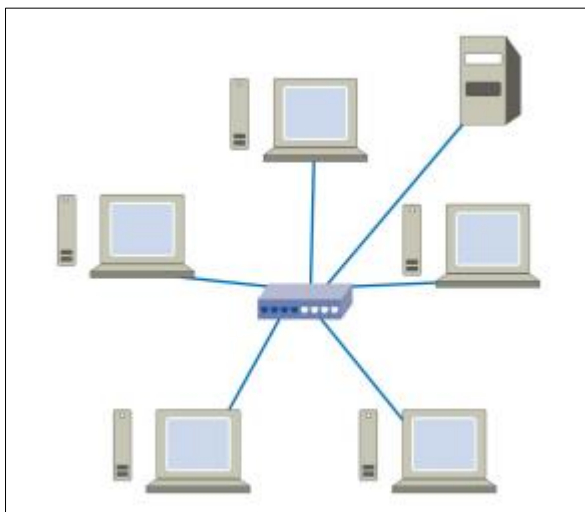


Fonte: CTISM

Rede Cliente-Servidor – Tipo de rede de computadores largamente utilizado, existe a presença de um ou mais servidores, o computador (cliente) que deseja utilizar um recurso ou realizar uma atividade deve solicitar a autorização do servidor, tem-se como características: custo mais elevado e implantação mais complexa, presença de pelo menos um servidor de rede, sua estrutura de segurança é melhor, não há tolerância a falhas, entre outros (TANENBAUM, 2003; KUROSE & ROSS, 2010; FRANCISCATTO *et al*, 2014).

A Figura 5 exemplifica a estrutura de uma rede do tipo cliente-servidor.

Figura 5: Exemplo de uma rede cliente/servidor



Fonte: CTISM (2014)

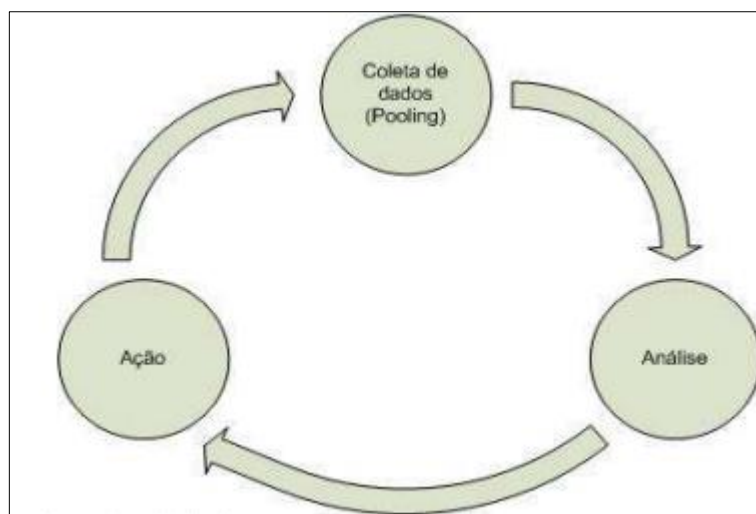
Com o crescimento da rede e dos serviços oferecidos na *internet*, como compras *online* e operações bancárias a segurança passou a ser essencial nas redes de computadores (TANENBAUM, 2003).

3.3 GERENCIAMENTO DE REDES

O gerenciamento de redes engloba as ações de fornecer, integrar e coordenar os *hardwares*, os *softwares* e também no que diz respeito da operação humana que atua monitorando, testando, configurando, consultando, analisando, avaliando e controlando a rede e seus recursos, a fim de garantir um bom desempenho, qualidade dos serviços prestados e operação em tempo adequado sobre um custo plausível (KUROSE, 2010).

De acordo com Bueno (2012), no gerenciamento trabalha-se com três pontos, exemplificado na Figura 6.

Figura 6: Ciclo do gerenciamento de rede



Fonte: Pandora FMS (2012)

Coleta de dados (*Pooling*) – Realiza a coleta de dados dos recursos a serem gerenciados. É executado por componente de *hardware* e *software*.

Análise – Os dados coletados serão analisados dentro dos padrões estabelecidos pelo administrador, verificando sua normalidade ou não.

Ação – A ação acontece após a análise dos dados, são exemplos de ações, o envio de um e-mail, um alarme visual no navegador de *Internet*, ou qualquer outra ação que for suportada ou compatível com a plataforma de gerenciamento utilizada.

3.4 SOFTWARE DE GERÊNCIA DE REDE

De acordo com Specialski (1999), mesmo uma rede pequena, necessita de ser gerenciada, pois está ação garante que os recursos estejam disponíveis com desempenho razoável. O crescimento da rede faz com que também se aumente a complexidade da mesma. Desta forma, se faz necessária a adoção de ferramentas automatizadas para seu controle e monitoramento.

Segundo Harnedy, (1997, p. 2 *apud* SPECIALSKI, 1999), a necessidade de um *software* de gerenciamento de redes é justificada pelos seguintes fatores:

- “• As redes e recursos de computação distribuídos estão se tornando vitais para a maioria das organizações. Sem um controle efetivo, os recursos não proporcionam o retorno que a corporação requer;
- O contínuo crescimento da rede em termos de componentes, usuários, *interfaces*, protocolos e fornecedores ameaçam o gerenciamento com perda de controle sobre o que está conectado na rede e como os recursos estão sendo utilizados;
- Os usuários esperam uma melhoria dos serviços oferecidos (ou no mínimo, a mesma qualidade), quando novos recursos são adicionados ou quando são distribuídos;
- Os recursos computacionais e as informações da organização geram vários grupos de aplicações de usuários com diferentes necessidades de suporte nas áreas de desempenho, disponibilidade e segurança. O gerente da rede deve atribuir e controlar recursos para balancear estas várias necessidades;
- À medida que um recurso fica mais importante para a organização, maior fica a sua necessidade de disponibilidade. O sistema de gerenciamento deve garantir esta disponibilidade;
- A utilização dos recursos deve ser monitorada e controlada para garantir que as necessidades dos usuários sejam satisfeitas a um custo razoável.”

O uso do *software* permite o monitoramento e o controle remoto de roteadores, *switches*, *hosts* e pontes, verificando sua situação, dados estatísticos da rede a qual pertencem, configurações e rotas de *interface* (COMER, 2007).

Atualmente os modelos de *software* mais aceitos são ISO (*International Organization for Standardization*) (BUENO, 2012).

Na Tabela 2, encontram-se as cinco áreas funcionais do gerenciamento de redes, estabelecidos pela ISO:

Tabela 2 - Resumo de modelo de gerenciamento ISO

Áreas	Descrição
Gerenciamento de Falhas	A falha é considerada uma condição de anormalidade que exige ação de gerenciamento. Tem como objetivo, registro, a detecção e reação no tratamento de falha. O protocolo <i>Simple Network Management Protocol</i>

	(SNMP) é utilizado para este fim.
Gerenciamento de Contabilização	O administrador da rede faz o controle da utilização de recursos pelos usuários, ou seja, define quotas e critérios para o seu uso.
Gerenciamento de Configuração	Está relacionado com o processo de inicialização da rede e a sua desabilitação total ou parcial, também é responsável por manutenções, atualizações, adições ou relacionamento de componentes. Em outras palavras deve ser capaz de identificar os componentes da rede e definir a conectividade dos mesmos e entre eles.
Gerenciamento de Desempenho	Faz o monitoramento das atividades da rede, bem como o controle de ajustes e troca dos recursos. O protocolo <i>Simple Network Management Protocol</i> (SNMP) tem papel fundamental, para gerenciar desempenho da internet.
Gerenciamento de Segurança	Protegem os recursos da rede e também as informações dos usuários, a política de segurança deve ser robusta e efetiva, faz a utilização de firewalls, chave e autoridades certificadoras.

Fonte: Specialski (1999)

3.5 MODELO DE GERENCIAMENTO SNMP

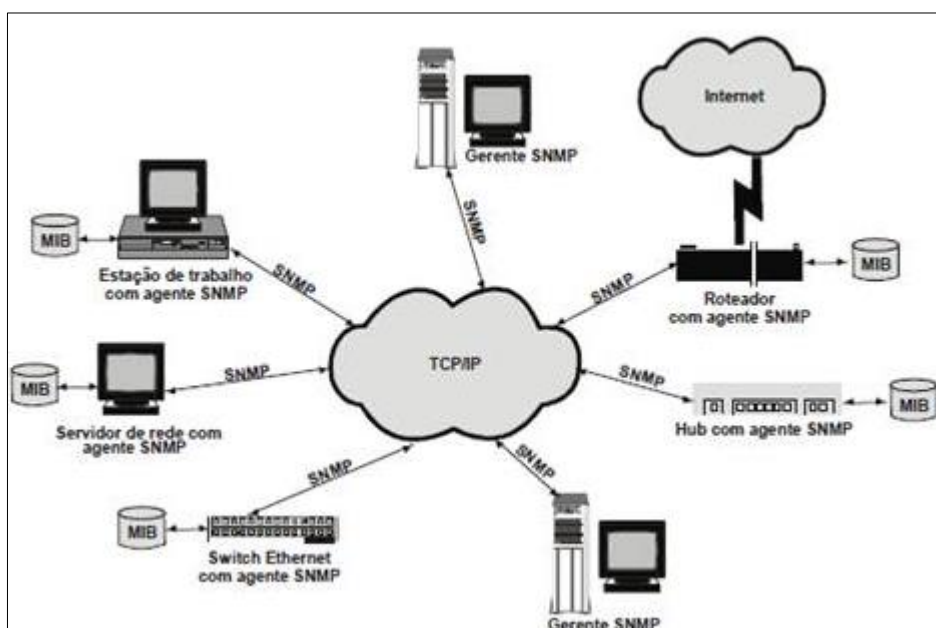
O modelo de gerenciamento SNMP, também é chamado de modelo de *internet*, possui uma interpelação genérica, e utiliza componentes básicos como, gerente, agente, entidade com função dupla, informações e protocolos de informações (DÉO, 2012).

Specialski (1999) descreve que o modelo arquitetural do SNMP, consiste em uma:

“coleção de estações de gerenciamento e elementos de rede. As estações de gerenciamento executam aplicações que monitoram e controlam os elementos de rede. Os elementos de rede são equipamentos tais como hospedeiros, *gateways*, servidores de terminais, e similares, que possuem agentes de gerenciamento, e que são responsáveis pela execução das funções de gerenciamento de rede, requisitadas pelas estações de gerenciamento. O protocolo SNMP é usado para transportar a informação de gerenciamento entre as estações de gerenciamento e os agentes existentes nos elementos de rede”.

A Figura 7 mostra algumas das possibilidades de interações entre o gerente e um agente, através do protocolo SNMP:

Figura 7: Arquitetura de gerência SNMP



Fonte: Teleco (2012)

O monitoramento por SNMP propicia uma visão geral da comunicação ente agente e gerente, as informações são colhidas pelos agentes e enviadas ao gerente, que transfere a aplicação do gerenciamento, que irá efetuar as ações adicionais que podem ser armazenamento em banco de dados, alimentação de gráficos, geração de alertas, entre outros (SANTOS, 2015).

O SNMP é definido como o protocolo de gerência de redes padrão do IETF (*Internet Engineering Task Force*) (IETF, 1990).

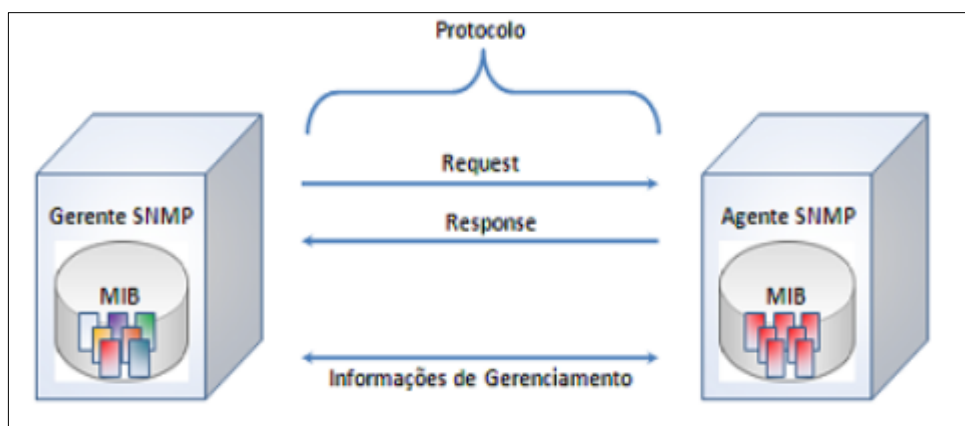
Segundo Contessa e Polina (2006), este protocolo pertence à camada de aplicações da ISO e faz uso na camada de transporte os serviços do protocolo UDP para o envio de mensagens através da rede IP utilizam as portas 161 e 162 e seus pacotes tem tamanho variável.

De acordo com TELECO (2012), os quatro componentes básicos do protocolo SNMP são:

- “Nós gerenciados (agentes) - são instalados nos dispositivos gerenciáveis da rede, que podem ser quaisquer componentes de *hardware* conectados a ela, tais como computadores (*hosts*), impressoras, *hubs*, *switches*, roteadores, entre outros. Os agentes interagem diretamente com a MIB e são responsáveis por responder às solicitações feitas pelos gerentes (*pollings*) através de ações (*responses*). Eles também podem enviar, assincronamente, informações (*traps*) aos gerentes, isto quando ocorre algum problema sério ou um evento relevante para o gerenciamento da rede;
- As estações de gerenciamento (gerentes) - são *softwares* executados em uma ou mais estações capazes de realizar tarefas de gerenciamento da rede, sendo responsáveis por enviar *pollings (requests)* às estações agentes e receber as respostas a estes *pollings (responses)*, podendo ainda acessar (*get*) ou modificar (*set*) informações nos agentes e receber, mesmo sem requisição, informações relevantes ao gerenciamento (*traps*);
- As informações de gerenciamento (MIBs) - um banco de dados lógico que armazena informações estatísticas de configuração e de *status*, relativas a todos os possíveis objetos gerenciáveis da rede.
- Protocolo de gerenciamento (SNMP). ”

A Figura 8 exemplifica o funcionamento do protocolo:

Figura 8: Protocolo SNMP



Fonte: Horst; Pires; Déo (2015)

De acordo com Horst; Pires; Déo (2015) as informações são armazenadas em MIBs, e transportadas através do protocolo SNMP.

4 ZABBIX

A Zabbix SIA define Zabbix como uma solução Open Source de monitoramento de performance definitiva. Apresenta um monitoramento avançado, alertas e características visuais ainda não encontrados em outros sistemas.

É utilizada com o intuito de monitorar a disponibilidade e o desempenho de aplicações, ativos e serviços de rede em todo mundo (HORST; PIRES; DÉO, 2015).

A versão mais recente da ferramenta é a 3.4, é utilizada para cumprir suas funções um sistema de gerenciamento de banco de dados (SGBD), para armazenar configurações e dados coletados (HORST; PIRES; DÉO, 2015).

4.1 HISTÓRIA DO ZABBIX

Esta ferramenta foi desenvolvida por Alexei Vladishev, que buscava um dispositivo capaz de realizar o monitoramento acessível financeiramente, de descomplicada manutenção e utilização (HORST; PIRES; DÉO, 2015). Foram lançadas sete versões desta ferramenta, sendo a última a 3.4, que será apresentada em setembro de 2017, a oitava versão da ferramenta é denominada Zabbix 4.0 LTS, e a expectativa é de que seja lançada até 2020 (Zabbix SIA)

4.2 COMPOSIÇÃO E ARQUITETURA

Segundo Horst; Pires; Déo (2015), o Zabbix é composto por diferentes módulos, os mais comumente utilizados estão indicados no Tabela 3:

Tabela 3 - Módulo Zabbix

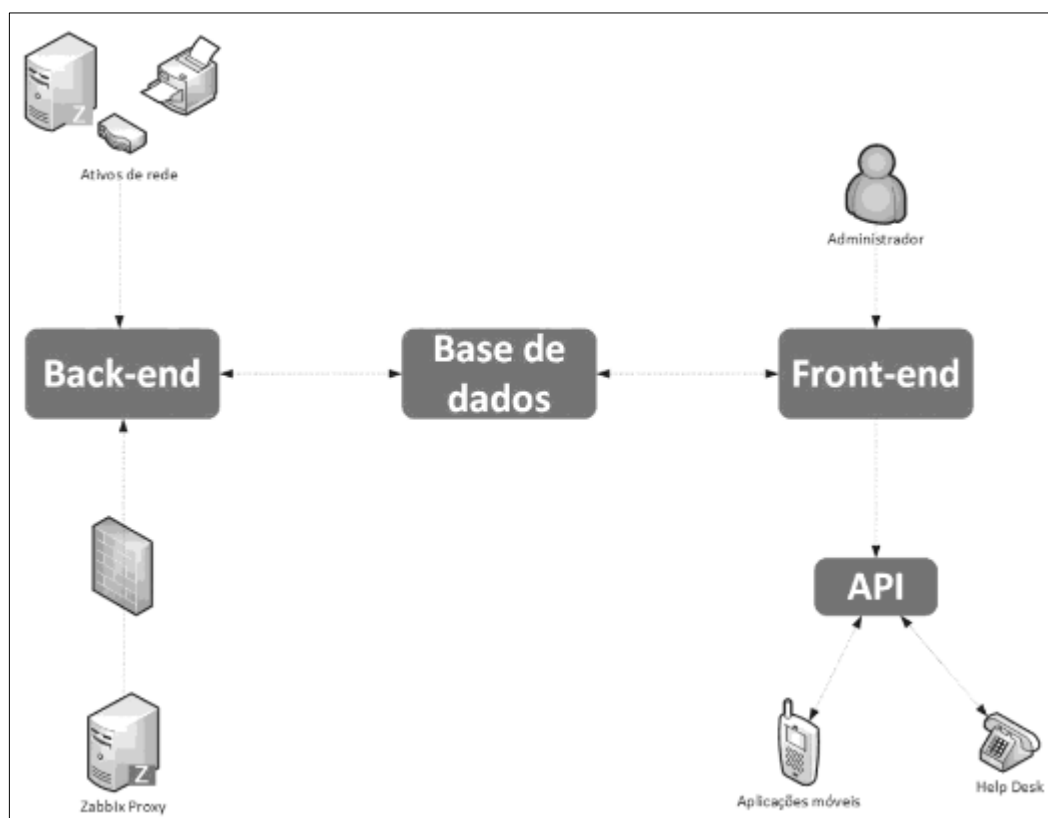
Módulo	Características
Servidor Zabbix	Faz a verificação remota dos serviços e é a ligação para os agentes e proxy Zabbix, trata as informações para mostrar relatórios e alertas com finalidade de executar ações pré configuradas.

Banco de dados	Os acessos ocorrem via servidor Zabbix e <i>interface web</i> .
<i>Interface web</i>	Ambiente de configuração do Zabbix para visualização das informações
Agente Zabbix	Faz o envio ao servidor de dados do equipamento, monitora a situação dos aplicativos, serviços e <i>hardware</i> .
<i>Proxy Zabbix</i>	Recebe dados de funcionamento e <i>status</i> em prol de um servidor Zabbix
Java Gateway	Suporte para aplicações JMX (<i>Java Management Extensions</i>) e o objetivo de restaurar os contadores do JMX (versão 2).

Fonte: Horst; Pires; Déo (2015)

As três camadas que compõem a arquitetura do Zabbix, são: aplicação, o banco de dados e *interface Web* (LIMA, 2014). A Figura 9 ilustra a arquitetura do Zabbix.

Figura 9: Arquitetura do Zabbix



Fonte: Lima (2014)

De acordo com Lima (2014), as funções das camadas são:

“*Back-end* – responsável por fazer a coleta dos dados nos ativos de rede;

Banco de dados – responsável por armazenar as informações coletadas pelo *back-end* e apresenta-las ao *Front-end*;

Interface Web – É representada pelo *Front-end*, a qual dá acesso a informações de monitoramento aos administradores e também fornece informações para aplicações que utilizam a API do Zabbix.”

4.3 DEFINIÇÕES ZABBIX

Segundo Zabbix, para termos muito usados na ferramenta as definições são:

Tabela 4 - Definições do Zabbix

Nome	Descrição
<i>Host</i>	Um elemento da rede que é monitorado através de IP ou DNS.
Grupo de <i>hosts</i>	Um conjunto lógico de <i>hosts</i> que pode incluir <i>hosts</i> ou <i>templates</i> . Os grupos de <i>hosts</i> podem ser utilizados para definições de acesso de diferentes grupos de usuários.
Item	Configuração para receber um dado de um <i>host</i> monitorado.
<i>Trigger</i>	Através de uma expressão lógica é definido limites de mudança de estado, que podem ser incidentes e é usado para analisar dados recebidos pelos itens. Quando o dado coletado está fora do limite atribuído a <i>trigger</i> passa do estado de "OK" para o estado de "PROBLEM". A partir do momento em que o dado retorna

	para o limite atribuído a <i>trigger</i> retorna para o estado de "OK".
Evento	Uma vez que ocorre algo como a mudança de estado de uma <i>trigger</i> ou descoberta de rede.
Ação	Uma configuração de reação a um evento que executa o envio de notificações ou a execução de comandos remotos.
Mídia	Um meio de entregar uma notificação, um canal de comunicação.
Notificação	Uma mensagem a respeito de algum evento ocorrido. As mensagens são encaminhadas para os usuários por meio das mídias.
Comando remoto	Um comando programado para ser executado automaticamente quando ocorrer uma condição pré-definida.
<i>Template</i>	Um conjunto de configurações pré-definidas que pode conter itens, <i>triggers</i> , aplicações, gráficos, regras de autobusca. O <i>template</i> pode ser aplicado a um <i>host</i> ou grupo de <i>hosts</i> .

Fonte: Próprio autor

4.4 MONITORAÇÃO PERSONALIZADA

Segundo Horst; Pires; Déo (2015), um dos mais importantes avanços trazidos pelo monitoramento do Zabbix é a possibilidade de criação automática de itens que se baseia em autobusca interna no *host* monitorado (LLD). É possível mapear

virtualmente qualquer coisa com o desenvolvimento de *scripts*, além disso, o Zabbix suporta uma quantidade muito significativa de protocolos e ambientes.

As funcionalidades de monitoração são realizadas por meio de *scripts* externos que são chamados de parâmetros do usuário (*User parameter*). (HORST; PIRES; DÉO ,2015)

O agente do Zabbix permite que os parâmetros dos usuários sejam definidos em dois locais: diretamente no arquivo *zabbix_agentd.conf* ou em algum diretório de *includes* que poderá conter diversos arquivos de configuração (HORST; PIRES; DÉO, 2015).

As formas de inclusão de novas funcionalidades estão descritas na Tabela 5.

Tabela 5 - Forma de inclusão de novas funcionalidades

Parâmetro	Objetivo
<i>Include</i>	Inclui todos os arquivos de determinada localização como parâmetros de usuário
<i>UserParameter</i>	Inclui um parâmetro de usuário específico Formato: <i>UserParameter</i> =<chave>, <comando do shell/script>

Fonte: Horst; Pires; Déo (2015)

5 DESENVOLVIMENTO

Para o servidor Zabbix foi usado um PowerEdge R720 2U Rack; fonte de energia 1100W AC *hot-plug*; sistema operacional FreeBSD 10.3; processador Intel Xeon E5-2660 com frequência de 2.20GHz; 32GB de memória RAM. Um HD de 300GB para o sistema; um HD de 300GB para o *log* do PostgreSQL; um SSD de 240GB para o banco PostgreSQL; um SSD de 240GB para o histórico do zabbix; um HD de 500GB para testar *backup* e um HD de 600 para *backup*.

Na Figura 10 pode ser observado as configurações de *hardware* e o sistema operacional utilizado para o servidor Zabbix.

Figura 10: Especificações de *hardware* e SO

```
PowerEdge R720 2U Rack|
FreeBSD zabbix 10.3-RELEASE-p11 FreeBSD 10.3-RELEASE-p11 #0: Mon Oct 24 18:49:24 UTC 2016
root@amd64-builder.daemonology.net:/usr/obj/usr/src/sys/GENERIC: amd64

CPU:
hw.machine: amd64
hw.model: Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz
hw.ncpu: 16

RAM 32GB

Disco:
DESKTOP\diego.martins@zabbix:~ % gpart show -l
=> ..... 34 584843197 mfid0 GPT (279G)
..... 34 ..... 1024 ..... 1 (null) (512K)
..... 1058 ..... 4193280 ..... 2 root (2.0G)
..... 4194338 104857600 ..... 3 swap (50G)
109051938 104857600 ..... 4 tmp (50G)
213909538 41943040 ..... 5 usr (20G)
255852578 20971520 ..... 6 var (10G)
276824098 16777216 ..... 7 pg_xlog (8.0G)
293601314 291241917 ..... 8 backup (139G)

=> ..... 34 584843197 mfid1 GPT (279G)
..... 34 584843197 ..... 1 pgxlog (279G)

=> ..... 34 467664829 mfid2 GPT (223G)
..... 34 467664829 ..... 1 pgbase (223G)

=> ..... 34 467664829 mfid3 GPT (223G)
..... 34 467664829 ..... 1 zhistory (223G)

=> ..... 34 1169686461 mfid4 GPT (558G)
..... 34 1169686461 ..... 1 backup2 (558G)

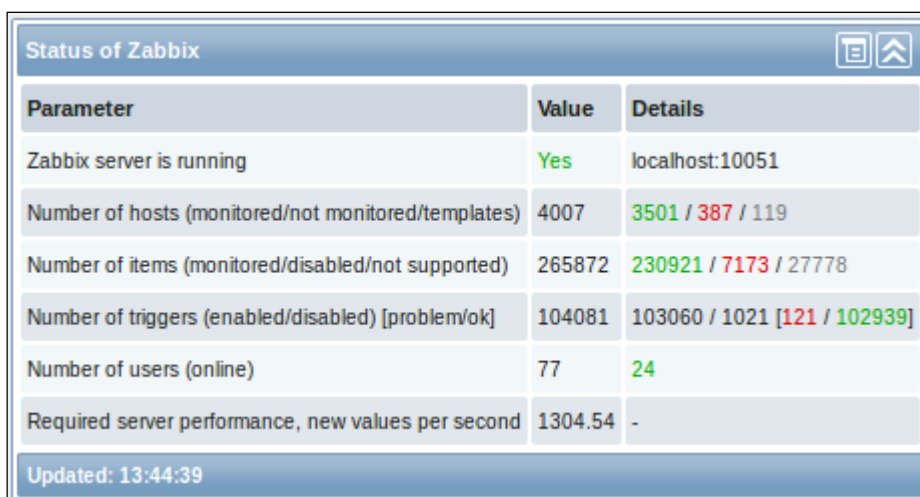
=> ..... 34 936640445 mfid5 GPT (447G)
..... 34 936640445 ..... 1 pgbase2 (447G)

Alimentação:
Auto-ranging Titanium efficiency, hot-plug redundant 1100W AC power supply;
```

Fonte: Próprio autor

Para o servidor foi usado a versão Zabbix 2.2 LTS. Na Figura 11 pode ser visualizadas as estatísticas do Zabbix quanto a quantidade de *hosts*, *itens*, *triggers*, usuários e quantidade de novos valores obtidos por segundo.

Figura 11: Informativo de *performance*



Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (monitored/not monitored/templates)	4007	3501 / 387 / 119
Number of items (monitored/disabled/not supported)	265872	230921 / 7173 / 27778
Number of triggers (enabled/disabled) [problem/ok]	104081	103060 / 1021 [121 / 102939]
Number of users (online)	77	24
Required server performance, new values per second	1304.54	-

Updated: 13:44:39

Fonte: Próprio autor

5.1 MONITORAMENTO DO ESTADO DA PORTA

Foi usado o *template padrão do Zabbix "Template SNMP Interfaces"*. Esse *template* possui a função *discovery > Item prototypes*. Essa opção cria macros para criação de item, usa a MIB e mapeia os *index* contidos na tabela da MIB. A Figura 12 mostra o exemplo de configuração para o Item *ifDescr*.

Figura 12: Configuração (*Item prototypes*)

The screenshot shows the Zabbix web interface for configuring a discovery rule. The page title is 'CONFIGURATION OF DISCOVERY RULES'. The breadcrumb trail is: Dashboard » Search » Configuration of items » Configuration of discovery rules » Configuration of triggers. The current configuration is for a discovery rule named 'Network interfaces' of type 'SNMPv2 agent'. The 'Key' field is highlighted with a red box and contains the value 'ifDescr'. Other fields include 'SNMP OID' (IF-MIB::ifDescr), 'SNMP community' ({SNMP_COMMUNITY}), 'Update interval (in sec)' (3600), and 'Filter' (Macro {#SNMPVALUE} Regexp @ExtremeInterfaces). The 'Enabled' checkbox is checked. At the bottom, there are buttons for 'Save', 'Clone', 'Delete', and 'Cancel'. The footer shows 'Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA' and 'Connected as 'diego.b''.

Fonte: Próprio autor

A Figura 13 apresenta os *Item prototypes* que vem configurados como padrão, e o nome do Item fica como descrito em *Name* e onde está `{#SNMPVALUE}` é substituído pelo *ifDescr*. Em outras palavras, a descrição da porta.

Figura 13: *Item prototypes*

The screenshot shows the Zabbix web interface for configuring item prototypes. The page title is "CONFIGURATION OF ITEM PROTOTYPES" and it shows "Item prototypes of Network interfaces". The table below lists the configured items:

Name	Key	Interval	History	Trends	Type	Applications	Status
Admin status of interface [#SNMPVALUE]	#AdminStatus[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled
Alias of interface [#SNMPVALUE]	#Alias[#SNMPVALUE]	3600	7		SNMPv2 agent	Interfaces	Enabled
Description of interface [#SNMPVALUE]	#Desc[#SNMPVALUE]	3600	7		SNMPv2 agent	Interfaces	Enabled
Inbound errors on interface [#SNMPVALUE]	#InErrors[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled
Incoming traffic on interface [#SNMPVALUE]	#InOctets[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled
Operational status of interface [#SNMPVALUE]	#OperStatus[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled
Outbound errors on interface [#SNMPVALUE]	#OutErrors[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled
Outgoing traffic on interface [#SNMPVALUE]	#OutOctets[#SNMPVALUE]	60	7	365	SNMPv2 agent	Interfaces	Enabled

Fonte: Proprio autor

A partir dos itens é possível criar as *triggers* para gerar eventos. Assim, na Figura 14 podemos visualizar como criar a *trigger* do Item descoberto *ifOperStatus[X460-24x Port 3]*, esse item foi descoberto no *host swx-bb-odessa01*. Esse item é para a porta 3 do *switch*. A função *last()* significa que estamos analisando o último dado coletado pelo item. Nesse caso estamos configurando uma porta que está com estado inativo (*down*). O retorno para porta *down* é 0, então a *trigger* foi criada com o nome "Documentar porta 3". Se o último dado for igual a 1, que é o retorno para porta ativa (*up*), a *trigger* gerará um evento para documentar a porta. Além de ser essencial para o diagnóstico, é possível manter no próprio Zabbix a documentação do *switch*. O *host* em questão é um *switch* da fabricante Extreme Networks do modelo X460, 24 portas ópticas mais 4 portas combo, que podem tanto ser ópticas como elétricas. Todas as portas com velocidade 100/1000 Mb/s.

Figura 14: *Trigger* (Documentar porta)

The screenshot displays the Zabbix web interface for configuring a trigger. The page title is 'ZABBIX' and the user is logged in as 'zabbix'. The navigation menu includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The current page is 'CONFIGURATION OF TRIGGERS' for the host 'swx-bb-odessa01'. The trigger configuration form is shown with the following details:

- Name:** Documentar porta 3
- Expression:** {swx-bb-odessa01:ifOperStatus[X460-24x Port 3].last()}=1
- Multiple PROBLEM events generation:**
- Description:** (empty text area)
- URL:** (empty text field)
- Severity:** Not classified, Information, Warning, Average, **High**, Disaster
- Enabled:**

Buttons at the bottom include 'Save', 'Clone', 'Delete', and 'Cancel'. The footer shows 'Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA' and 'Connected as 'diego.b''.

Fonte: Proprio autor

Na Figura 15 foi realizada a configuração da porta 19 do *host* swx-bb-odessa01. O nome "Port 19 Down: Link Eth3 Odessa-R3 (PPPoE)" significa que o equipamento que está ligado nessa porta é a porta 3 do *host* Odessa-R3 e o operador #1 significa diferente de 1, ou seja, quando a porta estiver inativa gerará um evento.

Figura 15: *Trigger (Porta Down)*

The screenshot displays the Zabbix web interface for configuring a trigger. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Configuration' section is active, showing a breadcrumb trail: 'Configuration of Items » Configuration of discovery rules » Configuration of triggers » Configuration of discovery rules » Configuration of triggers'. The main content area is titled 'CONFIGURATION OF TRIGGERS' and shows the configuration for a specific trigger. The trigger name is 'Port 19 Down: Link Eth3 Odessa-R3 (PPPoE)'. The expression is '{swx-bb-odessa01:ifOperStatus[X460-24x Port 19].last()}#1', which is highlighted with a red box. The trigger is set to 'High' severity and is enabled. The description is 'Link Eth3 Odessa-R3 (PPPoE)'. The interface also shows buttons for 'Save', 'Clone', 'Delete', and 'Cancel'.

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | Configuration | Administration | zabbix

Host groups | Templates | **Items** | Maintenance | Actions | Screens | Search

Slide shows | Maps | Discovery | IT services

History: Configuration of Items » Configuration of discovery rules » Configuration of triggers » Configuration of discovery rules » Configuration of triggers

CONFIGURATION OF TRIGGERS

« [Host list](#) **Host:** [swx-bb-odessa01](#) **Monitored** [Applications \(12\)](#) [Items \(617\)](#) [Triggers \(324\)](#)

[Graphs \(203\)](#) [Discovery rules \(3\)](#) [Web scenarios \(0\)](#)

Trigger | Dependencies

Name: Port 19 Down: Link Eth3 Odessa-R3 (PPPoE)

Expression: {swx-bb-odessa01:ifOperStatus[X460-24x Port 19].last()}#1

[Expression constructor](#)

Multiple PROBLEM events generation:

Description: Link Eth3 Odessa-R3 (PPPoE)

URL:

Severity:

Enabled:

Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA | Connected as 'diego.b'

Fonte: Próprio autor

Na Figura 16 podemos verificar o evento do *host* swx-bb-nsdore01 o nome da *trigger* “Port 33 Down: Link MPLS FastEthernet -> Desktop”.

Figura 16: Evento (Porta Down)

3. Switches Extreme	0	2	2	7	0	0	
4. Backbone/ Agregação	0	0					Close
5. Mikrotiks	0	0					
6. Switches PPPoE	0	3					
7. Cluster DSLAMs	0	0					

Host	Issue	Age	Info	Ack	Actions
swx-bb-nsdore01	Service Down: Tunel MPLS Fasternet NSDores -> Ouroverde Down	4h 49m 8s		Yes (1)	-
swx-bb-nsdore01	Port 33 Down: Link MPLS FastEthernet -> Desktop	4h 49m 37s		Yes (1)	-

Fonte: Próprio autor

Uma porta ativa, ao mudar para o estado de inativa, afeta a disponibilidade do serviço. Com o monitoramento, alarmes e eventos é possível uma tomada de decisão mais rápida. Uma porta inativa, ao mudar para o estado de ativa, afeta a confiabilidade. Monitorar o estado da porta é essencial para garantir maior segurança quanto a: confiabilidade e disponibilidade.

5.2 SCRIPT DE INFORMAÇÃO DO TRANSCEIVER

No *switch* da fabricante Extreme Networks algumas informações não estão disponíveis via SNMP. Portanto, essas só podem ser visualizadas via terminal por linha de comando, como por exemplo, as informações do *transceiver* (*interface* óptica de rede): temperatura, potência de sinal Tx e Rx em dBm, tensão e corrente. Como mostra na Figura 17, foi criado um *script* python e adicionado a rotina no crontab para ser executado a cada 5 minutos.

A variável *host* guarda uma lista de ip.

A variável *cmds* guarda uma lista de comandos.

Foi criado duas funções: *telnet* e *sfp*.

A função *telnet* é responsável por acessar os *switches* da variável *hosts* e executar os comandos que foram definidos na variável *cmds* o resultado dos comandos é guardado na variável *show*.

A função *sfp* é responsável por processar o conteúdo da variável *show* e retornar um dicionário contendo as informações do *transceiver*.

Foi criado um laço para repetir esse mesmo processo para todos os *hosts* e no laço a função *open* cria um arquivo cujo o nome é o ip do *host* no caminho

"/usr/local/etc/zabbix/externalscripts/extreme/" e salva o dicionário no arquivo (Figura 18). Sendo assim é criado um arquivo para cada host.

Figura 17: Script de acesso (Telnet)

```
#!/usr/local/bin/python
# -*- coding: utf-8 -*-
import sys, re, telnetlib

def telnet(host, cmds):
    user = "<usuario>"
    password = "<senha>"
    tn = telnetlib.Telnet(host, 23, 5)
    tn.read_until("login: ", 3)
    tn.write(user + "\n")
    if password:
        tn.read_until("password: ", 3)
        tn.write(password + "\n")
    show = []
    for cmd in cmds:
        tn.read_until("#", 1)
        tn.write(cmd + "\n")
        cli = tn.read_until("#", 2)
        show.append(str(cli))
    return show

def sfp(show):
    dic = {}
    for i in range(5, 11):
        port = re.findall('\d\d\d+', show[0].split('\n')[i])[0].split()
        if len(port) == 8:
            dic[port[0]] = {"temp" : "0", "txpower" : "0", "rxpower" :
                "0", "txbiascurrent" : "0", "voltage-aux1" : "0"}
        if len(port) == 7:
            dic[port[0]] = {"temp" : port[1], "txpower" : port[2],
                "rxpower" : port[3], "txbiascurrent" : port[4],
                "voltage-aux1" : port[5]}
    return dic

hosts = [<lista_de_hosts>]

for host in hosts:
    cmds = ["show ports 29-34 transceiver information", "show fdb stats"]
    show = telnet(host, cmds)
    dic = sfp(show)
    dic["mac"] = re.findall('Total: \d+', show[1])[0][7:]
    o = open(
        "/usr/local/etc/zabbix22/zabbix/externalscripts/extremes/%s.txt" %
        host, 'w')
    o.write(str(dic))
    o.close
```

Fonte: Próprio autor

Na Figura 18 possui o conteúdo do arquivo salvo pelo script.

Figura 18: Arquivo (lista de informações)

```
{'33': {'txpower': '0', 'voltage-aux1': '0', 'rxpower': '0',  
'temp': '0', 'txbiascurrent': '0'}, '32': {'txpower': '-1.72',  
'voltage-aux1': '3.31', 'rxpower': '-0.79', 'temp': '43.73',  
'txbiascurrent': '29.20'}, '31': {'txpower': '-1.65',  
'voltage-aux1': '3.32', 'rxpower': '-6.51', 'temp': '41.63',  
'txbiascurrent': '45.77'}, '30': {'txpower': '-1.55',  
'voltage-aux1': '3.34', 'rxpower': '-2.00', 'temp': '39.45',  
'txbiascurrent': '39.90'}, '29': {'txpower': '-1.41',  
'voltage-aux1': '3.35', 'rxpower': '-0.08', 'temp': '38.20',  
'txbiascurrent': '36.24'}, '34': {'txpower': '-1.48',  
'voltage-aux1': '3.29', 'rxpower': '-1.81', 'temp': '37.27',  
'txbiascurrent': '39.18'}, 'mac': '3102'}
```

Fonte: Próprio autor

No arquivo de configuração do Zabbix `/usr/local/etc/zabbix22/zabbix_server.conf` foi adicionada a linha `ExternalScripts=/usr/local/etc/zabbix22/zabbix/externalscripts/`, que significa que os itens *external check* do Zabbix, vão rodar nesse diretório. Na Figura 19 podemos ver o *script* que o zabbix usa para criar o item. Usando os dados que foram salvos no arquivo na etapa anterior (Figura 18).

O *script* recebe três ou quatro argumentos: O primeiro é o ip do *host*, o segundo é o número da porta (caso sejam apenas três argumentos, e o segundo argumento for o valor 'mac', o script retorna a quantidade de macs alocados na porta), o terceiro é o tipo do serviço (*txpower*, *voltage-aux1*, *rxpower*, *temp*, *txbiascurrent*), e o quarto é o *status*.

Figura 19: *Script external check (Transceiver information)*

```
#!/usr/local/bin/python
#-*- coding: utf-8 -*-
import sys, re

def read(host):
    o = open("/usr/local/etc/zabbix22/zabbix/externalscripts/extremes/%s.txt" % host, "r")
    dic = eval(o.read())
    o.close()
    return dic

if __name__ == '__main__':
    if len(sys.argv) in range(3, 6):
        host = sys.argv[1]
        port = sys.argv[2]
        if len(sys.argv) == 4:
            service = sys.argv[3]
        if len(sys.argv) == 5:
            service = sys.argv[3]
            status = sys.argv[4]
        else:
            print "usage: host port/mac [port:service] [status]"
            quit()

    dic = read(host)
    if len(sys.argv) == 3 and port == "mac":
        print dic[port]

    if len(sys.argv) == 4:
        sfp = dic[port][service]
        if sfp[-1:] == "***":
            print sfp[:-1]
        else:
            print sfp

    if len(sys.argv) == 5 and status == "status":
        sfp = dic[port][service]
        if sfp[-1:] == "***":
            print "1"
        else:
            print "0"
```

Fonte: Próprio autor

Os Itens são criados no *template* “Template DSK Switch Extreme” a fim de garantir que os Itens sejam usados em todos os *switches* que usam esse *template*. Na Figura 20 podemos visualizar alguns desses itens.

Figura 20: Itens (*Transceiver information*)

The screenshot shows the Zabbix web interface for configuring items. The page title is 'CONFIGURATION OF ITEMS' and it displays 74 items. The table below lists the items, their triggers, keys, intervals, and other configuration details.

Wizard	Name	Triggers	Key	Interval	History	Trends	Type	Applications	Status
<input type="checkbox"/>	Transceiver information TxBiasCurrent Status Port 29	Triggers (1)	extreme["{HOST.IP}",29,txbiascurrent,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information TxPower Port 30		extreme["{HOST.IP}",30,txpower]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Voltage-Aux1 Port 31		extreme["{HOST.IP}",31,voltage-aux1]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information TxPower Status Port 30	Triggers (1)	extreme["{HOST.IP}",30,txpower,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Temperature Status Port 30	Triggers (1)	extreme["{HOST.IP}",30,temp,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information RxPower Status Port 31	Triggers (1)	extreme["{HOST.IP}",31,rxpower,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Temperature Status Port 33	Triggers (1)	extreme["{HOST.IP}",33,temp,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Temperature Status Port 31	Triggers (1)	extreme["{HOST.IP}",31,temp,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information RxPower Status Port 29	Triggers (1)	extreme["{HOST.IP}",29,rxpower,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information TxBiasCurrent Port 29		extreme["{HOST.IP}",29,txbiascurrent]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Temperature Status Port 29	Triggers (1)	extreme["{HOST.IP}",29,temp,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information RxPower Port 32		extreme["{HOST.IP}",32,rxpower]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information Voltage-Aux1 Status Port 29	Triggers (1)	extreme["{HOST.IP}",29,voltage-aux1,status]	300	7	365	External check	Transceiver	Enabled
<input type="checkbox"/>	Transceiver information TxPower Status Port 29	Triggers (1)	extreme["{HOST.IP}",29,txpower,status]	300	7	365	External check	Transceiver	Enabled

Fonte: Próprio autor

Na Figura 21 foi feita a criação do Item. O tipo de verificação utilizada foi o *External check*. Na opção *key* é passado o nome do *script* criado (Figura 19), seguido das variáveis entre chaves.

Figura 21: Criação de Item (*Transceiver information*)

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | Configuration | Administration | zabbix

Host groups | Templates | Hosts | Maintenance | Actions | Search

Screens | Slide shows | Maps | Discovery | IT services

History: Custom screens » Search » Configuration of templates » Configuration of triggers » Configuration of items

CONFIGURATION OF ITEMS

« [Template list](#) **Template:** [Template DSK Switch Extreme](#) [Applications](#) (12) [Items](#) (74) [Triggers](#) (42)

[Graphs](#) (28) [Screens](#) (0) [Discovery rules](#) (3) [Web scenarios](#) (0)

Item

Name:

Type:

Key:

Type of information:

Data type:

Units:

Use custom multiplier:

Update interval (in sec):

Flexible intervals

Interval	Period	Action
No flexible intervals defined.		

New flexible interval

Interval (in sec)	<input type="text" value="50"/>	Period	<input type="text" value="1-7,00:00-24:00"/>	<input type="button" value="Add"/>
-------------------	---------------------------------	--------	--	------------------------------------

History storage period (in days):

Trend storage period (in days):

Store value:

Show value: [show value mappings](#)

New application:

Applications

- InterfacesInfo
- Nexthop
- Processors
- PW
- Sfp
- Transceiver**

Fonte: Próprio autor

Foi criado *triggers* para *status*. Assim que algum item esteja fora do limite gerará um evento, como pode ser visualizado na Figura 22.

Figura 22: Trigger (Transceiver information RxPower Status)

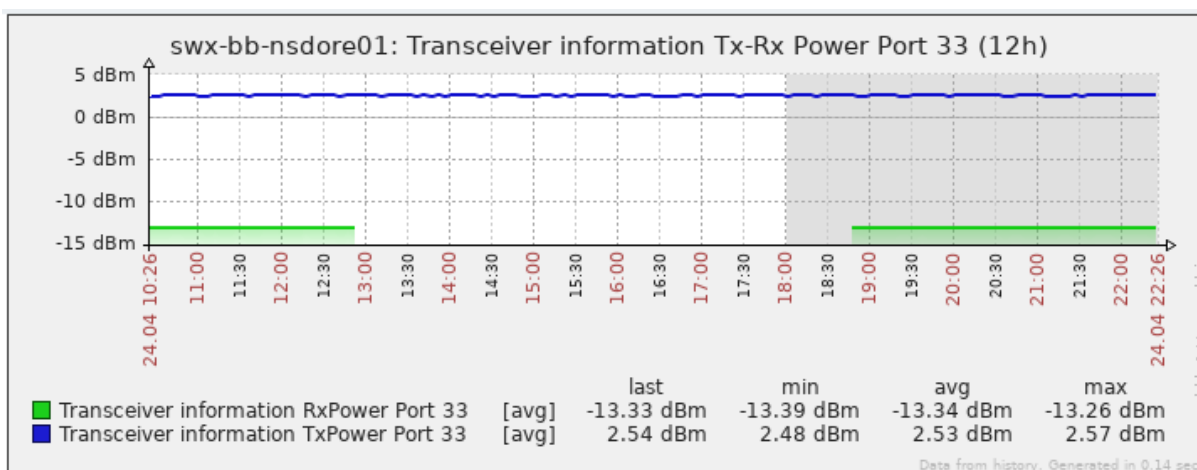
System status						
Host group	Disaster	High	Average	Warning	Information	Not classified
1. CPD Network	0	0	0	10	2	0
2. Monitoramento Elétrico	0	0	1	12	0	0
3. Switches Extreme	0	2	2	8	0	0
4. Backbone/ Agregação	0	0	0	Close		
5. Mikrotiks	0	0	0			
6. Switches PPPoE	0	3	0			
7. Cluster DSLAMs	0	0	1			

Host	Issue	Age	Info	Ack	Actions
swx-bb-merced02	CPU is too high on swx-bb-merced02	4m 4s		No	-
swx-bb-nsdore01	Transceiver information RxPower Status Port 33	4h 47m 43s		No	-

Fonte: Próprio autor

Com o gráfico da Figura 23 pode ser observado que existe uma interrupção devida ao evento ocorrido (Figura 22).

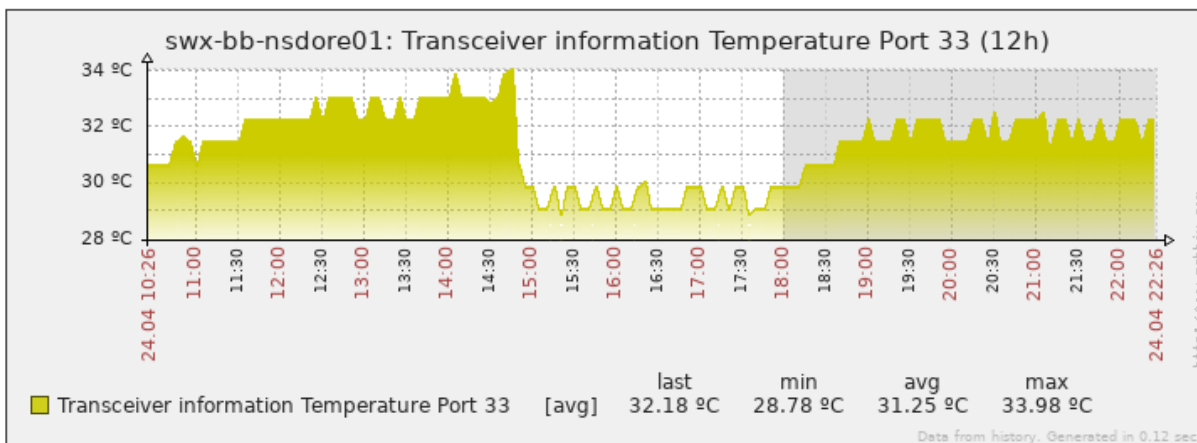
Figura 23: Gráfico de informação de potência de sinal do *transceiver*



Fonte: Próprio autor

A Figura 24 mostra o gráfico de temperatura do *transceiver*.

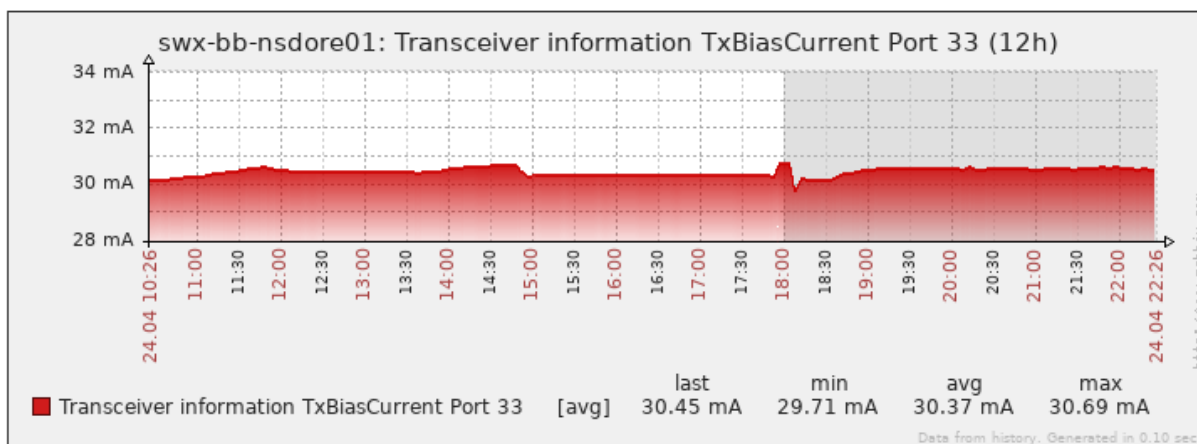
Figura 24: Gráfico de informação de temperatura do *transceiver*



Fonte: Próprio autor

A Figura 25 mostra o gráfico de corrente do *transceiver*.

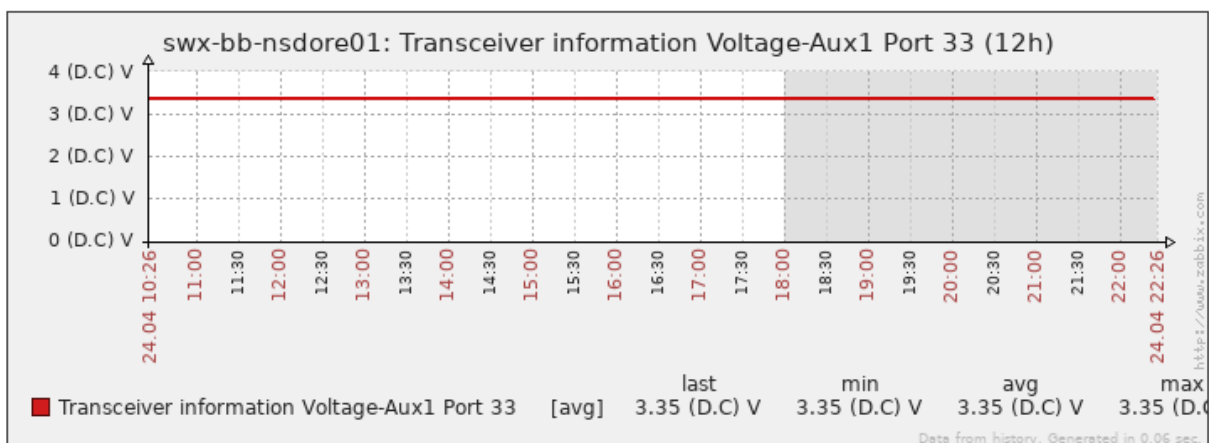
Figura 25: Gráfico de informação de corrente do *transceiver*



Fonte: Próprio autor

A Figura 26 mostra o gráfico de tensão do *transceiver*.

Figura 26: Gráfico de informação de tensão do *transceiver*



Fonte: Próprio autor

O evento gerado a Figura 22 ocorreu devido a um rompimento de fibra óptica. A figura 23 mostra que o gráfico de transmissão (potência de sinal) não foi interrompido, pois o *transceiver* continua funcionando, porém o de recepção (potência de sinal) foi interrompido devido à falta de comunicação. Na figura 24 é possível observar que o gráfico de temperatura também se alterou com uma diminuição de temperatura, pois não está recebendo a potência do laser. Na Figura 25 o gráfico de corrente aumentou e diminuiu, pois percebeu a alteração de potência e tentou compensar a diferença somente estabilizando após o retorno da comunicação. O gráfico de tensão (Figura 26) não se alterou, pois a tensão do *transceiver* continua a mesma.

O monitoramento do *transceiver* é importante para identificar problemas no próprio *transceiver* e também na fibra óptica, assim possibilitando uma manutenção preventiva ou corretiva com eficiência e maior agilidade, assim garantindo maior disponibilidade do serviço.

5.3 CHECKSUM /ETC/PASSWD

O Zabbix possui um *template* padrão para monitorar servidores Linux, o *template* “Temaplte OS Linux”. Esse *template* possui um item que auxilia o profissional de segurança da informação, que é o item Checksum of <diretório que deseja monitorar>. O monitoramento é feito através do agente Zabbix, e o diretório padrão que vem configurado é o /etc/passwd, que guarda as informações de usuário do Linux. A Figura 27 mostra o item padrão.

Figura 27: Item (*Checksum*)

Parent items [Template OS Linux](#)

Name

Type

Key

Host interface

Type of information

Data type

Units

Use custom multiplier

Update interval (in sec)

Flexible intervals

Interval	Period	Action
No flexible intervals defined.		

New flexible interval Interval (in sec) Period

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- General
- Memory
- Network interfaces
- OS
- Performance
- Processes
- Security

Fonte: Próprio autor

Na Figura 28 temos a trigger padrão para o item. Caso ocorra uma modificação no dado recebido pelo item ocorrerá um evento. Ou seja, se um usuário for criado, apagado ou modificado, o Zabbix irá informar.

Figura 28: Trigger (Checksum)

The screenshot shows the configuration page for a trigger in Zabbix. The parent trigger is 'Template OS Linux'. The trigger name is '/etc/passwd has been changed on {HOST.NAME}'. The expression is '{Web_server_teste:vfs.file.cksum[/etc/passwd].diff(0)}>0', which is highlighted with a red box. There is an 'Add' button next to the expression field. Below the expression field is the 'Expression constructor' link. The 'Multiple PROBLEM events generation' checkbox is unchecked. The 'Description' field is empty. The 'URL' field is empty. The 'Severity' is set to 'Warning'. The 'Enabled' checkbox is checked. At the bottom, there are buttons for 'Save', 'Clone', 'Delete', and 'Cancel'. The footer of the page reads 'Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA'.

Fonte: Próprio autor

Foi criado um usuário no host "Web_server_teste". A Figura 29 mostra o evento que foi gerado.

Figura 29: Evento (*Checksum*)

System status						
Host group	Disaster	High	Average	Warning	Information	Not classified
1_CPD Network	3	1	0	13	2	0
2_Monitoramento Elétrico	0	0	1	12		
3_Switches Extreme	2	3	2	8		
4_Backbone/ Agregação	0	0	0	1		
5_Mikrotiks	0	0	0	0		
6_Switches PPPoE	0	8	0	0		
7_Cluster DSLAMs	0	0	0	0		
9_Bases RF	0	1	4	9		
10_Conds Cab. Estruturado	0	2	0	0		
Cliente Pref. Indaiatuba	0	0	0	0		
Clientes Premium	0	11	0	2		
Clientes VIP	1	0	0	0		
Discovered hosts	0	3	0	0		
Headend	0	0	0	3		
LAN Services	0	3	0	0		
PMNO - Paco Municipal	0	0	0	0		

Host	Issue	Age	Info	Ack	Actions
swoperadoras	Inbound errors exceed limit on swoperadoras interface xe-0/0/45	58s		No	-
swoperadoras	Inbound errors exceed limit on swoperadoras interface ae7	58s		No	-
BRAS pppoe03	Inbound errors exceed limit on BRAS pppoe03 interface xe-2/0/3	4m 57s		No	-
Web_server_teste	/etc/passwd has been changed on Web_server_teste	5m 19s		No	-
FreeNAS	Free disk space is less than 20% on volume Real memory	8h 2m 12s		No	-
Web_server_teste	Free disk space is less than 20% on volume /	17h 3m 44s		No	-
BRAS pppoe03	Inbound errors exceed limit on BRAS pppoe03 interface xe-2/0/1	2d 49m		No	-
BRAS pppoe01	Inbound errors exceed limit on BRAS pppoe01 interface xe-0/0/1	2d 9h 10m		No	-
FreeNAS	Free disk space is less than 20% on volume /mnt/Storage/jails/backupcore/proc	8m 26d 18h		No	-
FreeNAS	Free disk space is less than 20% on volume /mnt/Storage/jails/backupsvr/proc	8m 26d 18h		No	-
FreeNAS	Free disk space is less than 20% on volume /mnt/Storage/jails/backupsvr/dev	8m 26d 18h		No	-
FreeNAS	Free disk space is less than 20% on volume /mnt/Storage/jails/backupcore/dev	8m 26d 19h		No	-
FreeNAS	Free disk space is less than 20% on volume /dev	10m 3d 18h		No	-

Fonte: Próprio autor

O evento gerado na Figura 29 é um indicativo que a integridade do sistema foi afetada, gerando um problema de segurança.

5.4 SCRIPT NOTIFICAÇÃO SMS

No arquivo de configuração do Zabbix `/usr/local/etc/zabbix22/zabbix_server.conf` foi adicionada a linha `AlertScriptsPath=/usr/local/etc/zabbix22/zabbix/alertscripts`.

Foi criado um *script* que envia mensagens e adicionado no servidor via terminal no caminho `/usr/local/etc/zabbix22/zabbix/alertscripts/zenvia_sms.sh`. A Figura 30 mostra o conteúdo do *script*.

O *script* recebe três argumentos, número de telefone, título da mensagem e conteúdo da mensagem, e envia essas informações através da *url*. O servidor da empresa Zenvia recebe as informações e envia a mensagem sms.

Figura 30: Script que envia mensagens SMS

```
#!/bin/sh
DST="$1"
SUBJ="$2"
MSG="$3"

# "dispatch=send" indica que é envio simples
URL="http://api.zenvia360.com.br/GatewayIntegration/msgSms.do?dispatch=send"
ACCOUNT="<username>"
CODE="<password>"

curl -s -d "&account=${ACCOUNT}&code=${CODE}&to=${DST}&from=${SUBJ}&msg=${MSG}" -> ${URL} >/dev/null

# Debug
#curl -d "&account=${ACCOUNT}&code=${CODE}&to=${DST}&from=${SUBJ}&msg=${MSG}" -> ${URL}
```

Fonte: Proprio autor

Foi configurado o número do telefone celular do usuário que irá receber a mensagem. Como mostra a Figura 31, na opção *Administration > Users > Media*: é inserido o nome da mídia a fim de identificação, o telefone com o código do país e o DD seguido do número de celular. Também é adicionada a política de atendimento do usuário que foi definida como 7 dias da semana e 24 horas por dia. Significa que o usuário receberá todas as mensagens a qualquer hora.

Figura 31: Administração de usuários (mídia)

The screenshot shows the Zabbix Administration interface. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Administration' section is active, and the 'Users' sub-section is selected. The 'Media' tab is active, showing a table of media configurations. The table has columns for 'Media', 'Name', 'Phone', 'Time', 'Policy', 'Status', and 'Action'. Two entries are visible, both for 'SMS_Zenvia' with phone numbers 5519982955964 and 5519993818376, and a policy of 'NIWAHD' and 'Enabled' status. Below the table are 'Add' and 'Delete selected' links, and 'Save', 'Delete', and 'Cancel' buttons.

Media	Name	Phone	Time	Policy	Status	Action
<input type="checkbox"/>	SMS_Zenvia	5519982955964	1-7,00:00-24:00	NIWAHD	Enabled	Edit
<input type="checkbox"/>	SMS_Zenvia	5519993818376	1-7,00:00-24:00	NIWAHD	Enabled	Edit

Buttons: Save, Delete, Cancel

Footer: Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA | Connected as 'diego.b'

Fonte: Próprio autor

O servidor Zabbix executa ações conforme os alarmes através das mídias, na opção *Administration > Media types*, quem pode ser Email, Jabber, mensagem SMS através de um modem 3G ou outra mídia. No caso, foi criada uma nova mídia como mostra a Figura 32.

Figura 32: Tipos de mídias

ZABBIX Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | Configuration | Administration | zabbix

General | DM | Authentication | Users | **Media types** | Scripts | Audit | Search

Queue | Notifications | Installation

History: Dashboard » Search » Dashboard » Configuration of user groups » Configuration of users

CONFIGURATION OF MEDIA TYPES [Create media type](#)

Media types

Displaying 1 to 4 of 4 found

<input type="checkbox"/>	Name	Type	Status	Used in actions	Details
<input type="checkbox"/>	Email	Email	Enabled	-	SMTP server: "smtp.desktop.com.br", SMTP helo: "zabbix.desktop.com.br"
<input type="checkbox"/>	Jabber	Jabber	Enabled	-	Jabber identifier: "jabber@company.com"
<input type="checkbox"/>	SMS	SMS	Enabled	-	GSM modem: "/dev/ttyS0"
<input type="checkbox"/>	SMS_Zenvia	Script	Enabled	SMS_Zenvia	Script name: "zenvia_sms.sh"

Enable selected

Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA | Connected as 'diego.b'

Fonte: Próprio autor

A Figura 33 mostra a tela de criação da mídia. O nome é para identificação, o tipo da mídia no caso *script* e o nome do *script* do servidor.

Figura 33: Criação da mídia SMS

The screenshot displays the Zabbix web interface for configuring a media type. The page title is "CONFIGURATION OF MEDIA TYPES". The breadcrumb trail is "History: Search » Dashboard » Configuration of user groups » Configuration of users » Configuration of media types". The form fields are as follows:

Name	<input type="text" value="SMS_Zenvia"/>
Type	<input type="text" value="Script"/>
Script name	<input type="text" value="zenvia_sms.sh"/>
Enabled	<input checked="" type="checkbox"/>

Buttons: Save, Delete, Cancel

Footer: Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA | Connected as 'diego.b'

Fonte: Próprio autor

Para definir o padrão da mensagem, este deve ser configurado. Na opção Configuration > Actions > na aba Action, são configuradas as variáveis que serão encaminhadas na mensagem. No exemplo da Figura 34 o nome é para identificação, Default subject: é o nome do host que está com problema, Default message: Foi configurado o *status* e o nome da *trigger*. Essas configurações foram definidas tanto na notificação de problema quanto na notificação de *recover*. Para ser mais fácil a identificação.

Figura 34: Configuração da ação (SMS)

The screenshot displays the Zabbix web interface for configuring an action. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Configuration' menu is active, and the 'Actions' sub-menu is selected. The breadcrumb trail shows the path: Configuration of user groups » Configuration of users » Configuration of media types » Configuration of host groups » Configuration of actions.

The main content area is titled 'CONFIGURATION OF ACTIONS' and features three tabs: 'Action', 'Conditions', and 'Operations'. The 'Action' tab is selected, showing the following configuration details:

- Name:** SMS_Zenvia
- Default subject:** {HOST.NAME1}
- Default message:** Trigger status: {TRIGGER.STATUS}
Trigger: {TRIGGER.NAME}
- Recovery message:** (checked)
- Recovery subject:** {HOST.NAME1}
- Recovery message:** Trigger status: {TRIGGER.STATUS}
Trigger: {TRIGGER.NAME}
- Enabled:** (checked)

At the bottom of the configuration form, there are four buttons: 'Save', 'Clone', 'Delete', and 'Cancel'. The footer of the page indicates 'Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA' and shows the user is connected as 'diego.b'.

Fonte: Próprio autor

Na aba *conditions* é necessário configurar as condições para a notificação. No exemplo foram configurados alertas (*Trigger*) da Label A á D com o operador OR, os *hosts* de E a AH também separados com operador OR, e Alertas e *hosts* separados com operador AND, como pode ser visualizado na Figura 6. Isso significa que as condições para ativar essa notificação, pelo menos uma *trigger* tem que coincidir com um *host*. Exemplo: Label C (Trigger name like is unavailable) e Label Q (Host = FW8), nessas condições geraria uma notificação como mostra na Figura 35.

Figura 35: Condições da ação

zabbix.desktop.com.br/zabbix/actionconf.php?form=update&actionid=45&sid=e3efc5

ZABBIX

Monitoring | Inventory | Reports | Configuration | Administration

Host groups | Templates | Hosts | Maintenance | Actions | Screens | Slide shows | Maps | Discovery | IT services

History: Dashboard » Configuration of scripts » Configuration of media types » Configuration of actions » Configuration of media types

CONFIGURATION OF ACTIONS

Action | Conditions | Operations

Type of calculation: (A or B or C or D) and
(E or F or G or H or I or J or K or L or M or N or O or P or Q or R or S or T or U or V or W or X or Y or Z or AA or AB or AC or AD or AE or AF or AG or AH)

Label	Name	Action
(A)	Trigger name like Yellow Alarm	Remove
(B)	Trigger name like Red Alarm	Remove
(C)	Trigger name like is unavailable	Remove
(D)	Trigger name like AC power failure - No Power	Remove
(E)	Host = BRAS pppoe02	Remove
(F)	Host = nb3-he-rack01	Remove
(G)	Host = MX104	Remove
(H)	Host = nb1-cp-rack00	Remove
(I)	Host = BRAS pppoe01	Remove
(J)	Host = nb9-cp-rack04	Remove
(K)	Host = nb8-cp-rack04	Remove
(L)	Host = nb11-cp-rack01	Remove
(M)	Host = ats-cp-stfc01	Remove
(N)	Host = nb2-cp-rack00	Remove
(O)	Host = nb4-cp-rack02	Remove
(P)	Host = swoperadoras	Remove
(Q)	Host = FW8	Remove
(R)	Host = nb2-he-rack01	Remove
(S)	Host = nb1-he-rack01	Remove
(T)	Host = nb1-he-sigmanet (sdbt)	Remove
(U)	Host = ats-he-rack06	Remove
(V)	Host = ats-he-rack05	Remove
(W)	Host = ats-he-rack04	Remove
(X)	Host = ats-he-rack03	Remove
(Y)	Host = ats-he-rack02	Remove
(Z)	Host = ats-he-rack01	Remove
(AA)	Host = MX80	Remove
(AB)	Host = FW5	Remove
(AC)	Host = FW4	Remove
(AD)	Host = FW1 OLD	Remove
(AE)	Host = FW7	Remove
(AF)	Host = FW3	Remove
(AG)	Host = FW9	Remove
(AH)	Host = FW2	Remove

New condition: like [Add](#)

[Save](#) [Clone](#) [Delete](#) [Cancel](#)

Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA

Fonte: Próprio autor

Na aba *Operations* são configurados os usuários e mídia usada para receber a notificação. No caso, para todos os usuários será usada a mídia SMS_Zenvia, que foi criada na Figura 36.

Figura 36: Operações da ação

Default operation step duration: (minimum 60 seconds)

Action operations

Steps	Details	Start in	Duration (sec)	Action
1	Send message to users: diego.b (Diego Bruno gr-ti) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: phimenoni (Paulo Menoni) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: alexandre.infra (Alexandre Justino gr-ti) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: lbraz (Leandro Braz) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: tbraga (Thiago Braga) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: wpereira (William Pereira gr-noc) via SMS_Zenvia	Immediately	Default	Edit Remove
1	Send message to users: klaus (Klaus Klaus) via SMS_Zenvia	Immediately	Default	Edit Remove

[New](#)

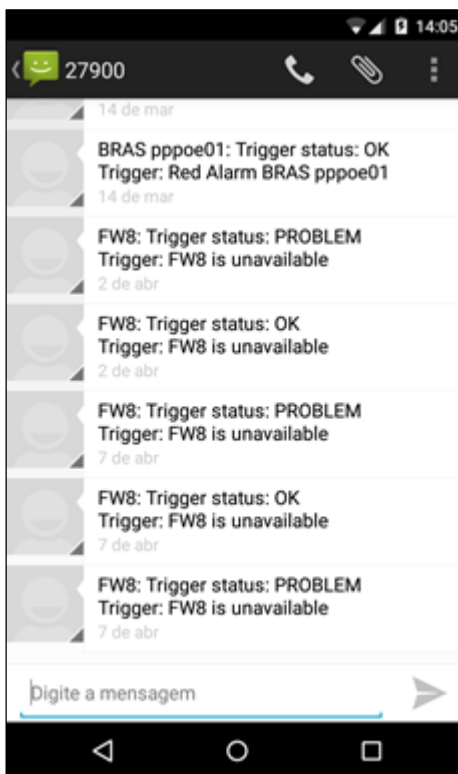
[Save](#) [Clone](#) [Delete](#) [Cancel](#)

Zabbix 2.2.15 Copyright 2001-2016 by Zabbix SIA | Connected as 'diego.b'

Fonte: Próprio autor

Na Figura 37, pode-se observar as mensagens SMS recebidas no celular.

Figura 37: Mensagens SMS recebidas



Fonte: Próprio autor

O script notificação sms contribui para uma maior disponibilidade dos serviços e tomada de ação mais rápida.

É possível criar escalabilidade de eventos para enviar notificações de tempos em tempos para diferentes níveis de funcionários, começando pelo analista até chegar ao gerente ou diretor.

As principais vantagens da ferramenta são: É um software de código aberto e gratuito, as atualizações são disponibilizadas no site e pode ser atualizado sem custos. Possui uma documentação bem completa, e é fácil encontrar material de apoio na internet.

A desvantagem do Zabbix: por ser uma ferramenta com muitos recursos as vezes não é muito atrativa para novos usuários. O primeiro contato aparenta ser uma ferramenta muito complexa e difícil de usar.

6. CONSIDERAÇÕES FINAIS

O Zabbix mostrou ser uma ótima ferramenta de monitoramento e muito simples de ser customizada. Foram desenvolvidos *scripts* a fim de monitorar serviços que não são possíveis serem monitorados com o agente padrão da ferramenta.

Com isso, foi viável criar monitoramento de portas do *switch* buscando garantir controle das *interfaces* que estão em uso e qual serviço está atrelado a porta, além de maior disponibilidade do serviço.

Através de *scripts* foi criado monitoramento de *status* de equipamentos. Também foi desenvolvido um sistema de encaminhamento de mensagens dos eventos da rede, buscando auxiliar na tomada de ações e aumentando a disponibilidade dos ativos da rede.

Usando o monitoramento padrão do Zabbix que monitora arquivos do sistema Linux foi possível ao administrador obter controle sobre a integridade do sistema e garantir segurança dos dados.

Apesar do Zabbix não ser uma ferramenta de segurança, ele possui alguns recursos de monitoramentos que foram usados nesse trabalho, e que permitiram a criação de mecanismos de segurança que atendem a uma empresa de *internet*.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. NBR ISO/IEC 17799 - **Tecnologia da informação**: código de prática para a gestão da segurança da informação. Rio de Janeiro: ABNT, 2005

ALBUQUERQUE, F. **TCP/IP Internet Protocolos & Tecnologias**. 3ª edição, 2001.

AMARAL, A.F.F. **Redes de Computadores**. Vitória/ES: Instituto Federal do Espírito Santo, 2012. 81 p.

BUENO, E. M. **Monitoramento de redes de computadores com uso de ferramentas de software livre**. Monografia de especialização. Departamento acadêmico de eletrônica. Universidade Federal do Paraná, 2012.

BONOMO, E. **Gerenciamento e monitoração de redes de computadores utilizando-se zabbix**. Monografia de especialização. Universidade federal de lavras, 2006.

COMER, D.E. **Rede de computadores e Internet**: abrange transmissão de dados, ligações inter-redes; Tradução Álvaro Strube de Lima. 4. ed. Porto Alegre: Bookman, 2007.

Contessa, D. F. ; Polina, E. R. **Gerenciamento de Equipamentos Usando o Protocolo SNMP**, 2006. Disponível em: <<http://docplayer.com.br/537322-Gerenciamento-de-equipamentos-usando-o-protocolo-snm.html>>. Acesso em: 22 abr. 2017.

DANTAS, M. L. **Segurança da informação**: uma abordagem focada em gestão de riscos. Olinda: Livro Rápido, 2011.

FERREIRA, Aurélio Buarque H. **Dicionário Aurélio da Língua Portuguesa**. 5ª edição, 2010.

FRANCISCATTO, R. Cristo, F. Perlin, T. **Rede de computadores**. Universidade Federal de Santa Maria, Colégio Agrícola de Frederico Westphalen, 2014. P-15-16.

HORST, A.H.S.; PIRES, A.S.; DÉO, A.L.B. De A a ZABBIX . São Paulo: Novatec Editora, 2015.

HUNECKE, M. **Rede de computadores**: Conceitos gerais. Casa do concurseiro. Disponível em: <<https://estudaquepassa.com.br/concursos/5492/nocoos-de-informatica-marcio-hunecke-redes-de-computadores-conceitos-gerais>>. Acesso em: 22 abr. 2017.

INTERNET ENGINEERING TASK FORCE. **RFC1157**, 1990. Disponível em: <<http://www.ietf.org/>>. Acesso em 22 abr. 2017.

KUROSE, J. F.; ROSS, K.W. **Redes de Computadores e a Internet**: uma abordagem TopDown. Tradução Opportunity Translations; Revisão técnica Waggner Zucchi. 5. ed. São Paulo: Addison Wesley, 2010.

LIMA, J.R. **Monitoramento de redes com Zabbix**: Monitore a saúde dos servidores e equipamentos de rede / Janssen dos Reis Lima. Rio de Janeiro: Brasport, 2014.

MENDES, Douglas Rocha. **Redes de computadores**: Teoria e prática. São Paulo: Novatec, 2007.

MORIMOTO, Carlos Eduardo. **Faixas de endereços IP, CIDR e máscaras de tamanho variável. Guia do Hardware**. [Online] 26 set. 2007. Disponível em: <<http://www.hardware.com.br/tutoriais/endereco-ip-cidr/pagina2.html>>. Acesso em: 22 abr. 2017.

SANTOS, S. S. N. **Monitoramento de redes**: análise e configuração do software zabbix. Trabalho de conclusão. Instituto federal de educação, ciência e tecnologia de são paulo – câmpus salto, 2015.

TANENBAUM, A.S., **Redes de computadores**. Tradução Vanderberg D. de Souza. Rio de Janeiro: Elsevier, 2003.

TCU, **Boas Práticas em Segurança da Informação**, 4ª edição, Brasília, 2012

TELECO, **Gerenciamento e Monitoramento de Rede II: Gerenciamento SNMP**, 2012. Disponível em:

http://www.teleco.com.br/tutoriais/tutorialgmredes2/pagina_2.asp

SÊMOLA, Marcos. **Gestão da segurança da informação: uma visão executiva**. 8º ed. Rio de Janeiro, Editora Campus, 2003, 156 p.

SPECIALSKI, e., S., Dra., **Gerência de Redes de Computadores e de Telecomunicações**, Universidade de Santa Catarina, Florianópolis, 1999.

Disponível em:

<<http://cassio.orgfree.com/disciplinas/gredes/ApostilaGerenciamento.pdf>>. Acesso em: 22 abr. 2017.