FACULDADE DE TECNOLOGIA ALCIDES MAYA - AMTEC CURSO TECNOLÓGICO EM REDES DE COMPUTADORES

JEAN GUEDES DUARTE

SOLUÇÃO PARA CONTROLE DE ACESSO EM REDES LAN USANDO PORT-SECURITY E NAGIOS.

Porto Alegre 2019

JEAN GUEDES DUARTE

SOLUÇÃO PARA CONTROLE DE ACESSO EM REDES LAN USANDO PORT-SECURITY E NAGIOS.

Projeto de Pesquisa apresentado como requisito parcial para obtenção do título de Tecnólogo em Redes de Computadores, pelo Curso de Redes da Faculdade de Tecnologia Alcides Maya - AMTEC

Orientador: Prof. Fagner Coin Pereira

Porto Alegre

2019

RESUMO

O port-security é uma alternativa de segurança importante que pode ser adotada nas camadas de acesso, onde ficam os switches que conectam os dispositivos finais. Esse recurso permite restringir o que podem ou não ter acesso à rede, através da verificação dos endereços físicos (MAC) de cada equipamento conectado à porta do switch. Entretanto, essa solução pode causar transtornos para o usuário final, principalmente para os que utilizam dispositivos móveis como notebooks, tablets ou celulares, uma vez que ao trocar o equipamento de local o mesmo ficará sem acesso à rede, até que seja realizada sua liberação, já que cada equipamento pode conectarse em uma porta por vez. Para uma alternativa, o presente projeto visa implementar uma solução para aplicar em uma instituição financeira, onde existe o port-security habilitado. Assim, será estudado a integração do port-security com a ferramenta de monitoramento Nagios, que também já é utilizado pela instituição, onde busca-se como resultado a liberação de forma breve e apenas dos equipamentos previamente autorizados.

Palavras-chaves: Port-security, Nagios, Instituição Financeira.

ABSTRACT

Port-security is an important security alternative that can be adopted at the access layers, where the switches that connect end devices are located. This feature allows you to restrict what may or may not have access to the network by checking the physical addresses (MAC) of each equipment connected to the switch port. However, this solution can cause inconvenience to the end user, especially those who use mobile devices, such as notebooks, since switching the equipment will have no access to the network until it is released, as each equipment can connect to one port at a time. To provide an alternative, this project aims to implement a solution to apply to a financial institution where port-security is already enabled. Thus, the integration of port-security with the NAGIOS monitoring tool will be studied, which is also already used by the institution, where it is sought as a result the brief release of only previously authorized equipment.

Keywords: Port-security, Nagios, Financial Institution

LISTA DE FIGURAS

LISTA DE SIGLAS

ACL - *Access Control List*
ARP - *Address Resolution Protocol*
CPDs – Centro de Processamento de Dados
CPU - *Central Processing Unit*
DDOS - *Distributed Denial of Service*
DNS - *Domain Name Service*
DOS - *Denial of Service*
GPL - *General Public License*
GUI - *Graphical User Interface*
HTTP - *Hypertext Transfer Protocol*
ICMP - *Internet Control Message Protocol*
IETF - *Internet Engineering Task Force*
IMAP - *Internet Message Access Protocol*
IP - *Internet Protocol*
LAN - *Local Area Network*
MAC - *Media Access Control*
MAN - *Metropolitan Area Networ*
MIB - *Management Information Base*
NEB - *Nagios Event Broker*
NNTP - *Network News Transfer Protocol*
NNTP - *Network News Transfer Protocol*
NRPE - *Nagios Remote Plug-in Executor* NSCA - *Daemon Nagios Service Check Acceptor* OSI - *Open Systems Interconnection.*
PDU - *Protocol data unit*
RFC - *Request for Comments*
SLA - *Service Level Agreement*
SMI - *Structure of Management Information*
SMTP - *Simple Mail Transfer Protoco*
SNMP - *Simple Network Management Protocol*
SSH - *Secure Shell*
TCP - *Transmission Control Protocol*
TI - Tecnologia da Informação
UDP - *User Datagram Protocol*
USM - *User Security Model*
VPN - *Virtual Private Network* WAN - *Wide Area Network*

SUMÁRIO

1 INTRODUÇÃO

Pode-se definir controle e gerenciamento de redes de computadores como a análise de todos os recursos materiais e/ou lógicos presentes na constituição da rede física de uma organização (PINHEIRO, 2002). Com o uso de computadores e o surgimento de sistemas distribuídos e do transporte de dados por meio de redes e recursos de comunicação, tornou-se clara a necessidade de ferramentas de segurança que protegessem as informações armazenadas no computador (STALLINGS, 2005). Um ataque é caracterizado pela realização de uma ameaça intencional (SOARES; LEMOS; COLCHER, 1995). Alguns tipos de ataques comuns são: o *spoofing* de DNS, interrupção de serviços e modificação de informações importantes.

O escopo deste projeto é apresentar como é possível melhorar a segurança nas redes locais utilizando a ferramenta port-security em switches Cisco, aumentando também o controle de quem acessa a rede, em segunda instância também será abordado como será realizado o monitoramento das ocorrências causadas pelo portsecurity, facilitando na identificação e agilizando no tratamento destas ocorrências.

Estas informações podem ser úteis tanto para a prevenção, como para a detecção e resolução de problemas que possam acabar com a disponibilidade da rede, e que por muitas vezes, acabará livrando a empresa ou instituição, de quem sabe, sofrer um duro golpe financeiro devido a falhas oriundas do mal funcionamento da rede.

1.1 Definição do Tema ou Problema

O tema desta pesquisa aborda a metodologia de implementação e monitoração da ferramenta de segurança “port-security” nos switches da Cisco em redes LAN, aumentando a segurança e o controle dos equipamentos conectados à rede.

Uma rede local é a interconexão de diversos dispositivos em uma rede de computadores que concede um meio de troca de informações entre esses dispositivos (STALLINGS, 2005). Uma rede que não contém controle sobre as suas camadas de acesso acaba deixando uma enorme falha em sua segurança possibilitando vários

tipos de ataques como “ARP *poisoning*” e “MAC *flooding*”, mais comuns em redes de grande porte. Um ataque é caracterizado pela realização de uma ameaça intencional (SOARES; LEMOS; COLCHER, 1995).

Com o uso de computadores e o surgimento de sistemas distribuídos e do transporte de dados por meio de redes e recursos de comunicação, tornou-se clara a necessidade de ferramentas de segurança que protegessem as informações armazenadas no computador (STALLINGS, 2005). Conforme BORGES (2013)
“Existem alguns programas chamados de *sniffers* de rede que, a grosso modo, podem capturar mensagens que trafegam por uma rede local de computadores, mesmo que elas não sejam endereçadas ao seu computador”. Um equipamento ligado à rede usando algum tipo de *Sniffer* como o “Wireshark”, por exemplo, poderá utilizá-lo junto a ataques como o ARP *Cache Poisoning.*

Com a ferramenta port-security da Cisco podemos limitar o acesso à rede de empresa que utilizam essa marca de switch, evitando assim vários tipos de ataques e equipamentos não autorizados na rede, aumentando a segurança e tendo um maior controle sobre os terminais de acesso. Porém, com a implementação do port-security resolve-se parte do problema, pois em uma empresa de grande porte, isso prejudicaria a mobilidade dos funcionários que usam notebooks dentro da empresa, gerando assim uma quantidade enorme de ocorrências a ser tratadas pelo administrador de redes. Diante disso faz-se necessária uma ferramenta que monitore as ocorrências geradas pela ferramenta port-security e faça a validação do MAC inserido na rede para sua liberação automática.

Com a ferramenta de monitoração do NAGIOS e scripts automatizados, pode-se monitorar as ocorrências geradas pelo port-security de todos os switches interligados na rede e realizar a liberação automática do equipamento (caso ele seja autorizado pela empresa) ou identificar com mais brevidade qual equipamento está tentando acessar a rede, resolvendo assim a outra parte do problema.

1.2 Delimitações do Trabalho

Neste projeto será realizado a implementação do port-security nos switches e utilizada a estrutura e os componentes da ferramenta de monitoramento Nagios

(ferramenta empregada no ambiente a ser monitorado) para alertar quando um dispositivo é bloqueado pelo port-security. Não será abordado comparativo do Nagios com outros sistemas de monitoramento, tampouco será utilizado switches de outros fabricantes, que não seja Cisco.

1.3 Objetivos

Os objetivos dividem-se em: geral e específicos.

1.3.1 Objetivo Geral

O principal objetivo é implementar o port-security nos switches da Cisco e adaptá-lo a ferramenta de monitoramento NAGIOS para analisar e disponibilizar com maior agilidade e eficiência o equipamento que está sendo inserido na rede, verificando se o mesmo tem ou não autorização de acesso à rede.

.

1.3.2 Objetivos Específicos

- Demonstrar a identificação de equipamentos inseridos na rede (capturando o MAC) através da ferramenta port-security da CISCO;
- Propor a integração do Nagios com o port-security como solução de monitoramento para o problema.

1.4 Justificativa

Existe certa dificuldade de manter a segurança em uma rede de grande porte, conforme (ZANI, 2014). Tanto o setor público quanto o setor privado, no que diz respeito à segurança da informação, possuem os mesmos desafios: rápida evolução das ameaças e das tecnologias, complexidade dos ataques, dificuldade para detectar incidentes rapidamente e diminuir o tempo de reação.

Como em uma empresa de grande porte a rotatividade de equipamentos é constante, o controle de quem acessa a rede se torna muito difícil, justifica-se esse projeto como um meio de melhorar a segurança nos terminais de acesso à rede com um protocolo (port-security) já existente no switch da Cisco e facilitar a identificação

de ativos com mais eficiência e brevidade utilizando ferramentas de monitoramento (Nagios) já implementadas no local, evitando assim mais custo para a empresa.

Conforme Dias (2018), com o port-security podemos também controlar o uso e a movimentação de equipamentos usados na rede, “A funcionalidade é bastante útil também em ambientes onde hosts e servidores precisam ser vinculados obrigatoriamente a uma porta (em ambientes como em CPDs e *Data Centers*) ou em localidades onde o usuário costuma migrar a estação sem comunicar a equipe de suporte.

2 REVISÃO BIBLIOGRÁFICA

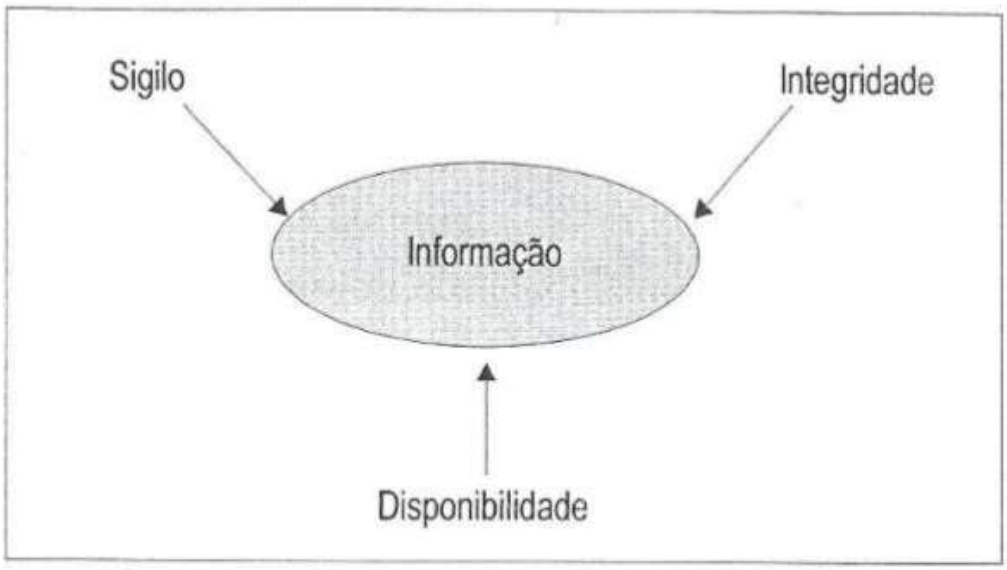
A implementação do port-security da Cisco junto ao seu monitoramento com o Nagios e o uso de *scripts* automatizados depende de diversos serviços e protocolos para seu devido gerenciamento. Os protocolos de gerência de redes são primordiais para coleta de dados de gerenciamento de redes, uma vez que a comunicação nas redes só é possível devido ao uso de protocolos de comunicação padronizados e estáveis (OLIVEIRA, 2007).

2.1 Segurança de Redes

A segurança de redes é uma parte essencial para a proteção da informação. Para isso três principais requisitos podem ser definidos, conforme a Figura 19 (NAKAMURA; GEUS, 2007):

- Confiabilidade/Sigilo;
- Integridade;
- Disponibilidade;

FIGURA 1 - Propriedades importantes de segurança



Fonte: (NAKAMURA; GEUS, 2007)

Um conjunto de condições que possibilita a violação de uma política de segurança explícita ou implícita é chamada de vulnerabilidade (SEACORD; HOUSEHOLDER, 2005). Para CERT.br (2006) na área de tecnologia da informação,

exemplos de vulnerabilidade poderiam ser falhas de configuração de programas, serviços ou equipamentos de redes. Assim as vulnerabilidades podem ser “pontos nos quais o sistema é suscetível a ataques e ameaças de segurança” (SCHWEITZE, 2005, P. 82).

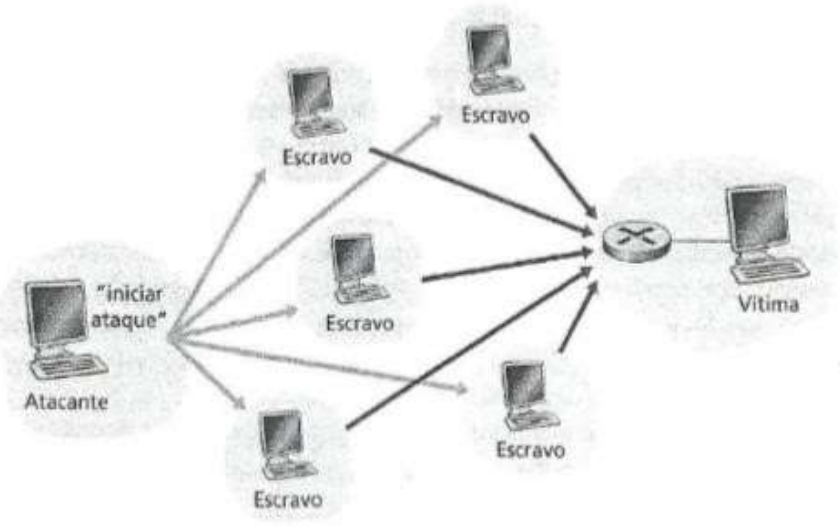
Há dois tipos de ameaças possíveis: ameaças acidentais, em que não há a intenção premeditada e ameaças intencionais. Essas ameaças ainda podem ser passivas ou ativas. Uma ameaça passiva não tem como resultado nenhuma modificação das informações espionadas. Já uma ameaça ativa resulta na modificação das informações (SOARES; LEMOS; COLCHER, 1995).

> Os ataques também podem ser classificados como ataques passivos ou ativos. Os ataques passivos não têm como intuito afetar os recursos da rede, somente tentam aprender e utilizar as informações que são trafegadas. Alguns exemplos desse tipo de ataque são o vazamento de conteúdo das mensagens, como conversas telefônicas, e-mails ou arquivos e o de análise de tráfego. Já os ataques ativos tentam afetar a operação da rede e alterar seus recursos, como modificar um fluxo de dados ou criar um fluxo falso. Alguns exemplos deste tipo de ataque são: o de falsidade, em que uma entidade se passa por outra diferente; de repetição, que envolve a retransmissão de unidades de dados capturadas; de modificação de mensagens, em que uma parte da mensagem legitima é alterada; e de negação de serviço (STALLINGS, 2005, p.219).

Um ataque de negação de serviço (*Denial-of-Service Attack* – DoS), é quando se gera uma grande quantidade de algum tipo de trabalho para a rede, hospedeiro ou outro componente da infraestrutura, fazendo com que o trabalho legítimo não possa ser realizado, tornando assim impossível a utilização do mesmo por um usuário autêntico da rede. (KUROSE; ROSS, 2006). Tanto para Kurose e Ross (2006), Nakamura e Geus (2007), como para Morimoto (2008) uma variante do ataque de DoS é o ataque de negação de serviço distribuído (*Distributed Denial of Service* – DDoS), representado na Figura 2, que tem como objetivo a invasão e coordenação de vários hosts distribuídos por um *hacker*, para realizar ataques simultâneos aos alvos escolhidos. Assim com a utilização de diversos tipos de vulnerabilidades em sistemas, o atacante consegue formar um *botnet*, instalando e executando um programa escravo em inúmeras máquinas, que continuam de forma aparentemente normal, executando

suas tarefas, aguardando o comando de seu mestre. Assim que a *botnet* estiver com um grande número de máquinas infectadas, com poucos comandos o atacante consegue lançar um ataque de DDoS em algum alvo escolhido. (MORIMOTO, 2008)

Figura 2 – Ataque DDos



Fonte: (KUROSE; ROSS, 2006)

Um tipo importante de ataque passivo é o *scanning* de vulnerabilidades. Para isto, pode ser utilizado uma ferramenta de *port scanner*. Um dos mais utilizados é o NMAP, que tem como funcionalidade, por meio do mapeamento de portas, a obtenção de informações de serviços que podem ser acessíveis. Assim o scanner de vulnerabilidades varre somente as vulnerabilidades específicas do que já foi identificado como alvo e os tipos de sistemas e serviços que neles são executados. Após checar os roteadores, serviços, *firewalls*, sistemas operacionais e outras entidades IP, alguns riscos podem ser analisados, como: compartilhamento de arquivos que não são protegidos por senhas; configurações incorretas; softwares desatualizados; falhas na camada de rede; configurações de roteadores potencialmente perigosas; checagem de senhas fáceis de serem adivinhadas; SNMP; e possibilidade de DoS (NAKAMURA; GEUS, 2007).

Os ataques citados acima têm sua origem de fora da rede local, porém muito mais comuns são os ataques internos, ou seja, que tem origem dentro da própria rede (CRONKHITE; MCCULLOUGH, 2001). Tanto para Felippetti (2008) como para Odom

(2003) algumas abordagens são essenciais para as políticas de segurança da rede local, como por exemplo, as listas de acesso (*Access List* - ACLs).

> Listas de acesso são, essencialmente, listas de condições que controlam o acesso. Uma vez citadas, podem ser aplicadas tanto ao tráfego entrante (inbound traffic) quanto ao tráfego sainte (outbound traffic), em qualquer interface. A aplicação de listas de acesso fará com que o router examine cada pacote atravessando uma determinada interface em uma determinada direção e tome as providências apropriadas. (FELIPPETTI, 2008, p.342):

Assim, as ACLs têm o objetivo de filtrar o tráfego indesejado da rede; encontrar pacotes com níveis de prioridades diferentes; e evitar que sistemas críticos sejam acessados por funcionários não designados (ODOM, 2003).

Outra forma de proteger a rede interna é através da configuração de Port Security, que permite restringir uma porta do switch a um conjunto de endereços MAC. O administrador da rede pode configurar de forma estática quais os endereços MAC são permitidos para cada porta específica ou de forma dinâmica, em que é necessário limitar o número de endereços MAC que serão aprendidos. Assim a porta fornece acesso a quadros somente dos endereços que forem considerados seguros (FROOM, 2010).

2.2 Monitoramento

Monitoramento é o processo de obter informações sobre elementos de um sistema computacional. Estas informações ajudam a entender a situação do sistema, sua configuração, estatísticas de uso e desempenho, informações sobre erros e sobre a topologia do sistema. O processo de monitoramento depende de técnicas para coletar, processar, armazenar e disponibilizar estas informações. Entretanto, a variedade de elementos que compõem um sistema computacional exigirá técnicas adequadas para cada classe de elemento (VERMA, 2009).

Segundo Zarpelão (2004) o monitoramento dos ativos de redes é uma avaliação contínua das variáveis operacionais, cujo principal objetivo é detectar antecipadamente anomalias com uma baixa taxa de falsos positivos, ou seja,

alarmesfalsos, garantindo assim um bom funcionamento e confiabilidade das redes de computadores monitoradas.

> Hoje, em uma empresa, é de extrema importância que um gerente de TI (Tecnologia da Informação) possua uma forma de monitorar e controlar componentes da rede. Para realizar essa tarefa é necessário um software de gerenciamento para rede capaz de monitorar e controlar qualquer componente possível de ser gerenciado, como servidores de serviços, roteadores e comutadores de camada 2, conhecidos como switch (STALLINGS,1999, p 619).

Diversas ferramentas estão disponíveis para o monitoramento de servidores, ativos de rede e serviços, tais como Monit, Cacti, Zabbix e Nagios. Muitas destas ferramentas trazem rotinas prontas para o monitoramento de serviços amplamente utilizados, restando para o administrador apenas compor sua ferramenta de monitoramento como blocos de construção.

2.2.1 Monitoramento e Controle

Segundo Fry e Nystrom (2009), a arquitetura geral dos sistemas de gerenciamento de redes apresenta quatro componentes básicos: “Os elementos gerenciados, as estações de gerência, os protocolos de gerenciamento e as informações de gerência. ” Os elementos gerenciados são dotados de um *software* chamado agente, que permite o monitoramento e controle do equipamento através de uma ou mais estações de gerência. A princípio, qualquer dispositivo de rede (impressoras, roteadores, repetidores, switches, etc.) pode ter um agente instalado.

Nas estações de gerência encontramos o *software* gerente, responsável pela comunicação direta desta estação com os agentes nos elementos gerenciados. Claro que para que aconteça a troca de informações entre o gerente e os agentes é necessário ainda um protocolo de gerência que será o responsável pelas operações de monitoramento e de controle (FRY; NYSTROM, 2009).

2.2.2 Monitoramento com Nagios

Andrade (2006) afirma que originalmente escrito sob o nome Netsaint, o “Nagios” foi criado e ainda é mantido por Ethan Galstad e sua equipe de mais de 150 desenvolvedores espalhados por todo o mundo, dedicados a desenvolver *plug-ins*, corrigir *bugs*, desenvolver uma interface WEB, produzir e traduzir a vasta documentação, entre outras atividades. Este software de monitoramento de redes é distribuído livremente, através da lei de Copyleft GPL. A habilidade em administrar ambientes com infraestrutura de WAN, LAN e MAN, e a interface gráfica – GUI utilizada lhe garantem desempenho comparável a sistemas comerciais existentes, como WhatsUp e BigBrother, assim como o Angel Network Monitor, o PIKT, o Autostatus e outros.

> Com o Nagios é possível monitorar serviços de rede como (SMTP, POP3, HTTP, NNTP entre outros), é capaz de monitorar recursos de servidores como (carga do processador, uso de disco, memória entre outros). É possível definir uma hierarquia de configuração, facilitando a identificação correta de qual equipamento causou o problema em uma rede complexa. Notifica através de e-mail ou torpedo SMS. O Desenvolvimento simples permite que o gerente de TI crie seus próprios plugins, que dependendo das necessidades pode ser realizado em Shell script, C, Perl, Python, PHP, C#. O plugin para o Nagios é um executável compilado ou um script, exemplo (Perl, shell), sendo executado na linha de comando para identificar o status de um servidor ou serviço. O uso dos plugins no é indispensável, pois sem isso o Nagios tornase uma ferramenta inútil, não realizando a recuperação de informação de serviços ou identificando se um servidor está ligado ou desligado. (COSTA, FELIPE. 2008, p 189)

Nagios é uma aplicação de monitoramento de redes de código aberto bastante popular. Ele permite monitorar tanto hosts quanto serviços, alertando o administrador quando ocorrerem problemas na rede. É utilizado por administradores de redes para que possam ter um controle sobre os serviços e equipamentos de sua rede. Foi idealizado inicialmente para ser utilizado em sistemas operacionais Linux e, a partir da versão 3.0.4, tornou-se compatível com outros Sistemas Operacionais, (KOCJAN, 2014).

Segundo Payne e Carboni (2007) Nagios utiliza dois tipos de checagem: ativo e passivo. Quando a iniciativa da checagem é tomada pelo processo do Nagios, esta checagem é chamada ativa. Outros processos podem informar para o Nagios o estado de um serviço, quando isto acontece, a checagem é chamada passiva. Para fazer uma checagem ativa, o Nagios executa um comando configurado na definição do serviço monitorado, captura a saída do comando e a armazena em suas estruturas de dados. Estes comandos utilizam plug-ins que podem ser desenvolvidos por qualquer um e incorporados na configuração do Nagios. Estes comandos são executados localmente no servidor Nagios.

Payne e Carboni (2007) ainda afirmam que o Nagios processa os dados deste arquivo em um intervalo de tempo configurado. Isto é possível apenas para processos locais. O *daemon* Nagios Service Check Acceptor (NSCA) permite que *hosts* remotos coloquem seus resultados para as checagens de serviço no "*External Command File*". O Nagios oferece um intermediador de eventos chamado Nagios Event Broker (NEB) que permite que qualquer um escreva um módulo para trabalhar com este intermediador. O NEB cria canais dedicados para cada tipo de evento e os módulos requisitam o registro nos canais que lhe interessam. Quando determinado evento ocorre, o Nagios procura os módulos que registraram interesse no canal do evento, chama o módulo e passa uma estrutura de dados com informações sobre o evento.

> A característicasprincipais do Nagios são: o monitoramento de serviços de rede como tráfego de dados de host e serviços que podem ser definidos pelo administrador da rede, além de monitorar serviços como SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol), HTTP (HyperText Transfer Protocol), NNTP (Network News Transfer Protocol), ICMP (Internet Control Message Protocol) e SNMP (Simple Network Management Protocol). O Nagios monitora também os recursos de servidores como logs do sistema, carga do processador, uso de memória e uso de disco. O Nagios também trabalha com plug-ins, permitindo adicionar novas funcionalidades ao mesmo. Os plug-ins podem ser desenvolvidos em qualquer linguagem, mas a grande maioria é desenvolvida em perl e python, (BENINI; DAIBERT, 2013, p 114)

2.3 Nagios Core

Conforme Ramiro Ferrão (2014) o Nagios Core é o sistema de monitoramento *Open Source* que permite às organizações identificar e resolver problemas de infraestrutura de TI antes que eles afetem os processos críticos de negócios. Nagios Core permite que você monitore toda a sua infraestrutura de TI para garantir que os sistemas, aplicativos, serviços e processos de negócios estão funcionando corretamente. No caso de uma falha, ele pode alertar os responsáveis técnico do problema, permitindo-lhes começar o processo de correção antes que as interrupções afetam os processos de negócios, usuários finais ou clientes. A Figura 3 exibe a comparação entre o Core e o XI.

Figura 3 – Comparativo entre o Nagios Core e XI

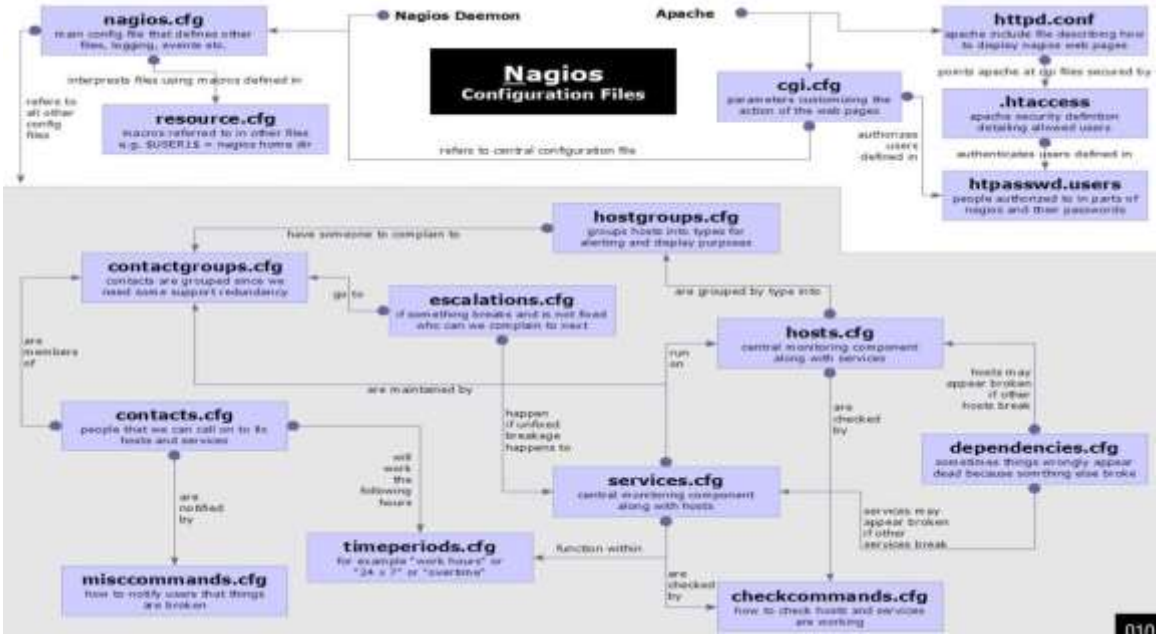
Feature	Nagios Core	Nagios XI
Infrastructure Monitoring		
Servers	✔	✔
Network Elements	✔	✔
Applications	✔	✔
System Metrics	✔	✔
Custom Services	✔	✔
Alerting		
Email	✔	✔
Mobile Phone	✔	✔
Custom Method	✔	✔
RSS Feed		✔
Per-User Notifications		✔
Reporting		
Basic Reports	✔	✔
Advanced Reports		✔
PDF, JPG and CSV Export		✔
Performance Graphs		✔
SLA Reports		✔
Email Reports		✔
Scheduled Reporting		✔
Capacity Planning		✔
Custom Report Creation		✔
User Interface		
User-Specific Customization		✔
Advanced Dashboards		✔
Session Authentication		✔
Instant Remote Host Access		✔
Custom Actions		✔
Sharable/Deployable Dashboards		✔
Custom Branded Interface Capabilities		✔
Distributed Monitoring		
Basic Capabilities	✔	✔
Advanced Capabilities		✔
Supported Plugins	4000+	4000+
Send & Receive SNMP Traps		✔
Third-Party Ticketing/Solution Integration		✔

Fonte: (NAGIOS CORE XI; CORE FEATURE COMPASION,2011)

2.3.5 Arquivos de configurações do Nagios

Principais arquivos de configuração do nagios: Na Figura 4, segue a topologia dos arquivos de configurações do Nagios.

Figura 4 – Mapeamento dos arquivos de configuração do Nagios.



Fonte: (NAGIOS3, 2011)

As informações das configurações do Nagios abaixo foram tiradas da sua documentação através do site oficial (Nagios documentation, 2018).

- /usr/local/nagios/etc/nagios.cfg: Inicia os serviços de monitoramento.

Nagios inclui uma documentação extensa própria, uma vez instalado no diretório /usr/local/nagios/share/docs, que pode também ser acessado pela interface web. Esta é sempre recomendada como uma fonte útil para maiores informações. Os arquivos de referência da configuração do diretório /etc/nagios terminam com -sample e são os que serão modificados em uma atualização do produto, ficando assim preservados os arquivos já customizados para produção. Arquivo com as configurações principais. O padrão é bem completo, somente altere os parâmetros: (Nagios Documentation, 2018)

check_external_commands=0 -> check_external_commands=1
date_format=us -> date_format=euro

- cgi.cfg: Configurações e permissões dos Programas CGI. Nele ficam as configurações de utilização de arquivos cgi pelo Nagios. Devem ser configurados os parâmetros de autorização de utilização da interface Web. Altere os campos para o nome do usuário cadastros no arquivo /usr/local/nagios/etc/htpasswd.users e assim terão acesso ao sistema, os parâmetros são:

```
authorized_for_system_information=usuario1, usuario2
authorized_for_configuration_information=usuario1, usuario2
authorized_for_system_commands= usuario1, usuario2
authorized_for_all_services=usuario1, usuario2
authorized_for_all_hosts=usuario1, usuario2
authorized_for_all_service_commands=usuario1, usuario2
authorized_for_all_host_commands=usuario1, usuario2
```

- hosts.cfg: Cadastro e informações dos hosts. O objeto host descreve um dos nós de rede que está sendo monitorado. Nagios espera o endereço IP como um parâmetro (ou um nome de domínio completo) e o comando para identificar se o host está ativo. A definição de host é novamente referenciada na no arquivo que trata do serviço. (NAGIOS, DOCUMENTATION, 2018). Exemplo consta no anexo A.

- Hostgroup.cfg: Cadastro dos grupos que os hosts farão. Alguns hosts podem ser combinados em um grupo. Esta configuração simplificaria o processo, pois grupos de hosts podem ser especificados ao invés de hosts únicos quando definir os serviços (o serviço vai então existir para cada membro do grupo). Em adição, Nagios representa os hosts de um grupo de host juntos em uma tabela na apresentação web, que também ajuda para tornar mais clara a visualização. (NAGIOS, 2018). Exemplo consta no anexo B.

- contacts.cfg: Cadastro e informações dos contatos que receberão as notificações enviadas pelo Nagios. Exemplo consta no anexo C.

- contactsgroups.cfg: Cadastro dos grupos de contatos. Notificação de eventos em hosts e serviços são feitas para um grupo de contato. Uma ligação direta entre host/serviço e uma pessoa de contato não é possível. (NAGIOS DOCUMENTATION, 2018). Exemplo consta no anexo D.

- services.cfg: Cadastro e informações dos serviços. Os serviços individuais a serem monitorados são definidos como objetos de serviço. Um serviço nunca existe independente de um host. Assim é possível ter alguns serviços com o mesmo nome, contanto que eles pertençam a hosts diferentes. Na linguagem do Nagios, um serviço é sempre um par host serviço que serão monitorados. (NAGIOS, 2018). Exemplo consta no anexo E.

- hostextinfo.cfg: Definição das imagens que representarão cada host. Objetos de informação estendida de host são opcionais e definem um gráfico específico e /ou URL, que o Nagios adicionalmente inclui em seus gráficos de saída. A URL pode se referir à página web que provê informação adicional no host. (NAGIOS, 2018). Exemplo consta no anexo F

- timeperiods.cfg: Contém informações de períodos de monitoramentos. Exemplo consta no anexo G.

- commands.cfg: Contém os comandos que devem ser executados pelo Nagios. Nagios sempre chama programas externos pelos objetos de comando. Além dos plugins, programas de mensagens incluem envio de e-mails ou SMS. (NAGIOS, 2018). Exemplo consta no anexo H.

2.3.6 Plug-ins do Nagios

O Nagios é um sistema de monitoramento vazio sem seus *plug-ins*, em outras palavras, para adicionar funcionalidades ao Nagios é necessária a instalação de *plugins*. Segundo Kofller (1999), *plug-ins* são executáveis compilados ou scripts desenvolvidos em Schell Script, Perl, entre outros, utilizados na linha de comando para checar o estado de um host ou serviço. Assim os *plug-ins* possuem um papel importante para o desempenho das funções da ferramenta Nagios. Eles são

aplicativos intermediários entre o Nagios e as estações a serem monitoradas. O Nagios é um aplicativo que em sua arquitetura permite o acréscimo de novos *plug-ins* com novas funcionalidades, desenvolvidos em paralelo e podem ser incrementados como uma atualização conforme a necessidade existente.

2.3.7 NagiosQL

O NagiosQL é uma interface WEB, desenvolvida em PHP, que permite fazer toda a administração do Nagios. Com esta ferramenta, o utilizador pode rapidamente e de forma simples criar toda a sua configuração sem a necessidade de recorrer aos ficheiros de texto que fazem parte da estrutura do Nagios (PINTO, 2015). Com ele é possível gerenciar e configurar de forma mais eficiente a ferramenta Nagios, visto que a configuração via terminal é bastante complexa.

2.4 Port-Security

O port-security é uma funcionalidade da camada 2, utilizada para limitar o acesso de equipamentos não autorizados na rede, Apesar do port-security poder trabalhar com o 802.1X e/ou autenticação baseada em endereços MAC, ele pode ser configurado de maneira simples para validar e permitir o funcionamento de endereços MAC registrados localmente na porta do Switch sua configuração é feita diretamente no switch.

> O recurso port-security permite o controle mais rígido dos dispositivos que se conectam a sua rede. É uma ferramenta de segurança importante que pode ser adotada nas camadas de acesso das redes de computadores onde ficam os switches que conectam os dispositivos terminais. Esse recurso restringe os computadores (e outros dispositivos terminais) que podem ou não ter acesso à rede cabeada da empresa por meio da verificação dos endereços físicos (MAC) dos quadros que chegam nas interfaces do switch. (AGILITY NETWORKS, 2016, p11 e 12)

Com o port-security podemos também controlar o uso e a movimentação de equipamentos usados na rede, “A funcionalidade é bastante útil também em ambientes onde hosts e servidores precisam ser vinculados obrigatoriamente a uma

porta (em ambientes como em CPDs e *Data Centers*) ou em localidades onde o usuário costuma migrar a estação sem comunicar a equipe de suporte” (DIAS, 2018).

> Outro objetivo do port security é prevenir ataques de flooding de endereços MAC. Os ataques de flooding de endereços MAC consiste em forçar o Switch a popular a sua tabela de endereços MAC originando inúmeras mensagens com endereço de origem falsos para superpopular a tabela MAC e forçar o Switch a atuar como hub. (DIAS, 2018, p. 17).

2.5 Demais Tecnologias

2.5.1 Snmp

O protocolo SNMP teve seu início a partir de 1988, como um *feedback* às necessidades de gerenciamento da rede Internet. Conforme Morishita e Moreira (1997), o protocolo foi implementado para ser uma solução provisória, enquanto se esperava o desenvolvimento de um novo protocolo mais completo (CMOT).

Esse novo protocolo, o CMOT, foi abandonado por não evoluir, deixando o SNMP como a única solução de padronização, sendo adotado por todos os fabricantes de equipamento de rede. É um protocolo pertencente à camada de aplicação da arquitetura TCP/IP e utiliza na camada de transporte os serviços do protocolo UDP para enviar suas informações através da rede IP. Ele é usado em sistemas de gerenciamento de redes a fim de monitorar dispositivos que exijam atenção por parte de seus administradores.

De acordo com Soares (2012), o SNMP possui a capacidade de ser aplicado aos mais variados sistemas operacionais, tais como sistemas Unix, Windows, Linux e MAC-OS, além de impressoras, modems, racks e fontes de alimentação, dentre outros, não se restringindo somente a dispositivos físicos, mas com a possibilidade de serem introduzidos em softwares (servidores web e banco de dados, por exemplo). Morishita e Moreira (1997) afirmam que o protocolo SNMP é responsável por transportar as informações entre os agentes e os gerentes, e libera as seguintes classes de operações:

• GET: Que permite a obtenção, por parte da estação de gerenciamento, dos Valores associados aos objetos da MIB;

• SET: que possibilita que a estação de gerenciamento efetue alterações dos valores dos objetos em um dado agente;

• TRAP: que autoriza um agente notificar a estação de gerenciamento sobre algum evento importante.

Conforme Esteves (2013), à primeira variante do SNMP foram adicionadas novas funcionalidades, que permitem monitorar cada vez mais recursos dos equipamentos, com maior e mais segurança. De acordo com Mauro e Schmidt (2001), a essência do SNMP é uma coleção simples de operações (e das informações adquiridas por essas operações) que permitem ao administrador modificar o estado de alguns dispositivos baseados em SNMP. Segundo Goeten (2001), atualmente existem três versões do protocolo SNMP disponíveis para utilização: SNMPv1, SNMPv2c e SNMPv3.

2.5.1.1 Snmpv1

Esta é a versão primária do protocolo SNMP e passou a ser utilizada a partir de meados de 1990 É definida em três RFCs e é padronizada pelo IETF, conforme segue:

1. RFC 1155: define a *Structure of Management Information* (SMI), linguagem utilizada para definir as informações, com o propósito de descrever e nomear os objetos que serão gerenciados;
2. RFC 1212: define a descrição (sintaxe) mais concisa, mas inteiramente consistente com o SMI;
3. RFC 1157: define o *Simple Network Management Protocol* (SNMP).

Abreu e Pires (2014) consideram fraca a segurança desta versão, pois as senhas são baseadas em *community strings*, que são simples *passwords*, *strings* em formato texto aberto, que permitem que qualquer ferramenta de gerência que conheça esta *string* obtenha acesso aos dados deste dispositivo. A vulnerabilidade da utilização de *community strings* está no fato delas serem enviadas sem criptografia.

Dependendo da configuração, através da *community strings* é possível ler e/ou alterar facilmente informações da MIB no agente.

Conforme Santos (2009), para minimizar o risco de captura da informação na *community strings* por usuários não autorizados, é indicada a utilização de *firewalls* para proteger a comunicação SNMP entre os dispositivos e VPNs (Virtual Private Networks) para assegurar a criptografia do tráfego ou então modificar as *communitys* regularmente.

2.5.1.2 Snmpv2

O SNMPv2 acarreta em algumas vantagens, como melhorias na eficiência e no desempenho. Segundo Stallings (1999), após alguns anos de uso da primeira versão do protocolo SNMPv1, algumas deficiências passaram a ser percebidas e as necessidades de melhoria foram identificadas, o que levou ao desenvolvimento da versão do SNMPv2, aprimorada nos quesitos definições de objetos da tabela MIB, procedimentos do protocolo e segurança.

A segunda versão do SNMP é também denominada SNMPv2c (*community string-based SNMPv2*). Ela é definida pelas RFCs 1905, 1906 e 1907 e pelo IETF com o status de experimental. Esta versão além da vantagem no quesito segurança, também teve melhorias nas operações de protocolo com a criação das mensagens *InformRequest* e *GetBulkRequest*, que permitem a comunicação entre gerentes, facilitando a gerência descentralizada da rede e também a otimização da recuperação de informações de equipamentos que são monitorados quando necessário, o que tornou possível o gerenciamento distribuído.

> O protocolo SNMP, refere-se a um conjunto de padrões para gerenciamento que inclui um protocolo, uma especificação de estrutura de dados, e um conjunto de objetos de dados. Este protocolo hoje já está na sua segunda versão oficial, chamada de SNMPv2. (LIMA, 1997, p. 1).

2.5.1.3 Snmpv3

Conforme LIMA (1997) o SNMP versão 3 foi criado para suprir uma necessidade de padronização com as variações do SNMPv2, que tentavam criar soluções de segurança para o protocolo. A versão 3 (SNMPv3) é definida nas RFCs 1905, 1906, 1907, 2570, 2571, 2572, 2573, 2574, 2575, 2576 e 2786. No SNMPv3 ocorreu a inclusão de uma autenticação rigorosa e comunicação privativa entre as entidades gerenciadas.

Além das questões de segurança, o projeto do SNMPv3 também objetivou uma padronização de implementação das entidades (agente/gerente), modularizando suas funcionalidades, o que facilita a evolução de alguns mecanismos do protocolo sem exigir que novas versões sejam lançadas. Outros objetivos eram a manutenção de uma estrutura simples, a facilitação da integração com outras versões e, sempre que possível, o reaproveitamento das especificações existentes.

O SNMPv3 incorporou o SMI e o MIB do SNMPv2, assim como também utilizou as mesmas operações do SNMPv2, apenas com uma reescrita da norma para uma compatibilização da nomenclatura. De acordo com Mauro e Schmidt, a única alteração relevante da terceira versão do protocolo SNMP (SNMPv3) trata os problemas de segurança que se manifestaram nas versões anteriores (SNMPv1 e SNMPv2). A versão não sofreu alterações no protocolo nem sequer recebeu operações novas, e permanece com suporte para todas as operações definidas nas versões anteriores do protocolo.

Para realizar essas tarefas, SNMPv3 traz o conceito de “principal”, que´é uma entidade pela qual os serviços são providos ou onde o processamento ocorre. O SNMPv3 e modular. Cada entidade SNMP possui uma única máquina SNMP, sendo que uma máquina SNMP implementa funções para enviar e receber mensagens, autenticar e criptografar dados, e realizar o controle de acesso.

Os módulos, ou processadores, do SNMPv3 são apresentados a seguir:

Despacho: prove suporte a versões diferentes do SNMP. Esse processador´e responsável por aceitar PDUs de aplicações para transmitı-los através da rede e entregar PDUs que chegam para as aplicações. Também é responsável por mensagens de saída para o subsistema de processamento de mensagens para

preparar as mensagens, bem como o inverso, e receber as mensagens e repassa-as ao subsistema de processamento de mensagens, para que este possa extrair a PDU entrante;

Subsistema de Segurança: conhecida como Modelo de Segurança do Usuário (do inglês: User Security Model - USM), prove os serviços de segurança, como autenticação e criptografia de mensagens (BLUMENTHAL; WIJNEN, 1999);

Subsistema de Controle de Acesso: prove um conjunto de serviços de autorização que uma aplicação pode utilizar verificar permissões de acesso;

Gerador de Comando: cria as PDUs de requisição get, get-next, getbulke set,e processa as PDUs de resposta;

Receptor de Comandos: recebe PDUs de requisição destinados ao sistema local, realiza o controle de acesso, e executa a devida operação de protocolo. Por fim, gera o PDU de resposta;

Gerador de Notificação: monitora condições e eventos específicos do sistema, e gera traps ou informs de acordo com os valores encontrados. Deve ser configurado no sistema para onde enviar as mensagens;

2.5.2 PuTTY

O PuTTy é um software de emulação de terminal de código livre. Suporta SSH, destinado a suportar o acesso remoto a servidores via shell seguro e a construção de "túneis" cifrados entre servidores. Também suporta conexão direta, telnet, rlogin e por porta serial. Sua utilização será para acesso e configuração dos switches que compõem a rede.

2.5.3 VirtualBox

VirtualBox é uma ferramenta de virtualização multiplataforma, permitindo a criação de diversos sistemas operacionais - dentro de máquinas virtuais - independentemente da plataforma. (VIRTUALBOX, 20-).

No contexto deste trabalho, esta ferramenta foi a escolhida como plataforma para criação do ambiente virtual, para criação do Servidor Nagios.

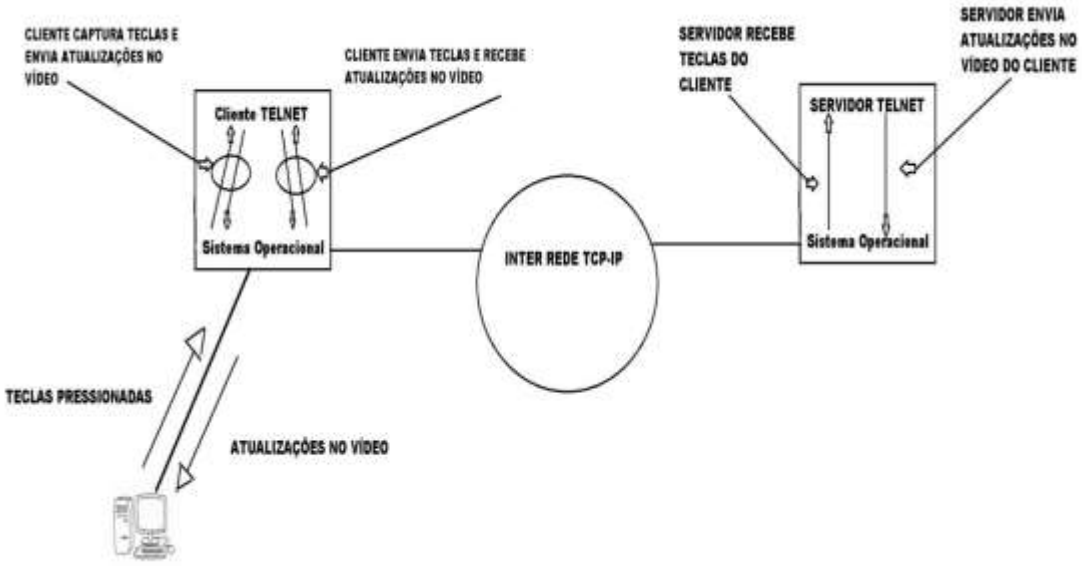
2.5.4 Telnet

O protocolo TELNET oferece a capacidade de *logon* remoto, que permite que um usuário em um terminal ou computador pessoal efetue *logon* com um computador remoto e funcione como se estivesse conectado a esse computador. O protocolo foi desenvolvido para funcionar com terminais simples, no modo de rolagem. TELNET, na realidade é implementado em dois módulos: O usuário TELNET interage como modo de I/O (input e output) do terminal para se comunicar com o terminal local. Ele converte as características dos terminais reais para o padrão de rede e vice-versa. (STALLINGS, 2005).

O TELNET interage com uma aplicação, atuando como um manipulador de terminal substituto, de modo que os terminais remotos apareçam como locais à aplicação. O tráfego de terminais entre o Usuário e o Servidor TELNET é transportado em uma conexão TCP. Ensinam SOARES, LEMOS e COLCHER (1995) que o protocolo TELNET permite que um usuário utilizando uma máquina “A” estabeleça uma sessão interativa com uma máquina “B” na rede. A partir daí todas as teclas pressionadas na máquina “A” são repassadas para a máquina “B” como se o usuário estivesse utilizando um terminal ligado diretamente a ela.

O Telnet pode ser usado para a pesquisa de informações e transferência de arquivos, tudo depende do que o computador ao qual estamos conectados permitir que façamos. Ele também é muito usado por operadores de sistemas (Sysop's) a fim de fazer algum tipo de manutenção (muitas pessoas pensam que o Sysop de nosso provedor sai de casa toda vez que tem algum problema nos servidores, estão muito enganadas, muitas vezes ele faz a manutenção de casa mesmo, via Telnet.).

Figura 5 – Exemplo de conexão via telnet



Fonte:Elaborada pelo autor

A Figura 5 ilustra o funcionamento de uma conexão TELNET. Quando o programa TELNET começa a ser executado em uma máquina, ela passa a funcionar como cliente TELNET. Junto com o comando que dispara a execução do TELNET, o usuário informa o nome, ou o endereço IP, da máquina à qual deseja se conectar.

Segundo Soares, Lemos e Colcher (1995), uma conexão TCP é estabelecida, e a máquina de destino assume o papel de servidor TELNET. Uma vez estabelecida a conexão, todas as teclas pressionadas pelo usuário são capturadas pelo cliente TELNET e enviadas, através da conexão TCP, ao processo servidor TELNET na máquina remota. O servidor processa as teclas e envia de volta para o cliente os caracteres que devem ser mostrados no vídeo do terminal.

2.5.5 Switch

Os switches, muitas vezes são confundidos com os hubs, mas a semelhança só está na aparência, na verdade os switches são inteligentes e ao contrário dos hubs que operam por difusão enviando quadros para todas as portas, enviam quadros de dados somente para a porta de destino do quadro. Para tal é possível apontar suas vantagens como, aumento de desempenho da rede já que o meio físico vai se manter

livre; mais de uma comunicação pode ser estabelecida simultaneamente, desde que envolvam portas de destino que não estejam sendo usadas em outra comunicação (TORRES, 2001, p. 346).

Uma forma de demonstrar as vantagens de um switch em uma rede é outra comparação com o hub. Enquanto um hub de 24 portas de 100 Mbps compartilha esta velocidade proporcionalmente para todas as portas, ou seja, levando em consideração a utiliza go de todas as portas, cada uma trabalhará aproximadamente a 4,5 Mbps, gerando um gargalo na rede. Já com o uso do switch, cada porta vai trabalhar a 100 Mbps dedicados e não vai gera£ broadcast para todas as portas. A origem se comunica com o destino diretamente através de um circuito fechado ponto a ponto (DIOGENES, 2004, p. 162).

Para que estes circuitos fechados possam ocorrer, existem dois modos de transmissão de *frames*, o *store and forward* e *cut-through*. O primeiro aceita e analisa o pacote inteiro antes de encaminhá-lo para a porta de saída, guardando cada quadro em um buffer. Este método permite detectar alguns erros, evitando a sua propagação pela rede. O endereço do destino e da fonte é lido e filtros são aplicados antes do frame ser passado adiante. A latência ocorre enquanto o quadro está sendo recebido. O switch é capaz de checar o frame inteiro a procura de erros, os quais permitem maior detecção de erros. Já no segundo o quadro é remetido pelo switch antes que frame inteiro seja recebido. No mínimo o frame de destino deve ser lido antes que o frame possa ser enviado. Este modo decrementa a latência da transmissão, mas também reduz a detecção de erros, trad. (CCNA 3, 2003).

> Os switches conseguem enviar quadros diretamente para as portas de destino porque eles são dispositivos que aprendem. Quando uma máquina envia um quadro para a rede através do switch, o switch lê o campo de endereço MAC de origem do quadro e anota em uma tabela interna o endereço o MAC da placa de rede do micro que está conectado aquela porta. Assim quando o switch recebe um quadro para ser transmitido, ele consulta essa sua tabela. Se o endereço o MAC de destino constar nessa tabela, ele sabe para qual porta deve enviar o quadro. No entanto se o endereço MAC do quadro for desconhecido pelo switch, isto é, ele não sabe qual porta deve entregar o quadro, ele gera um processo conhecido como inundação, flooding. Ele envia o quadro para todas as suas portas, menos para a porta de origem do quadro. Nesse momento o switch opera igual a um hub. É importante notar que o switch também desaprende endereços MAC. Após um

> determinado período de tempo sem perceber qualquer quadro de um determinado endereço MAC, por exemplo. Cinco minutos, o switch elimina esse endereço de sua tabela. Isso permite que a estrutura física da rede seja alterada e o switch mantenha a sua capacidade de aprendizado, mantendo a rede funcionando, por exemplo, se uma máquina for trocada de porta (TORRES, 2001. p. 349 e 350).

2.5.6 Groovy

Conforme Freitas (2011) Groovy é uma linguagem dinâmica baseada na jvm, totalmente integrado com Java (há quem defina simplesmente como Java em versão script). Também fornece várias simplificações comparadas ao padrão da linguagem Java, além de recursos avançados como closures, properties, regex e suporte nativo a listas e mapas. Groovy permite que classes e métodos sejam alterados em tempo de execução, não necessita de ponto e vírgula para separar os comandos (exceto se estes comandos estiverem na mesma linha).

James Strachan falou sobre o desenvolvimento do Groovy pela primeira vez em seu blog em agosto de 2003. Em março de 2004, Groovy foi enviado ao *Java Community Process* (JCP) como JSR 241 e aceito. Diversas versões foram lançadas entre 2004 e 2006. Depois que o processo de padronização através do JCP começou, a numeração de versão mudou, e uma versão chamada "1.0" foi lançada em 2 de janeiro de 2007. Depois de vários betas numerados como 1.1, em 7 de dezembro de 2007, Groovy 1.1 Final foi lançado e imediatamente renumerado como Groovy 1.5 para refletir as várias mudanças que foram feitas (KOENIG, LAFORGE e GLOVER, 2008).

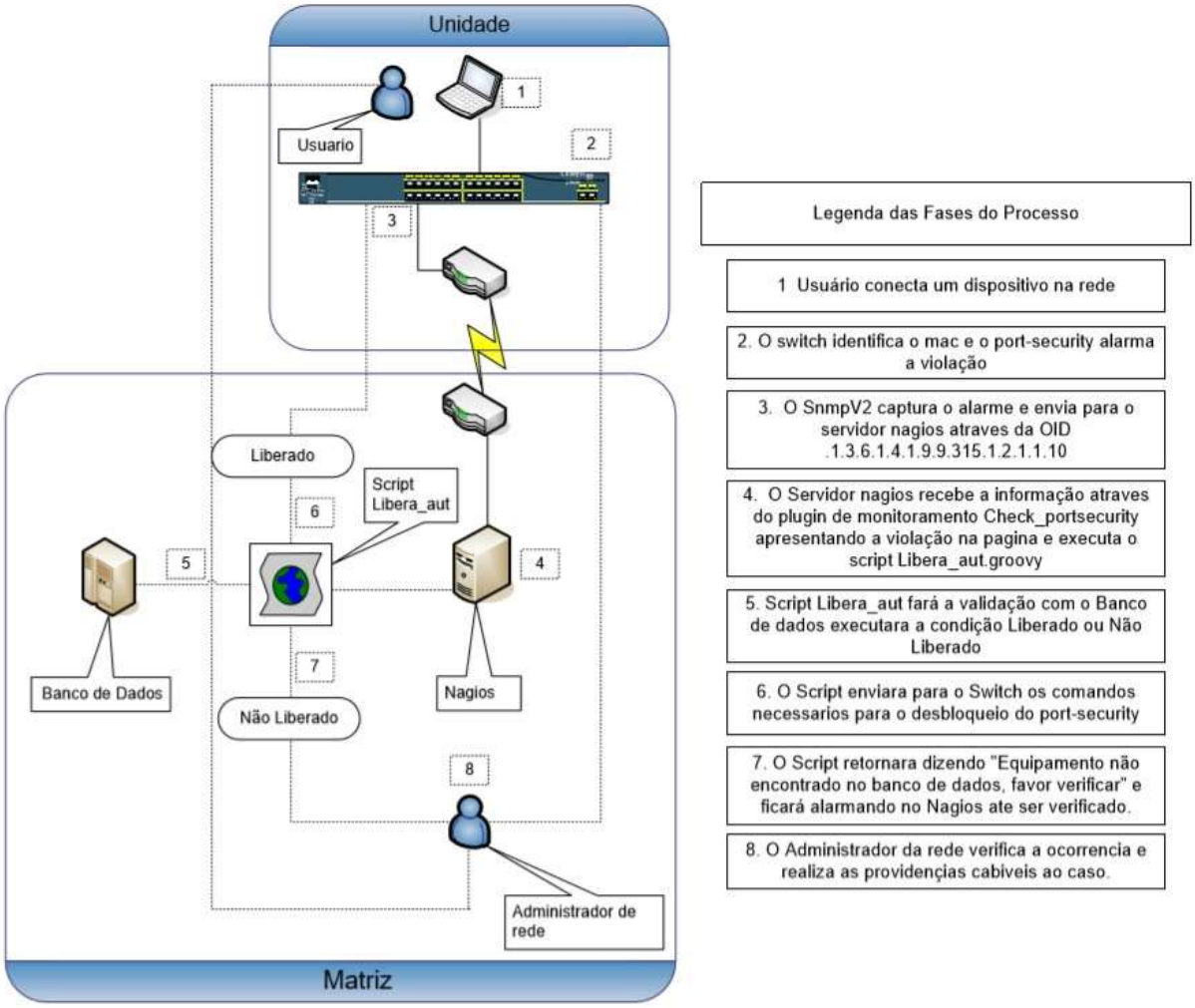
3 DESCRIÇÃO DA SOLUÇÃO

A realização deste trabalho propõe aumentar a segurança de uma rede corporativa controlando o acesso através dos MAC’s dos equipamentos e realizar o monitoramento dos mesmos para agilizar o desbloqueio no mesmo.

Para isto será necessário realizar a implementação da ferramenta de monitoramento NAGIOS, o desenvolvimento de um plug-in para a captura das informações de Port-security, um script em shell script que fara a liberação do equipamento caso ele seja autorizado para acesso à rede e configurar as portas dos switches habilitando o Port-Security.

Na figura 6, é exibido como será o fluxo do processo.

Figura 6 – Fluxograma baseado no processo



Fonte: Elaborada pelo autor

4 METODOLOGIA

Será realizada uma pesquisa exploratória com a finalidade de obter um maior conhecimento sobre o assunto, e com isso, ampliar as minhas possibilidades e aumentar as chances de êxito na solução proposta.

- Criação de ambiente de testes utilizando equipamentos físicos e virtuais;
- Implantação da ferramenta de monitoramento Nagios;
- Implantação da ferramenta port-security;
- Cadastramento dos switches no nagios;
- Configuração de *plug-in* que que faça o monitoramento do port-security;
- Criação de um *script* que faça a liberação caso o equipamento esteja autorizado.

5 IMPLEMENTAÇÃO

A implementação foi realizada seguindo a ordem dada na metodologia. Abaixo segue um resumo do que será abordado neste capítulo.

- No item 5.1 Apresentação do Ambiente Físico
- No item 5.2 Instalação do servidor virutal
- No item 5.3 Aborda sobre a instalação do Nagios
- No item 5.4 Implementação do port-security nos switches
- No item 5.5 Cadastros dos switches no Nagios
- No item 5.6 Configuração do plug-in de monitoração
- No item 5.7 Inserção do script de liberação

5.1 Ambiente de testes - Físico

O primeiro procedimento a ser realizado será a preparação do ambiente de teste, para isto será utilizado um computador com Windows 10 e virtualbox instalado, 3 Switches CISCO 2960 e um notebook para testes, conforme Figura 7.

Figura 7 – Ambiente de teste

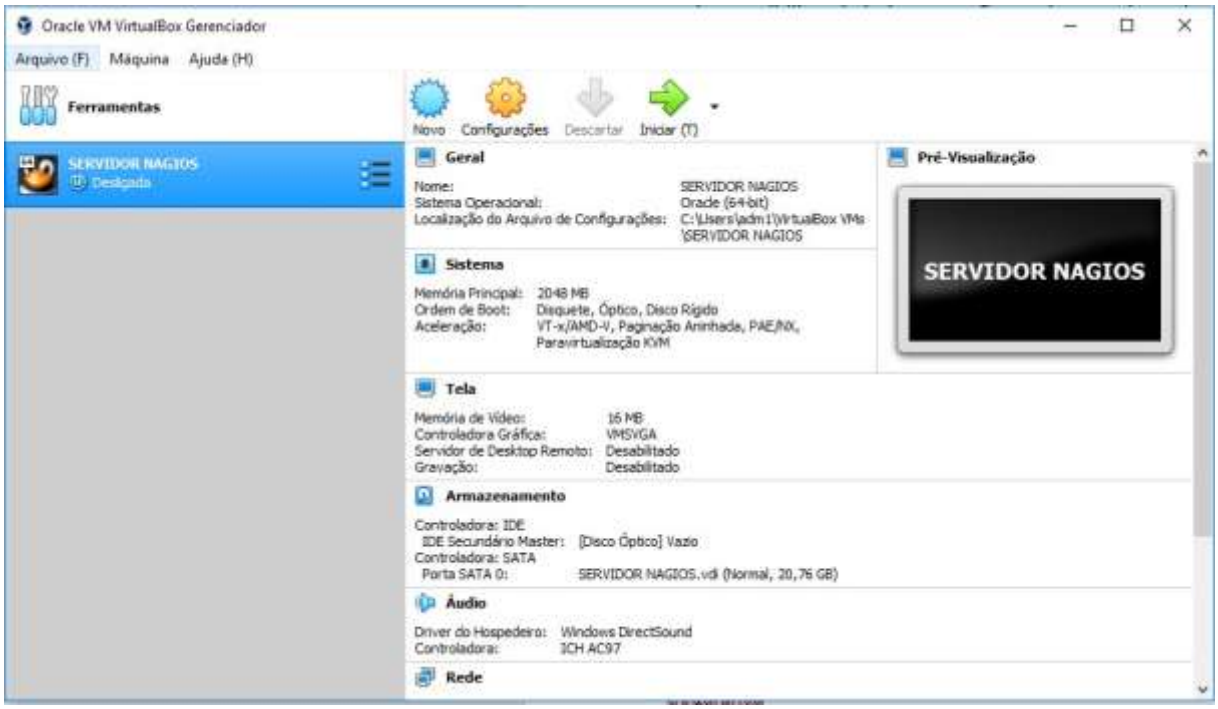
Fonte: Elaborada pelo autor

5.2 Ambiente virtual

Foi realizada a instalação do Oracle Virtual Box 6.0 e utilizado suas configurações padrões. Foi utilizado, como solução de virtualização o Oracle Virtual Box, sendo assim possibilitando baixo custo financeiro para homologação e testes.

Com a instalação do Virtual Box, foi possível criar uma máquina virtual e por questões de familiaridade foi utilizado o Sistema Operacional Debian Server 10.1.0. Este servidor será utilizado para a instalação da ferramenta de monitoração Nagios.

Figura 8 - Interface do VirtualBox



Fonte: Elaborada pelo autor.

5.3 Implantação do NAGIOS

Embora a configuração do Nagios pode se tornar demorada, existe a opção de somente modificar uma pequena parte dela e conseguir deixar o sistema ativo e executando. Por sorte, muitos parâmetros no Nagios já estão definidos com valores de padrão aceitáveis.

Nagios inclui uma documentação extensa própria, uma vez instalado no diretório /usr/local/nagios/share/docs, que pode também ser acessado pela interface web. Esta é sempre recomendada como uma fonte útil para maiores informações.

5.3.1 Pré-Requisitos

Antes da instalação do Nagios, se faz necessária a instalação de alguns softwares como Apache2, Openssl, mcrypt, nmap, inetd, gd, libpng, libjpeg e atualização do sistema. Para instalar os pré-requisitos será utilizado os comandos abaixo.

```
#apt-get updade
#apt install php7.3
#apt-get install -y autoconf gcc libc6 make wget unzip apache2 apache2-utils php libgd-dev
#apt-get install libapache2-mod-php5
#apt-get install build-essential
#apt-get install libgdb-xpm-dev
```

5.3.2 Instalação do Nagios

Neste capítulo se mostrado como o Nagios foi instalado, seguindo sua documentação localizado em: ***https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#Ubuntu***

Para iniciar a instalação do Nagios será criado a conta “nagios” e com respectiva senha “nagios”.

Criando usuário:

```
snagios@SNAGIOS:~$ su
root@SNAGIOS:/home/nagios#/usr/sbin/useradd–m nagios
```

Alterando senha:

```
root@SNAGIOS:/home/nagios# passwd nagios
```

Criado um grupo para permissão de acesso externo através da interface web.

```
root@SNAGIOS:/home/nagios# /usr/sbin/goupadd nagios
```

Realizado o download dos pacotes do Nagios (versão 4.4.5) e de seus Plugins (versão 2.2.1) disponíveis em:

Nagios:

http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.4.5.tar.gz

Pluguins:

http://www.nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz

Após os downloads serem concluídos foi realizado o acesso a pasta “tmp” e feita a descompactação dos dois arquivos baixados

```
root@SNAGIOS:/home/nagios# cd /tmp root@SNAGIOS:/home/nagios/tmp#
tar xvzf nagios-4.4.5.tar.gz  root@SNAGIOS:/home/nagios/tmp#  tar xvzf
nagios-plugins-2.2.1.tar.gz
```

Depois de realizada a descompactação dos dois arquivos foi realizado o acesso ao diretório “nagios-4.4.5”.

```
root@SNAGIOS:/home/nagios/tmp#  cd nagios-4.4.5
```

Realizada a compilação do nagios e a configuração de alguns parâmetros do apache.

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# ./configure --with-httpd
conf=/etc/apache2/sites-enabled root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make
all root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make install
```

Após a realização das configurações dos parâmetros foi realizada a compilação e a instalação do Nagios.

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make install-daemoninit
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make install-commandmode
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make install-config
```

Instalando o Apache:

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#make install-webonf
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#a2enmod rewrite
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5#a2enmod cgi
```

Criando um usuário no apache para acesso a interface web:

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# htpasswd -c
/usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Configurando Firewall:

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# iptables -I INPUT -p tcp --destination-port
80 -j ACCEPT root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# apt-get install -y iptables-
persistent
```

Iniciando serviço do apache:

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# systemctl restart apache2.service
```

Iniciando serviço do Nagios:

```
root@SNAGIOS:/home/nagios/tmp/nagios-4.4.5# systemctl start nagios.service
```

Realizada a instalação do Nagios com sucesso, conforme Figura 9:

Figura 9 – Página inicial do Nagios

Fonte: Elaborada pelo autor.

5.3.3 Instalação dos Plug-ins do Nagios

Na sequência, foi instalado o *plug-ins* do Nagios para carregamento das funções utilizadas no projeto.

Compilando e Instalando:

```
root@SNAGIOS:/home/nagios/tmp/# cd nagios-plugins-2.2.1
root@SNAGIOS:/home/nagios/tmp/nagios-plugins-2.2.1#../toosl/setup
root@SNAGIOS:/home/nagios/tmp/nagios-plugins-2.2.1#./configure
root@SNAGIOS:/home/nagios/tmp/nagios-plugins-2.2.1# make
root@SNAGIOS:/home/nagios/tmp/nagios-plugins-2.2.1# make install
```

5.3.4 Instalação do NagiosQL

Baixando NagiosQL

```
root@SNAGIOS:/home/nagios/tmp#wget
http://sourceforge.net/projects/nagiosql/files/nagiosql/NagiosQL%203.2.0/nagiosql_320.tar.gz
```

Descompactando NagiosQl e copiando seu conteúdo para a o Nagios

```
root@SNAGIOS:/home/nagios/tmp/# tar -zxvf nagiosql_320.tar.gz
root@SNAGIOS:/home/nagios/tmp/# cp -R nagiosql32/ /usr/local/nagios/share/nagiosql
```

Adicionando permissões ao usuário apache (www-data) na pasta nagiosql e todos os arquivos dentro

```
root@SNAGIOS:/home/nagios/tmp/#chown
www-data.www-data /usr/local/nagios/share/nagiosql –R
root@SNAGIOS:/home/nagios/tmp/#
```

Alterando o grupo da pasta “etc” e adicionando permissão de escrita nas pastas “etc” e “/bin/nagios”

```
root@SNAGIOS:/home/nagios/tmp/# chgrp -R nagcmd /usr/local/nagios/etc/
root@SNAGIOS:/home/nagios/tmp/# chmod -R g+w /usr/local/nagios/etc/
root@SNAGIOS:/home/nagios/tmp/# chown nagios.nagcmd /usr/local/nagios/bin/nagios
```

Adicionando permissões dos usuários “www-data” e “nagios” ao grupo “nagcmd”

```
root@SNAGIOS:/home/nagios/tmp/# usermod -a -G nagcmd www-data
root@SNAGIOS:/home/nagios/tmp/# usermod -a -G nagcmd nagios
```

Após realizar todos os procedimentos acima o apache foi reinicializado

```
root@SNAGIOS:/home/nagios/tmp/# /etc/init.d/apache2 restart
```

Iniciando a instalação do nagiosQL através da interface web pela url abaixo e conforme a figura 10.

```
http://localhost/nagios/nagiosql/install/index.php
```

Figura 10 – Interface de instalação web do nagiosQL

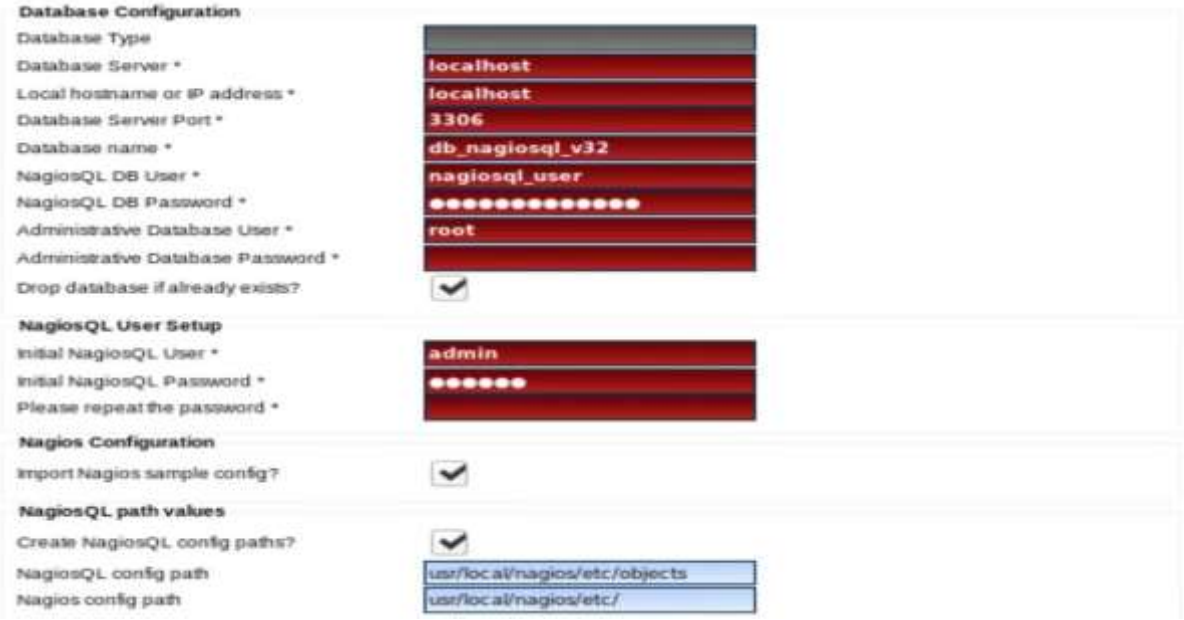
Fonte: Elaborada pelo autor.

As Figuras 11, 12 e 13 exibem os passos necessários para a conclusão da instalação do NagiosQL. Entre elas, é exibido, também, a criação do Banco de Dados:

Figura 11 – Requisitos DO NagiosQL

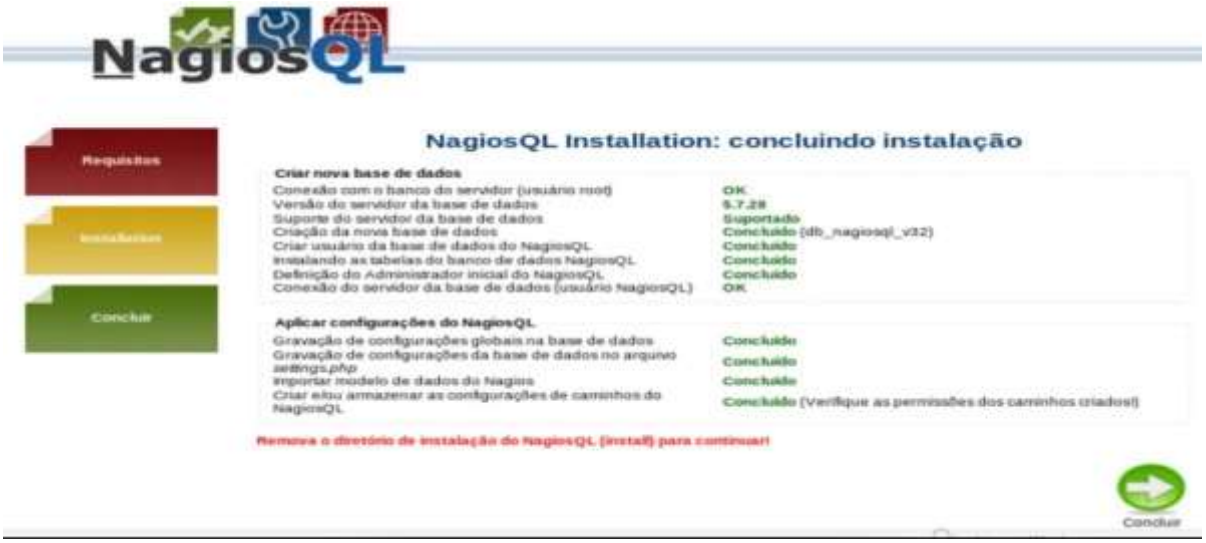
Fonte: Criada pelo autor.

Figura 12 - página de criação do banco de dados do NagiosQl



Fonte: Elaborada pelo Autor

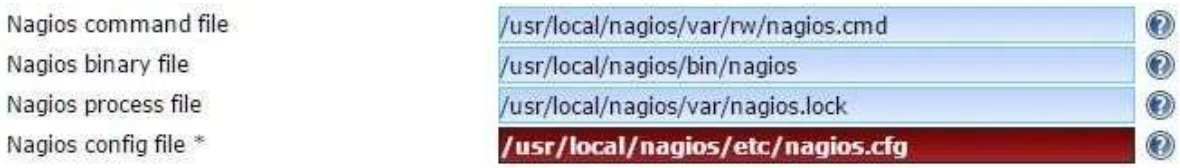
Figura 13 – NagiosQl instalação concluída



Fonte: Elaborada pelo Autor.

Após a instalação o último passo para deixar o NagiosQL configurado para receber os hosts e serviços de monitoramento é alterar o caminho de configuração do Nagios na ferramenta NagiosQL indo em “administration/config targets” e alterar o campo “nagios config. file” para o caminho “/usr/local/nagios/etc/nagios.cfg” conforme mostra a Figura 14:

Figura 14 – Caminho de armazenamento de host



Fonte: Elaborada pelo Autor

5.4 Implementação da ferramenta Port-Security

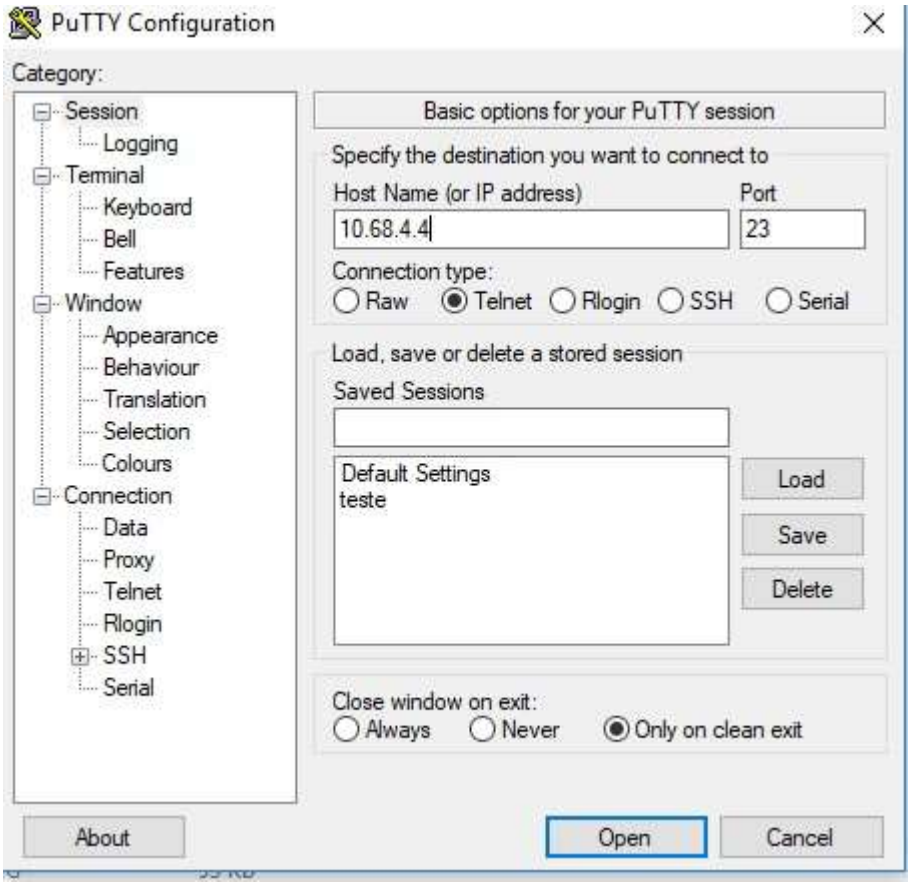
Primeiramente nesta fase de testes a implementação do Port-Security foi aplicada nos 3 switches reservados para testes sendo eles:

- Switch1
- Switch2
- Switch3

5.4.1 Configuração do Switch1

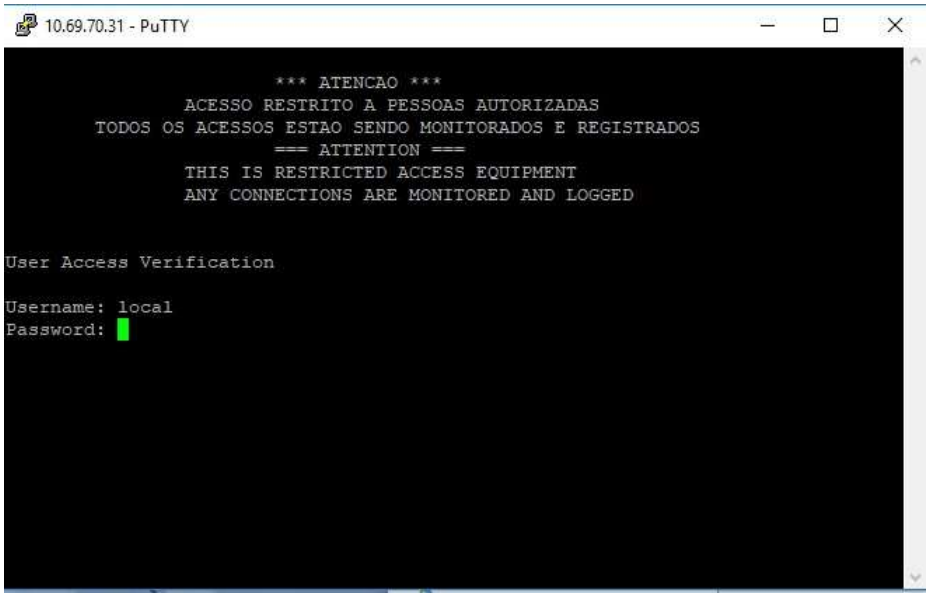
Para configuração do Switch1, foi realizado o acesso via telnet pelo programa putty, através do IP de gerência dado como “10.68.4.4”, conforme Figura 15. Será solicitado o usuário “local” e senha “local”, já pré configurado no switch conforme Figura 16:

Figura 15: Interface do Programa Putty



Fonte: Elaborada pelo autor.

Figura 16: Tela de Login do Switch1



Fonte: Elaborada pelo autor.

Realizando a Habilitação do snmp no switch

```
Switch1# snmp-server enable traps port-security
Switch1# snmp-server host 10.68.4.129 public
```

Aplicando as configurações do port-security nas portas do switch

```
Switch1# int range Fa0/2 – Fa0/24
Switch1(config-if-range)# switchport port-security
Switch1(config-if-range)# switchport port-security maximum 2
Switch1(config-if-range)# switchport port-security violation restrict
Switch1(config-if-range)# switchport port-security mac-address sticky Switch1(config-if-range)#exit
```

Após a aplicação das configurações nas portas elas deverão estar conforme o exemplo da Figura 17:

Figura 17: Configuração da interface Fa0/24 do Switch1

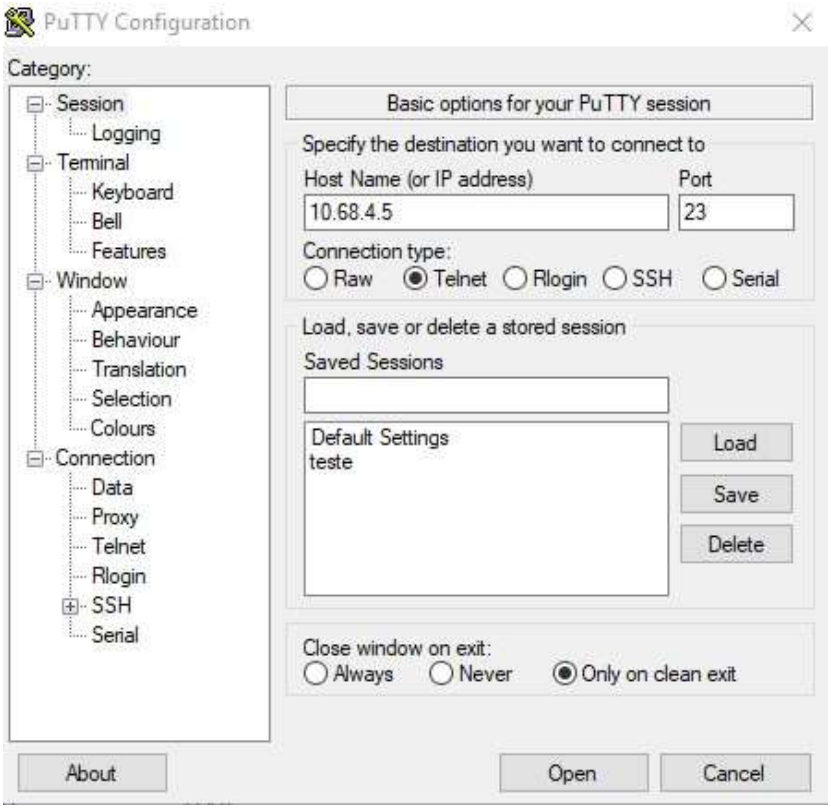
```
Building configuration...

Current configuration : 225 bytes
!
interface FastEthernet0/24
 description TESTE DE PORT-SECURITY
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 switchport port-security violation restrict
 spanning-tree portfast
end
```

Fonte: Elaborada pelo autor.

5.4.2 Configuração do Switch2

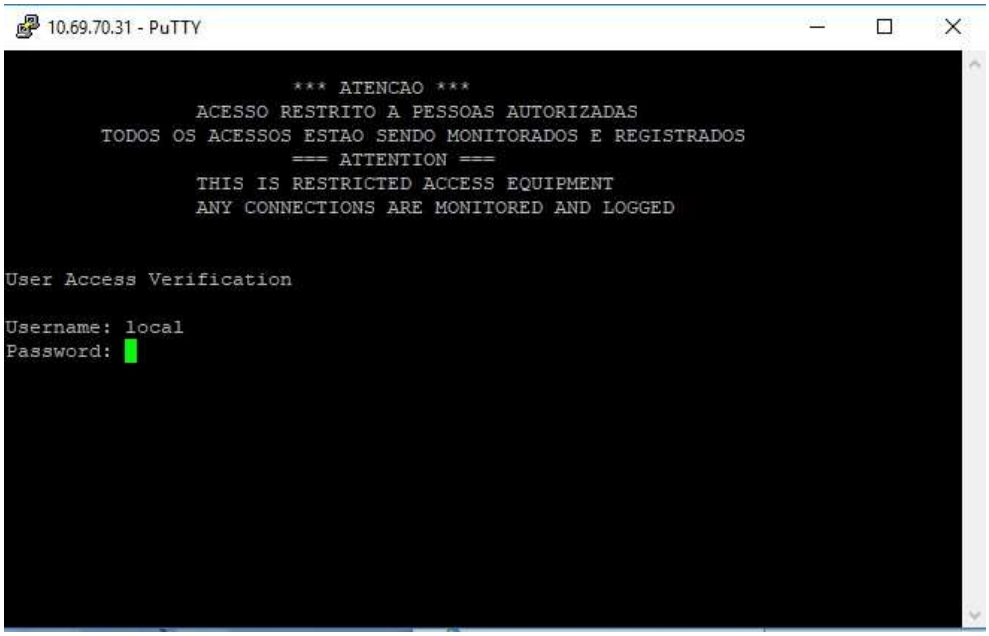
Para configuração do Switch2, foram realizados os passos conforme o Item 5.4.1, onde o acesso via telnet se deu através do IP de gerência dado como “10.68.4.5”, conforme Figura 18. Será solicitado o usuário “local” e senha “local”, já pré configurado no switch, conforme Figura 19:

Figura 18: Interface do Programa Putty



Fonte: Elaborada pelo autor.

Figura 19 - Tela de Login do Switch2



Fonte: Elaborada pelo autor.

Realizando a Habilitação do snmp no switch

```
Switch2# snmp-server enable traps port-security
Switch2# snmp-server host 10.68.4.129 public
```

Aplicando as configurações do port-security nas portas do switch

```
Switch2# int range Fa0/2 – Fa0/24
Switch2(config-if-range)# switchport port-security
Switch2(config-if-range)# switchport port-security maximum 2
Switch2(config-if-range)# switchport port-security violation restrict
Switch2(config-if-range)# switchport port-security mac-address sticky Switch2(config-if-range)#exit
```

Após a aplicação das configurações nas portas elas deverão estar conforme o exemplo da Figura 20:

Figura 20 - Configuração da interface Fa0/24 do Switch2

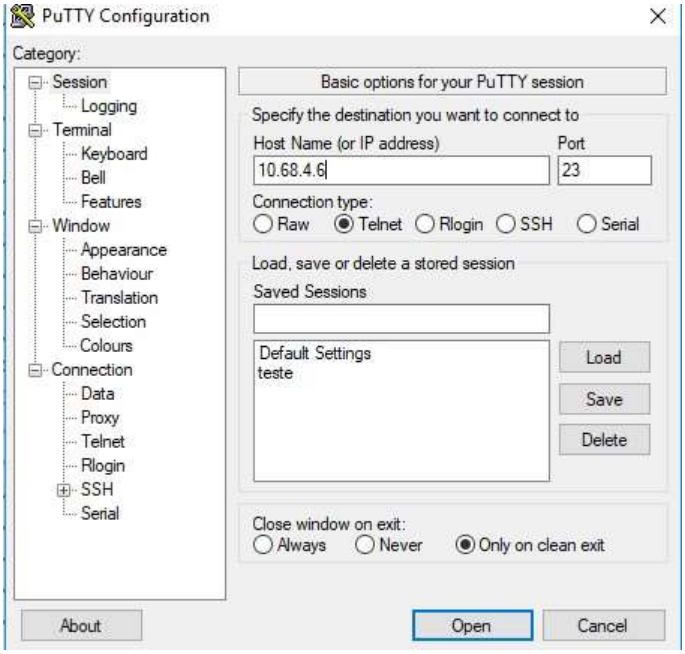
```
Building configuration...

Current configuration : 225 bytes
!
interface FastEthernet0/24
 description TESTE DE PORT-SECURITY
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 switchport port-security violation restrict
 spanning-tree portfast
end
```

Fonte: Elaborada pelo autor

5.4.3 Configuração do Switch3

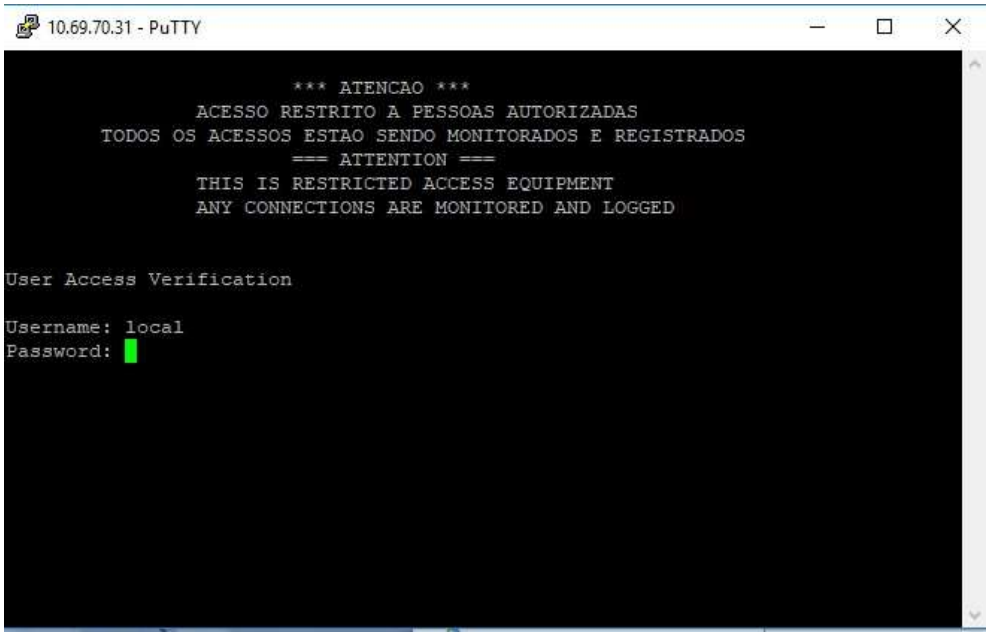
Para configuração do Switch3, foram realizados os mesmos passos anteriores, onde o acesso via telnet se deu através do IP de gerência dado como “10.68.4.6, conforme Figura 21. Será solicitado o usuário “local” e senha “local”, já pré configurado no switch conforme mostra a Figura 22:

Figura 21: Interface do Programa Putty



Fonte: Elaborada pelo autor

Figura 22: Tela de Login do Switch3



Fonte: Elaborada pelo autor

Realizando a Habilitação do snmp no switch

```
Switch3# snmp-server enable traps port-security
Switch3# snmp-server host 10.68.4.129 public
```

Aplicando as configurações do port-security nas portas do switch

```
Switch3# int range Fa0/2 – Fa0/24
Switch3(config-if-range)# switchport port-security
Switch3(config-if-range)# switchport port-security maximum 2
Switch3(config-if-range)# switchport port-security violation restrict
Switch3(config-if-range)# switchport port-security mac-address sticky Switch3(config-if-range)#exit
```

Após a aplicação das configurações nas portas elas deverão estar conforme o exemplo da Figura 23:

Figura 23: Configuração das interfaces dos Switchs

```
Building configuration...

Current configuration : 225 bytes
!
interface FastEthernet0/24
 description TESTE DE PORT-SECURITY
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 switchport port-security violation restrict
 spanning-tree portfast
end
```

Fonte: Elaborada pelo autor

5.5 Cadastrado dos Switches no Nagios

Para realizar o cadastramento dos equipamentos no Nagios, basta acessar a página do NagiosQL, através do caminho http://localhost/nagios/nagiosql e seguir conforme a Figura 24:

Figura 24 – Inclusão de Hosts

Fonte: elaborada pelo autor

Após selecionar a opção “Novo”, será exibido uma página igual a Figura 25, onde será necessário acrescentar as informações de “Nome”, “Endereço”, “Descrição” e “Nome de Exibição”.

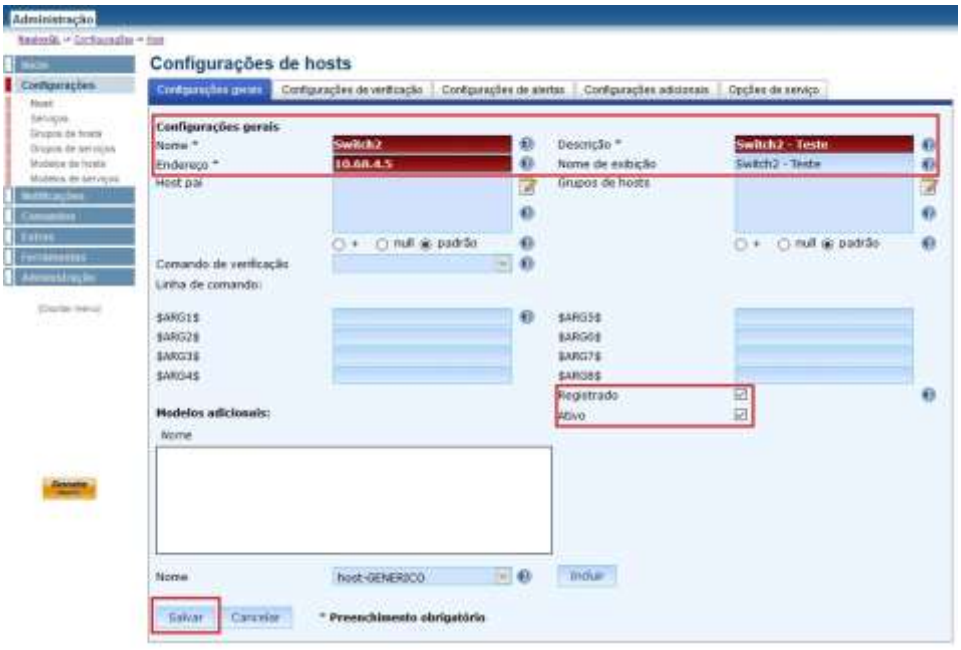
–

Figura 25 Inserção do Switch1

Fonte: elaborada pelo autor.

O mesmo processo foi realizado para os demais switches, conforme Figuras 26 e 27:

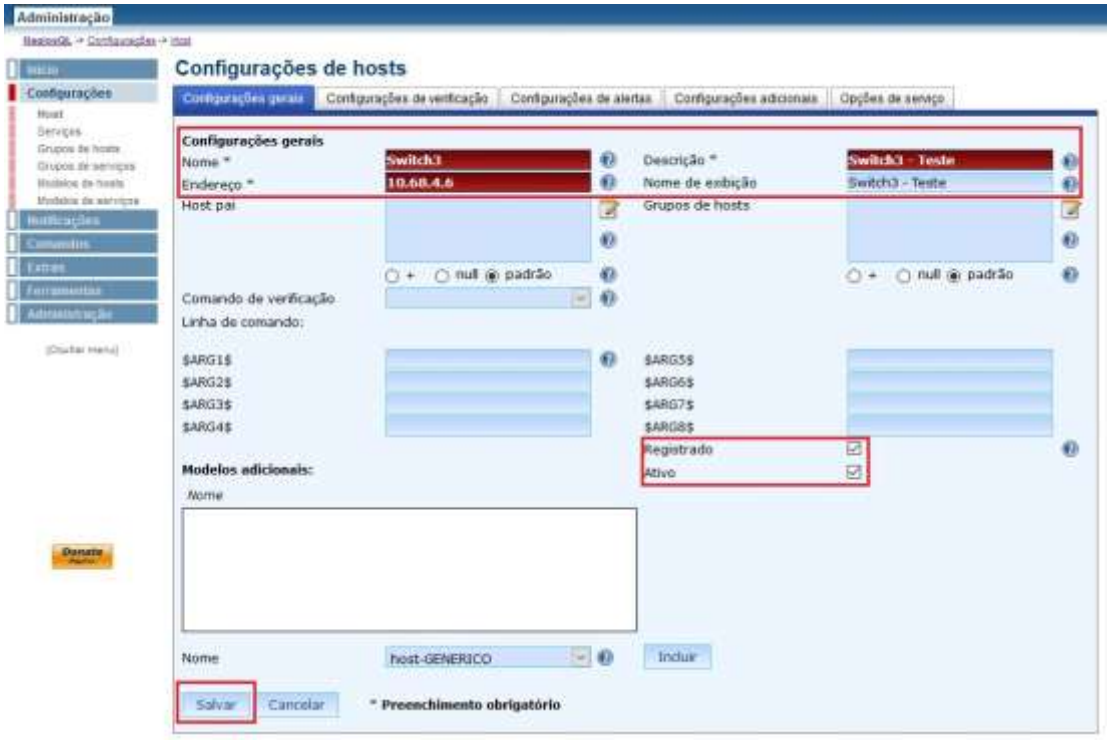
Figura 26 – Inserção do Switch2



Fonte: elaborada pelo autor.

Figura 27 Inserção do Switch3

–



Fonte: elaborada pelo autor.

Ao finalizar, situação dos switches ficará idêntica a Figura 28 em que apresenta, com destaque para o campo “Arquivo”, onde todos estarão como “Não Gravado”:

Figura 28 – Não gravado

Fonte: elaborada pelo autor.

Para realizar a gravação dos arquivos, exibida na Figura 28, é necessário realizar os passos exibidos na Figura 29:

Figura 29 Passos de gravação

–

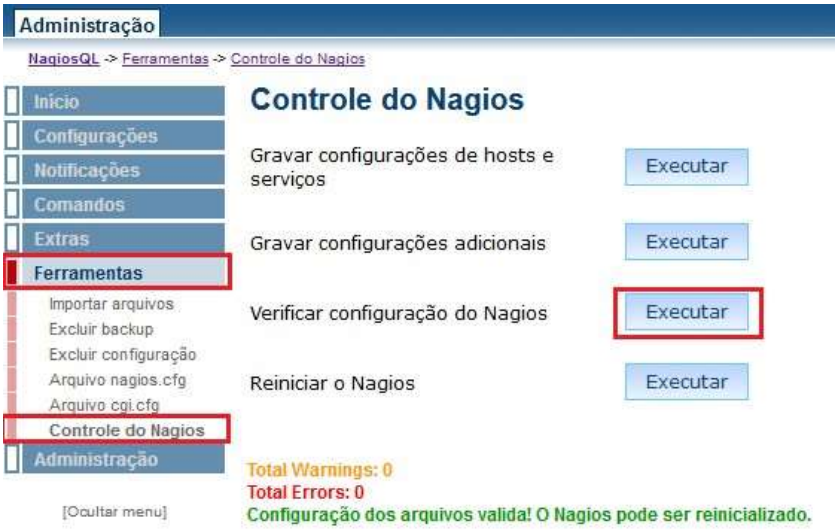
Fonte: elaborada pelo autor

Também se faz necessário a gravação de configurações adicionais, conforme Figura 30, e realizar a verificação das configurações, a fim de confirmar que estão todas corretas como exibido na Figura 31:

Figura 30 – Gravação adicional

Fonte: elaborada pelo autor.

Figura 31 – Verificação de Configuração do Nagios



Fonte: elaborada pelo autor

Por fim, o Nagios está pronto para ser reiniciado, onde entrarão em vigor as alterações realizadas, como exibido na Figura 32:

Figura 32 – Reiniciando do Nagios

Fonte: elaborada pelo autor

Após o Nagios ser reinicializado, os equipamentos aparecerão conforme status exibido na Figura 33. Assim, os equipamentos já estão sendo monitorados pela Ferramenta, e estão disponíveis conforme Figura 34.

Figura 33 – Arquivo gravado

Fonte: elaborada pelo autor

Figura 34 - Monitoração dos Switches

Fonte: elaborada pelo autor

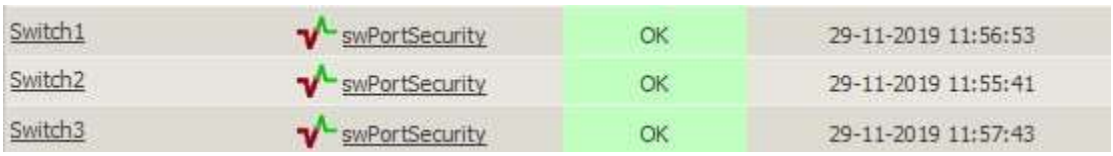
5.6 Configuração do Plug-In Port-Security

O *plug-in* do Port-Security foi adicionado dentro do servidor Nagios, no caminho padrão da instalação da ferramenta, sendo ele o /usr/local/nagios/libexec/. O *plug-in* utilizado foi o check_portsecurity.sh disponível em:

https://github.com/jeanguedes/TCC/issues/1#issue-538630942.

Após a adição do *plug-in*, foi realizada a configuração via NagiosQL dos switches, adicionando-os para monitorar a solução de Port-Security. Assim, os mesmos ficaram monitoráveis no Nagios corretamente, conforme Figura 35:

Figura 35 – Monitoração Port-security



Fonte: elaborada pelo autor

5.7 Script Liberação Automática

O Script para a liberação automática dos equipamentos permitidos foi elaborada em groovy, conforme Apêndice A. Para que o mesmo seja executado, foi necessária a habilitação do serviço “Event Handlers” do Nagios. A habilitação deste serviço foi realizada dentro do plug-in do Port-Security, conforme abaixo:

```
cat /usr/local/nagios/etc/objects/services/check_portsecurity.cfg.| more
define service {
    #NAGIOSQL_CONFIG_NAME        switch_port_sec
hostgroup_name                   switch-UNIDADE-PORT-SECURITY
service_description              swPortSecurity      use
                        servico-generico      check_command
                  check_port_sw      event_handler_enabled      1
    event_handler                  libera_aut
}
```

Após isto, foi realizada a adição do comando “libera_aut”, no caminho /usr/local/nagios/etc/objects/commands.cfg. Nele, foi apontado o caminho do script, conforme abaixo:

```
cat /usr/local/nagios/etc/objects/commands.cfg | more
define command {
    command_name          libera_aut
    command_line          /usr/local/nagios/libexec/eventhandlers/libera_aut.groovy
$HOSTADDRESS$
"     register
1
}
```

6 VALIDAÇÃO

O método de validação do projeto foi baseado em captura de telas, comprovando o pleno funcionamento do ambiente, atingindo os objetivos propostos.

Assim, a Figura 36 exibe a Ferramenta de Monitoração exibe a identificação de equipamento inserido na rede, o qual ficou bloqueado devido a solução do PortSecurity:

Figura 36 – Identificação Equipamento Nagios

Host	Serviço	Estado	Última verificação	Duração	Tentativa	Informação do estado
Switch1	swPortSecurity	WARNING	02-12-2019 15:09:16	0d 0h 05m 40s	5/5	Marca: cisco, Modelo: WS-C2960-24TT-L, Série: FOC1136X0Z5 **Erros de violacão** detectado em **1** interface(s). **Verificar violação de segurança** **Interface:** FastEthernet0/8 - **Qtd Max MacAddr:** 1 - **Qtd Violacão:** 38 - **MacAddress:** 4C-72-B9-A9-F5-65

Fonte: elaborada pelo autor

Já a Figura 37 exibe a atuação do *Script*, de forma automática, para liberação do equipamento inserido na rede, ressaltando que o processo foi realizado pelo usuário de teste para a execução do *Script* (SACSSP05):

Figura 37 – Execução do Script

```
Dec  2 15:01:49: %SEC-6-IPACCESSLOGS: list 80 permitted 10.68.4.67 2 packets
Dec  2 15:08:06: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 4c72.b9a9.f565 on port FastEthernet0/8.
Dec  2 15:08:15: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 4c72.b9a9.f565 on port FastEthernet0/8.
Dec  2 15:08:26: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 4c72.b9a9.f565 on port FastEthernet0/8.
Dec  2 15:08:43: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 4c72.b9a9.f565 on port FastEthernet0/8.
Dec  2 15:09:11: %SYS-5-CONFIG_I: Configured from console by SACSSP05 on vty1 (10.123.20.191)
Dec  2 15:09:13: %SYS-5-CONFIG_I: Configured from console by SACSSP05 on vty1 (10.123.20.191)
Dec  2 15:09:13: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed state to down
Dec  2 15:09:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to down
Dec  2 15:09:17: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed state to up
Dec  2 15:09:18: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to up
Dec  2 15:09:49: %SEC-6-IPACCESSLOGS: list 80 permitted 10.123.20.191 2 packets
Dec  2 15:09:49: %SEC-6-IPACCESSLOGS: list 80 permitted 10.68.4.67 2 packets
```

Fonte: elaborada pelo autor

A Figura 38 exibe a ferramenta de monitoração, após a execução do *script* de liberação, onde percebe-se que a solução foi executada com sucesso:

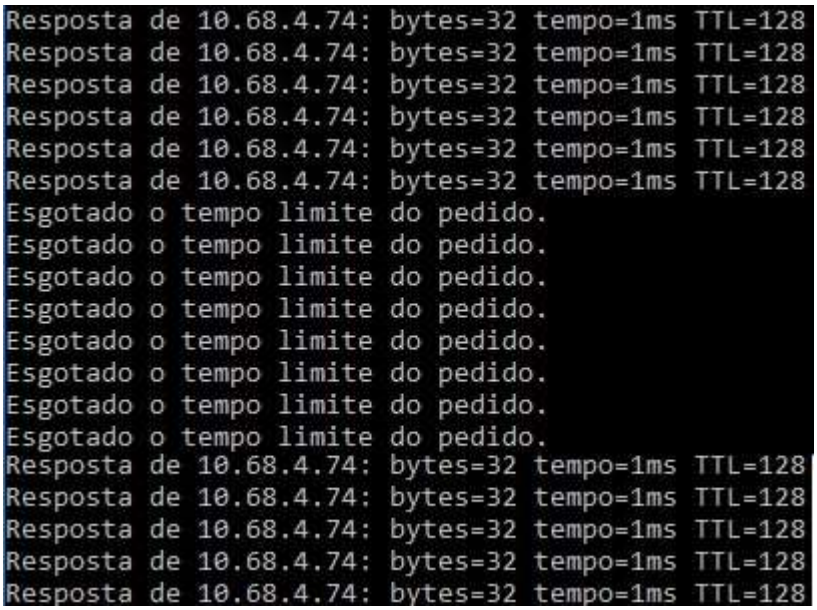
Figura 38 – NAGIOS Após Execução do Script

Host	Serviço	Estado	Última verificação	Duração	Tentativa	Informação do estado
Switch1	swPortSecurity	OK	02-12-2019 15:39:22	0d 0h 30m 46s	1/5	Marca: cisco, Modelo: WS-C2960-24TT-L, Série: FOC1136X0Z5

Fonte: elaborada pelo autor

Por fim, a Figura 39 mostra, apenas para fins de validação, o comportamento do equipamento previamente liberado, ao trocar de ponto lógico com a execução do *script* apresentado nas Figuras 37 e 38:

Figura 39 – Teste de Conectividade



Fonte: elaborada pelo autor

7 CONCLUSÃO

Este trabalho teve como finalidade analisar possíveis alternativas para implementação de uma solução de liberação automática de equipamentos previamente autorizados, integrando ela à ferramenta de monitoração já em produção da empresa. Muito embora não tenha sido objetivo avaliar desempenho e comparação da ferramenta de monitoração Nagios com outras soluções, pôde-se confirmar que há possibilidade de automatização da solução, fornecendo agilidade para o usuário final, sem colocar em risco a segurança dos sistemas.

Outra vantagem é a possibilidade de identificação brevemente e de forma antecipada de equipamentos que estão tentando acesso à rede, que não estão autorizados. Isso facilita para as equipes de operações na investigação da ocorrência (caso seja equipamento intruso, de fato) e/ou liberação de novo equipamento que não estava na base de dados.

Como tema futuro, é possível prosseguir com as configurações descritas neste projeto para os demais equipamentos da rede, contemplando a rede como um todo. Também é possível realizar a criação de página WEB que exibe o histórico de todos os alarmes e liberações realizadas (ou não), para fins de auditoria. Por fim, pode-se ampliar a utilização dos *scripts* para contemplar equipamentos de diferentes fabricantes.

8 REFERÊNCIAS BIBLIOGRÁFICA

ABREU, Fabiano Rocha; PIRES , Herbert Domingues. **Gerência de Redes** . Disponível em: <http://www.midiacom.uff.br/~debora/redes1/pdf/trab042/SNMP.pdf>.

Acesso: 22 nov. 2018.

AGILITYNETWORKS, **Port-security proteção de switches de acesso** Disponível em: <http://www.agilitynetworks.com.br/blogdaagility/port-security-protecao-deswitches-de-acesso> Acesso em: 11 Jun. 2018

ANDRADE, Hetty Alves de. **Nagios como solução de monitoramento de rede**. 2006. Disponível em: http://www.ginux.ufla.br/files/mono-HettyAndrade.pdfAcesso em: 12/06/2018.

BARRETT, D.J; SILVERMAN, R.E; BYRNES, R.G. SSH, **the Secure Shell The Definitive Guide**. 2005. Disponível em: <http://basie.exp.sis.pitt.edu/~christomer/lis2600/readings/SSH_Second_Edition.pdf> Último acesso em: 08/12/2018

BENINI, R. A.; DAIBERT, M. S. Monitoramento de Redes de Computadores - Trabalhando com a ferramenta Nagios. **Infra Magazine**, São Paulo, n. 1, 2013.

BORGES, CRISTIANO **Sniffers de Rede** Disponível em: <http://naticomseguranca.blogspot.com.br/2013/03/sniffers-de-rede.html> Acesso em: 03 abril. 2018

CISCO SYSTEMS. CISCO NETWORKING ACADEMY PROGRAM. CCNA 3: **Switching Basics and Intermediate Routing v3.0, Cisco Systems Incorporation**, 2003.

COSTA, Felipe. **Ambiente de Rede Monitorado: com Nagios e Cacti**. Única Rio de Janeiro: Editora Ciência Moderna, 2008. 189 p.

DIAS, DIEGO **Guia Básico para Configuração de Switches – Segurança.** 2018

DIERK, K; LAFORGE, G; GLOVER, A, **Groovy in Action.** 2008.

DIOGENES, Y, Certificação C/SCO: CCNA 4.0 — **Guia de Certificação Para o Exame #640-801. 3**. ed. Rio de Janeiro: Axcel Books, 2004

ESTEVES, Antonio Matheus Benaion. **Sistema de monitoramento de redes baseado nos protocolos SNMP e Spanning Tree**. Rio de Janeiro, 2013 Dissertação () - Centro Brasileiro de Pesquisas Físicas, 2013

FERRÃO, Ramiro **Diferenças entre o Nagios Core e o XI (Free e Paga)** Disponível em: <http://nagios-br.com/diferencas-entre-o-nagios-core-e-o-xi-free-epaga> último acesso em: 27/11/2018

FREITAS, S. **Groovy – O que é? Como funciona? Vale a pena?.** Disponível em: < https://saviofreitas.wordpress.com/2011/03/20/groovy-o-que-e-como-funciona-vale-apena/> Acesso em: 05 nov. 2019

FRY, C.; NYSTROM, M. **Security Monitoring**. Sebastopol: O'Reilly Media, Inc, 2009.
KOCJAN, W. **Learning Nagios 4. 2ª**. ed. Birmingham: Packt Publishing Ltd, 2014.

LIMA, Michele Mara de A. Espíndula. **Introdução a Gerenciamento de Redes TCP/IP**. Disponível em: http://www.rnp.br/newsgen/9708/n3-2.html. Acessado em: 15 nov. 2011.

LUIZ FERNANDO G., SOARES. **Redes de Computadores:das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 2012.

MAURO, Douglas R.; SCHMIDT, Kevin J. **Essential**: SNMP. United States of America: O'REILLY, 2001.

MORISHITA, Fábio Teruo ; MOREIRA, Edson dos Santos (Org). **Uma avaliação evolutiva dos protocolos de gerenciamento da Internet e suas implementações: SNMPvl, SNMPv2 e SNMPv3**. Tese (Ciências de Computação). Universidade Federal de São Carlos SãoPaulo,1997.

NagiosXi.Disponível em: http://nagios-br.com/nagios-xi <último acesso em 11/11/2018>

Nagios Documentation. Disponível em: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/objectdefinitions.html <último acesso em 11/11/2018>

ODOM, W. **Cisco CCNA: guia de certificação do exame**. 3ª Ed. Rio de Janeiro: Alta Books, 2003.

O'DONAVAN, B.; **GitHub: Nagios_plugins.** Disponível em https://github.com/barryo/nagios-plugins/blob/master/check_portsecurity.pl. Acesso em: 12/09/2019.

PAYNE, B. D.; CARBONE, M. D. P. de A.; LEE, W. **Secure and Flexible Monitoring of Virtual Machines, In: ANNUAL COMPUTER SECURITY APPLICATIONS CONFERENCE, 23**., 2007, Miami Beach, FL. Proceedings… [S.l.:s.n], 2007. p. 385397.

PEREIRA, Hermano Filipe Domingues. **Avaliação de ferramentas de monitorização e gestão de redes**. 2009.

PINHEIRO, J. M. S. **Utilizando os Padroes de Cabeamento, PROJETO DE REDES,2004**. Disponível em: http://www.projetoderedes.com.br/artigos/artigo utilizando_os_padroes. cabeamento.php. Acesso em: 05/11/2018.

PINTO, P. **NagiosQL – A interface gráfica para gerir o Nagios - 2015**. Disponível em: https://pplware.sapo.pt/tutoriais/networking/nagiosql-a-interface-grafica-paragerir-o-nagios/. Acesso em: 01/11/2019.

ROSS, Keith W.; KUROSE, Jim. Redes de Computadores e A Internet - Uma Abordagem Top-Down - **6ª Ed. 2013**

SANTOS, Cinthia Cardoso dos. **Gerenciamento de redes com a utilização de softwarelivre**.2009. Disponível em:<http://www3.iesampa.edu.br/ojs/index.php/sistemas/article/viewFile/442/374> Acesso: 22 nov. 2018.

SOARES, L. F. G.; LEMOS, G.; COLCHER, S. **Redes de Computadores das LANs, MANs e WANs às Redes ATM. 2ª Ed. Rio de Janeiro**: Campus, 1995.

SOARES, L.F.G; LEMOS, G; COLCHER, S; **Rede de computadores: das LANs MANs e WAMs às redes ATM.** Rio de Janeiro: Campus, 1995.

STALLINGS, W. **Redes e Sistemas de Comunicação de Dados: Teoria e Aplicações Corporativas. 2ª Ed. Rio de Janeiro: Elsevier**, 2005, 362p.

STALLINGS, William. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**. 3 ed. California: Addison Wesley Longman, 1999. 619 p.

TORRES, Gabriel. **REDES DE COMPUTADORES. CURSO COMPLETO**. Rio de Janeiro: Axcel Books, 2001.

VERMA, D. C. **Monitoring. In: VERMA, D. C. Principles of Computer Systems and Network Management**. Nova Iorque: Springer, 2009. p. 111-134.

VIRTUALBOX. Virtualização. 20–. Disponível em: https://www.virtualbox.org/manual/ch01.html>. Acesso em: 04/09/2018

VIVAOLINUX. **O MAC Flood! E agora?.** Disponível em: <https://www.vivaolinux.com.br/artigo/MAC-Flood-E-agora>. Acesso em: 09/06/2018

ZARPELÃO, B. B. **Detecção de Anomalias e Geração de Alarmes em Redes de Computadores, Trabalho de Conclusão de Curso de Ciência da Computação**, Universidade Estadual de Londrina, 2004.

APÊNDICE A – SCRIPT LIBERAÇÃO

Abaixo, é apresentado o script “Libera_aut”, que foi utilizado para liberação automática dos equipamentos permitidos:

libera_aut.groovy:

```
import domain.main.entity.Alarme
import java.sql.*; import
groovy.sql.Sql



log.debug("Alarme a ser liberado: {}", alarme) log.debug("Classe
factory: {}", factory)

alarme = (Alarme)

def macAddress = alarme.macAddress.split("[^\\d\\w]").join("") macAddress =
macAddress.substring(0,4) + "." + macAddress.substring(4,8) + "." +
macAddress.substring(8)

def sql = Sql.newInstance('jdbc:mysql://localhost:3306/SIINV', 'ATIVOSCAIXA',
'INVENTCAIXA', 'com.mysql.jdbc.Driver')

sql.firstRow("select count(*) from maquinas where mac = '" & macAddress + "'")

if (sql == 0) {
        log.error("Equipamento não encontrado em nosso banco de dados, Verificar
Ocorrencia!")
        System.exit(1)
}

def success = false

def fullResults = ["SSH": ["DESABILITADO"], "TELNET": []]

["TELNET"].each {
  try {
```

```
        if (!success) {
    def protocol = it
    def connection =
    factory.connect(proto
    col, [
            "HOST"                          : alarme.hostname,
                    "CREDENTIALS"   : credentials,
                    "PROMPT"                : protocol == "TELNET" ? alarme.hostname.substring(0,
alarme.hostname.indexOf('.')) + "#" : null,
                    "USER_PROMPT"           : "Username: ",
                    "PASSWORD_PROMPT" : "Password: ",
                    "LINE_END"        : protocol == "TELNET" ? "\r\n" : "\n"
        ])

        def result = fullResults[it]
    result << connection.send([
    """configure terminal\r\n
    interface $(alarme.iface)\r\n
    shutdown\r\n                end\r\n""",

            "clear port-security sticky interface $(alarme.iface}",
            "clear port-security sticky interface $(alarme.iface} access",
    "clear port-security sticky address $(macAddress}",

            """configure terminal\r\n
    interface $(alarme.iface}\r\n
    no shutdown\r\n                     end\r\n""",
            "echo ${macAddress}"
        ])

        if (it == "TELNET") {           result
    << connection.write("exit")
        }

        log.debug("Result: {}", result)

        connection.close()
    success = true
      }
```

```
   } catch (Exception ex) {        log.error("Erro ao conectar {} usando {}: {}",
alarme.hostname, it, ex,.getMessage(), ex)    }
}
```

fullResults

ANEXO A – EXEMPLO DE HOST.CFG

EX.

define host{

event_handler_enabled Essa diretiva é usada para determinar se o manipulador de eventos para esse serviço está ativado ou não. Valores: 0 = desabilitar o manipulador de eventos de serviço, 1 = habilitar o manipulador de eventos de serviço.
flap_detection_enabled Esta diretiva é usada para determinar se ou não a detecção de flap está habilitada para este serviço
max_check_attempts Esta diretiva é usada para definir o número de vezes que o Nagios tentará o comando de verificação de serviço se ele retornar qualquer Estado que não seja um estado OK. Definir esse valor como 1 fará com que o Nagios gere um alerta sem repetir novamente a verificação de serviço. notification_interval Esta diretiva é usada para definir o número de "unidades de tempo " para aguardar antes de notificar novamente um contato que esse serviço ainda está em um estado não-OK
notification_options Esta diretiva é usada para determinar quando as notificações para o serviço devem ser enviadas para fora. As opções válidas são uma combinação de um ou mais dos seguintes: w = enviar notificações em um estado de aviso, u = enviar notificações em um estado desconhecido, c = enviar notificações em um estado crítico, r = enviar notificações em recuperações (OK estado), f = enviar notificações quando t o serviço começa e pára de bater e s = envia notificações quando o tempo de inatividade programado começa e termina. notification_period 24x7 Esta diretiva é usada para especificar o nome abreviado do período de tempo durante o qual as notificações de eventos para este serviço podem ser enviadas para contatos. Nenhuma notificação de serviço será enviada durante períodos que não sejam cobertos pelo período de tempo.
notifications_enabled Esta diretiva é usada para determinar se ou não as notificações para este serviço estão habilitadas. Valores: 0 = desabilitar notificações de serviço, 1 = habilitar notificações de serviço.
process_perf_data Essa diretiva é usada para determinar se o processamento de dados de desempenho está ativado ou não para esse host. Valores: 0 = desabilitar o processamento de dados de desempenho, 1 = habilitar o processamento de dados de desempenho.
retain_status_information Esta diretiva é usada para determinar se as informações relacionadas ao status sobre o host são mantidas ou não nas reinicializações do programa. Isso é útil apenas se você tiver ativado a retenção de estado usando a diretiva retain_state_information. Valor: 0 = desabilitar a retenção de informações de status, 1 = habilitar a retenção de informações de status
retain_nonstatus_information Esta diretiva é usada para determinar se as informações sem status sobre o host são retidas ou não nas reinicializações do programa. Isso é útil apenas se você tiver ativado a retenção de estado usando a diretiva retain_state_information. Valor: 0 = desativa a retenção de informações que não são de status, 1 = habilita a retenção de informações que não são de status register 0

}

ANEXO B – EXEMPLO DE HOSTGROUP.CFG

EX. define

hostgroup{

hostgroup_name Essa diretiva é usada para definir um nome abreviado usado para identificar o grupo de hosts. alias Esta diretiva é usada para definir um nome ou descrição mais longa usada para identificar o grupo de hosts. Ele é fornecido para permitir que você identifique mais facilmente um determinado grupo de hosts.

contact_groups Esta é uma lista dos nomes abreviados dos grupos de contatos que devem ser notificados sempre que houver problemas (ou recuperações) com esse host. Vários grupos de contatos devem ser separados por vírgulas. Você deve especificar pelo menos um contato ou grupo de contatos em cada definição de host.

members Esta é uma lista das descrições dos serviços (e os nomes de seus hosts correspondentes) que devem ser incluídos neste grupo. Os nomes de host e serviço devem ser separados por vírgulas. Essa diretiva pode ser usada como uma alternativa à diretiva de grupos de serviços em definições de serviço. O formato da diretiva de membro é o seguinte (observe que um nome de host deve preceder um nome / descrição de serviço): }

ANEXO C – EXEMPLO DE CONTACT.CFG

EX. define

contact{

contact_name Essa diretiva é usada para definir um nome abreviado usado para identificar o contato. Ele é referenciado nas definições do grupo de contato. Nas circunstâncias corretas, a macro $ CONTACTNAME $ conterá esse valor.

alias Esta diretiva é usada para definir um nome ou descrição mais longa para o contato. Nas circunstâncias dos direitos, a macro $ CONTACTALIAS $ conterá esse valor. Se não for especificado, o contact_name será usado como o alias.

service_notification_period Esta diretiva é usada para especificar o nome abreviado do período de tempo durante o qual o contato pode ser notificado sobre problemas de serviço ou recuperações. Você pode pensar nisso como um tempo "on call" para notificações de serviço para o contato. Leia a documentação sobre períodos de tempo para mais informações sobre como

isso funciona e problemas potenciais que podem resultar de uso. indevido.Periodos definidos em TIMEPERIODS.CFG

host_notification_period Essa diretiva é usada para definir uma lista dos nomes abreviados dos comandos usados para notificar o contato de um problema ou recuperação do host. Vários comandos de notificação devem ser separados por vírgulas. Todos os comandos de notificação são executados quando o contato precisa ser notificado. A quantidade máxima de tempo que um comando de notificação pode ser executado é controlada pela opção notification_timeout.Periodos definidos em TIMEPERIODS.CFG

service_notification_options w,u,c,r Esta diretiva é usada para definir os estados do host para os quais as notificações podem ser enviadas para esse contato. As opções válidas são uma combinação de um ou mais dos seguintes: d = notificar nos estados do host DOWN, u = notificar nos estados do host UNREACHABLE, r = notificar nas recuperações do host (estados UP), f = notificar quando o host iniciar e parar de flapping e s = envia notificações quando o tempo de inatividade programado do host ou do serviço começa e termina. Se você especificar n (nenhum) como uma opção, o contato não receberá nenhum tipo de notificação de host. host_notification_options d, u, r (d = down / u = notificar / r = recuperações /n = nenhum).

service_notification_commands notify-by-email Esta diretiva é usada para definir uma lista dos nomes abreviados dos comandos usados para notificar o contato de um problema de serviço ou recuperação. Vários comandos de notificação devem ser separados por vírgulas. Todos os comandos de notificação são executados quando o contato precisa ser notificado. A quantidade máxima de tempo que um comando de notificação pode ser executado é controlada pela opção notification_timeout.

host_notification_commands host-notify-by-email Essa diretiva é usada para definir uma lista dos nomes abreviados dos comandos usados para notificar o contato de um problema ou recuperação do host. Vários comandos de notificação devem ser separados por vírgulas. Todos os comandos de notificação são executados quando o contato precisa ser notificado. A quantidade máxima de tempo que um comando de notificação pode ser executado é controlada pela opção notification_timeout.

}

EX.

ANEXO D – EXEMPLO DE CONTACTGROUP.CFG

define contactgroup{

contactgroup_name grupo Esta diretiva é um nome curto usado para identificar o grupo de contato. alias Esta diretiva é usada para definir um nome ou descrição mais longa usada para identificar o grupo de contato.

members This optional directive is used to define a list of the short names of contacts that should be included in this group. Multiple contact names should be separated by commas. This directive may be used as an alternative to (or in addition to) using the contactgroups directive in contact definitions. }

EX.

ANEXO E – EXEMPLO DE SERVICES.CFG

define service{

hostgroup_name Esta diretiva é usada para especificar o (s) nome (s) abreviado (s) do (s) grupo (s) de host em que o serviço é "executado" ou está associado. Vários grupos de hosts devem ser separados por vírgulas. O hostgroup_name pode ser usado em vez de, ou além da, diretiva host_name. service_description Essa diretiva é usada para definir a descrição do serviço, que pode conter espaços, traços e doispontos (ponto e vírgula, apóstrofos e aspas devem ser evitados). Nenhum dos dois serviços associados ao mesmo host pode ter a mesma descrição. Os serviços são identificados exclusivamente com suas diretivas host_name e service_description check_command Essa diretiva é usada para especificar o nome abreviado do comando que o Nagios executará para verificar o status do serviço. A quantidade máxima de tempo que o comando de verificação de serviço pode ser executado é controlado pela opção service_check_timeout. }

EX.

ANEXO F – EXEMPLO DE HOSTEXTINFO.CFG

define hostextinfo{

host name Essa variável é usada para identificar o nome abreviado do host que os dados estão associados.

icon_image Essa variável é usada para definir o nome de uma imagem GIF, PNG ou JPG que deve ser associada a esse host. Esta imagem será exibida no status e informações estendidas CGIs. A imagem vai ficar melhor se for 40x40 pixels de tamanho

icon_image_alt Essa variável é usada para definir uma cadeia de caracteres opcional que é usada na marca ALT da imagem especificada pelo argumento < icon_image >. A marca ALT é usada no status, informações estendidas e statusmap CGIs.

vrml_image Essa variável é usada para definir o nome de uma imagem GIF, PNG ou JPG que deve ser associada a esse host. Essa imagem será usada como o mapa de textura para o host especificado no CGI statuswrl

statusmap_image Essa variável é usada para definir o nome de uma imagem que deve ser associada a este host no statusmap CGI. Você pode especificar uma imagem JPEG, PNG e GIF)

register 0

}

EX.

ANEXO G – EXEMPLO DE TIMEPERIODS.CFG

```
define timeperiod{ timeperiod_name 24x7 alias
24 Horas por dia, 7 Dias da Semana sunday
00:00-24:00 monday          00:00-24:00 tuesday
00:00-24:00 wednesday       00:00-24:00
thursday        00:00-24:00 friday          00:00-
24:00 saturday        00:00-24:00
}
```

EX.

ANEXO H – EXEMPLO DE COMMANDS.CFG

EX. define

command{

command_name Esta diretiva é o nome abreviado usado para identificar o comando. Ele é referenciado nas definições de contato, host e serviço (nas diretivas de notificação, verificação e manipulador de eventos), entre outros locais.

command_line Essa diretiva é usada para definir o que é realmente executado pelo Nagios quando o comando é usado para verificações de serviço ou host, notificações ou manipuladores de eventos. Antes da linha de comando ser executada, todas as macros válidas são substituídas por seus respectivos valores. Consulte a documentação sobre macros para determinar quando você pode usar diferentes macros. Observe que a linha de comando não está entre aspas. Além disso, se você quiser passar um sinal de dólar ($) na linha de comando, você tem que escapar com outro sinal de dólar.