**Gerenciamento de Rede**

**Definição** - É o controle de qualquer objeto passível de ser monitorado numa estrutura de recursos físicos e lógicos de uma rede – Essencial para garantir a alta disponibilidade da rede – Satisfazer exigências operacionais .

• Desempenho • Qualidade de Serviço • Custo – Necessário o monitoramento, teste, configuração e diagnóstico dos componentes da rede.

**FCAPS** - Modelo de Gerenciamento de Rede

• Criado pela ISO e refinado pela ITU-T

• Divide o gerenciamento de redes em 5 tarefas : Fault Management – Configuration Management – Accounting Management – Performance Management – Security Management.

**SNMP**

• Simple Network Management Protocol

• Framework para o gerenciamento de rede utilizando o conjunto de protocolos TCP/IP

• Camada de aplicação

• Monitora dispositivos produzidos por diferentes fabricantes

• Não segue o modelo cliente – servidor convencional

• Padrão IETF

**SNMP Community String** é como o id do usuário ou senha que admite o acesso às estatísticas do roteador ou aplicativo, que por padrão é **public.**

**SNMPv2** **incluiu** mecanismos de segurança, como a **criptografia** e **autenticação das mensagens.**

**SNMPv3** **melhorou** a segurança das versões anteriores, incluindo **confidencialidade, integridade e autenticidade**. Ademais, reforçou a **CRIPTOGRAFIA SIMÉTRICA** **(DES, 3DES, AES),** adicionou a **AUTENTICAÇÃO (baseada em comunidade (SNMPv2c) ou em usuário (SNMPv2u))** e o **CONTROLE DE ACESSO (baseado em visões).** Além disso o SNMPv3 é **compatível com as versões anteriores.** No SNMPv3, a autenticação é baseada em HMAC-MD5 ou HMAC-SHA e a encriptação em DES ou 3DES ou AES.

Para a coleta de informações é utilizado a porta 161/ UDP via requisição ou 162/UDP via trap. O SNMP define 5 tipos de pacotes: GetRequest, GetNextRequest,  GetBulkRequest, Trap, InformRequest, Response e Report. **GETBULK** foi introduzido no SNMPv2 e facilita o acesso a grandes quantidades de informações relacionadas sem iniciar as operações GETNEXT repetidas. A PDU de GETBULK consiste em 7 campos que começam com o tipo de PDU e ID de reposta.

**Tipos de mensagens geradas pelo SNMP (desde a v2):**

* **GetRequest**: pega o valor de uma ou mais instâncias de objetos MIB (gerente a agente);
* **GetNextRequest**: pega o valor da próxima instância de objeto MIB na lista ou tabela (gerente a agente);
* **GetBulkRequest**: pega valores em grandes blocos de dados, por exemplo, valores em uma grande tabela (gerente a agente);
* **InformRequest**: informa à entidade gerenciadora remota valores da MIB que são remotos para seu acesso (gerente a gerente);
* **SetRequest**: define valores de uma ou mais instâncias de objetos MIB (gerente a agente);
* **Response**: gerada em resposta a GetRequest, GetNextRequest, GetBulkRequest e SetRequest PDU ou InformRequest (agente a gerente ou gerente a gerente); e
* **Trap**: informa ao gerente um evento excepcional (agente ao gerente).

A versão 2 do protocolo de gerência de rede SNMP (SNMPv2) é incompatível com a versão 1 (SNMPv1).(Correto).  
Os formatos das mensagens são diferentes (Correto, mas o PDU continua o mesmo).  
Há dois novos tipos de mensagens na SNMPv2 que não existem na SNMPv1: GetBulk e Inform (Correto, GetBulk - recupera grande quantidade de dados e Inform - um gerente envia informações para outro gerente).

O USM (User-based security Model) foi projetado de forma a proteger os pacotes do SNMPv3 das ameaças provenientes de disfarces, modificação de mensagens, modificação do fluxo de mensagens e vazamento de informação.

Isto é alcançado através da implementação do conceito de múltiplos usuários, onde cada um tem suas chaves secretas para autenticação (*authKey*) e privacidade (*privKey*). Estas chaves não são registradas na MIB do agente, não podendo ser acessadas diretamente por funções SET ou GET.

Basicamente são definidos três tipos de MIBs**: MIB II, MIB experimental, MIB privada**.  
  
A MIB II, que é considerada uma evolução da MIB I, fornece informações gerais de gerenciamento sobre um determinado equipamento gerenciado. Através das MIB II podemos obter informações como: número de pacotes transmitidos, estado da interface, entre outras.  
  
A MIB experimental é aquela em que seus componentes (objetos) estão em fase de desenvolvimento e teste, em geral, eles fornecem características mais específicas sobre a tecnologia dos meios de transmissão e equipamentos empregados.  
  
MIB privada é aquela em que seus componentes fornecem informações específicas dos equipamentos gerenciados, como configuração, colisões e também é possível reinicializar, desabilitar uma ou mais portas de um roteador.

A  **MIB**é organizada como uma árvore (**hierarquicamente**), na qual cada nó tem um **nome** e um **número**. O **número de qualquer nó**na MIB inclui o **número do respectivo nó pai**separado com um **ponto** seguido por **seu próprio número** e assim por diante. Tanto os nomes quanto os números foram estabelecidos de forma padronizada na **MIB**. Por exemplo: "**Internet**" está diretamente abaixo de "**dod**" que, por sua vez, está abaixo de "**org**" que, por sua vez, está abaixo de "**iso**". Portanto teremos: **iso.org.dod.Internet**, em que cada um desses nomes tem seu próprio número.

* **iso = 1**
* **org = 1.3**
* **dod = 1.3.6**
* **Internet** **= 1.3.6.1**

Na **MIB-II**, abaixo de **iso.org.dod.Internet**temos outros 4 nós, um dos quais é "mgmt (1.3.6.1.2)". Poderíamos dizer que mgmt é a raiz da subárvore onde estão localizados os elementos de interesse para gerência de redes. São **11** nós na subárvore **mgmt**, o primeiro deles é o **system**(**1.3.6.1.2.1**). O **system**contem **objetos gerenciáveis**relacionados aos **sistemas**que rodam nos **dispositivos gerenciados**em uma **rede gerenciada**.

|  |
| --- |
| Grupos                              Informações |
| system (1)                        Sistema de operação dos dispositivos da rede |
| interfaces (2)                    Interface da rede com o meio físico |
| address translation (3)      Mapeamento de endereços IP em endereços físicos |
| ip (4)                                  Protocolo IP |
| icmp (5 )                            Protocolo ICMP |
| tcp (6)                                Protocolo TCP |
| udp (7)                               Protocolo UDP |
| egp (8)                               Protocolo EGP |
| cmot (9)                             Protocolo CMOT |
| transmission (10)               Meios de transmissão |
| snmp (11)                          Protocolo SNMP |

O protocolo SNMP opera na porta **161** por padrão. A porta **162** é denominada **SNMPTRAP**. Um trap SNMP é usado para reportar uma notificação ou para outros eventos assíncronos sobre o subsistema gerido. As portas utilizadas são sempre no destino!

A porta 161/UDP é utilizada SEMPRE no destino. O destino deve estar escutando na porta 161/UDP. Isto NÃO significa que o destino é sempre o Agente SNMP, pois a comunicação entre o Agente SNMP e o Gerente SNMP pode ocorrer de modo bidirecional. Logo, o Gerente SNMP pode ser o destino da comunicação (p. ex.: quando o Agente SNMP envia um GetResponse ao Gerente SNMP).

Por sua vez, a porta 162/UDP é utilizada SEMPRE pelo agente SNMP para comunicações via traps, onde o destino SEMPRE será o Gerente SNMP, pois não há mensagens de respostas haja vista que o trap é unidirecional. Assim, o Gerente SNMP deve estar escutando na porta 162/UDP, a fim de receber a notificação da ocorrência de um evento.

O conjunto de padrões SNMP [27] é formado por:

* **Um conjunto de especificações de estrutura e identificação para as informações gerenciais.** Este padrão é chamado S***MI (Structure of Management Information)*.** Estas especificações são encontradas no RFC 1155 [24]. A SMI baseou-se na linguagem ASN.1
* **Um protocolo de comunicação entre o gerente e o agente**, chamado simplesmente de **SNMP (Simple Network Management Protocol).** A especificação deste protocolo e apresentada no RFC 1157 [24].
* Uma **base de informações gerenciais que especifica quais variáveis são mantidas pelos elementos de rede.** Esta base de informações é denominada **MIB (Management Information Base)**, e a sua segunda versão, MIB-II, está especificada pelo RFC 1213 [24].

A SMI tem duas versões: a Structure of Management Information Version 1 (SMIv1) (RFC 1155) que **define precisamente como objetos gerenciados são nomeados e especifica seus tipos de dados associados** e a Structure of Management Information Version 2 (SMIv2) (RFC 2578) que prevê melhorias para o SNMPv2.

A definição de objetos gerenciados pode ser dividida em três atributos:

* **Nome**: O nome ou identificador de objeto (OID) define um único objeto gerenciado. Os nomes aparecem geralmente em duas formas: número ou “legível”. Em ambos os casos, os nomes são extensos e inconvenientes.
* **Tipo e Sintaxe**: Um objeto gerenciado é definido usando um subconjunto do Abstract Syntax Notation One (ASN.1). A ASN.1 é uma forma de especificar a forma como os dados são representados e transmitidos entre agentes e gerentes, no âmbito do SNMP. O lado bom do ASN.1 é que a notação é independente da máquina.
* **Codificação**: Uma única instância de um objeto gerenciado é codificada em uma string de octetos usando o Basic Encoding Rules (BER). O BER define como os objetos são codificados e decodificados para que eles possam ser transmitidos através de um meio de transporte, tais como a Ethernet.

**Comandos**

* **GetRequest:** O comando GetRequest é uma requisição do gerente ao agente para que o valor de uma variável ou uma lista de variáveis seja retornado. As variáveis-alvo são especificadas no campo Enlace de variáveis. O agente então retorna a requisição com uma mensagem do tipo Response, com os valores correntes das variáveis requisitadas.
* **SetRequest:** Este comando define um novo valor de uma variável ou de uma lista de variáveis. Novamente, uma resposta do tipo Response chega ao gerente com os valores correntes das variáveis em questão.
* **GetNextRequest:** Um comando do gerente para solicitar variáveis disponíveis. Quando um comando deste tipo é recebido pelo agente, a resposta Response ao gerente incrementa o OID do objeto, expondo o valor da variável seguinte. Dessa forma, o MIB completo de um agente pode ser recuperado iterativamente se GetNextRequest for emitido com OID=0.
* **GetBulkRequest:** Este comando é uma versão otimizada de GetNextRequest. Introduzido na SNMPv2, é um comando de múltiplas chamadas de GetNextRequest.
* **Response:** Este termo não indica propriamente um comando, e sim uma mensagem de resposta contendo valores de variáveis como resposta a uma solicitação ou confirmação de uma mudança pelo gerente.
* **Trap:** Dispositivos gerenciados podem enviar notificações SNMP para seus gerentes quando certos eventos ocorrem. Um exemplo de notificação importante que um SNMP deve receber seria de uma falha em algum dos roteadores. **Trap** é exatamente um dos dois tipos de notificação que o SNMP suporta. O envio de uma mensagem **Trap** permite que um agente notifique o sistema de gerenciamento para a ocorrência de qualquer evento relevante em qualquer instante de tempo. No entanto, não há qualquer confirmação por parte do gerente do recebimento da Trap.
* **InformRequest:** Um gerente SNMP que recebe uma notificação **InformRequest** retorna ao agente emissor uma Response reconhecendo o recebimento de sua notificação. ***A PDU*InformRequest é usada por uma entidade gerenciadora para comunicar a outra entidade gerenciadora informações MIB remotas à entidade receptora.**

O **Zabbix** é uma ferramenta “open source” ou de código aberto que vem ganhando cada vez mais o mercado de ferramentas de monitoramento. Essa ferramenta é capaz de monitorar toda a infraestrutura de rede além da possibilidade de se estender o monitoramento às aplicações e serviços da rede.

Por ser open source se torna uma ferramenta multiplataforma, podendo ser instalado e utilizado nos mais diversos sistemas operacionais como Linux, Windows, FreeBSD, Mac OS X, entre outros. Reparem que os agentes também são multiplataformas.

Percebam que o acesso à ferramenta é muito simples e pode se dar por intermédio do próprio navegador (Browser) ou via cliente específico da ferramenta. Além disso, a instalação, configuração e mapeamento da rede é bem prática e simples, apesar de ser uma ferramenta extremamente versátil e robusta.

Outras características importantes do ZABBIX são:

• Possibilidade de monitorar serviços simples sem a necessidade de instalação de agentes (http, pop3, imap e ssh);

• Suporte nativo ao principal protocolo de gerenciamento de rede: SNMP;

• Integração com banco de dados (MySQL, Oracle, PostgreeSQL ou SQLite);

• Suporte à arquitetura de 32 e 64 bits;

• Capacidade de disparar alertas via e-mail, Jabber ou SMS;

Um ponto que é bastante interessante aos administradores de rede é a capacidade de customização de scripts de monitoramento e templates para mapeamento dos objetos. Ou seja, caso se possua diversos equipamentos de um mesmo fabricante, pode-se criar um template padrão e importa-lo para cada objeto, otimizando bastante a montagem e mapeamento da infraestrutura na ferramenta. Lembrando que é uma ferramenta com interface gráfica não só para visualização, mas também para configuração e gerência.

**Subagente: o** subagente é um pedaço de *software* rodando num componente SNMP que implementa a funcionalidade de gerenciamento de informações , esse é definido por uma MIB específica de um subsistema específico: por exemplo, a camada de ligação da *ethernet*.

Alguns recursos do subagente são: Recolher a informação a partir de objetos gerenciados; Configurar os parâmetros dos objetos gerenciados; Responder às solicitações dos gestores; Gerar alarmes ou armadilhas (*traps*);