UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

ANÁLISE DE VULNERABILIDADES ATRAVÉS DE *SCANNERS* DETECTORES

TRABALHO DE CONCLUSÃO DE CURSO

Patrícia Monego Buzzatte

Santa Maria, RS, Brasil

2014

TCC/REDES DE COMPUTADORES/UFSM, RS MONEGO, Patrícia Buzzatte Graduada 2014

ANÁLISE DE VULNERABILIDADES ATRAVÉS DE *SCANNERS* DETECTORES

Patrícia Monego Buzzatte

Trabalho de Conclusão de Curso (TCC) do Curso Superior de Tecnologia em Redes de Computadores, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Tecnólogo em Redes de Computadores**

Orientador: Prof. Ms. Renato Preigschadt de Azevedo

Santa Maria, RS, Brasil

2014

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Conclusão de Curso

ANÁLISE DE VULNERABILIDADES ATRAVÉS DE *SCANNERS* DETECTORES

elaborado por
Patrícia Monego Buzzatte

COMISSÃO EXAMINADORA

Renato Preigschadt de Azevedo, Ms.
(Presidente/Orientador)

Murilo Cervi, Dr. (UFSM)

Simone Regina Ceolin, Dra. (UFSM)

Santa Maria, 04 de julho de 2014.

RESUMO

TRABALHO DE CONCLUSÃO DE CURSO
CURSO SUPERIOR DE TECNOLOGIA EM REDES COMPUTADORES
UNIVERSIDADE FEDERAL DE SANTA MARIA

ANÁLISE DE VULNERABILIDADES ATRAVÉS DE *SCANNERS* DETECTORES

AUTOR: PATRÍCIA MONEGO BUZZATTE

ORIENTADOR: RENATO P. DE AZEVEDO

Data e local da Defesa: Santa Maria, 04 de Julho de 2014

Atualmente é impossível imaginar o mundo sem as redes de computadores para a transmissão das informações. Entretanto para que essas possam funcionar de modo seguro, preciso e confiável, é necessário garantir a integridade e disponibilidade dos dados que por ela trafegam. Nesse ponto alguns fatores são pontuais para que isso ocorra efetivamente, por exemplo: um *firewall* com suas regras devidamente configuradas, chaves criptográficas fortes, usuários com permissões apropriadas para sua função, monitoramento das redes através de sistemas de detecção de intrusão são algumas rotinas que devem ser mantidas. Profissionais de segurança em redes costumam empregar ferramentas que auxiliam na detecção de vulnerabilidades como um mecanismo auxiliar na avaliação da segurança interna da rede. Este trabalho tem como objetivo apresentar *scanners* detectores de vulnerabilidades capazes de realizar a varredura das redes domésticas e corporativas e apresentar relatórios indicando os pontos mais propensos às vulnerabilidades, bem como portas abertas, serviços ativos e *patches* ausentes.

Palavras-chave: Redes de Computadores. Segurança. Vulnerabilidade.

ABSTRACT

COMPLETION OF COURSE WORK
SUPERIOR COURSE OF TECHNOLOGY IN COMPUTER NETWORKS
FEDERAL UNIVERSITY OF SANTA MARIA

VULNERABILITY ANALYSIS THROUGH SCANNER DETECTORS

AUTHOR: PATRÍCIA MONEGO BUZZATTE

ADVISOR: RENATO P. DE AZEVEDO

Defense Place and Date: Santa Maria, July 04, 2014

Currently it is impossible to imagine the world without computer networks for the transmission of information. However, for these to function so secure, accurate and reliable, it is necessary to ensure the integrity and availability of data that passes through it. At this point some factors are specific to that effectively occur, for example, a firewall properly configured your rules, strong cryptographic keys, users with appropriate permissions to its function of monitoring networks through intrusion detection systems are some routines that must be maintained. Professional network security usually employ tools that assist in detecting vulnerabilities as a mechanism to assist in evaluating internal network security. This work has aims to present detectors scanners capable of performing vulnerability scans of domestic and corporate networks and report indicating the points more likely to vulnerabilities and open ports, active services and missing patches.

Keywords: Network Computer. Security. Vulnerability.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACL	-	*Access Control List*
AES	-	*Advanced Encryption Standard*
CBC	-	*Cipher Block Chaining*
CTISM	-	*Colégio Técnico Industrial de Santa Maria*
DES	-	*Data Encryption Standard*
DNS	-	*Domain Name System*
DoS	-	*Denial-of-Service Attack*
EAP	-	*Extensible Authentication Protocol*
FTP	-	*File Transfer Protocol*
HIDS	-	*Host Intrusion Detection System*
HTML	-	*Hyper Text Markup Language*
HTTP	-	*Hypertext Transfer Protocol*
ICMP	-	*Internet Control Message Protocol*
IDS	-	*Intrusion Detection System*
IMAP	-	*Internet Message Access Protocol*
IP	-	*Internet Protocol*
MAC	-	*Media Access Control*
MD5	-	*Message-Digest algorithm 5*
MIB	-	*Management Information Base*
MITM	-	*man-in-the-middle*
NASL	-	*Nessus Attack Scripting Language*
NDIS	-	*Network Intrusion Detection System*
NVT	-	*Network Vulnerability Test*
OMP	-	*OpenVas Management Protocol*
OTP	-	*OpenVas Transfer Protocol*
PDF	-	*Portable Document Format*
PDU	-	*Protocol Data Unit*
PHP	-	*Personal Hypertext Preprocessor*
POP3	-	*Post Office Protocol*
RC4	-	*Rivest Cipher 4*
RSA	-	*Rivest, Shamir, Adleman*
SHA-1	-	*Secure Hash Algorithm*
SMTP	-	*Simple Mail Transfer Protocol*
SNMP	-	*Simple Network Management Protocol*
SSH	-	*Secure shell*
SSL	-	*Security Sockets Layer*
TCP	-	*Transmission Control Protocol*
TLS	-	*Transport Layer Security*
UDP	-	*User Datagram Protocol*
UFSM	-	*Universidade Federal de Santa Maria*
XML	-	*eXtensible Markup Language*

SUMÁRIO

INTRODUÇÃO

A *internet* tornou-se indispensável à grande maioria da população. Ela é utilizada para realizar diversas atividades do dia a dia, tais como: fazer transações bancárias, compras *online*, redes sociais, entre outras atividades. O alto grau de conectividade além de grandes benefícios inseriu em ambientes virtuais incidentes que comprometem a segurança das redes, fazendo com que massivos investimentos em ferramentas de proteção contra invasores acompanhem este crescimento (KUROSE, 2006).

Para Nakamura e Geus (2007), ambientes de redes, quando não bem configurados, podem apresentar falhas passíveis de ataques internos ou externos que podem comprometer o seu bom funcionamento, tornando-o mais lento e acessível às pessoas não autorizadas, através da exploração de vulnerabilidades, que são *bugs* na implementação. Ataques exploram 'brechas' existentes em qualquer nível relacionado à proteção da informação que são: sistema operacional, serviços e protocolos, rede e telecomunicações, aplicação, usuários e organização (NAKAMURA; GEUS, 2007).

Para estruturação de um ambiente de rede seguro é preciso analisar alguns pontos básicos na configuração das políticas de segurança. Estas, por sua vez, fornecem um conjunto de regras, leis e práticas destinadas à gestão da segurança. Criptografia, assinatura digital, autenticação e controle de acesso são alguns dos mecanismos utilizados para implementação destas políticas, pois provém um conjunto de ferramentas gerenciáveis (DUMONT, 2006). Às ferramentas citadas anteriormente, pode-se somar ainda os sistemas de detecção de intrusão (IDS) que monitoram o tráfego da rede, e equipamentos de restrição e controle de tráfego como *firewall*, utilizados para reforçar a segurança e deixar o ambiente mais seguro.

De acordo com Kurose (2006), proteger a comunicação e os recursos da rede é o fator primordial para definir uma comunicação segura. Sendo assim, a segurança da rede não envolve apenas sua proteção, mas também a detecção de falhas, ataques à infra-estrutura e reações a serem tomadas. O monitoramento das ameaças torna-se necessário para que se detectem mudanças na rede. Através de *scanners* detectores de vulnerabilidades é possível realizar diversos testes na rede e procurar falhas de segurança. Os *Scanners* são programas de varredura de rede utilizados para detectar

vulnerabilidades em sistemas, sua funcionalidade consiste em procurar por falhas de segurança na rede para corrigi-las antes que sejam exploradas por intrusos, obtendo alguma vantagem ou causando prejuízo (MOREIRA et al., 2008).

O presente trabalho tem como objetivo principal apresentar *softwares* capazes de identificar possíveis vulnerabilidades em redes de computadores. Realizar um estudo sobre as características essenciais de cada ferramenta, e efetuar testes que serão realizados junto à Universidade Federal de Santa Maria (UFSM), pretende-se verificar quais das ferramentas apresentadas possui o melhor desempenho e o maior número de vulnerabilidades detectadas.

A organização deste trabalho está segmentada em Capítulos. O Capítulo 2 apresenta a revisão bibliográfica, com conceitos relacionados. O Capítulo 3 tem por escopo a proposta de desenvolvimento do trabalho. O Capítulo 4 engloba os testes e os resultados obtidos destacando as vulnerabilidades mais encontradas. Por fim, no Capítulo 5, encontram-se as considerações finais deste estudo, bem como, as propostas de trabalhos futuros. Por fim, no Capítulo 6 as referências bibliográficas deste trabalho.

1 REVISÃO BIBLIOGRÁFICA

Este Capítulo descreve os conceitos fundamentais para a compreensão dos temas abordados no presente estudo, bem como elaborar uma pesquisa detalhada e realizar o levantamento dos principais conceitos que auxiliem na elaboração do mesmo.

A seguir são apresentados alguns conceitos importantes para elaboração deste trabalho. A Seção 2.1 expõe o sistema da Segurança da Informação, a Seção 2.2 descreve as principais ameaças em um sistema ou rede de computadores, a Seção 2.3 descreve alguns tipos de vulnerabilidades em redes de computadores, a Seção 2.4 apresenta tipos de ataques, e finalmente, nas Seções 2.5, 2.5.1, 2.5.2, 2.5.3, 2.5.4 são apresentados as formas de prevenção contra ataques e as formas de proteção.

1.1 Segurança da Informação

Segundo Tanembaum (2003), a segurança em redes de computadores nas primeiras décadas de sua existência não necessitou de muitos cuidados, pois era utilizada principalmente por pesquisadores universitários para correio eletrônico e funcionários de empresas que precisavam compartilhar impressoras. Mas o notório crescimento no uso de meios de comunicação, em especial a *internet*, tornou possível o acesso aos mais diversos serviços dos mais distintos locais, através de um computador ou um telefone celular conectado a *internet* (CARISSIMI et al., 2009).

Confiabilidade, integridade e disponibilidade são propriedades fundamentais para segurança de redes. Toda a informação deve chegar aos usuários de forma integra e confiável, para isso, todos os elementos de rede por onde os dados serão transmitidos até chegar ao seu destino final devem estar disponíveis (NAKAMURA; GEUS, 2007). A Norma de Segurança da Informação (NBR ISO/IEC 17799:2000) define as propriedades fundamentais da segurança de redes, como:

- Confiabilidade: garantir que a informação será acessada somente por pessoas autorizadas;

- Integridade: garantir que as informações serão mantidas nos seu estado original, sem alterações;
- Disponibilidade: garantir que usuário autorizado tenha acesso às informações sempre que necessário.

Desta forma a segurança não deve ser considerada apenas uma forma de proteção, mas sim a capacidade de prevenção, monitoração e respostas a um eventual incidente (NAKAMURA; GEUS, 2007).

Para Moraes (2008) existem soluções, tecnologias e dispositivos que podem garantir um ambiente seguro, com conhecimento adequado e aplicabilidade correta, e podem prevenir ataques como, por exemplo, acesso não autorizado ou uso não autorizado de computadores e redes.

1.2 Ameaças

Ameaças são elementos que tem a condição de explorar vulnerabilidades e causar problemas a sistemas de informação e redes de computadores, incluindo fraudes eletrônicas, espionagem, sabotagem, vandalismo, incêndio e inundação (NBR ISO/IEC 17799:2000).

Sêmola (2003) classifica ameaça quanto à sua intencionalidade, que são:

- Naturais: Ameaças causadas por fenômenos da natureza, como incêndios, enchentes, terremotos, tempestades, poluição, entre outros;
- Involuntários: Ameaças inconscientes, geralmente causadas pelo desconhecimento, como erro, acidente, falta de conhecimento dos ativos;
- Voluntárias: Ameaças propositais, causada por *hackers*, invasores, criadores e disseminadores de vírus, entre outros.

Moreira et al. (2008) destaca os principais tipos de ameaças que comprometem o correto funcionamento de programas que auxiliam no gerenciamento das organizações. Dentre elas estão:

a) Vírus: Programas desenvolvidos para destruir dados ou sistemas de computador;
b) *Worms*: Possui a capacidade de auto-replicação, exploram vulnerabilidades e não necessita da interação com usuário;

c) *Bots*: Propaga-se automaticamente explorando vulnerabilidades ou falhas na configuração de softwares, e pode ser controlado remotamente.
d) *Keyloggers*: Programas que registram as teclas digitadas pelo usuário, incluindo senhas, cartão de credito e números da conta;
e) *Screenlogger*: Programa capaz de armazenar a posição do cursor e a tela apresentadas no monitor.

Impedir que ameaças explorem pontos fracos e afete os princípios básicos da segurança (integridade, disponibilidade, confiabilidade) é um dos objetos da segurança da informação, a fim de não provocar danos a rede (MÓDULO, 2006).

1.3 Vulnerabilidades

Ferreira e Araujo (2006) definem vulnerabilidade como qualquer característica que um sistema possa apresentar que permita a usuários não autorizados assumir o controle sobre ele, ou o impeça de operar corretamente. Vulnerabilidades são resultados de falhas nas implementações de sistemas operacionais, serviços, aplicativos e protocolos, que podem ser exploradas por atacantes para executar ações maliciosas, como invadir um sistema, acessar informações confidencias, disparar ataques contra outros computadores ou tornar um serviço inacessível (NAKAMURA; GEUS, 2007).

Nakamura e Geus (2007) apresentam alguns dos protocolos que permitem que suas vulnerabilidades sejam exploradas, entre eles destacam-se, ICMP (*Internet Control Message Protocol*), UDP e TCP. Um exemplo de ataque que explora o ICMP é o *smurf*, que segundo Thomas (2007), desabilita um host ou a rede alvo e consome todos os seus recursos. O autor apresenta ainda o ataque ping-pong o qual envia uma mensagem UDP falsa para a porta de serviço encarregada a responder a qualquer pacote enviado. Outro ataque referido pelo autor é chamado de *SYN Flood*, que consiste basicamente em sobrecarregar um serviço com muitas requisições de endereços IP falsos.

Sêmola (2003) afirma que fragilidades associadas a ativos que manipulam ou processam informações ao serem exploradas, podem ocasionar incidentes de segurança. As vulnerabilidades de *hardware*, como falhas de recursos tecnológicos e de *softwares*, como a configuração ou instalação incorreta, são alguns dos exemplos que afetam os princípios de segurança da informação. A norma ISO/ISC 27005 apresenta uma lista

com exemplos de vulnerabilidades, as quais se destacam: vulnerabilidade de *hardware*, *software*, de rede, de pessoal, de instalações e da estrutura organizacional.

Rodrigues (2010) destaca algumas falhas que podem ocorrer deixando vulnerável um sistema de informação, que são:

a) *Bugs*: Erro no funcionamento de um *software*, normalmente devido a falhas de programação durante a fase de desenvolvimento, que podem ocasionar falhas na segurança da rede;

b) Não cumprimento das regras básicas de segurança: Utilizadores do sistema, pelas suas ações, podem causar diversas vulnerabilidades, tais como acesso a serviços duvidosos ou inseguros, senhas com fácil decifração, entre outros;

c) Desinteresse dos Administradores pela Segurança: Falta de políticas de segurança.

As ameaças e as vulnerabilidades podem mudar rapidamente, pois uma vulnerabilidade não causa prejuízo sozinha, necessita que haja uma ameaça capaz de explorá-la. A norma ISO-IEC 27005 aponta que o monitoramento constante torna-se necessário para que seja possível detectar essas mudanças. Somente após identificar as vulnerabilidades, será possível dimensionar os riscos ao qual o ambiente estará exposto e assim definir medidas de segurança apropriadas para sua correção (MÓDULO, 2006).

1.4 Ataques

Ataque é uma ação inteligente que ameaça a segurança através da violação da política de segurança de um sistema ou com intuito de invadir serviços de segurança (RFC 2828). Para Carissimi et al. (2009) ataque é denominado como qualquer ação que tenha como objetivo afetar a correta operação de um sistema de computação pela alteração de seu funcionamento, dos dados que possui, ou através de acesso indevido a recursos e aos próprios dados do sistema.

Melo e Gervilla (2010) dividem tentativas de ataque em três categorias: interno, externo e físico. Ataques internos são aqueles que ocorrem dentro de organizações, são praticados através do acesso indevido há dados de funcionários. Ataques externos são provenientes de fatores externos da rede, por meio da *internet* um usuário não

autorizado ou ilegítimo tem acesso ao sistema (RFC 2828). Ataques físicos ocorrem quando o invasor tem acesso físico às organizações, ou seja, são roubados equipamentos, software ou fitas magnéticas. Além do roubo é possível executar uma série de ações maliciosas ou destrutivas, tais como, copiar documentos confidencias, obter informações privilegiadas, modificar arquivos importantes ou aumentar privilégios de alguns usuários (NAKAMURA; GEUS, 2007).

Carissimi et al. (2009) classifica ainda ataque em dois grandes grupos, que são:

- Ataque Passivo: É aquele que deseja obter acesso indevido a uma informação, mas não afeta o funcionamento do sistema como um todo. Através da observação ou monitoramento das transmissões de dados, passa a ter acesso a informações não autorizadas como, por exemplo, senhas de acesso.
- Ataque Ativo: Altera ou prejudica o funcionamento normal do sistema, através da utilização de códigos executáveis capazes de corromper arquivos, controlar aplicações e instalar vírus.

Stallings (2005) apresenta dois tipos de ataques passivos, que são o vazamento de conteúdo de mensagens e a análise de tráfego, os quais são muito difíceis de detectar por não envolver qualquer alteração nos dados. Porém, é possível impedir o sucesso desses ataques utilizando criptografia. Os ataques ativos modificam o fluxo de dados ou criam um fluxo falso, e podem ser subdivididos em quatro categorias: de falsidade, de repetição, de modificação de mensagem e de negação de serviço. Para prevenir perfeitamente esses ataques seria necessário um monitoramento constante na proteção física de equipamentos e de linhas de comunicação, então o objetivo é detectar esses ataques e recuperar o prejuízo ou atraso causado por ele.

Nakamura e Geus (2007) explicam que Ataque de Negação de Serviços (*DoS - Denial-of-Service Attack*) são recursos explorados de forma agressiva impossibilitando o usuário legítimo de utilizá-lo. Nos ataques de negação de serviço (DoS), um computador é utilizado para tirar de operação um serviço ou computador conectado a *internet*, os métodos utilizados para isso podem ser: através de uma grande sobrecarga no processamento dos dados, um grande tráfego de dados para uma rede, ou a retirada de serviços importantes de um provedor, impossibilitando o acesso dos usuários (BAUER, 2006).

Segundo Kurose (2006), antes de atacar uma rede os invasores procuram saber os endereços IP das máquinas pertencentes à mesma, quais sistemas operacionais elas utilizam e os serviços que o sistema oferece. Conhecer o terreno e coletar informações

sobre o alvo a ser atacado, sem ser notado ou descoberto, é um passo indispensável para um ataque de sucesso. Técnicas e ferramentas podem ser utilizadas para obtenção de informações importantes do sistema entre elas, as quais destacam-se (NAKAMURA; GEUS, 2007):

a) *Dumpster:* é a verificação do lixo em busca de informações sobre a organização ou a rede da vítima;
b) *Packet sniffing:* é a captura de informações valiosas através do fluxo de pacotes na rede;
c) *Port scanning:* ferramenta utilizada para obtenção de informações referente aos serviços disponíveis, através do mapeamento das portas TCP e UDP;
d) *Scanning de vulnerabilidades:* realiza testes na rede a procura de falhas na segurança.

Nakamura e Geus (2007) complementam, ainda, que essas ferramentas citadas acima fazem parte também do arsenal de defesa usado para análises de segurança, que tem como objetivo identificar os pontos fracos e inseguros para executar correções e melhorias necessárias. Em outras palavras, a utilização das mesmas ferramentas/técnicas utilizadas pelos invasores servem de fortes aliadas para que os administradores consigam detectar falhas e manter um ambiente seguro em relação a intrusos.

1.5 Prevenções Contra Ataques

Prevenção é um conjunto de medidas que tem por objetivo reduzir a probabilidade de concretização das ameaças existentes, através de um conjunto de medidas que visa abranger os sistemas de informação com a capacidade de inspeção, detecção, reação e reflexão (SILVA et al., 2003).

Definir uma política de segurança é o ponto inicial para a segurança de uma rede. Nela é especificado desde o tipo de tráfego de dados, permitido através do *firewall* até os procedimentos de emergência a serem tomados em caso de algum incidente na segurança. O *firewall* é o primeiro meio de defesa capaz de impedir a exposição das informações aos ataques externos. Através da utilização do recurso de monitoramento é possível detectar eventuais sobreposições das barreiras, que pode ser automatizadas por

meio dos sistemas de detecção de intrusões (IDSs).

Somado aos dois recursos anteriores incluem-se também a criptografia, que proporciona um maior grau de confiabilidade das informações que trafegam na rede. Todo esse conjunto de ferramentas, quando bem empregadas, pode contribuir para formação de ambientes seguros (NAKAMURA; GEUS, 2007).

1.5.1 Políticas de Segurança

Ferreira e Araujo (2006) definem política de segurança como um conjunto de normas, métodos e procedimentos utilizado para a manutenção da segurança da informação, a fim de assegurar que as informações e os serviços importantes para organização recebam a proteção apropriada garantindo a sua confidencialidade, integridade e disponibilidade. A Figura 1 ilustra os fatores para garantir o sucesso da política de segurança.

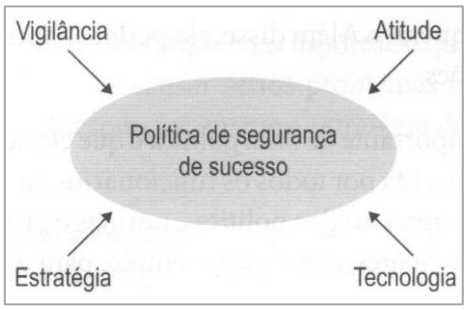


Figura 1- Fatores de Sucesso da Política de Segurança.
Fonte: Nakamura e Geus, 2007.

A política de segurança é um documento organizado com informações cada vez mais detalhadas sobre procedimentos, práticas e padrões a serem aplicados em determinadas circunstâncias, sistemas ou recursos. Sua implantação deve ocorrer formalmente, podendo ser composta por várias políticas inter-relacionadas, como a política de senhas, de *backup*, de contratação e instalação de equipamentos e *softwares*.

Esse processo passa por algumas etapas as quais se destacam: elaboração, aprovação, implementação, divulgação e manutenção (Tribunal de Contas da União, 2012).

A RFC 2196 (1997) cita alguns componentes fundamentais para uma boa política de segurança, dentre as quais destacam-se:

- Política de privacidade: define a privacidade relacionada a aspectos como monitoramento de e-mail, *logs* de atividades e acesso à arquivos de usuários;
- Política de acesso: direitos e privilégios para proteger a organização de danos, especificando a conduta do usuário, pessoal e gerente;
- Política de contabilidade: define responsabilidade de usuário;
- Política de autenticação: define uma política de senha efetiva e estabelece uma conduta para autenticação de acesso remoto e o uso de dispositivos de autenticação.

Segundo Thomas (2007), políticas de segurança permitem e capacitam a todos na empresa, definindo claramente a responsabilidade de cada funcionário, através de políticas e processos para cada departamento dentro de sua organização, determinando os recursos que precisam ser protegidos e as medidas que serão tomadas para proteger a rede. As políticas de segurança devem conter, de modo mais claro possível, as punições e os procedimentos adotados no caso do seu não cumprimento, com a finalidade de evitar abusos e para que usuários tenham consciência de que a política de segurança é importante para o sucesso da organização (NAKAMURA; GEUS, 2007).

1.5.2 Sistema de Detecção de Intrusão (IDS)

Sistema de detecção de intrusão (*Intrusion Detection System* - IDS) tem como objetivo detectar atividades suspeitas, impróprias, incorretas ou anômalas na rede, como por exemplo, ataques realizados por meio de portas legítimas permitidas, que não podem ser protegidos pelo *firewall* (NAKAMURA; GEUS, 2007). Segundo Pinheiro (2007) IDS são sistemas automáticos que funcionam em tempo real, analisando o tráfego de rede e detectando tentativas não autorizadas de acesso a infra-estrutura lógica, baseando-se em ataques conhecidos e verificando alterações de comportamento

do tráfego de dados, capaz de fornecer diferentes alertas e relatórios que podem ser analisados para políticas e planejamento de segurança.

A Figura 2 demonstra que sistema de detecção de intrusão trabalha de forma integrada, capaz de detectar, analisar e responder a atividades suspeitas.

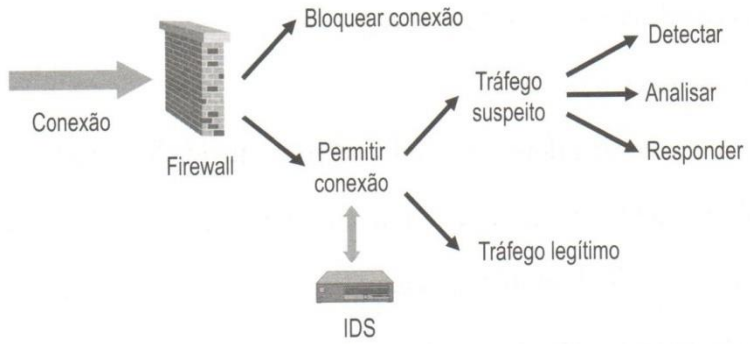


Figura 2- Funções de um IDS.
Fonte: Nakamura Geus, 2007.

Um sistema de detecção de intrusão exige uma quantidade significativa de recursos, por esse motivo os IDSs têm de ser instalados em locais estratégicos, configurados adequadamente e monitorados, pois poderão lidar com uma quantidade surpreendente de tráfego irregular da rede, tornando-se mais úteis quando colocados próximos de ativos importantes (CHESWICK et al., 2005).

Pinheiro (2007) apresenta duas formas básicas de IDS disponíveis que são: baseados em host (HIDS) e baseados em rede (NIDS).

- *Host Intrusion Detection System* (HIDS): um sistema capaz de monitorar, detectar e responder a atividades de usuário, sistema e ataques em um determinado *host* (THOMAS, 2005);
- *Network Intrusion Detection System* (NIDS): um sistema que monitora o tráfego do segmento de rede com a interface de rede, atuando em modo promíscuo, a detecção é realizada através da captura e análise dos cabeçalhos e conteúdo dos pacotes (NAKAMURA; GEUS, 2007).

Os NIDS podem ser divididos em duas partes, sendo elas: os sensores que são colocados em pontos estratégicos da infra-estrutura e analisam todo o tráfego do segmento de rede que estão inseridos, e o gerenciamento que é responsável pela

administração dos sensores e definindo os tipos de respostas para cada evento suspeito (PINHEIRO, 2007).

Nakamura e Geus (2007) afirmam que ao detectar tentativas de ataques externos ou internos, dependendo da localização dos IDS, permite que o administrador de segurança tenha conhecimento do que esta acontecendo na rede e possa tomar medidas para solucionar o problema, sempre de acordo com a política de segurança da empresa.

1.5.3 *Firewall*

O *firewall* controla todo o tráfego de dados através da verificação das informações que entram e saem da rede a fim de garantir que não ocorram acessos não autorizados (PINHEIRO, 2007). Segundo Carissimi et al. (2009) *firewall* é um processo que aplica uma série de regras de filtragem em um tráfego na *internet*, as quais se baseiam na política de segurança da organização, rejeitando pedidos e respostas, vinculados a serviços não autorizados.

Segundo Cheswick et al. (2005) *firewalls* podem filtrar vários níveis diferentes em uma pilha de protocolos de redes, sendo três categorias principais: *filtragem* de pacotes, *gateways* de circuito e *gateways* de aplicação, cada um deles é caracterizado pelo nível de protocolo que controla.

Filtragem de pacotes é um procedimento que examina individualmente os datagramas IP, tomando decisões com base nos campos de endereço de origem e de destino, tipo de protocolo e interpretando seu conteúdo (CARISSIMI et al., 2009). O filtro de pacotes trabalha na camada de rede e de transporte, isso faz com que ele seja simples, flexível e transparente ao usuário, garantindo assim seu melhor desempenho (NAKAMURA; GEUS, 2007).

Gateways de circuito trabalham no nível TCP, verificando todos os pedidos de conexão TCP e os segmentos transmitidos, e tem como tarefa analisar o fluxo de *bytes* recebidos, e se estiverem em conformidade com a política implementada, direcioná-lo para a saída. Um *firewall* desta categoria pode controlar o estabelecimento de conexões TCP para evitar ataques de negação de serviço (DoS) e inspecionar segmentos mal formados que visam explorar vulnerabilidades de sistemas operacionais (CARISSIMI et al., 2009).

Gateways de aplicação também conhecido com *proxy* de aplicação, funcionam de forma ativa sobre a conexão TCP e no protocolo de aplicação, onde são criadas duas conexões TCP uma entre o cliente e o *firewall* e outra entre o *firewall* e o servidor, assim o *gateway* de aplicação recebe e analisa as PDUs (*Protocol Data Unit*) do protocolo, aplicando regras de filtragem, verificando sua validade, sua correta formação, e verificando o cumprimento da política estabelecida (CARISSIMI et al., 2009).

Nakamura e Geus (2007) apresentam as principais tecnologias de *firewalls* e suas variações, que são:

- *Proxy*: registra todo o tráfego, seja ele com origem interna ou externa, ativando um sistema de alarme quando um tráfego não apropriado estiver em andamento;
- *Firewall* pessoal: atua diretamente no próprio equipamento do usuário, é capaz de controlar o acesso aos recursos, bloquear determinada conexão, monitorar todo o tráfego gerado e criar *logs* de todos os acessos do sistema;
- *Firewall* reativo: incluem funções de detecção de intrusão e alarmes, pode policiar acessos e serviços, mudar configurações de suas regras de filtragem de modo dinâmico, enviar mensagens aos usuários e ativar alarmes;
- Filtro de pacotes dinâmico baseado em estados: verifica somente o primeiro pacote de cada conexão, de acordo com as regras de filtragem, se o pacote for aceitos os demais são filtrados utilizando-se as informações da tabela de estados;
- *Firewalls* híbridos: misturam os elementos de três tecnologias, garantindo a proteção dos *proxies*, a segurança do filtro de pacotes e o desempenho do filtro de pacotes com base em estados.

A escolha do *firewall* que será instalado depende das exigências exatas de proteção e gerenciamento, assim como o tamanho da rede e do que será protegido pelo *firewall* (THOMAS, 2007).

1.5.4 Criptografia

Ferreira e Araujo (2006) definem criptografia com um conjunto de técnicas que permitem embaralhar as informações transmitidas entre hosts, de modo a impedir que o conteúdo dessas informações seja lido no meio do caminho, usada para autenticar a

identidade de usuário e manter o sigilo das comunicações pessoais e comerciais. Um algoritmo criptográfico é considerado eficiente se não existem facilidades que permitam a recuperação das informações, se o número de chaves possíveis for suficientemente grande, permite que ataques de força bruta se tornem impraticáveis (DUMONT, 2006).

O algoritmo de chave privada ou simétrica tem como característica a rapidez na execução, mas possui problema de distribuição e gerenciamento de chaves, onde é necessário para cada tipo de comunicação e para cada mensagem o uso de chaves secretas diferentes. Os algoritmos de chave pública ou assimétrica podem possibilitar, além do sigilo, integridade, não-repúdio e autenticidade, porém possui o problema de desempenho: a comunicação ocorre por meio de dois pares de chaves diferentes, uma privada e uma pública para cada entidade, o que exige maior poder de processamento. Desta forma os dois tipos de algoritmos geralmente são utilizados em conjunto explorando as melhores características de cada um (NAKAMURA; GEUS, 2007).

Dumont (2006) explica que a criptografia possibilita a criação de túneis criptográficos que podem ser usados para tornar a comunicação em rede mais segura, protocolos seguros como SSH e SSL possibilitam a encriptação, autenticação e integridade dos dados, o qual permite que vários serviços possam ser tunelados, e assim proteger a informação mesmo quando transferida através de redes públicas inseguras.

Para Tanenbaum (2003) algoritmos criptográficos usam transformações complexas que envolvem substituições e combinações para transformar o texto simples em texto cifrado. Os algoritmos criptográficos podem ser divididos em algoritmo de chave simétrica e chave pública, os quais se destacam o DES e RSA, e para documentos que precisam de assinaturas digitais são empregados algoritmos como MD5 ou SHA-1. O gerenciamento de chaves públicas pode ser implementado com o emprego de certificados, sendo essas ferramentas usadas para proteger o tráfego da rede.

2 TRABALHOS RELACIONADOS

A aplicação de técnicas e ferramentas utilizadas em testes de segurança é um campo cada vez mais explorado por estudiosos na área de segurança de redes de computadores. Na literatura há diversas propostas que visam identificar e auxiliar a correção das vulnerabilidades encontradas em ambientes de redes de computadores.

A análise de vulnerabilidade é de extrema importância para as organizações, pois apresentam as ameaças que as rodeiam diariamente (MOREIRA et al., 2008). As falhas de segurança têm origens distintas e podem ser ou não intencionais, sendo que sempre que acontece uma falha na rede, existe um sério compromisso com a integridade das informações envolvidas (SANTOS; SILVA, 2012).

De acordo com Martinelo e Bellezi (2014), manter o ambiente computacional das empresas atualizado e protegido das vulnerabilidades é um trabalho minucioso. A manutenção da qualidade da rede é auxiliada por *softwares* de análise de vulnerabilidades, que automatizam e facilitam o rastreamento dessas vulnerabilidades.

Segundo Wanner (2003), as ferramentas para varredura de vulnerabilidades foram desenvolvidas para automatizar a busca por falhas em serviços e em estações de trabalho existente em uma rede de computadores, elas verificam se nos dispositivos há vulnerabilidades exploráveis. As vulnerabilidades podem ser várias, como por exemplo: os servidores que permitem o acesso de pessoas não autorizadas, as aplicações desatualizadas presentes na rede, os privilégios dados aos utilizadores que podem ser alterados sem autorização, as palavras-passe dos computadores que podem não ser suficientemente fortes, entre outras muitas vulnerabilidades (SANTOS; SILVA, 2012).

As ferramentas fazem uma verificação da rede e geram relatórios sobre as vulnerabilidades de um computador ou uma rede. Na base de dados das ferramentas constam informações sobre ataques, vulnerabilidades, falhas e atualizações (MELO; GERVILLA, 2010). Os *softwares* fornecem as informações necessárias que tornam viáveis à utilização de ferramentas de defesa como *firewalls*, ou removem vulnerabilidades por falta de atualização de *softwares* ou ainda por motivos desconhecidos (CORSO, 2009).

3 TRABALHO PROPOSTO

Este Capítulo apresenta na Seção 3.1 o conceito de *scanners* de vulnerabilidades, que são *softwares* utilizados no suporte à segurança das redes, juntamente com a proposta de desenvolvimento deste estudo. Para o completo entendimento do propósito apresentado neste trabalho se faz necessário a explanação dos três *softwares* selecionados para o estudo, bem como, suas principais características e funcionalidades. Através das Seções 3.2, 3.3, 3.4 é possível entender como são estruturados os *softwares* escolhidos.

3.1 *Scanners* de Vulnerabilidades

As atuais medidas de segurança não garantem total eficácia contra todos os possíveis ataques. O processo de avaliação das vulnerabilidades do sistema e das potenciais ameaças é componente essencial de qualquer programa de gerência de segurança da informação.

Há necessidade de se empregar práticas para evitar eventuais falhas nos componentes de redes, sistemas e comportamento pessoal, que venham a comprometer o ambiente de redes. Entre as práticas de segurança já citadas, administradores de redes têm utilizado os testes de penetração para avaliar a segurança de sua rede.

Os testes de penetração ou *pentest* como também é chamado, consiste em uma série de testes realizados na rede em busca de falhas que intrusos possam explorar. É considerado eficiente quando consegue desvendar falhas em um sistema antes que intrusos mal intencionados as obtenham (JUNIOR, 2010).

O teste para o descobrimento de vulnerabilidades pode ser classificado como caixa preta (*Black Box*), caixa branca (*White Box*) ou ainda caixa cinza (*Gray Box*). O teste de caixa preta define que não existe qualquer conhecimento prévio da infra-estrutura a ser testada. E o teste de caixa branca assume que o testador possui total conhecimento da infra-estrutura a ser testada, incluindo diagrama da rede, endereço IP e qualquer informação complementar (COUTINHO, 2011). O teste caixa cinza é a

mesclagem dos dois tipos de testes citados a cima, o testador tem conhecimento parcial da infra-estrutura ou informações parciais relativas à rede (MORENO, 2014).

De acordo com Junior (2010) na realização de teste de penetração são definidas algumas etapas:

1. Levantamento de informações;
2. Identificação de vulnerabilidades;
3. Verificação e exploração das vulnerabilidades;
4. Obtenção de acesso privilegiado;
5. Manter o acesso;
6. Limpar as evidências.

De acordo com Ramos (2013), a ordem em que as fases são realizadas é importante, porque o resultado de uma etapa vai ser utilizado na etapa seguinte. A Figura 3 ilustra a metodologia a ser seguida por um teste de penetração.

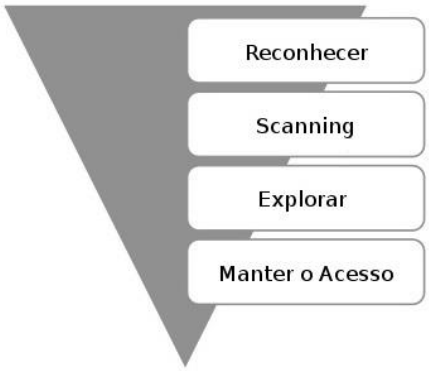


Figura 3- Metodologia de um Teste de Penetração.
Fonte: Ramos, 2013.

Para a realização desse trabalho será abordada a etapa de identificação de vulnerabilidades, a qual verifica e analisa as vulnerabilidades que existem em um determinado sistema, que podem vir a causar potencias problemas de segurança. Entre outras técnicas, utilizam *scanners* detectores de vulnerabilidades para descobertas de informações mais detalhadas da rede.

Os *Scanners* de vulnerabilidade têm como objetivo procurar por falhas em serviços, aplicativos, sistemas operacionais que representam um risco à segurança de uma rede quando utilizado por pessoas não autorizadas.

De acordo com Wanner (2003), um *scanner* de vulnerabilidade tem que apresentar os seguintes requisitos:

- Um banco de dados de vulnerabilidades sempre atualizado;
- Um relatório de vulnerabilidade preciso, evitando falsos positivos ou falsos negativos;
- Fornecer informações relevantes dos problemas encontrados e das maneiras de corrigi-los.

Através de uma lista de falhas conhecidas, o *scanner* de vulnerabilidade verifica se o sistema está ou não executando um serviço com problemas. O *scanner* é capaz de detectar erros comuns de configuração, configuração e senhas-padrão como, por exemplo, *softwares* com configuração de fábrica, combinações obvias de usuários e senha e vulnerabilidades divulgadas (ULBRICH; DELLA VALE, 2004).

Existem várias aplicações que executam a análise de vulnerabilidades de uma rede. Neste trabalho, a abordagem dos testes que serão realizados será do tipo caixa cinza, em redes já conhecidas, mas sem o conhecimento de sua infra-estrutura e topologia. A seguir são detalhadas as ferramentas utilizadas nesse trabalho: LanGuard, Nessus e o OpenVAS.

3.2 LanGuard

De acordo com Santos e Silva (2012), LanGuard foi criado em 2000 pela empresa GFI Software, é um *scanner* de segurança que oferece solução de gerenciamento de *patches* (atualização de segurança) e auditoria de rede, tem como objetivo assegurar o bom funcionamento de uma rede local. Possui versão disponível apenas para *Microsoft Windows*.

O LanGuard oferece o gerenciamento de *patches* para a *Microsoft*, *Mac OS X*, *Linux*, para os sistemas operacionais e aplicativos, possui a capacidade de detectar vulnerabilidades da rede antes de serem expostas. Realiza a avaliação de vulnerabilidades a fim de descobrir ameaças de segurança, para isso, são realizadas mais de 50.000 avaliações de vulnerabilidades através de sua rede, incluindo ambientes virtuais, móveis e dispositivos de rede como, por exemplo, *switches*, roteadores, pontos de acessos e impressoras.

Através de auditoria de rede é possível analisar o estado de segurança da rede, identificar os riscos e o grau de exposição. Oferece uma visão completa de aplicativos instalados, *hardwares* em sua rede, dispositivos móveis e o estado de aplicações de segurança, como, antivírus, *anti-spam*, *firewall*, entre outros, e que podem representar riscos à segurança da rede (GFI LANGUARD, 2014). A Figura 4 demonstra o funcionamento do LanGuard.

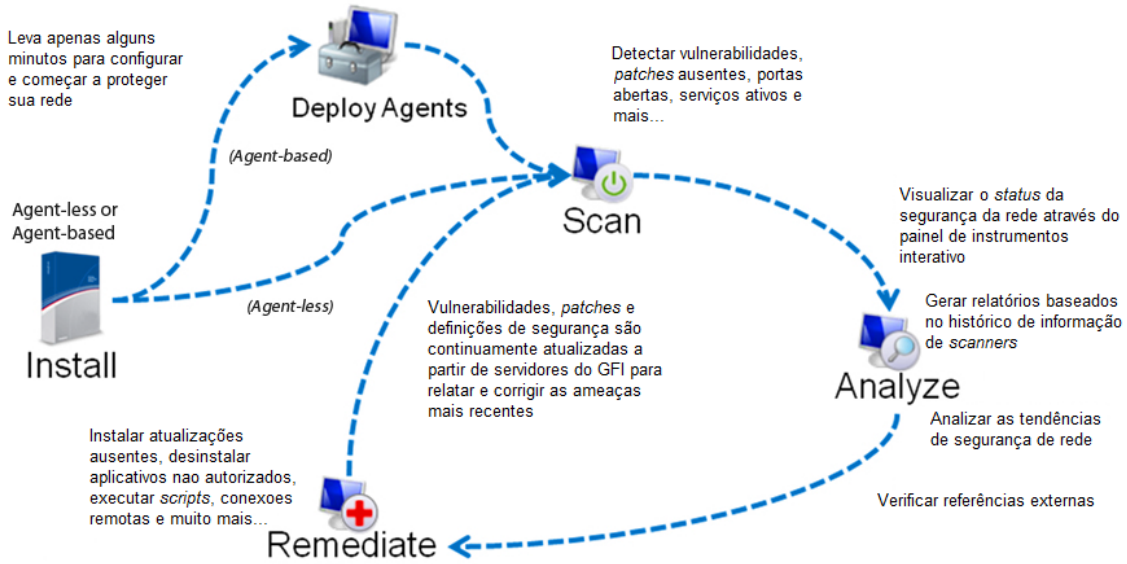


Figura 4- Funcionamento do LanGuard.
Fonte: Guia de instalação e configuração LanGuard, 2014.

Segundo o Guia de instalação e configuração do LanGuard (2014), é obtida a proteção da rede local através dos seguintes passos:

- Identificação de pontos fracos no sistema e da rede através de um banco de dados abrangente de vulnerabilidades;
- Auditorias de todos os ativos de *hardware* e *software* da rede, permitindo criar inventários detalhados dos ativos;
- *Download* automático e instalação remota de serviços *pack* (pacote de atualização de segurança) e *patches* para sistemas operacionais;
- Desinstalação automática de *software* não autorizado.

O LanGuard é capaz de gerar automaticamente texto e relatório gráficos baseados em informações obtidas a partir de verificações de segurança de rede e exportar os

resultados em XML, HTML e PDF. Os principais tipos de relatórios são: Relatórios gerais que fornecem relatórios técnicos detalhados, e relatórios de conformidade legal que fornece informações sobre o sistema e auditoria de rede cumprindo as normas, leis e regulamento da rede.

Na realização desse estudo serão gerados relatórios gerais baseados no *status* das vulnerabilidades, o qual destaca as informações estatísticas relacionadas às vulnerabilidades detectas nos computadores de destino, podendo ser agrupadas da seguinte forma: nome dispositivo, gravidade da vulnerabilidade, data/hora, categoria.

3.3 Nessus

De acordo com Santos e Silva (2012) Nessus foi lançado em 1998, sob licença GPL (*General Public Licence*) / GNU. Até a versão 2.2 era uma alternativa *open-source*, a partir da versão 3.0 a *Tenable Network Security* decidiu fechar o código-fonte, pois outras empresas passaram a incorporar recursos do Nessus em seus produtos proprietários e desenvolver versões modificadas da solução oferecida por eles, porém existe ainda uma versão para uso sem fins comerciais.

O *Nessus Vulnerability Scanner* é uma excelente ferramenta usada para identificar vulnerabilidades e falhas na rede local. Realiza uma varredura de portas e detecta servidores ativos, não somente nas portas padrões, mas em todas as portas, simulando invasões para detectar vulnerabilidades (MELO; GERVILLA, 2010). Funciona em diversas plataformas como, *Windows*, *FreeBSD*, *Linux* e *Mac OS*, composto por uma arquitetura cliente/servidor, sendo o servidor responsável pela varredura da rede e detecção de falhas de segurança, e o cliente prover a interface ao usuário, permitindo o mesmo analisar as vulnerabilidades.

O Nessus permite realizar auditorias remotas e determinar se a rede foi invadida ou usada de maneira indevida. Verifica a presença de vulnerabilidades, descobre dados sensíveis, realiza o gerenciamento de *patches* e análise de vulnerabilidades. O Nessus é capaz de descobrir uma ampla variedade de dispositivos físicos e virtuais em sua rede identificando sistemas operacionais, aplicações, banco de dados e serviços em execução nos dispositivos ativo da rede.

O Nessus é baseado em *plugins*, que são programas menores que provém funcionalidades especificas capazes de verificar a existência de uma determinada vulnerabilidade. Os *plugins* são escritos na linguagem *Nessus Attack Scripting Language* (NASL), os *plugins* contém informações de vulnerabilidades e um conjunto genérico de ações e correções para testar a presença do problema na rede (TENABLE NETWORK SECURITY, 2014).

Tenable oferece quatro códigos de ativação sendo um deles sem custo, que são:

- Nessus *Home*: Permite escanear sua rede pessoal com mesmo desempenho que um assinante Nessus, destinado a usuários domésticos com capacidade de escanear até 16 IPs;
- Nessus: capaz de escanear ilimitados IPs, possui um campo maior de verificações de segurança de rede;
- Nessus *Enterprise*: Fornece uma solução para facilitar a manutenção da segurança do ambiente de modo colaborativo entre as equipes de segurança, auditoria, proprietários dos sistemas e administradores de rede;
- Nessus *Enterprise Cloud*: Possui as mesmas características do Nessus Enterprise, porém, agrega a funcionalidade de gerenciamento e armazenamento de informações na nuvem. Dessa forma resultados, políticas e programações detalhadas podem ser acessadas e compartilhadas a qualquer momento e qualquer local pelos usuários.

O Nessus oferece uma descrição detalhada da(s) vulnerabilidade(s) detectada(s) e o(s) passos a ser seguido para eliminá-la(s), através de relatórios em HTML, XML, LaTeX e texto simples. A exibição desses relatórios pode ser feita de diferentes maneiras, ou seja: remedições sugeridas – Nessus resume as ações a serem tomadas para o endereço com a maior quantia de vulnerabilidades na rede; vulnerabilidades agrupadas por *plugin* – lista cada vulnerabilidade encontrada durante o escaneamento dos *hosts* afetados; vulnerabilidades agrupadas por *host* – lista cada *host* encontrado durante o escaneamento e suas vulnerabilidades associadas.

3.4 OpenVAS

O OpenVAS é um sistema para avaliação de vulnerabilidades de código aberto, distribuído sob a licença GPL *General Public license* (*GNU*), e é um *fork* [1] livre do buscador de vulnerabilidade Nessus . O propósito inicial do projeto era permitir o livre desenvolvimento do agora proprietário *Nessus Security Scanner*. Atualmente o OpenVAS possui uma comunidade crescente e conta com a contribuição de indivíduos de todo o mundo que auxiliam nas melhorias e na documentação do *software*, e possui versões disponíveis para as distribuições do *Linux*: *Debian*, *Fedora*, *OpenSuse*, *RedHat*, além de versão para *Windows*.

Segundo Schwarzer (2011) OpenVAS oferece um ambiente completo de avaliação de segurança, com uma série de serviços e componentes que podem ser organizados em diversas formas para construir um ambiente de avaliação adequado a rede. OpenVAS assim com o Nessus segue o modelo cliente/servidor, o servidor é o componente central que contém as funcionalidades utilizadas para execução de um grande número de testes, é responsável pelo agendamento e pela execução de buscas. O cliente é constituído de uma interface gráfica onde é possível configurar as atividades, como buscas e acessar os resultados.

O *scanner* de segurança é acompanhado de uma alimentação diária atualizada de testes de vulnerabilidades de rede que são chamados de *Network Vulnerability Test* (NVTs). Os NVTs são rotinas de testes que verificam a presença de uma vulnerabilidade em um sistema, desenvolvidos na linguagem de *script* do Nessus (NASL). De acordo com Brown e Galitz (2010) assim como o Nessus o OpenVAS permite a criação de seus próprios *plugins* (NVTs) para verificação de segurança, podem ser usados para necessidades especificas como, por exemplo, testar vulnerabilidades em sistemas próprios que são alterados com frequência.

A Figura 5 ilustra a arquitetura do OpenVAS que é composta por uma estrutura de vários serviços e ferramentas.

[1] fork é uma derivação com base em um *software* ou S.O. Acontece quando um desenvolvedor inicia um projeto independente com base no código-fonte de um projeto já existente.

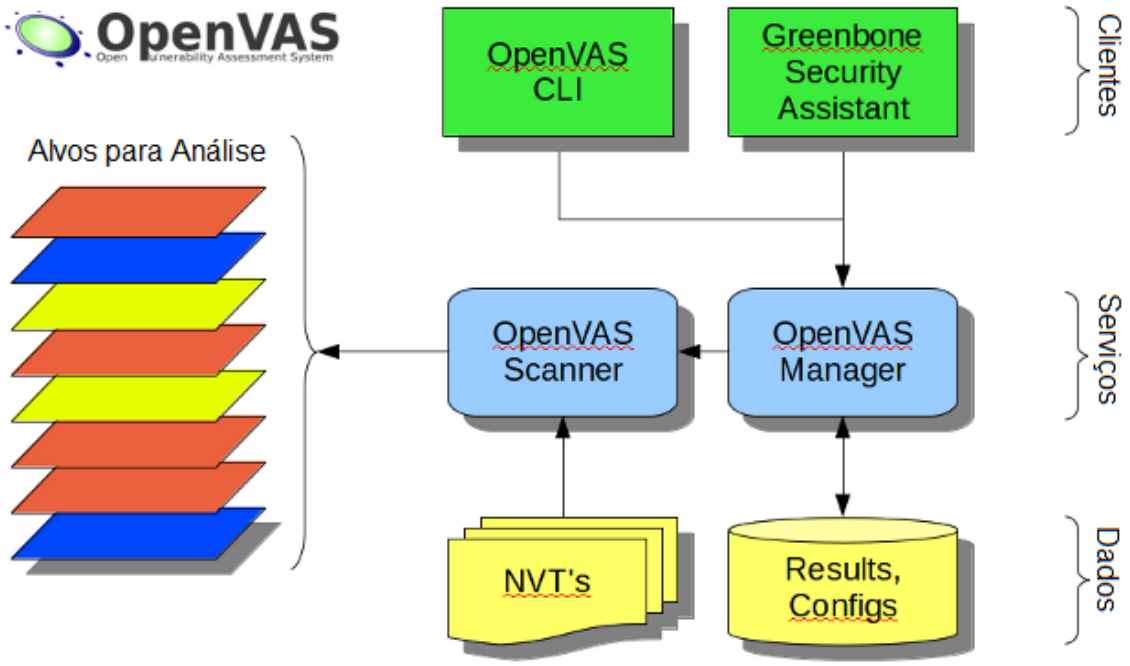


Figura 5- Arquitetura OpenVAS.
Fonte: openvas.org, 2014.

O OpenVAS Manager é o serviço central para o controle do ambiente, estabelece e gerencia as varreduras de vulnerabilidades, utiliza os protocolos *OpenVAS Management Protocol* (OMP) que possibilita o cliente se conectar ao manager e *OpenVAS Transfer Protocol* (OTP) que controla a execução da varredura, e armazena todos os dados de varreduras anteriores em sua base de dados interna, administrando clientes e novos serviços. O *Greenbone Security Assistant* é responsável pela interface *web* oferecida para o usuário, é possível rodar em qualquer navegador. O OpenVAS *Scanner* que executa nos alvos os testes NVTs. E por fim OpenVAS CLI contém a linha de comando da ferramenta OMP (OPENVAS, 2014).

O OpenVAS possui algumas ferramentas de segurança integradas, as quais incluem-se o nmap (*scanner* de portas), nikito (teste de servidor *web*), ike-scan (varreduras em servidores IPsec), entre outros. Após completar a varredura da rede OpenVAS oferece um relatório listando os detalhes com base nas portas, serviços encontrados em sua rede, ele destaca as vulnerabilidades com prioridades alta, moderada e baixa, sendo possível exportá-lo em vários formatos, incluindo HTML, XML e PDF.

4 TESTES E RESULTADOS

Após a apresentação e entendimento do funcionamento dos principais recursos dos programas listados anteriormente, foram selecionadas duas redes de computadores para a realização de testes práticos, com intuito de analisar o desempenho e os modelos de relatórios apresentados por cada uma. Os resultados relativos às vulnerabilidades de cada rede escaneada serão visualizados de maneiras diferentes em cada *software*.

4.1 Ambiente de teste

Para a realização dos testes foi escaneada a rede de servidores da UFSM (Universidade Federal de Santa Maria) e rede de servidores do CTISM (Colégio Técnico Industrial de Santa Maria), com IPs (*Internet Protocol*), 200.18.45.0/26 e 172.17.8.0/24 respectivamente. O Languard e o OpenVAS foram aplicados na rede de servidores do CTISM, enaquanto o Nessus e o OpenVAS tiveram seus recursos testados na rede de servidores da UFSM.

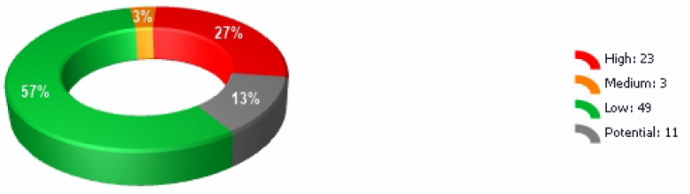
Esta etapa tem como objetivo apresentar as ferramentas na prática, seus resultados e suas vantagens na utilização. Serão utilizadas máquinas virtuais com os seguintes sistemas operacionais: *Windows*, *Ubuntu* e *Debian*, a fim de satisfazer as configurações necessárias para a instalação de cada *software*. Máquinas virtuais são imprescindíveis em ambientes de testes, uma vez que possibilitam a instalação e virtualização de vários sistemas operacionais e suas configurações necessárias para o correto funcionamento de programas específicos dentro de um sistema hospedeiro. Para esse trabalho foi escolhido o *software* VMware para realizar as virtualizações.

4.2 Teste com o LanGuard

A ferramenta LanGuard foi instalada em uma máquina virtual com sistema operacional Windows 7 Ultimate. A versão utilizada foi o LanGuard 2014 *version* 11.02, disponível para realização de teste por trinta dias. O ambiente é apresentado ao usuário através de um aplicativo disponível para *download* no endereço: http://www.gfi.com/downloads. O *software* não necessita de nenhum servidor *web* para seu funcionamento, uma vez instalado é preciso apenas utilizar o atalho na área de trabalho ou *menu* iniciar. Para esta ferramenta foi definido um intervalo de IPs para que o *scanner* pudesse identificar as máquinas na rede de servidores do CTISM.

O relatório do LanGuard, como já foi mencionado, apresentou não somente o modo texto mas também o recurso gráfico através de percentuais e total dentro de cada percentual das vulnerabilidades identificadas. A Figura 6 mostra ainda, o numero dos dispositivos encontrados e os resultados obtidos através do escaneamento da rede com seus respectivos nomes, vulnerabilidades detectadas e grau de risco das vulnerabilidades encontradas.

Vulnerability Distribution by Severity



Vulnerability Distribution by Computer

Computer/IP	High	Medium	Low	Potential
ANIMATI02	0	0	1	0
CTISMEAD01	0	0	2	1
DSPACE	0	0	2	0
EAD01	0	0	3	1
EAD03	0	0	2	0
EAD04	0	0	5	0
EAD05	0	0	2	0
EAD08	0	0	1	0
EADPOLI01	0	0	1	1
FIREWALL	0	1	2	0
IAS-UFSM	0	0	1	0
INFRA	0	0	3	0
IRRIGA	0	0	1	0
MULTI01	0	0	3	2
MULTISERVER01	0	0	1	0
MULTISERVER02	0	0	2	0
MULTISERVER05	0	0	3	1
MULTIWEB	0	0	3	0
RPA-DELL	23	1	6	4
SAMBASERVER	0	0	4	0
ST-UFSM-00	0	1	1	1

Figura 6- Resultados obtidos pelo LanGuard.

Para realizar a análise das vulnerabilidades serão destacados dois dispositivos com vulnerabilidades de médio risco (*Medium*), vulnerabilidades potenciais (*Potential*) e vulnerabilidades de baixo risco (*Low*). Nesta varredura podemos observar também que entre os dispositivos listados o que apresenta os maiores níveis de risco é a máquina em que o *software* de varredura estava executando, sendo detectadas vulnerabilidades de alto risco para a segurança do computador.

No dispositivo identificado como FIREWALL, foi detectada uma vulnerabilidade de risco médio, a qual foi destacada para esse estudo, pelo fato de que se não solucionada, pode ser explorada por algum indivíduo mal intencionado. E duas vulnerabilidades de baixo risco, totalizando três vulnerabilidades encontradas neste dispositivo. O relatório nos traz a informação que o serviço *Simple Network Management Protocol* (SNMP) está habilitado e ainda esclarece que várias vulnerabilidades foram relatadas em implantações SNMP de vários fornecedores. A Figura 7 ilustra o gráfico gerado pelo relatório do LanGuard.

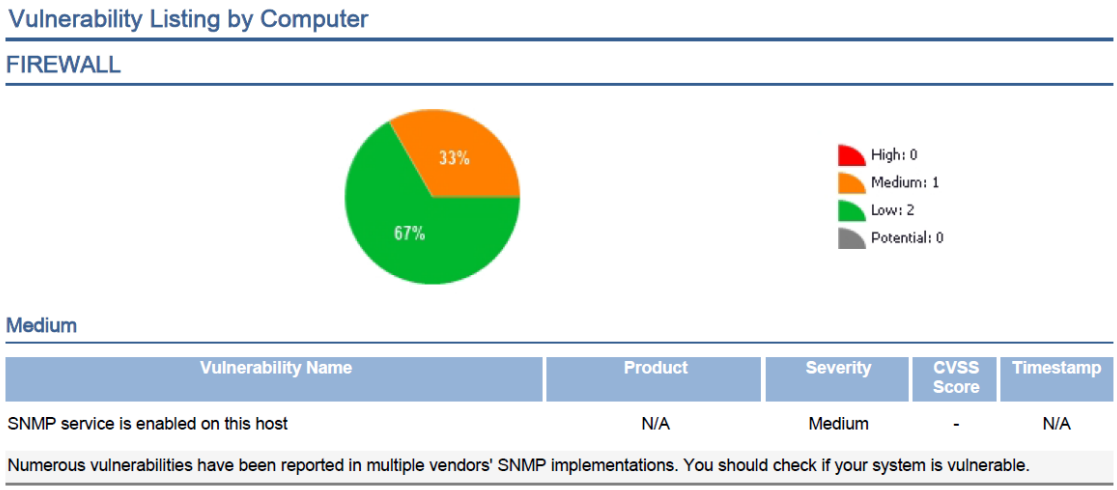


Figura 7- Vulnerabilidades do FIREWALL.

O SNMP é um protocolo de gerenciamento remoto largamente utilizado para gerenciamento e monitoramento de dispositivos de rede. Este protocolo possibilita modificar as configurações através do gerenciamento remoto do dispositivo de rede.

O protocolo SNMP tem uma arquitetura cliente-servidor. A comunicação entre o cliente e o servidor é realizada através de uma mensagem chamada *Protocol Data Unit* (PDU) que são unidades de dados. O dispositivo que responde às requisições é chamado de Agente (servidor) SNMP, as informações que ele fornece são organizadas em *Management Information Base* (MIBs), conforme o Gerente (cliente) faz os pedidos obtém uma grande quantidade de informações sobre o dispositivo.

Para que se faça possível a troca de informações o SNMP usa o protocolo de transporte UDP, e prevê uma autenticação, que é realizada através da *string de community* utilizada pelo Agente SNMP para identificar a que tipo de dados o Gerente SNMP terá acesso podendo ser diferente para leitura e escrita (CONTESSA; POLINA, 2014). SNMP *community string* são transmitidas em texto plano e há capacidade de se obter informações sobre o seu conteúdo, e pode ser utilizado para ganhar acesso a recursos baseados em SNMP.

O SNMP torna-se vulnerável, pois, muitas vezes é instalado automaticamente em muitos dispositivos de rede com configuração usando os nomes padrão da

comunidade, com "public" como a seqüência de leitura e "private" como a seqüência de gravação. Esta configuração padrão do SNMP pode permitir que atacantes obtenham uma grande quantidade de informações da rede interna, reconhecendo o sistema, sendo capaz de reconfigurar ou desligar remotamente os equipamentos, podendo causar assim uma negação de serviço (DoS). As MIBs fornecem informações como o nome do sistema, localização, contatos, e às vezes até mesmo números de telefone que pode ser muito útil em engenharia social. Um invasor pode ainda usar o sistema de contato para obter uma senha de um usuário desativado (SANS, 2014).

Para garantir a implementação segura do protocolo SNMP pode ser realizada a configuração simples deste protocolo utilizando a versão 3 (SNMPv3), a qual usa o conceito de grupos SNMP, os usuários SNMP possuem senhas de autenticação, utilizando-se de parâmetros de autenticação como MD5 ou SHA-1 e criptografia, que podem utilizar parâmetros como, por exemplo, DES (CABRAL et al., 2012).

Outra recomendação seria filtrar e controlar o acesso, aceitar pacotes SNMP apenas de *hosts* específicos, permitindo que somente IPs autorizados tenham acesso às requisições respondidas pelo Agente SNMP. Além da instalação de *patches* e atualização de versões (NAKAMURA; GEUS, 2007).

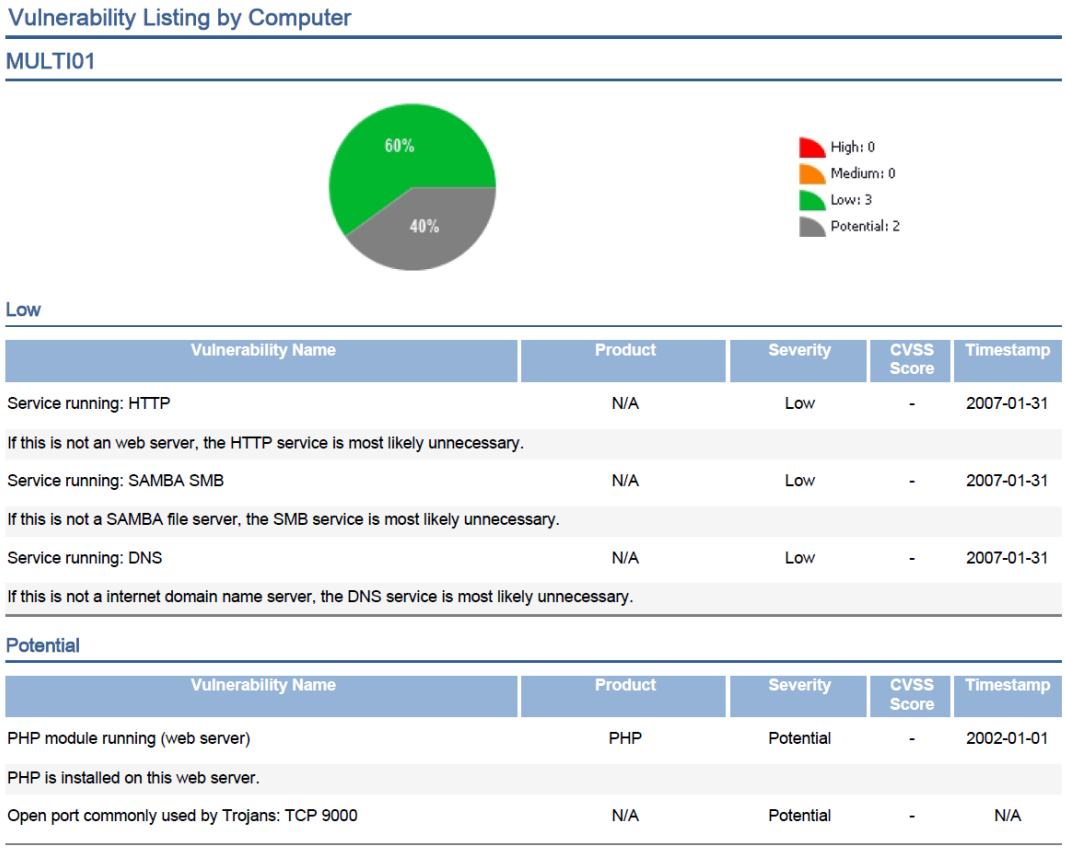
No dispositivo MULTI01 foi possível observar que 60% possuem vulnerabilidades de baixo risco (*Low*) e 40% de vulnerabilidades potenciais (*Potential*). As vulnerabilidades de baixo risco estão associadas a serviços que estão rodando no dispositivo, os quais são: Serviços de funcionamento HTTP e DNS, e serviço em execução SAMBA. Para os três serviços detectados, o *software* informou que se o dispositivo não possuir servidor *web*, servidor de nomes e servidor de arquivos SAMBA, não se faz necessário os serviços listados estarem ativos. Com o objetivo de evitar que esse dispositivo fique exposto a possíveis ataques, pode-se isolar o servidor *web* de outros serviços públicos, como o servidor DNS.

Vulnerabilidades potenciais encontradas podem se tornar um risco ao ambiente de rede se não solucionadas. As vulnerabilidades que LanGuard detectou e classificou como potenciais são:

- Módulo PHP em execução, serviço esse que deve ser instalado no servidor *web*, e se o mesmo não for um servidor *web* não se faz necessário;

- Porta TCP 9000 aberta, que pode ser explorada por *hackers* para obter acesso remoto e para utilização de *trojans*.

De acordo com Nakamura e Geus (2007), *trojans* são *softwares* legítimos que possuem códigos escondidos e executam atividades não previstas, quando utilizado executa funções ilegais, como, por exemplo, enviar mensagens e arquivos para *hackers* ou abrir portas de entrada para futuras invasões. A Figura 8 apresenta o gráfico com as vulnerabilidades encontradas nesse dispositivo.



Vulnerability Listing by Computer

MULTI01

Low

Vulnerability Name	Product	Severity	CVSS Score	Timestamp
Service running: HTTP	N/A	Low	-	2007-01-31
If this is not an web server, the HTTP service is most likely unnecessary.				
Service running: SAMBA SMB	N/A	Low	-	2007-01-31
If this is not a SAMBA file server, the SMB service is most likely unnecessary.				
Service running: DNS	N/A	Low	-	2007-01-31
If this is not a internet domain name server, the DNS service is most likely unnecessary.				

Potential

Vulnerability Name	Product	Severity	CVSS Score	Timestamp
PHP module running (web server)	PHP	Potential	-	2002-01-01
PHP is installed on this web server.				
Open port commonly used by Trojans: TCP 9000	N/A	Potential	-	N/A

Figura 8: Vulnerabilidades no MULTI01

Semelhante às vulnerabilidades encontradas nos dispositivos apresentados na Figura 7 e na Figura 8, as vulnerabilidades detectadas nos demais dispositivos não são diferentes. Vulnerabilidades de baixo risco baseiam-se em serviços como HTTP, SSH, DNS, PostgreSQL, SMTP, entre outros, o LanGuard informa que o se os serviços que estão rodando não estiverem em um servidor que ofereça esses serviços, não se fazem

necessários. Para maior segurança da rede o administrador pode verificar se esses serviços estão rodando em seus servidores específicos, se caso não estiverem, a sugestão da ferramenta é desabilitá-los, pois já apresentam um risco a rede.

As vulnerabilidades classificadas nos demais dispositivos como potenciais apresentam as mesmas vulnerabilidades encontradas no dispositivo exposto, o qual informa serviço PHP rodando na máquina e portas abertas que podem se tornar alvos de atacantes. As classificadas de risco médio apresentam vulnerabilidades no protocolo SNMP e as vulnerabilidades de alto risco como já citadas foram detectadas no *host* em que estava rodando a ferramenta.

4.3 Testes com Nessus

A ferramenta de escaneamento de redes Nessus oferece uma versão gratuita destinada a uso doméstico, a qual permite o usuário escanear sua rede pessoal. A varredura pode ser realizada em uma rede com até dezesseis endereços IP por *scanner*, com a mesma velocidade e avaliação profunda das vulnerabilidades encontradas na rede que um assinante Nessus desfruta. Para obter a liberação de acesso aos recursos que o Nessus oferece, é necessário registrar-se no *site* oficial da *Tenable Network Security* para assim receber um código de ativação e utilizá-lo.

O Nessus Home *Version* 5.2.5 foi utilizado para a realização dos testes de varredura de rede nesse estudo. Instalado em uma máquina virtual rodando o Sistema Operacional *Ubuntu* 12.04, foi escaneada a rede de servidores da UFSM, com a limitação de dezesseis IPs para cada rede. O Nessus oferece uma interface *web*, que após receber o código de ativação poderá ser acessado pelo endereço http://localhost:8834. As atualizações dos *plugins* do Nessus são automáticas, e são realizadas enquanto a ferramenta está em funcionamento.

O relatório do Nessus apresenta as vulnerabilidades encontradas pelos *plugins* em uma lista onde os *hosts* são organizados pelo maior nível de risco encontrado. Os níveis de riscos são classificados em: *Critical*, *High*, *Medium*, *Low* e *Info*. A *Info* não é um fator de risco, mas trás informações sobre os testes realizados no *host* específico. O relatório do Nessus ainda oferece informações detalhadas de cada *host*, informa a data e

hora de início e fim do *scan*, e é capaz de identifica o nome de domínio do servidor DNS, o sistema operacional que esta sendo executado.

Na rede de servidores da UFSM, mesmo com o número limitado de IPs escaneados, o Nessus conseguiu detectar dezesseis *hosts*, sendo assim, possível detectar vulnerabilidades de risco alto (*High*), médio (*Medium*) e baixo (*Low*).

A Figura 9 demonstra o resultado obtido no *host* de IP 200.18.45.1, o qual apresenta vulnerabilidade de alto risco, no servidor SNMP, devido ao fato de utilizar o a configuração da *Community public*.

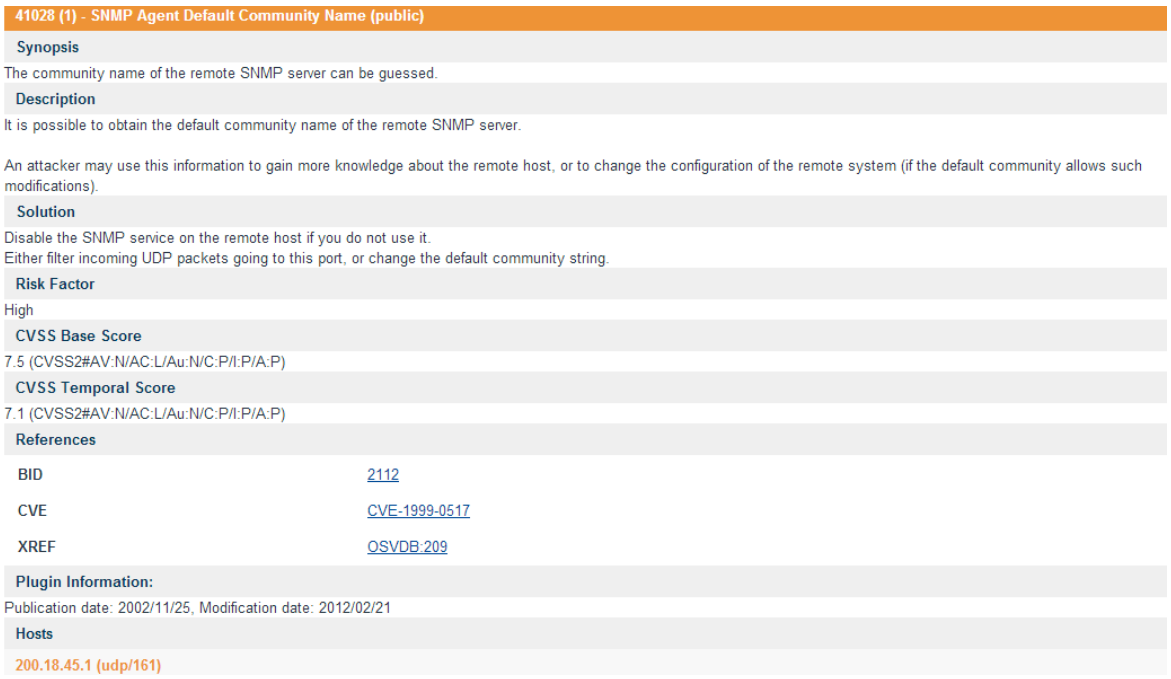


Figura 9- Nessus alto risco UFSM.

Como já mencionado anteriormente o SNMP é um protocolo utilizado para gerenciamento e monitoramento de dispositivos de rede, projetado para oferecer uma *interface* padrão para recuperar e configurar opções de configuração em *hosts* individuais, armazenando e recuperando estatísticas e enviando alertas em respostas a eventos.

O risco de manter o nome da comunidade padrão do servidor SNMP remoto possibilita a um invasor obter informações sobre o *host* para alterar as configurações do sistema. Como solução para a vulnerabilidade encontrada o Nessus sugere desativar o

servidor SNMP do *host* remoto, se o mesmo não estiver sendo utilizado, ou filtrar pacotes UDP de entrada indo para a porta UDP/161, que são utilizados para o transporte de informações, ou ainda, alterar a sequência de comunidade padrão.

A comunidade SNMP pode ser configurada com três nomes, que são: *read-only*, *read-write* e *trap*. A *string* de comunidade *read-only* permite ler valores de dados, a *string* de comunidade *read-write* permite ler e modificar valores de dados, e a *string* de comunidade *trap* permite receber notificações do agente (GOMES, 2002). Os nomes das comunidades são como senhas que possibilitam qualquer aplicativo baseado em SNMP, com a capacidade de reconhecer a *string* da comunidade, obter o acesso à informação de gerenciamento de um dispositivo.

A Figura 10 ilustra a vulnerabilidade de risco médio encontrada em oito *hosts* diferentes em três portas TCP: tcp/80, tcp/8080, tcp/3128, em que está rodando um servidor *proxy* HTTP.

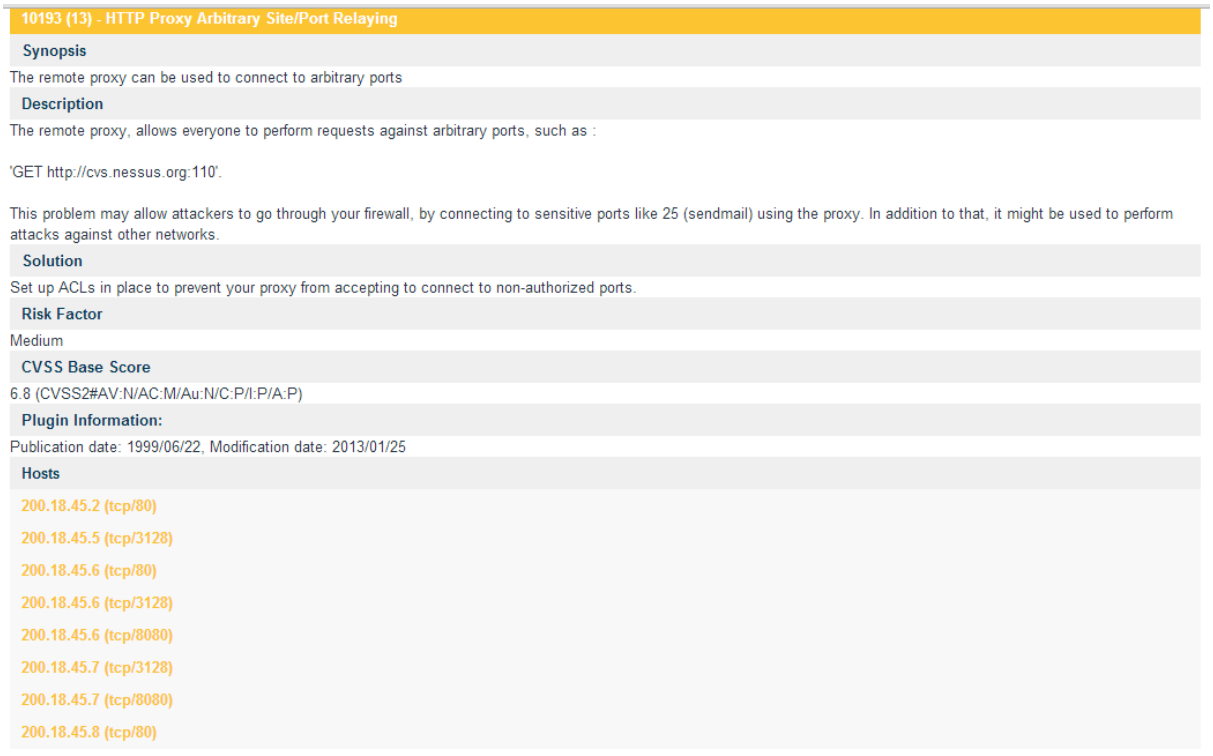


Figura 10- Nessus médio risco UFSM.

Os testes realizados pelo Nessus nestes dispositivos destacaram uma vulnerabilidade no *proxy* remoto. O *proxy* permite que usuários realizem pedidos, GET, permitindo um atacante através do *firewall* conectar-se a portas sensíveis, como por exemplo, a porta 25 utilizada pelo servidor de correio eletrônico, *sendmail*, ou pode ser utilizado para realizar ataques a outra redes.

A solução oferecida pelo Nessus seria a configuração de *Access Control List* (ACLs) no local para evitar conexões às portas não autorizadas. ACLs é a lista de controle de acesso que define quem tem permissão de acesso a certos serviços, de maneira que possibilita um servidor permitir ou negar determinada tarefa definido para cada usuário ou grupo.

Uma das vulnerabilidades de baixo risco que o relatório exibiu é encontrada em seis dispositivos diferentes, a Figura 11 ilustra a vulnerabilidade detectada.

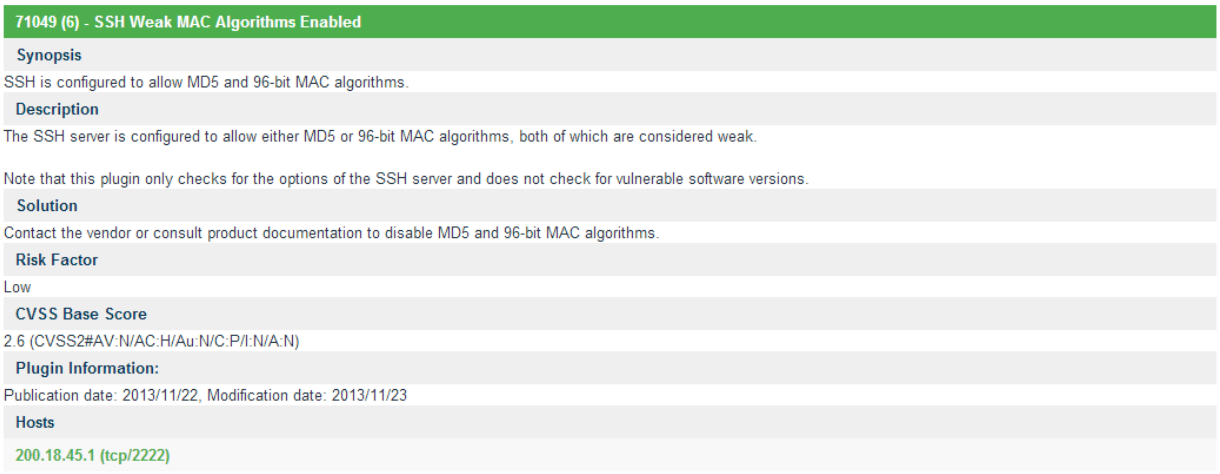


Figura 11- Nessus baixo risco UFSM.

A partir do relatório, o Nessus informou que o servidor SSH está configurado para permitir configuração de algoritmos MAC de 96 *bits* e MD5, sendo os dois considerados fracos. O algoritmo MD5 é um algoritmo de *hash* 128 *bits*, muito utilizados na verificação de integridade de arquivos e *logins*. Pode se tornar vulnerável se duas *strings* diferentes produzirem o mesmo *hash*. Para sanar essa vulnerabilidade Nessus sugere desativar o algoritmo MD5 e MAC de 96 *bits*.

4.4 Testes com OpenVAS

Para realização dos testes de varredura de vulnerabilidades, foi utilizado o *software* OpenVAS 5, instalado em uma máquina virtual com sistema operacional Debian 7.2.0. O programa pode ser instalado de três maneiras distintas: através de pacotes binários, ou seja, o usuário instala o sistema direto do repositório do fabricante; pode ser compilado o código-fonte através dos pacotes necessários; ou executar como *virtual appliance*, que possibilita a importação de uma imagem OVA para uma máquina virtual, *VirtualBox* por exemplo. Após instalado, o OpenVAS é acessado através de uma interface *web*, disponível em http://localhost:9392. Por se tratar de uma ferramenta gratuita, o número de *hosts* que foram detectados é maior do que no Nessus, pois para este trabalho a versão utilizada do Nessus é limitada a dezesseis computadores.

Com essa ferramenta realizou-se o escaneamento das redes de servidores da UFSM e de servidores do CTISM. O relatório gerado pelo OpenVAS informa a data de início e fim da varredura, descreve os resultados encontrados para cada *host* identificado e oferece recomendações a fim de corrigir os problemas detectados. Para obter maior desempenho da ferramenta na detecção de vulnerabilidades, sugere-se atualizar os *plugins* através do comando *openvas-nvt-sync*, inserido no terminal, que atualmente conta com mais de 35.000 NVTs.

Através da varredura da rede de servidores da UFSM foram detectados cinquenta e um *hosts*, como mostra a Figura 12.

1 Result Overview

Host	Most Severe Result(s)	High	Medium	Low	Log	False Positi
200.18.45.0	Severity: Log	0	0	0	5	0
200.18.45.1	Severity: High	1	0	0	24	0
200.18.45.2	Severity: Medium	0	2	0	33	0
200.18.45.3	Severity: Medium	0	1	0	27	0
200.18.45.4	Severity: Log	0	0	0	22	0
200.18.45.5	Severity: High	2	1	3	33	0
200.18.45.6	Severity: Medium	0	2	0	27	0
200.18.45.7	Severity: Medium	0	2	1	30	0
200.18.45.8	Severity: Medium	0	2	0	28	0
200.18.45.9	Severity: High	2	5	2	45	0
200.18.45.10 (itautecmoodle.proj.ufsm.br)	Severity: Log	0	0	0	16	0
200.18.45.11	Severity: Medium	0	2	0	30	0
200.18.45.12	Severity: Medium	0	2	0	30	0
200.18.45.13	Severity: Medium	0	2	4	30	0
200.18.45.14 (SERVERIPHONE)	Severity: Medium	0	4	0	52	0
200.18.45.15	Severity: Medium	0	1	0	20	0
200.18.45.16	Severity: High	26	7	1	39	0
200.18.45.17	Severity: Medium	0	1	0	21	0
200.18.45.18	Severity: Medium	0	2	1	29	0
200.18.45.19 (WIN-F22COA5GLCM)	Severity: Medium	0	3	3	30	0
200.18.45.20	Severity: Medium	0	2	1	34	0
200.18.45.21 (eadpoli01.proj.ufsm.br)	Severity: Medium	0	1	0	23	0
200.18.45.22	Severity: Medium	0	1	0	28	0
200.18.45.23 (statistica.proj.ufsm.br)	Severity: Medium	0	6	1	47	0
200.18.45.24	Severity: Medium	0	2	2	27	0
200.18.45.25	Severity: Log	0	0	0	5	0
200.18.45.26	Severity: Log	0	0	0	5	0
200.18.45.27 (tupi.proj.ufsm.br)	Severity: Log	0	0	0	17	0
200.18.45.28 (multiserver06.proj.ufsm.br)	Severity: High	2	1	1	70	0
200.18.45.29	Severity: Medium	0	1	0	23	0
200.18.45.40	Severity: Medium	0	1	0	11	0
200.18.45.41	Severity: Log	0	0	0	5	0
200.18.45.42	Severity: Log	0	0	0	5	0
200.18.45.44	Severity: Medium	0	1	0	21	0
200.18.45.46	Severity: Log	0	0	0	21	0
200.18.45.47	Severity: Medium	0	1	0	16	0
200.18.45.48	Severity: Medium	0	1	0	16	0
200.18.45.49	Severity: Medium	0	1	0	14	0
200.18.45.50	Severity: Medium	0	1	0	21	0
200.18.45.51	Severity: Medium	0	1	0	8	0
200.18.45.52	Severity: Medium	0	1	0	21	0
200.18.45.53	Severity: Medium	0	1	0	12	0
200.18.45.54	Severity: Medium	0	2	0	7	0
Host	Most Severe Result(s)	High	Medium	Low	Log	False Positiv
200.18.45.55	Severity: Medium	0	1	0	17	0
200.18.45.56	Severity: Medium	0	1	0	12	0
200.18.45.57	Severity: Log	0	0	0	4	0
200.18.45.58	Severity: Log	0	0	0	16	0
200.18.45.59	Severity: Medium	0	1	0	21	0
200.18.45.60	Severity: Log	0	0	0	21	0
200.18.45.61	Severity: Medium	0	1	0	17	0
200.18.45.62	Severity: Medium	0	1	0	21	0
Total: 51		33	69	20	1157	0

Figura 12- Relatório UFSM OpenVAS.

O relatório do OpenVAS apresenta o risco de ataques em três níveis *High*, *Medium*, *Low*. Outro campo apresentado no relatório é o *Log*, que contém a informação que é retornada pelo *plugin*, que apresenta informações mais detalhadas sobre os eventos encontrados durante o escaneamento. Para demonstrar os resultados encontrados com a utilização dessa varredura será apresentado um exemplo de cada nível de risco.

Entre os *hosts* detectados foi escolhido o que possui IP 200.18.45.16 que apresentou o maior número de vulnerabilidade de risco *High*. As vulnerabilidades encontradas pelo OpenVAS neste *host*, foram no serviço Moodle e na versão PHP que o servidor utiliza, e que deve ser atualizada. A Figura 13 apresenta o detalhamento de uma vulnerabilidade de alto risco encontrada neste *host*.

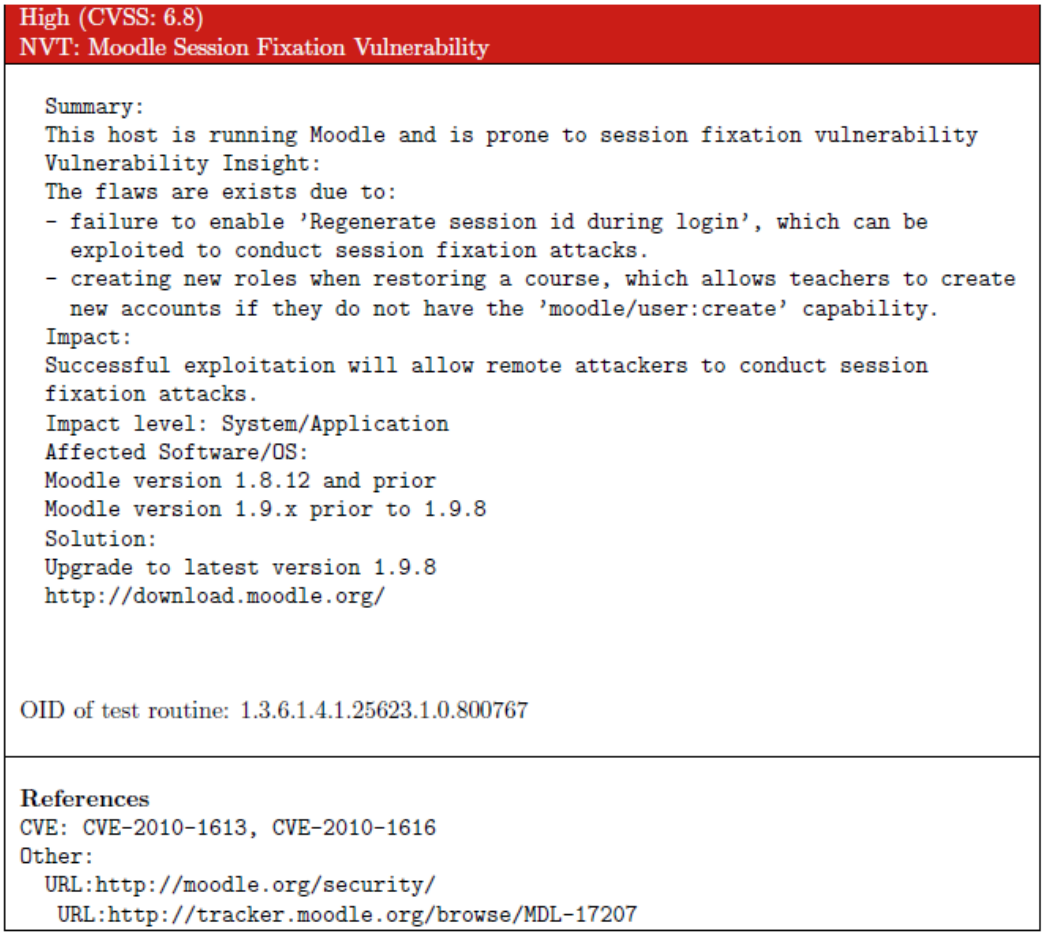


Figura 13- OpenVAS alto risco UFSM

Este servidor que está executando o Moodle foi o que apresentou maior propensão a exploração de vulnerabilidades de alto risco, através dos testes o OpenVAS detectou que este *host* está propenso a vulnerabilidade de fixação de sessão, e essas falhas são devido ao fato de não permitir reestruturar o ID da sessão durante o *login* e de os professores poderem criar novas contas e papéis dentro do Moodle, após restaurarem um curso sem ter a devida permissão no diretório: 'moodle/user:create'. O identificador de sessão é usado na comunicação de rede para identificar uma sessão, e será concedido a um usuário na primeira visita ao *site* para assim manter o estado das sessões dos usuários. O ID da sessão é uma sequência de *bytes* escolhida pelo servidor para identificar uma sessão ativa ou uma sessão reiniciável.

No ataque de fixação de sessão o *hacker* impõe para a vítima o identificador de sessão de sua conveniência antes que ele realize a autenticação em um *site*. Uma das vulnerabilidades que possibilita este ataque é o fato de que muitos servidores *web* atribuem um ID de sessão para os usuários no inicio da sessão, e após a autenticação mantem este ID de sessão inalterado, ao invés de criar um novo ID de sessão, que venha a representar a sessão do usuário após o *login* (BORGES, 2014).

A vulnerabilidade apresentada na Figura 14 retrata a situação de uma falha de risco *Medium* no *host* com IP 200.18.45.18.

2.19.2 Medium https (443/tcp)

Medium (CVSS: 4.3)
NVT: Check for SSL Weak Ciphers

```
Summary:
This routine search for weak SSL ciphers offered by a service.
Vulnerability Insight:
These rules are applied for the evaluation of the cryptographic strength:
- Any SSL/TLS using no cipher is considered weak.
- All SSLv2 ciphers are considered weak due to a design flaw within the SSLv2 pr
↪otocol.
- RC4 is considered to be weak.
- Ciphers using 64 bit or less are considered to be vulnerable to brute force me
↪thods
  and therefore considered as weak.
- 1024 bit RSA authentication is considered to be insecure and therefore as weak
↪.
- CBC ciphers in TLS < 1.2 are considered to be vulnerable to the BEAST or Lucky
↪ 13 attacks
- Any cipher considered to be secure for only the next 10 years is considered as
↪ medium
- Any other cipher is considered as strong
Solution:
The configuration of this services should be changed so
that it does not support the listed weak ciphers anymore.
Weak ciphers offered by this service:
  SSL2_RC4_128_MD5
  SSL2_RC4_128_EXPORT40_WITH_MD5



OID of test routine: 1.3.6.1.4.1.25623.1.0.103440
```

Figura 14- OpenVAS médio risco UFSM

O relatório especifica que encontrou uma vulnerabilidade de risco médio através dos testes de vulnerabilidade de rede (NVTs) realizado para verificar a segurança das codificações fracas utilizadas pelo SSL, e avaliar eficácia da criptografia oferecida por um serviço. As regras aplicadas pelos NVTs apresentaram uma série de informações sobre aplicação de codificação fracas na utilização do SSL, que são:

- A utilização dos protocolos SSL/TLS sem nenhuma cifra é considerado fraco;
- Todas as cifras SSLv2 são consideradas fracas devido a uma falha de projeto no âmbito do protocolo SSLv2;
- O algoritmo de criptografia RC4 é considerado fraco;
- Cifras usando 64 bits ou menos são considerados vulneráveis aos métodos de força bruta;

- 1024 bits na autenticação RSA são considerados inseguro;
- Cifras CBC em TLS 1.2 são consideradas vulneráveis ao BEAST ou *Lucky 13 attacks*;
- Qualquer cifra considerada segura somente para os próximos 10 anos.

O protocolo SSL provê privacidade da comunicação sobre a *internet*. O protocolo fornece criptografia de dados e autenticação entre um cliente e um servidor *web*, com o propósito de garantir a autenticidade, confiabilidade e integridade através das conexões TCP/IP. O protocolo SSL está localizado entre as camadas de aplicação e transporte, o que permite que ele trabalhe a cima dos protocolos HTTP, FTP, Telnet, SMTP, IMAP, POP3 a fim de garantir uma comunicação confiável.

Segundo Kurose (2006) o SSL tem as seguintes características: Autenticação do servidor SSL, onde permite que o usuário confirme a identidade de um servidor; Autenticação do cliente SSL, permite que o servidor confirme a identidade de um usuário; Sessão SSL criptografada, na qual toda a informação enviada entre browser e servidor é criptografada pelo *software* remetente.

O SSL implementa duas novas camadas formadas por dois protocolos, o *SSL Handshake Protocol* e o *SSL Record Protocol*. O *SSL Handshake Protocol* faz a autenticação entre cliente servidor, permite a negociação de um algoritmo de criptografia e as chaves criptográficas. A criptografia é aplicada de forma simétrica, usando entre outros os métodos AES ou RC4. Toda a troca de dados é feita com verificação da integridade baseada em *hash*, como por exemplo, SHA-1, MD5 e outros, para aumentar a segurança do processo inicial, e faz uso de certificados com formato X.509. O *SSL Record Protocol*, utilizado para encapsular todas as mensagens dos demais protocolos das camadas superiores, fragmenta os dados em blocos, podendo comprimi-las, aplica um código MAC, encripta e transmite o resultado.

A versão 2.0 do SSL foi lançada em 1995, mas possui falhas no seu projeto. A correção destas originou a ultima versão do SSL, a versão 3.0, lançada dois anos depois, e que serviu de base para protocolo *Transport Layer Securit*y (TLS) versão 1.0.

O SSLv2, por possuir falhas, torna-se suscetível a ataques que permitam romper ou espionar a comunicação. Uma das falhas desta versão é no processo de *Handshake*, o qual permite um ataque *man-in-the-middle* (MITM), onde o *hacker* se coloca entre o usuário e o servidor, a fim de capturar os pacotes, modificá-los e reenviá-los para ambos os lados da conexão (NAKAMURA; GEUS, 2007). O BEAST e Lucky 13 exploram uma vulnerabilidade que usam criptografia *Cipher Block Chaining* (CBC) como

criptografia de dados em determinadas configurações em servidores *web*, quando explorado permite a realização de ataques MITM.

Como solução para as vulnerabilidades citadas pelo OpenVAS, alterar a configuração desses serviços, a fim de garantir que não utilize as cifras fracas listadas, seria uma forma de proteger a rede de possíveis ataques. O OpenVAS ainda fornece as cifras utilizadas por esses serviços e classificadas como fracas que são: SSL2_RC4_128_MD5 e SSL2_RC4_128_EXPORT40_WITH_MD5.

Para vulnerabilidades de baixo risco o *host* com IP 200.18.45.19 permitiu uma solicitação não autenticada do *Scanner* OpenVAS no servidor *proxy* HTTP conforme pode ser visto na Figura 15.

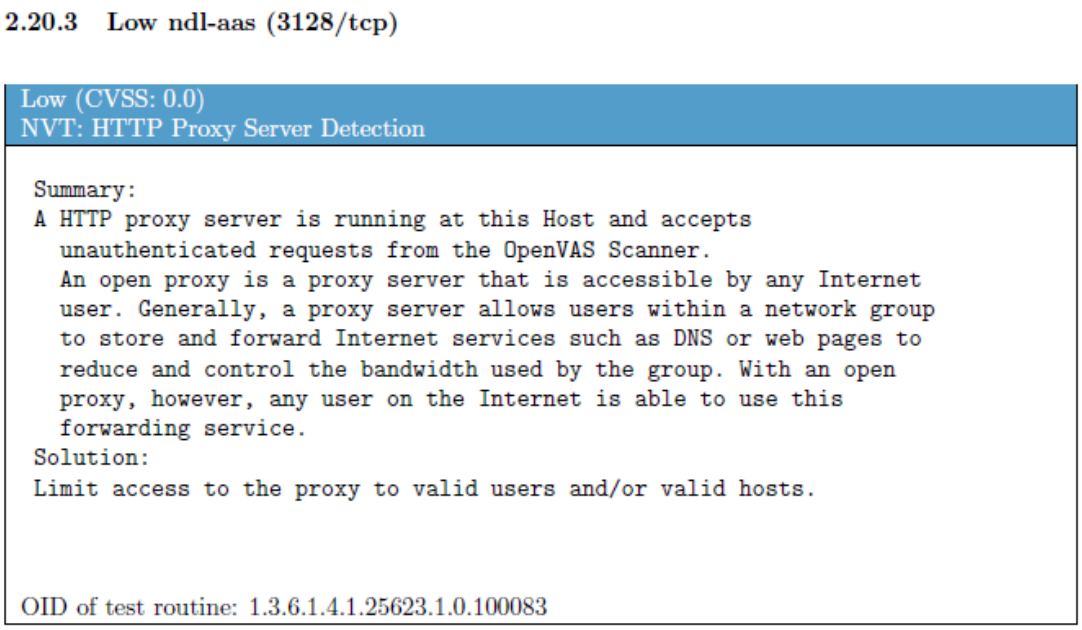


Figura 15- OpenVAS baixo risco UFSM

O OpenVAS informa que neste *host* está sendo executado um servidor *proxy* HTTP, e que o servidor *proxy* com configurações que não solicitam a autenticação do cliente pode se tornar acessível por qualquer usuário da *internet*.

O *proxy* desempenha a função de intermediário da conexão do cliente com o servidor externo. Como a conexão direta entre usuário interno e o servidor externo não é permitida, todas as requisições que são feitas ao servidor passarão pelo *proxy* que traduz as informações e repassa para o usuário, mascarando o IP do *host* interno e garantindo

assim mais segurança à rede interna. Para o cliente se conectar ao servidor *proxy*, primeiramente deve realizar a autenticação, após a autenticação o cliente envia sua requisição ao *proxy* que retransmite ao servidor.

Para o *host* analisado neste relatório de escaneamento o OpenVAS, apresenta que geralmente um servidor *proxy* aberto permite a usuários mal intencionados, dentro de um grupo de rede, armazenar serviços de *internet* e redirecionar páginas *web* ou de DNS para reduzir a largura de banda utilizada e monitorar os acessos dos usuários desse grupo. Com um *proxy* aberto, qualquer usuário na *internet* é capaz de usar este serviço de encaminhamento. Em outras palavras, *proxy* aberto em um computador é uma área da qual o *host* está vulnerável para *hackers* manipularem conteúdos dentro de um grupo de rede. Para solucionar as vulnerabilidades listadas pelo OpenVAS, a ferramenta sugere que seja limitado o acesso ao *proxy* para usuários válidos e /ou *hosts* válidos. Essa medida resulta em um aumento considerável da segurança da navegação *web*.

Na rede de servidores do CTISM o OpenVAS detectou duzentos e quatorze *hosts*, sendo que seis destes apresentaram vulnerabilidade de alto risco (*High*), treze com risco médio (*Medium*) e quinze com baixo risco (*Low*). Entretanto, as vulnerabilidades apresentadas para esta rede possuem soluções bastante simples, segundo a sugestão apresentado pelo OpenVAS. A Figura 16 apresenta o *host* com IP 172.17.8.31, o qual possui vulnerabilidades de alto risco, na versão PHP 5.3.6, que sofre vulnerabilidade de estouro de inteiros e erros de estouro de *buffer*.

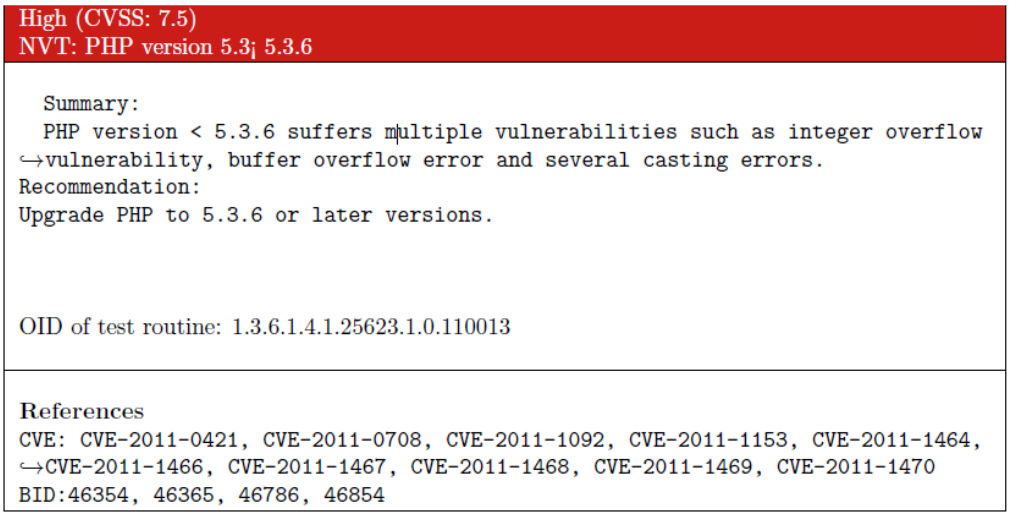


Figura 16: OpenVAS alto risco CTISM

Assim como estouro de *buffer*, os estouros de inteiro são erros de programação. Um inteiro é um tipo de dados que pode conter valor numérico, e precisa ter a capacidade de determinar se o valor numérico armazenado é positivo ou negativo. O inteiro com sinal armazena 1 ou 0 no *bit* mais significativo de seu primeiro *byte*. O estouro de inteiro ocorre porque os valores que podem ser armazenados excedem o valor suportado pelo próprio tipo de dado. Isto ocorre por causa de uma incompatibilidade de inteiros com sinal/sem sinal encontrada na versão PHP 5.3.6 ou menor.

Os ataques de estouro de inteiros possibilitam ataques de estouro de *buffer*. Este por sua vez ocorre quando um usuário ou processo tenta colocar mais dados em um *buffer* do que foi alocado anteriormente, causando uma violação na segmentação. Este tipo de comportamento pode ser explorado para se obter acesso ao sistema alvo, através do envio de um bloco de dados maior do que o *buffer* pode suportar. O *hacker* pode gerar assim uma negação de serviço, ou ainda fazer o sistema alvo executar um código malicioso (MCCLURE et al., 2012). Para solucionar os problemas encontrados, a ferramenta recomenda atualizar a versão atual PHP 5.3.6 por versões posteriores.

A Figura 17 destaca uma vulnerabilidade de médio risco no *host* de IP 172.17.8.4, onde foi detectada uma fraqueza no servidor *web* Apache. Este, quando configurado para usar a diretiva *FileETag*, está sujeito a vulnerabilidades que permitem ao atacante obter informações confidencias a respeito do servidor. Devido à maneira pela qual a diretiva *FileEtag* gera cabeçalhos de resposta Apache *Etag*, devolvida a um cliente, contem o número *inode* do arquivo. A diretiva *FileEtag* representa quais elementos serão usados para montar o *Etag* de cada arquivo.

2.5.1 Medium http (80/tcp)

Medium (CVSS: 4.3)
NVT: Apache Web Server ETag Header Information Disclosure Weakness

```
Summary:
A weakness has been discovered in Apache web servers that are
configured to use the FileETag directive. Due to the way in which
Apache generates ETag response headers, it may be possible for an
attacker to obtain sensitive information regarding server files.
Specifically, ETag header fields returned to a client contain the
file's inode number.
Exploitation of this issue may provide an attacker with information
that may be used to launch further attacks against a target network.
OpenBSD has released a patch that addresses this issue. Inode numbers
returned from the server are now encoded using a private hash to avoid
the release of sensitive information.
Solution:
OpenBSD has released a patch to address this issue.
Novell has released TID10090670 to advise users to apply the available
workaround of disabling the directive in the configuration file for
Apache releases on NetWare. Please see the attached Technical
Information Document for further details.
Information that was gathered:
Inode: 265039
Size: 163
```

OID of test routine: 1.3.6.1.4.1.25623.1.0.103122

References

```
CVE: CVE-2003-1418
BID:6939
Other:
  URL:https://www.securityfocus.com/bid/6939
```

Figura 17- OpenVAS médio risco CTISM

Para contornar esta vulnerabilidade a OpenBSD lançou um *patch* de correção com o intuito de resolver esse problema. Através da codificação dos números de *inode* retornado do servidor, que utiliza um *hash* privado para evitar a divulgação de informações sensíveis. A empresa Americana Novell lançou uma atualização, TID10090670, a qual aconselha os usuários aplicá-la como uma solução alternativa ao invés de desabilitar a diretiva no arquivo de configuração Apache.

No *host* de IP 172.17.8.3 o OpenVAS destacou uma das vulnerabilidades de baixo risco encontradas. A Figura 18 apresenta o resultado do relatório, com a vulnerabilidade detectada no *host*, o qual está executando um servidor Radius (*Remote Authentication Dial In User Service*).

2.4.1 Low radius (1812/udp)

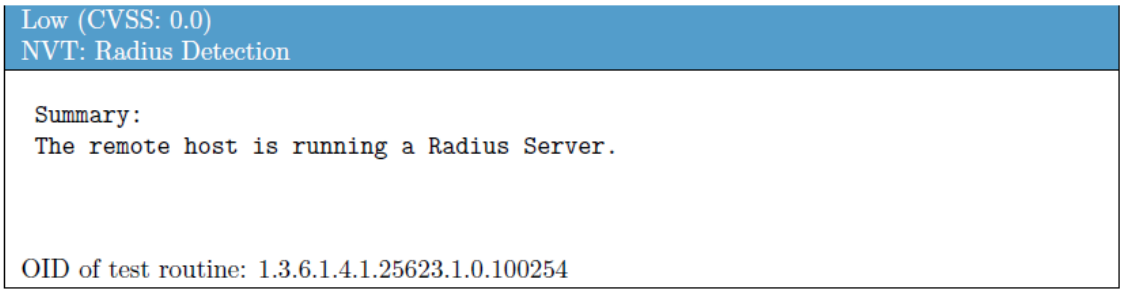


Figura 18- OpenVAS baixo risco CTISM

O servidor Radius é usado para autenticação, autorização e contabilização de terminais de acesso discado que utilizam o protocolo Radius, e para autenticação de usuários em redes sem fio via *Extensible Authentication Protocol* (EAP). O protocolo EAP é usado para transmitir informações de autenticação entre um cliente na estação de *wi-fi* e o servidor de autenticação Radius.

De acordo com o Centro de Coordenação CERT Advisory CA-2002-06 (CERT/CC) duas vulnerabilidades foram identificadas em diversas implementações do protocolo Radius, podendo ser exploradas remotamente, possibilitando a execução de códigos arbitrários com os mesmos privilégios, do usuário *root*, atribuídos ao servidor do Radius, ou resultar em uma negação de serviço (DoS). Para solucionar essas vulnerabilidades o Centro de Coordenação CERT sugere aplicar um *patch* ou atualizar para a versão específica pelo fornecedor.

4.2 Comparação

Os relatórios apresentados pelas ferramentas de escaneamento evidenciaram diversas vulnerabilidades, em diversos níveis de risco. Dessa forma é possível analisar as possíveis soluções sugeridas pelos *softwares* e aplicá-las para correção das falhas de modo que seja possível sanar todas as vulnerabilidades encontradas ou evitar futuros problemas. Entretanto, alguns itens apontados como falhas ou vulnerabilidades são na

verdade configurações necessárias para que serviços funcionem de maneira correta, a exemplo do servidor do moodle apresentado na seção 4.4, que pode funcionar da maneira descrita, para assegurar seu melhor desempenho ao oferecer o serviço à comunidade acadêmica.

Tabela 1- Comparativo entre LanGuard e OpenVAS

Risco/*Software*	**LanGuard**	**OpenVAS**
Baixo	49	15
Médio	3	13
Alto	23	6

A Tabela 1 apresenta um comparativo entre todos os resultados obtidos com o LanGuard o qual ao todo encontrou vinte e um *hosts* e o OpenVAS que detectou duzentos e quatorze *hosts* no total, com o escaneamento realizado na rede de servidores do CTISM.

Nota-se uma diferença acentuada entre o total de riscos classificados como baixo e alto entre o LanGuard e o OpenVAS, apesar deste último ter encontrado um número superior de *hosts*. O contrário acontece para os riscos categorizados como médio, onde o OpenVAS encontra um número superior de vulnerabilidades. Dentre as vulnerabilidades de médio risco listadas pelo LanGuard, destacam-se as vulnerabilidades no serviço SNMP, em função da configuração padrão do dispositivo, diferentemente das encontradas pelo OpenVAS, o qual destacou vulnerabilidades no servidor *web* Apache devido as suas configurações para prover acesso às requisições solicitadas. O LanGuard apresenta o serviço HTTP repetidamente como uma vulnerabilidade de baixo risco, e o serviço FTP, que permite a usuários transferirem arquivos facilmente de um sistema para outro, foi o mais significativo nos dispositivos escaneados pelo OpenVAS com o mesmo risco.

Tabela 2- Comparativo entre Nessus e OpenVAS

Risco/*Software*	**Nessus**	**OpenVAS**
Baixo	12	8
Médio	21	26
Alto	1	8

A Tabela 2 apresenta os resultados obtidos através dos escaneamento realizados com as ferramentas Nessus e OpenVAS na rede de servidores da UFSM. Será levado em consideração o fato de que a versão do Nessus utilizada possibilita escanear até dezesseis IPs, deste modo, para uma comparação mais precisa, serão avaliados os dispositivos com a mesma faixa de IP em ambos os *softwares*. Diferentemente dos resultados anteriormente expostos na Tabela 1, que explana a comparação entre LanGuard e OpenVAS, os resultados deste relatório apresentaram um intervalo menor entre o número de registros obtidos nos dois *softwares* comparados. O fator de risco que apresentou maior distanciamento nos resultados foi o de alto risco, onde o Nessus apontou uma única vulnerabilidade, mencionado na seção 4.3, também encontrada no mesmo *host* pelo OpenVAS. Entre os oito resultados de alto risco localizados nos dispositivos pelo OpenVAS, destaca-se na grande maioria o serviço de HTTP, o qual sugeriu atualizar a versão PHP que está em execução.

Observou-se que para as vulnerabilidades de nível médio e baixo risco houve pouca disparidade nos resultados apresentados. Dentre os registros apontados como risco médio apresentados pelo relatório da ferramenta Nessus destaca-se a presença de várias portas em um mesmo *host* com a mesma vulnerabilidade, o que possibilita ao atacante requisitar acesso a portas distintas, a partir de *proxy* remoto, de maneira arbitrária. Este mesmo cenário se faz presente em outros dispositivos. No OpenVAS a maioria das vulnerabilidades de risco médio é referente a informações gerais do TCP, onde os *hosts* aceitam requisições de pacotes IP de qualquer origem, e que podem ser utilizados por um atacante que tenha por finalidade burlar a filtragem de pacotes mal projetada.

As vulnerabilidades classificadas como baixo risco apresentadas pelo relatório das ferramentas, destacou para Nessus vulnerabilidades no algoritmo de criptografia utilizado no servidor SSH presentes em mais de um *host* em portas distintas, que são:

tcp/22, tcp/223, tcp/2222 e tcp/2223. Entre as vulnerabilidades listadas pelo OpenVAS com baixo fator de risco, uma se destacou pelo fato de apresentar a possibilidade de contribuição na melhoria do *software* através do contato com a equipe de desenvolvimento do sistema. O relatório apontou um servidor desconhecido rodando na porta tcp/3317 e orientou o administrador da rede, caso o serviço seja conhecido, relatar ao suporte do OpenVAS.

Um detalhe importante surgiu com a conclusão dos escaneamentos das redes em questão: houve diferença no modo como cada *software* tratou as informações coletadas e gerou os seus respectivos relatórios, ou seja, a interpretação de cada registro gerou modelos diferentes para a apresentação do relatório final. Por exemplo, uma vulnerabilidade encontrada no OpenVAS, Figura 19, foi apresentada apenas como um *log*, enquanto no Nessus, Figura 20, a mesma vulnerabilidade foi classificada em médio risco.

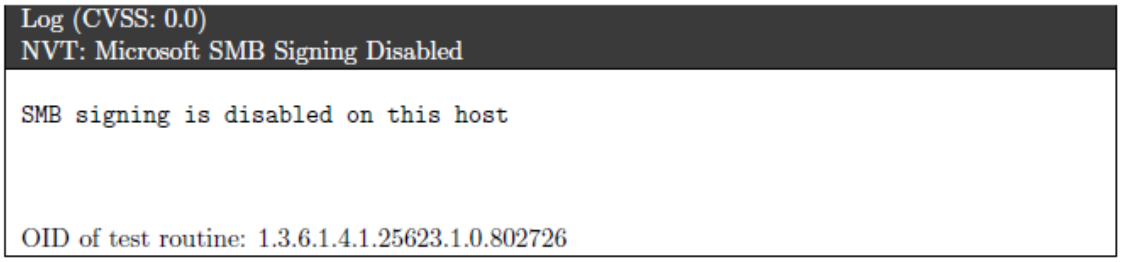
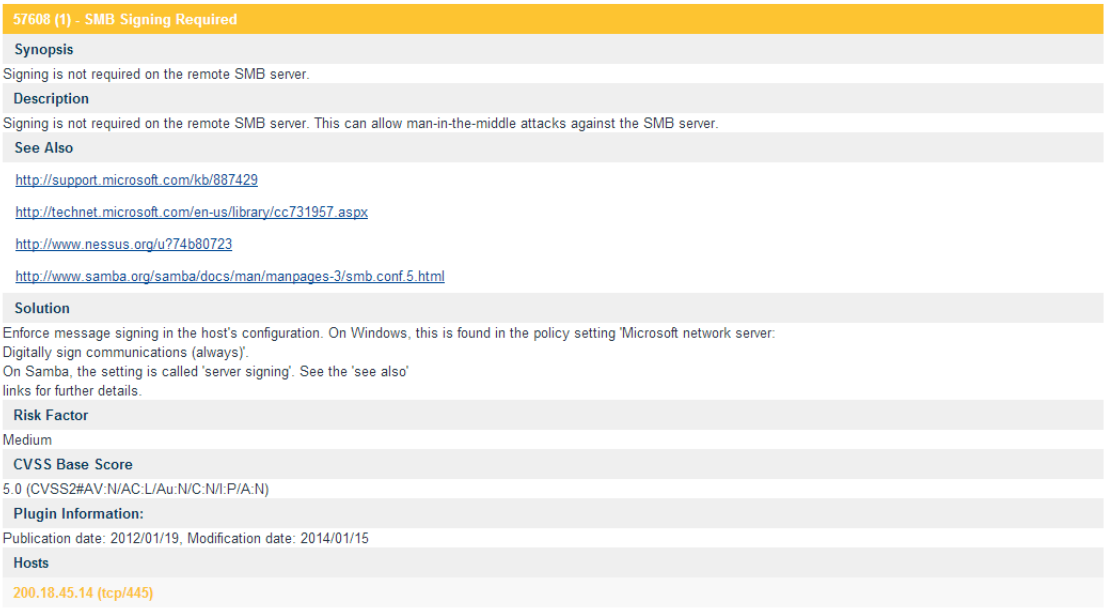


Figura 19- *Log* do OpenVAS.



57608 (1) - SMB Signing Required

Synopsis

Signing is not required on the remote SMB server.

Description

Signing is not required on the remote SMB server. This can allow man-in-the-middle attacks against the SMB server.

See Also

http://support.microsoft.com/kb/887429

http://technet.microsoft.com/en-us/library/cc731957.aspx

http://www.nessus.org/u?74b80723

http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html

Solution

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server:
Digitally sign communications (always)'.
On Samba, the setting is called 'server signing'. See the 'see also'
links for further details.

Risk Factor

Medium

CVSS Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

Plugin Information:

Publication date: 2012/01/19, Modification date: 2014/01/15

Hosts

200.18.45.14 (tcp/445)

Figura 20- Vulnerabilidade do Nessus.

5 CONSIDERAÇÕES FINAIS

Considerando o exposto neste trabalho, pode-se concluir que a alta disponibilidade, a exigência da integridade e confidencialidade dos dados é de fundamental importância para a segurança da informação.

No entanto, observa-se ainda que é possível identificar vulnerabilidades em um sistema e verificar a tolerância a ataques que uma rede suporta, mas, infelizmente, não se pode afirmar que uma rede é totalmente segura. As vulnerabilidades e ameaças descobertas são solucionadas, ao mesmo tempo em que novas vulnerabilidades são exploradas e técnicas de ataques são criadas. Várias ferramentas surgiram para prevenir e auxiliar na tarefa de identificar problemas de segurança nas redes, essas por sua vez apresentadas no Capítulo 2 deste trabalho.

Diante desse cenário, testes que verificam a segurança da organização como, por exemplo, os *pentest*, e a análise de vulnerabilidades, tornam-se uma necessidade quando se trata de segurança de redes. Através da frequente análise e avaliação da segurança da rede, um profissional de segurança é capaz avaliar o nível de segurança da sua rede e adotar medidas de segurança adequadas em tempo hábil. Os scanners detectores de vulnerabilidades são uma das ferramentas utilizadas que podem ser incorporadas na rotina dos administradores de redes.

Os *softwares*, como o LanGuard, o Nessus e o OpenVAS, utilizados nesse trabalho, auxiliam na descoberta de vulnerabilidades em ativos de redes, bem como sistemas operacionais em execução, portas abertas, serviços ativos e a necessidade de atualização de *patches*. Sistemas, como os citados anteriormente, auxiliam também na solução das vulnerabilidades apontadas, visto que dispõem de relatórios que descrevem vulnerabilidades, classificando-as em grau de risco, e sugerem soluções para as vulnerabilidades apontadas.

Através dos testes realizados com estas ferramentas que são capazes de listar possíveis vulnerabilidades, bem como vulnerabilidades propriamente ditas, nos *hosts* escaneados é possível entender a importância da utilização de ferramenta que realizam a varredura de rede auxiliando dessa forma o administrador na manutenção da segurança. A grande maioria das vulnerabilidades encontradas apresentou problemas de *patches* ausentes, onde o servidor em execução utiliza uma versão desatualizada do serviço que

oferece. As falhas apresentadas pelas ferramentas podem ser corrigidas seguindo as referências apresentadas por órgãos regulamentadores, como, por exemplo, o *Common Vulnerabilities and Exposures* (CVE). O CVE é um dicionário gratuito e público que contém informações sobre vulnerabilidades, e tem como objetivo padronizar as informações sobre uma vulnerabilidade. Deste modo, através da pesquisa realizada no presente estudo, é possível encontrar muitas medidas de segurança que possam servir como base de informações na prevenção de ataques e na descoberta precoce de vulnerabilidades na rede.

Para trabalhos futuros, poderá ser analisada a possibilidade de realizar testes de escaneamento da rede juntamente com outras ferramentas que auxiliem na detecção de vulnerabilidades. Nesta análise, poderá ser realizada uma pesquisa sobre diversas ferramentas capazes de detectar ameaças a redes de computadores juntamente com possíveis ataques as redes. Ainda poderá ser realizada a análise detalhada por cada CVE encontrada pelo *software*, e também a análise de possíveis falsos positivos que possam existir.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ABNT NBR ISO/IEC 17799:2000: **Tecnologia da Informação – Código de Prática para Gestão da segurança de Informação**. Rio de Janeiro, 2001.

ABNT NBR ISO/IEC 27005: **Tecnologia da Informação- Técnicas de Segurança- Gestão de Risco de Segurança da Informação**. Rio de Janeiro, 2008.

BAUER, C. A. **Política de segurança da informação para redes corporativas.** Trabalho de conclusão de curso – Centro Universitário Feevale, 2006.

BRASIL. Tribunal de Contas da União. **Boas Práticas de Segurança da Informação/ Tribunal de Contas da União**. 4. ed. Brasília: TCU, Secretaria de Fiscalização de Tecnologia da Informação, 2012.

BORGES, A. **Ataque de fixação de sessão**. Revista Linux. 2014. Disponível em: < http://www.linux-magazine.com.br/images/uploads/pdf_aberto/LM_92_14_15_02_col-alexborges.pdf >. Acessado em: 2 Jun. 2014.

BROWN, T; GALITZ, G. **O farejador de vulnerabilidades OpenVAS**. Linux Magazine, São Paulo, Abr. 2010.

CABRAL, L.; NÓBREGA, A.; SOARES, R. **Efetuando uma Implementação Segura do Protocolo SNMP em Roteadores Cisco: Da teoria à prática**. Unibratec, Recife, 2012.

CARISSIMI, A. S.; ROCHOL, J.; GRANVILLE, L. Z. **Redes de computadores: Volume 20 da Série Livros didáticos informática UFRGS.** Porto Alegre: Bookman, 2009.

CERT.org (2002). **Software Engineering Institute.CERT/CC advisories: Vulnerabilities in Various Implementations of the RADIUS Protocol**. Disponível em: < http://www.cert.org/historical/advisories/CA-2002-06.cfm >. Acessado em: 13 Jun. 2014.

CHESWICK, W. R.; BELLOVIN, S. M.; RUBIN, A. D. **Firewalls e Segurança na Internet: Repelindo o Hacker ardiloso**. 2 ed. Porto Alegre: Editora Bookman Companhia, 2005.

CONTESSA, D.F.; POLINA, E.R. **Gerenciamento de Equipamentos Usando o Protocolo SNMP**. Departamento de Pesquisa – CP Eletrônia S.A. Disponível em: <http://petry.pro.br/arquivos/Artigo_Gerenciamento_SNMP_MRTG.pdf >. Acessado em: 25 Mai. 2014.

CORSO, A. A. **Instalação e Utilização de um Sistema de Detecção de Intrusão**. Trabalho de Graduação- Universidade do Rio Grande do Sul, Porto Alegre, 2009.

COUTINHO, J.C.P. **Processo de Testes de Vulnerabilidades em Componentes MVC para CMS JOOMLA**. Monografia – Engenharia de Sistemas, ESAB, 2011.

DUMONT, C. E. S. **Segurança Computacional – Segurança em Servidores Linux em Camadas**. Monografia- Universidade Federal de Lavras, 2006.

FERREIRA, F. N. F.; ARAUJO, M. T. **Políticas de Segurança da Informação: Guia Prático para Embalagem e Implementação.** Rio de Janeiro: Editora Ciência Moderna Ltda., 2006.

GFI NETWORK SECURITY. Disponível em: < http://www.gfi.com/languard/>. Acesso em: 15 Mar. 2014.

GOMES, G. C. **Habilitação de Informações de Gerenciamento SNMP para o Sistema Wireless da UFLA**. Monografia de graduação- Universidade Federal de Lavras, Lavras-MG, 2002.

JUNIOR, O. S. **Roteiro para a Realização de Testes de Penetração em Cenários TURN-KEYS**. Trabalho de Conclusão de Curso- Universidade do vale do Itajaí, Itajaí-SC, 2010.

HULBRICH, H.C; DELLA VALLE, J. **Universidade H4CK3R – Desvende todos os segredos do submundo dos hackers**. 4. ed. São Paulo: Digerati Comunicação e tecnologia Ltda, 2004.

KUROSE, J. F., ROSS K. W. **Redes de Computadores e a Internet: uma abordagem top-down.** 3. Ed. São Paulo: Pearson Addison Wesley, 2006.

MARTINELO, C. A. G.; BELLEZI, M. A. **Análise de Vulnerabilidades com OpenVAS e Nessus**. Departamento de Computação- Universidade Federal de São Carlos, São Paulo, 2014.

MCCLURE, S.; SCAMBRAY, J.; KURTZ, G. **Hackers Expostos: Segredo e Solução para a Segurança de Redes**. 7 ed. Porto Alegre: Bookman Companhia Editora Ltda, 2012.

MELO, C. B. S.; GERVILLA, L. C. **Um estudo sobre técnicas de detecção de vulnerabilidade em ambientes de redes sem fio.** 2010. Trabalho de graduação – Faculdade de Tecnologia de São José dos Campos, 2010.

MÓDULO. **Curso Básico de Segurança da Informação** (Academia Latino-Americana de Segurança da Informação). Módulo Security, 2006.

MORAES, A. F. de. **Redes de computadores:** fundamentos. 7. Ed. São Paulo: Editora Érica, 2010.

MOREIRA et al. **Scanners de Vulnerabilidades Aplicados a Ambientes Organizacionais**. Revista Eletrônica da Faculdade Metodista Granbery: Jul/Dez, 2008.

MORENO, D. **Tipos de PenTest**. Disponível em: < http://www.100security.com.br/tipos-de-pentest/ >. Acessado em: 7 Mai. 2014.

NAKAMURA, E. T.; GEUS, P. L de. **Segurança de redes em ambientes cooperativos**. São Paulo: Novatec Editora, 2007.

OPENVAS. Disponível em: < http://www.openvas.org/index.html >. Acessado em 25 Mai. 2014.

PINHEIRO, J. M. **Ameaças e Ataques aos Sistemas de Informação: Prevenir e Antecipar.** 5. ed. Rio de Janeiro: Cadernos UniFOA, 2007.

RAMOS, J.J.A. **Sistema Automático para Realização de Testes de Penetração**. Dissertação para mestrado – Escola Superior de Tecnologia e Gestão, IPBeja, 2013.

RFC 2196 (1997). **Manual de segurança do site** (RFC 2196). Disponível em: < http://tools.ietf.org/html/rfc2196>. Acessado em: 12 Abr. 2014

RFC2828 (2010). **Glossário de Segurança de Internet** (RFC 2828). Disponível em: < http://www.ietf.org/rfc/rfc2828.txt>. Acessado em: 3 Abr. 2014.

RODRIGUES, P.E.B. **Segurança Informática de Redes e Sistemas (Abordagem Open-Source)**. Dissertação para obtenção de grau de Mestre – Universidade de Trás-os-Monte e Alto Douro, 2010.

SANS, **Intrusion Detection**. Disponível em: www.sans.org. Acessado em: 9 Jun. 2014.

SANTOS, M. C. D.; SILVA, J. R. **Avaliação de Diferentes Ferramentas para Realização de Testes de Segurança em Computadores e em Redes Locais (Lan's)**. Mestrado em Ciencia da Computação- Universidade do Porto (FEUP), 2012.

SÊMOLA, M. **Gestão da Segurança da Informação – Uma visão Executiva**. Rio de Janeiro: Editora Campus, 2003.

SCHWARZER, S. **OpenVAS 4 Análise detalhada**. Linux Magazine, São Paulo, Out. 2011.

SILVA, P. T.; CARVALHO, H.; TORRES, C. B. **Segurança dos Sistemas de Informação – Gestão Estratégica da Segurança Empresarial.** Lisboa: Centro Atlântico Ltda., 2003.

STALLINGS, W. **Redes e Sistemas de comunicação de Dados: Teoria e Aplicação Corporativas**. Rio de Janeiro: Elsevier, 2005.

TANENBAUM, A. S. **Redes de Computadores.** 4. ed. Rio de Janeiro: Elsevier, 2003.

TENABLE NETWORK SECURITY. Disponível em: < http://www.tenable.com/products/nessus>. Acessado em: 21 Mai. 2014.

THOMAS, T. **Segurança de Redes – Primeiros Passos.** Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

ULBRICH, H.C.; DELLA VALLE, J. **Universidade Hacker**. 4. ed. São Paulo: Digerati Comunicação e Tecnologia Ltda, 2004.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. **Grupo de Trabalho de Redes**. RFC 2196 [on line].1997.Disponível: <http://penta.ufrgs.br/gereseg/rfc2196/>

WANNER, P.C.H. **Ferramenta de Injeção de Falhas para Avaliação de Segurança**. Dissertação de Mestrado- Instituto de Informática, UFRGS, 2001.