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

ANÁLISE DE SOFTWARE DE ESCANEAMENTO DE VULNERABILIDADES EM REDES DE COMPUTADORES

TRABALHO DE CONCLUSÃO DE CURSO

Abrelino de Castro Bitencourt

Santa Maria, RS, Brasil

2014

ANÁLISE DE SOFTWARE DE ESCANEAMENTO DE VULNERABILIDADES EM REDES DE COMPUTADORES

Abrelino de Castro Bitencourt

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

Orientadora: Prof. Dra. Simone Regina Ceolin

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 SOFTWARE DE ESCANEAMENTO DE VULNERABILIDADE EM REDES DE COMPUTADORES

elaborado por
Abrelino de Castro Bitencourt

COMISSÃO EXAMINADORA

Simone Regina Ceolin, Dra.
(Presidente/Orientadora)

Rodrigo Castro Gil, Bel. (UFSM)

Thiago Cassio Krug, Bel. (UFSM)

Santa Maria, 11 de dezembro de 2014.

RESUMO

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

ANÁLISE DE SOFTWARE DE ESCANEAMENTO DE VULNERABILIDADES EM REDES DE COMPUTADORES

AUTOR: ABRELINO DE CASTRO BITENCOURT

ORIENTADORA: SIMONE REGINA CEOLIN

Data e Local da Defesa: Santa Maria, 11 de dezembro de 2014.

O crescente avanço tecnológico tornou o uso de computadores tanto no ambiente empresarial quanto para uso doméstico, indispensável. Assim como, o nível de conectividade a diferentes recursos é cada vez mais amplo. Esse cenário impacta em uma maior proliferação de ameaças a segurança da informação, esta que é um dos ativos mais importantes para qualquer organização, precisando estar protegida, garantindo seus princípios de confidencialidade, integridade e disponibilidade. Estabelecer um crescimento, com uma segurança equivalente, é para os profissionais da área de redes questão crucial nos dias de hoje. O apoio de técnicas e ferramentas que o auxiliem nesse trabalho se torna fundamental. Este trabalho propõe uma análise de *scanners* de vulnerabilidades, *software* que atuam na área preventiva de descoberta de possíveis pontos suscetíveis a falhas ou ameaças, visando descobrir dentre suas funcionalidades, a contribuição que podem trazer para a área segurança de redes.

Palavras-Chave: *Scanners*. Vulnerabilidades. Segurança de Redes.

ABSTRACT

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

VULNERABILITY SCANNERS SOFTWARE ANALYSIS IN COMPUTER NETWORKS

AUTHOR: ABRELINO DE CASTRO BITENCOURT

ADVISER: SIMONE REGINA CEOLIN

Date and Place of defense: Santa Maria, December 11th 2014.

The increasing technological advances became the use of computers in business environment and in the household, essential. Also, the level of connectivity to various technological resources is getting wider. This scenario impacts in an increasingly proliferation of threats to information security. This is one of the most important assets for any organization, needing to be protected, ensuring its principles of confidentiality, integrity and availability. Establishing a growth, with an equivalent security is a crucial question for professionals in the area of computer networks today, as well as the support techniques and tools to assist the professional in this work, are fundamental. This paper proposes a vulnerability scanners analysis, software working in the preventive area of discovery of possible susceptible points to failure or threats, aimed at discovering among its features, the contribution they can bring to the security area networks.

Keywords: Scanners. Vulnerability. Network Security.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABNT	-	Associação Brasileira de Normas Técnicas
DNS	-	*Domain Name System*
DoS	-	*Denial of Service*
GPL	-	*General Public License*
HTTP	-	*HyperText Transfer Protocol*
IANA	-	*Internet Assigned Numbers Authority*
IP	-	*Internet Protocol*
IPv6	-	*Internet Protocol version 6*
ISSO	-	*Internacional Organization for Standardization*
LAN	-	*Local Area Network*
NASL	-	*Nessus Attack Scripting Language*
NVT	-	*Network Vulnerability Tests*
OMP	-	*OpenVAS Management Protocol*
OTP	-	*OpenVAS Tranfer Protocol*
OVAL	-	*Open Vulnerability and Assessment Language*
SANS	-	*SysAdmin Audit Networking and Security Institute*
SNMP	-	*Simple Network Management Protocol*
SSL	-	*Secure Socket Layer*
TCP	-	*Transmission Control Protocol*
UDP	-	*User Datagram Protocol*

SUMÁRIO

1 INTRODUÇÃO

O crescente avanço tecnológico, aliado ao aumento da conectividade dos computadores à rede mundial contribuiu para o crescimento dos incidentes de segurança. O número de vulnerabilidades encontradas vem crescendo a cada ano e elas são normalmente utilizadas por atacantes que exploram suas fragilidades quando essas não são corrigidas por empresas ou organizações, segundo dados do Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.BR, 2014). A razão principal da realização de testes de segurança em sistemas computacionais é a de identificar vulnerabilidades e consequentemente repará-las, esta análise visa reduzir o risco em relação aos incidentes de segurança, é necessário detectar essas possíveis falhas e corrigi-las para garantir que a rede esteja em um nível de segurança adequado.

Uma vulnerabilidade é definida como uma condição que, quando explorada por um atacante, pode resultar em uma violação de segurança (PINHEIRO; KON, 2005 apud MARTINELO; BELLEZI, 2014). As vulnerabilidades são originadas de falhas na maioria das vezes não intencionais. Estas falhas podem ser:

- Físicas: Acesso aos ativos por pessoas não autorizadas;
- Naturais: Desastres naturais;
- Humanas: Decorrência de erro humano;
- *Hardware:* Falhas no hardware;
- *Software:* Falhas de configuração.

Devido ao grande número de vulnerabilidades em aplicações e sistemas que colocam em risco a segurança da informação, é necessário ao administrador algum auxílio, seja ele de componentes de *software*, aplicativos ou programas. Dentre essas opções de auxílio estão os *scanner*s de vulnerabilidades, os quais o presente trabalho pretende analisar as principais versões disponíveis, seu funcionamento e comparar resultados dos mesmos.

1.1 Objetivo geral

Obter conhecimentos sobre *scanners* de vulnerabilidades bem como sua importância, com motivação do crescente aumento da conectividade tanto de usuários domésticos quanto de organizações. O que gera proporcionalmente maior exposição destes a um ambiente cada

vez mais inseguro, suscetíveis a falhas de *software*, *hardware* ou até mesmo alvo de pessoas mal intencionadas.

1.2 Objetivos específicos

A partir dos conceitos sobre segurança, conhecer as principais técnicas de segurança com ênfase no escopo deste trabalho, analisar s*oftware* que possibilitem a técnica de varredura de vulnerabilidades, tendo em seus resultados uma possibilidade para que as vulnerabilidades sejam controladas e suspensas buscando prevenção às ações de caráter prejudicial ao sistema.

A organização deste trabalho está segmentada em Capítulos. O Capítulo 2 apresenta a revisão bibliográfica, com a base para o entendimento do que será visto no decorrer do trabalho. O Capítulo 3 relata os trabalhos relacionados a este estudo. O Capítulo 4 tem por objetivo a proposta de ferramentas de desenvolvimento deste trabalho. O Capítulo 5 tem como foco as análises práticas, em conjunto com os resultados obtidos. Por fim, no Capítulo 6, encontram-se as considerações finais e sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Para o revisão bibliográfica procurou-se levantar um estudo sobre os principais conceitos necessários para um bom entendimento do trabalho proposto.

2.1 Redes de Computadores

De acordo com Tanenbaum e Wetherall (2011), rede de computadores é um conjunto de computadores autônomos interconectados por uma única tecnologia. Dois computadores estão interconectados quando podem trocar informações. A conexão não precisa ser feita por um fio de cobre, também podem ser usadas fibras ópticas, microondas, ondas de infravermelho e satélites de comunicações. O compartilhamento de recursos tem como objetivo tornar todos os programas, equipamentos e dados ao alcance de todas as pessoas na rede, independente da localização física do recurso e do usuário. Prover a comunicação confiável entre os vários sistemas de informação, melhorar o fluxo e o acesso às informações, bem como agilizar a tomada de decisões administrativas facilitando a comunicação entre seus usuários.

O uso das redes vem, a cada dia, se tornando um recurso indispensável em todos os locais onde existe um conjunto de computadores. Com o crescimento da Internet abrangendo todos os ramos de atividade, aumentou ainda mais a necessidade da ligação dos computadores em redes, entretanto, é importante conhecermos as vantagens e as desvantagens do uso das redes e também os cuidados que devemos tomar para evitarmos os problemas.

2.2 Internet

A Internet é uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo. Há pouco tempo, esses dispositivos eram basicamente computadores de mesa, estações de trabalho e os assim chamados servidores que armazenam e transmitem informações, como páginas *web* e mensagens de e-mail. No entanto cada vez mais sistemas finais modernos da Internet: como TVs, console de jogos, telefones celulares, dispositivos de sensoriamento ambiental e de segurança estão sendo conectados a rede (KUROSE; ROSS, 2013).

Tamanha evolução tornou a Internet essencial para muitas instituições, incluindo grandes e pequenas empresas, universidades e órgãos do governo. Muitas pessoas também

contam com a Internet para suas atividades profissionais, sociais e pessoais. Mas atrás de toda essa utilidade existe o lado obscuro, em que "vilões" tentam causar problemas em nosso cotidiano danificando nossos computadores conectados à Internet, violando nossa privacidade e tornando inoperantes os serviços da Internet dos quais dependemos (SKOUDIS; LISTON, 2006).

2.3 Segurança da informação

A segurança de redes é uma parte essencial para a proteção da informação, fazendo com que uma boa estratégia leve em consideração, diferentes aspectos que se direcionem a realidade do ambiente ao qual se deseja manter seguro. Segurança da informação pressupõe a identificação de diversas vulnerabilidades e a gestão dos riscos associados a diversos ativos da informação de uma corporação, independentemente de sua forma ou do meio em que são compartilhados ou armazenados, digital ou impresso. O objetivo da segurança é garantir a confidencialidade, a integridade e a disponibilidade destes ativos de informação de uma corporação, segundo a norma ABNT NBR ISO/IEC 27002:2005:

> "Preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas" (ISO/IEC 27002, 2005, p.1).

Os objetivos fundamentais da segurança da informação são conceituados por Stallings (2010), da seguinte forma:

- **Confidencialidade** - propriedade que limita o acesso a informação tão somente às entidades legítimas, ou seja, àquelas autorizadas pelo proprietário da informação;
- **Integridade** - propriedade que garante que a informação manipulada mantenha todas as características originais estabelecidas pelo proprietário da informação, incluindo controle de mudanças e garantia do seu ciclo de vida (nascimento, manutenção e destruição);
- **Disponibilidade** - propriedade que garante que a informação esteja sempre disponível para o uso legítimo, ou seja, por aqueles usuários autorizados pelo proprietário da informação.

Segundo Kubota (2012), o grande problema não é o acesso à informação e, sim, gerenciá-la a fim de extrair o que é relevante e descartar o que não for útil. Com o desenvolvimento da Internet e seu poder de conectar pessoas criou grandes oportunidades econômicas e sociais tanto para pessoas bem-intencionadas quanto para as mal-intencionadas.

Porém o grande volume de produção, armazenamento e transferência de dados entre diferentes dispositivos e entre diversas redes resulta em um aumento significativo das ameaças e vulnerabilidades à segurança da informação. Não há dúvida de que o aumento da tecnologia significa agilidade dos sistemas e proporciona conforto e comodidade para os usuários, todavia significa criar mais pontos de fragilidade.

2.4 Gestão de segurança

A importância com a segurança cresce rapidamente quando se leva em consideração o aumento da complexidade das conexões, característico de ambientes empresariais e organizacionais. Para Nakamura e Geus (2007), a segurança é inversamente proporcional as funcionalidades, ou seja, quanto maiores as funcionalidades, como serviços, aplicativos e demais facilidades, menor é a segurança do ambiente. Isso pode ser explicado devido à maior abrangência dos pontos de ataque decorrentes de alguns fatores como:

- Exploração da vulnerabilidade em sistemas operacionais, aplicativos, protocolos e serviços;
- Exploração dos aspectos humanos das pessoas envolvidas;
- Falha no desenvolvimento e implementação da política de segurança.

Assim, quanto maior o número de sistemas, maior a responsabilidade dos administradores e maior a probabilidade de existência de brechas que podem ser exploradas. A previsão de todas as brechas é impraticável, principalmente porque o fator humano está envolvido, o que significa, por exemplo, que até mesmo a escolha de senha de cada usuário influi no nível de segurança do ambiente, o objetivo de uma eficaz gestão é, portanto, equilibrar a segurança com os riscos, a fim de minimizar os impactos que uma falha de segurança pode causar.

2.4.1 Ameaças a segurança

Incidentes de segurança da informação vêm aumentando consideravelmente ao longo dos últimos anos e assumem as formas mais variadas, como, por exemplo: infecção por vírus, acessos não autorizados, ataques *denial of service* contra redes e sistemas, furto de informação proprietária, invasão de sistemas, fraudes internas e externas, uso não autorizado de redes sem fio, entre outras.

O Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.BR, 2014), mantido pelo NIC.br do Comitê Gestor da Internet no Brasil é responsável por tratar incidentes de segurança em computadores que envolvam redes conectadas à Internet brasileira. O CERT.BR atua como um ponto central para notificações de incidentes de segurança no Brasil, provendo a coordenação e o apoio no processo de resposta a incidentes, suas atividades têm como objetivo estratégico aumentar os níveis de segurança e de capacidade de tratamento de incidentes das redes conectadas à Internet no Brasil.

Em recentes estatísticas o CERT.BR comprova o número elevado de incidentes atingindo o número total de 352925 no ano de 2013, conforme a Figura 1. Dentre estes 46,86% são ataques do tipo *Scan*: que são notificações de varreduras em redes de computadores, com o intuito de identificar quais computadores estão ativos e quais serviços estão sendo disponibilizados por eles, os dados completos podem ser vistos na Figura 2.

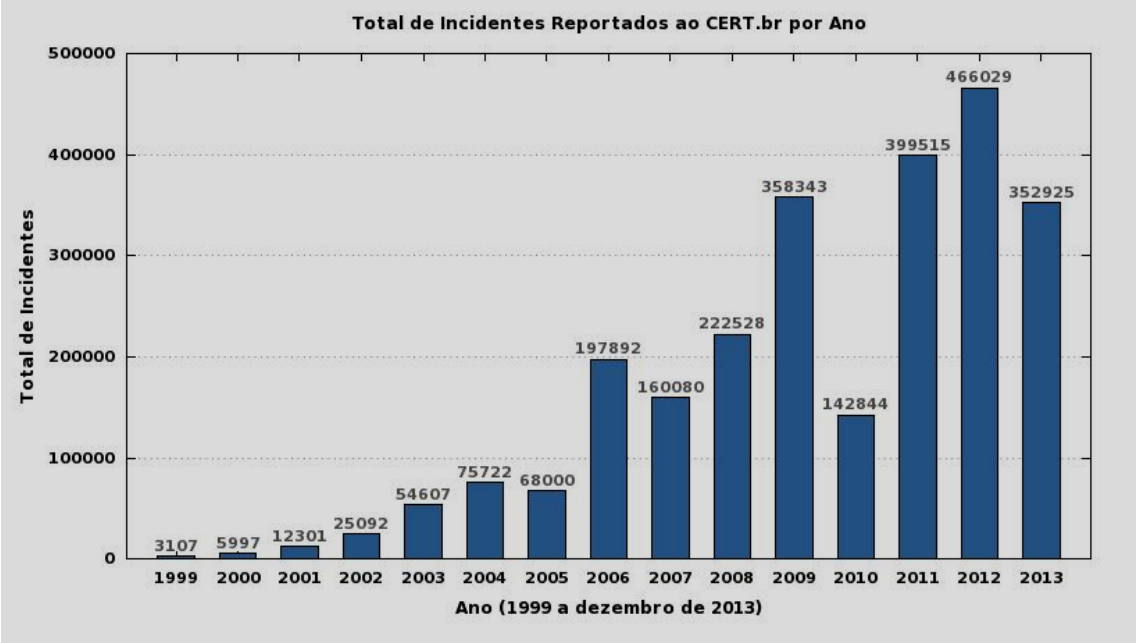


Figura 1: Incidentes reportados ao Cert.br de 1999 a dezembro 2013. (CERT.BR)

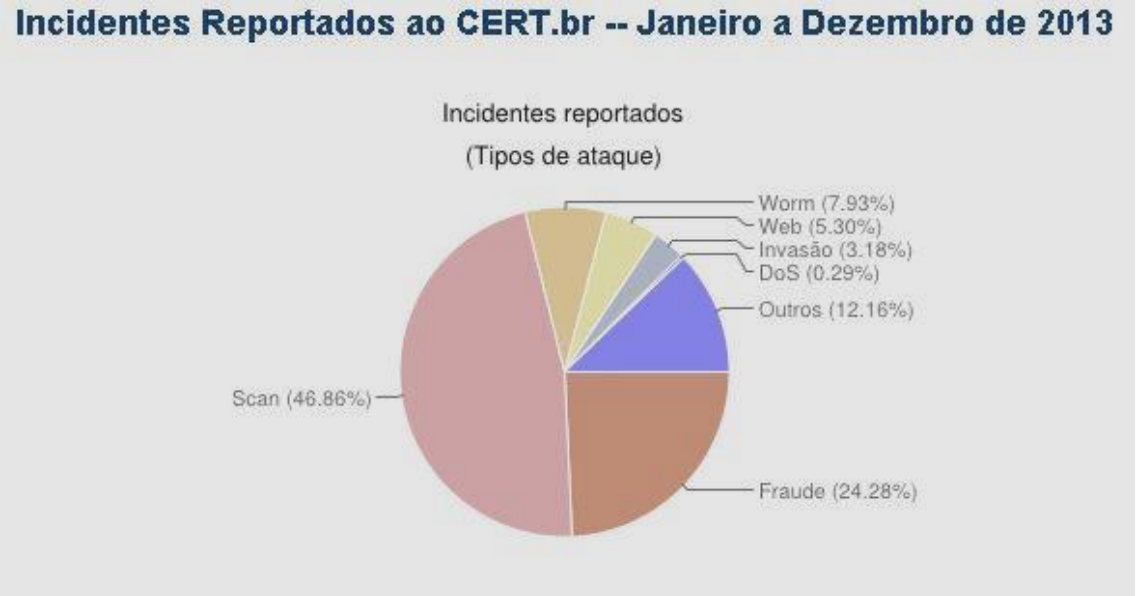


Figura 2: Tipos de incidentes reportados no ano de 2013. (CERT.BR)

2.5 Exploração de vulnerabilidades

Uma vulnerabilidade é definida como uma condição que, quando explorada por um atacante, pode resultar em uma violação de segurança (CERT.ORG, 2014). A vulnerabilidade é um ponto onde o sistema computacional é suscetível a ataques. Já a ameaça é uma violação de segurança em potencial. Esta violação em potencial pode levar a exploração de uma vulnerabilidade.

Essas vulnerabilidades são oriundas de falhas na maioria das vezes não intencionais, a origem destas pode ser:

- **Físicas:** acesso aos ativos por pessoas não autorizadas, devido à falta de controle de acesso ou violação intencional do acesso a fim de causar danos;
- ***Hardware:*** indisponibilidade no sistema ou perda de dados decorrentes a problemas no *hardware*;
- **Naturais:** comprometimento da segurança dos dados armazenados ou sistemas em função de desastres naturais;
- **Humanas:** o operador de sistema utilizar erroneamente uma função, prejudicando o funcionamento do mesmo;
- ***Software:*** falhas de programação ou configuração de *software*, abrindo brechas a serem exploradas.

Conhecer o ambiente e coletar informações sobre o alvo, se possível, sem ser notado, é o primeiro passo para a exploração bem sucedida do ambiente em que se quer obter informações. Pode-se dizer que todos os sistemas computacionais conectados à Internet estão sujeitos a ataques. O ataque, na maioria das vezes, é direcionado ao ponto onde o sistema é mais frágil, ou seja, aquele ponto que não conta com mecanismos de segurança adequados ou possui alguma falha. Nakamura e Geus (2007) definem as principais técnicas e ferramentas para obtenção de informações em: *dumpster diving* ou *trashing*, engenharia social, ataques físicos, informações livres, *packet sniffing*, *firewalking*, *port scanning* e *scanning* de vulnerabilidades.

2.5.1 *Dumpster diving* ou *trashing*

Técnica na qual o lixo é verificado em busca de informações sobre a organização, ou rede do alvo, como nomes de contas e senhas, informações pessoais e confidenciais. Uma característica importante é que se trata de uma ação legal, pois as informações são coletadas de material já descartado. Rascunhos com anotações, comprovantes de pagamentos, notas fiscais são alguns exemplos de materiais que podem ser explorados para obtenção de alguma informação.

2.5.2 Engenharia social

Segundo a Cert.br (2012), é técnica por meio da qual uma pessoa procura persuadir outra a executar determinadas ações. Uma prática de má-fé, usada por golpistas para tentar explorar a ganância, a vaidade e a boa-fé ou abusar da ingenuidade e da confiança de outras pessoas, a fim de aplicar golpes, ludibriar ou obter informações sigilosas e importantes.

2.5.3 Ataque físico

O ataque permite um acesso direto ao sistema, o que facilita as ações podendo ocorrer o roubo do próprio equipamento, ocasionando o acesso a informações privilegiadas como *e-mails*, documentos, com o objetivo de uso futuro das informações ou simplesmente danificar ou ocasionar transtorno ao sistema.

Devem ser tomadas medidas para impedir perdas, danos, furto ou comprometimento de ativos e interrupção das atividades de uma organização. O acesso físico deve ser protegido

com a criação de um perímetro de segurança física, incluindo controles de entrada física, segurança em escritórios, salas e instalações, proteção contra ameaças externas e do meio ambiente e acesso público (ISO/IEC 27002, 2005).

2.5.4 Informações livres

Segundo Nakamura e Geus (2007), diversas informações podem ser obtidas livremente, principalmente na própria internet. Consideradas como não intrusivas, pois não podem ser detectadas ou alarmadas, as técnicas incluem consultas a servidores DNS, análise de cabeçalhos de e-mail e busca de informações em redes sociais, grupos de discussão. Por meio delas, detalhes sobre sistemas, topologia e usuários podem facilmente serem obtidos.

2.5.5 *Packet Sniffing*

Interceptação de tráfego, ou *sniffing*, é uma técnica que consiste em inspecionar os dados trafegados em redes de computadores, por meio do uso de programas específicos chamados de *sniffers*. Segundo a Cert.br (2012), esta técnica pode ser utilizada de forma legítma, por administradores de redes, para detectar problemas, analisar desempenho e monitorar atividades maliciosas relativas aos computadores ou redes por eles administrados. Maliciosa, para capturar informações sensíveis, como senhas, números de cartão de crédito e o conteúdo de arquivos confidenciais que estejam trafegando por meio de conexões inseguras, ou seja, sem criptografia.

2.5.6 *Firewalking*

O *firewalking* é técnica implementada em uma ferramenta similar ao *traceroute* e pode ser utilizada para a obtenção de informações sobre uma rede protegida por um *firewall*. Essa técnica permite que pacotes passem por portas em um *gateway*, além de determinar se um pacote com várias informações de controle pode passar pelo *gateway*. Pode-se ainda mapear roteadores encontrados antes do *firewall*, (NAKAMURA; GEUS, 2007).

2.5.7 *Port Scan*

Os *port scanners* são ferramentas utilizadas para obtenção de informações referentes aos serviços que são acessíveis e definidas por meio do mapeamento das portas TCP e UDP.

De acordo com a Cert.br (2012), varredura em redes, ou *scan*, é uma técnica que consiste em efetuar buscas minuciosas em redes, com o objetivo de identificar computadores ativos e coletar informações sobre eles como, por exemplo, serviços disponibilizados e programas instalados. Com base nas informações coletadas é possível associar possíveis vulnerabilidades aos serviços disponibilizados e aos programas instalados nos computadores ativos detectados. A varredura em redes pode ser usada de forma:

Legítima: por pessoas devidamente autorizadas, para verificar a segurança de computadores e redes e, assim, tomar medidas corretivas e preventivas;

Maliciosa: por atacantes, para explorar as vulnerabilidades encontradas nos serviços disponibilizados e nos programas instalados para a execução de atividades maliciosas.

2.5.8 *Scanning* de vulnerabilidades

Por último, mas não menos importante o *scanning* de vulnerabilidades realiza diversos tipos de testes na rede, à procura de falhas de segurança, seja em protocolos, serviços, aplicativos ou sistemas operacionais. Para essa técnica são utilizados *software* chamados *scanner*s de vulnerabilidades, esses *scanners* podem analisar riscos, pela checagem de roteadores, servidores, firewalls, sistemas operacionais e outras entidades IP. Segundo Nakamura e Geus (2007) os principais riscos são:

- Compartilhamento de arquivos que não são protegidos por senhas;
- Configuração incorreta;
- *Software* desatualizado;
- Pacotes TCP que podem ter seus números de sequência adivinhados;
- *Buffer overflows* em serviços, aplicativos e no sistema operacional;
- Falhas no nível de rede do protocolo;
- Configurações de roteadores potencialmente perigosas;
- Checagem de cavalos de Tróia;
- Checagem de senhas fáceis de serem adivinhadas;
- Configurações de serviços;
- SNMP;
- Possibilidade de negação de serviço (*DoS*);
- Configuração da política dos navegadores.

A técnica *scanning* deve ser utilizada para demonstrar possíveis problemas de segurança e alertar para a necessidade de um melhor planejamento com relação aos ativos da rede. As consultorias de segurança utilizam constantemente essa técnica para justificar uma melhor proteção e, assim, vender seus serviços, aproveitando-se de uma importante funcionalidade dos *scanners*, que é a capacidade de emitir relatórios, sendo capazes de realizar a avaliação técnica dos riscos encontrados pelo *scanning*.

2.6 Importância da análise de vulnerabilidades

Em uma rede de computadores com diversos usuários, manter os ativos livres de vulnerabilidades é um trabalho minucioso para os administradores de rede. Segundo Martinelo e Bellezi (2014), dentre as dificuldades encontradas pelos responsáveis por manter a rede em segurança, estão o fato da liberdade do usuário em instalar novos *software*, desconhecimento da prevenção contra *malwares*, pois muitos através de e-mails falsos e mensagens em redes sociais instalam códigos maliciosos, a diversidade de versões de *software* e sistemas operacionais, impossibilidade dos profissionais de TI de estarem cientes de todas as vulnerabilidades descobertas, principalmente as mais recentes. Fatores estes tornam indispensável ao administrador ferramentas que o auxilie a manter o parque computacional o menos vulnerável possível. A motivação principal deste trabalho é analisar ferramentas disponíveis para atender este requisito da segurança e um realizar comparativo de resultados do uso das mesmas.

No Capítulo 3, será tratado o processo de desenvolvimento deste estudo, como os trabalhos relacionados e no Capítulo 4, a proposta deste estudo e ferramentas necessárias para a execução do mesmo.

3 TRABALHOS RELACIONADOS

Este estudo apresenta uma análise de aspectos referente à segurança em redes de computadores, com ênfase no uso de *software* de descoberta de vulnerabilidades. Para nortear os rumos de pesquisa, na literatura foram encontrados alguns estudos com escopo semelhantes.

Pasa (2013) propõe uma avaliação de ferramentas de análise de segurança como justificativa ao crescente número de ameaças que cercam ambientes corporativos, equiparando instalação, usabilidade, eficiência e resultados dos scanners de vulnerabilidades Nessus e OpenVAS. Conclui em seu estudo os principais pontos positivos e negativos de cada *software* salientando como diferencial o OpenVAS ter sua versão livre de custos.

Proposta semelhante é feita por Martinelo e Bellezi (2014) que em estudo recente abordam os principais tipos de vulnerabilidades e ataques conhecidos. Defendem o uso de *software* no auxilio de descobertas possíveis ameaças, com uma comparação de resultados do escaneamento entre Nessus e OpenVAS. Concluem como positivo o uso de ambos principalmente porque também servem de auxilio na prevenção e seus relatórios sugerem possíveis soluções aos riscos detectados.

Moreira et al. (2008) defendem a análise de riscos como processo essencial de qualquer programa de gerência de segurança da informação. Para o estudo foram utilizados os *scanners* Nessus e Languard, em conjunto com a funcionalidade de descoberta de portas do *network mapper* (Nmap), em uma rede heterogenea intencionalmente vulnerável. Tendo por objetivo encontrar medidas cabíveis e de fácil aplicação, para as diferentes situações de riscos detectadas.

Em comum entre o presente trabalho e os demais citados está a visão de que no cenário atual onde as ameaças aumentam em ritmo igual ao avanço de recursos computacionais se justifica a importância de toda ferramenta que venha de alguma forma contribuir na manutenção da segurança.

4 TRABALHO PROPOSTO

Este Capítulo descreve a proposta de desenvolvimento deste estudo, uma análise dos *software* que servem de auxílio aos administradores de redes na busca por pontos fracos suscetíveis a falhas ou ataques.

Cada vulnerabilidade implica na possibilidade de uso indevido de um sistema. Os passos necessários para explorar essa fraqueza, ou mesmo o código (programa) que pode tirar proveito da vulnerabilidade é descrito como *exploit*. Um *exploit* surge apenas quando há uma vulnerabilidade, mas podem existir vulnerabilidades para as quais não exista *exploit* (MALERBA, 2010).

Uma solução comumente aplicada envolve o uso de *scanners* de vulnerabilidade de rede. De acordo com Holm et al. (2011) um *scanner* de vulnerabilidade de rede é um dispositivo ou software que é usado para digitalizar a arquitetura de uma rede e relatar quaisquer vulnerabilidades identificadas. O procedimento normal de verificação de uma rede com uma ferramenta de avaliação de vulnerabilidade geralmente envolve três partes: digitalização em rede, varredura de vulnerabilidades e análise de vulnerabilidade.

Alguns *software* existentes que atendem ao requisito no processo de escaneamento de redes são: Retina, *Microsoft Baseline Security Analyzer* (MBSA), Core Impact, GFI Languard, QualysGuard, Nexpose, *Open Vulnerability Assessment System* (OpenVAS) e Nessus. Os três últimos citados serão analisados por este trabalho, Nessus por ser referência para as demais ferramentas, tendo citações em grande parte dos estudos do gênero. OpenVAS por ser a principal ferramenta *opensource* e Nexpose como uma nova opção *scanner* que está em constante evolução.

4.1 Nessus

O *scanner* de vulnerabilidades Nessus é um dos *softwares* mais conhecidos na área de segurança e análise de vulnerabilidades sendo um dos principais produtos do seu gênero em todo o setor de segurança e conta com o apoio de organizações profissionais de segurança da informação, como o *SysAdmin Audit, Networking and Security Institute* (SANS). Ele realiza uma varredura de portas, detectando ativos e simulando invasões para detectar possíveis vulnerabilidades. Desenvolvido pela empresa *Tenable Network Security* teve seu código aberto até o ano de 2005 e três anos depois teve sua licença modificada ganhando uma versão

comercial. Sua versão gratuita foi mantida, porém restrita ao uso doméstico (SECTOOLS, 2014). O site oficial do Nessus (2014) descreve a evolução em um breve histórico:

- 1998- Lançamento, disponível para Linux e 50 *plugins* escritos em linguagem C;
- 2000- Torna-se disponível para plataformas Windows e lançado o versão 1.0;
- 2001- Adicionado suporte a SSL;
- 2002- Número de *plugins* chega 1000 e criada a Tenable *Network Security*;
- 2006 - Suporte ao MAC OS X;
- 2008- Suporte para alvos IPv6 e número de plugins chega a 20000;
- 2011- Suporte a 20 diferentes distribuições Unix/Lunix;
- 2012- Suporte a detecção de vulnerabilidades em dispositivos móveis e 15.000 clientes;
- 2013 – O *software* completa 15 anos, com mais de 54.000 *plugins* disponíveis;

A Figura 3 trás a evolução do número de *plugins* ao longo dos anos.

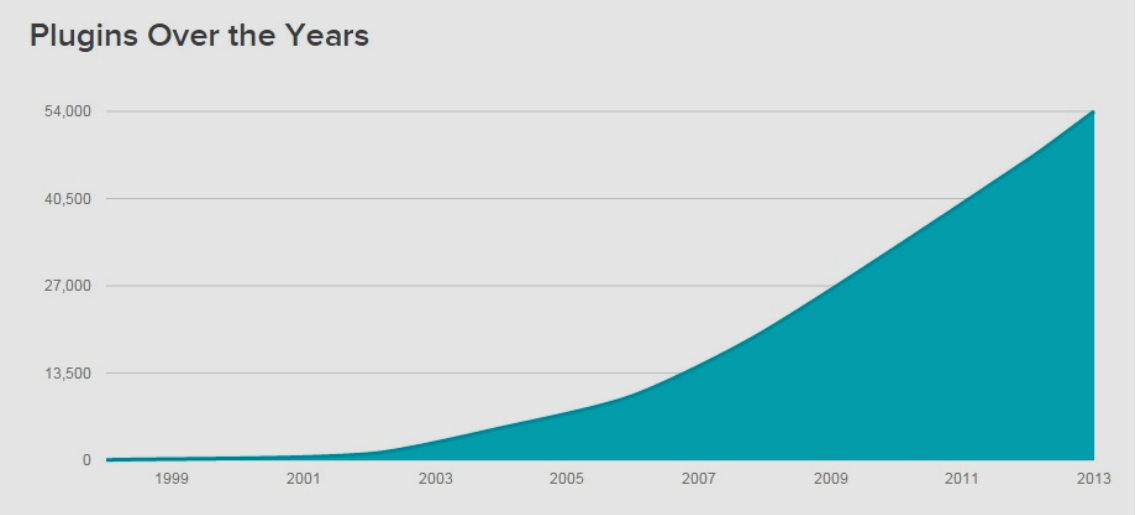


Figura 3: Evolução do número de *plugins* ao longo dos anos. (NESSUS, 2014)

4.1.1 Arquitetura

Sua arquitetura cliente-servidor oferece a flexibilidade de instalar o *scanner* (servidor) em compatibilidade de diferentes sistemas operacionais: Windows, Mac OS, Linux, FreeBSD e Solaris. Para o cliente é possível conectar-se a interface gráfica via *http*, o que significa que é possível utilizá-lo de qualquer navegador em distintas plataformas. De acordo com Martinelo e Bellezi (2014) o *scanner* Nessus utiliza *Nessus Attack Scripting Language* (NASL), linguagem criada especificamente para a criação de testes de segurança, conhecidos

por *plugins* ou *Network Vulnerability Tests* (NVT), na Figura 4 é ilustrada a visão de arquitetura Nessus.

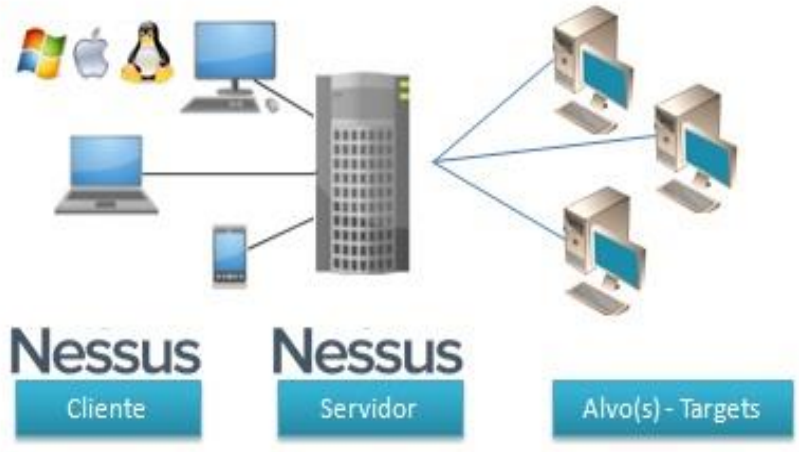


Figura 4: Visão da arquitetura Nessus.

4.1.2 Instalação

A *Tenable Network Security* após fechar o código fonte de seu produto desenvolveu versões *Professionals* pra venda as empresas. Para usuários comuns interessados apenas em aprender, desenvolveu a versão *home*, que é gratuita, mas precisa de um cadastro no site Nessus (2014) com um endereço de e-mail, através do mesmo é informada a chave de registro do *Nessus Home Feed*. Esta chave dá acesso ao download das atualizações dos plugins, parte integrante do programa. Há dois tipos de chaves:

- ***Professional Feed***: esta normalmente é usada por empresas e custa 1.200 dólares por ano, oferece suporte técnico, atualizações de vulnerabilidades em tempo real, auditoria de dados sensíveis e outros benefícios;
- ***Home Feed***: esta licença é gratuita e seu benefício são as atualizações em tempo real, é utilizada normalmente por usuários interessados em aprender sobre a ferramenta.

Existem várias maneiras de instalar o Nessus e existem várias plataformas onde isso é possível, como já foi visto no item 4.1.1 deste Capítulo. Para a instalação o desenvolvedor disponibiliza um manual completo de instalação e configurações, na plataforma Windows a instalação é muito rápida e intuitiva (PASA, 2013).

A Figura 5 ilustra um dos passos de instalação Nessus via plataforma Windows.

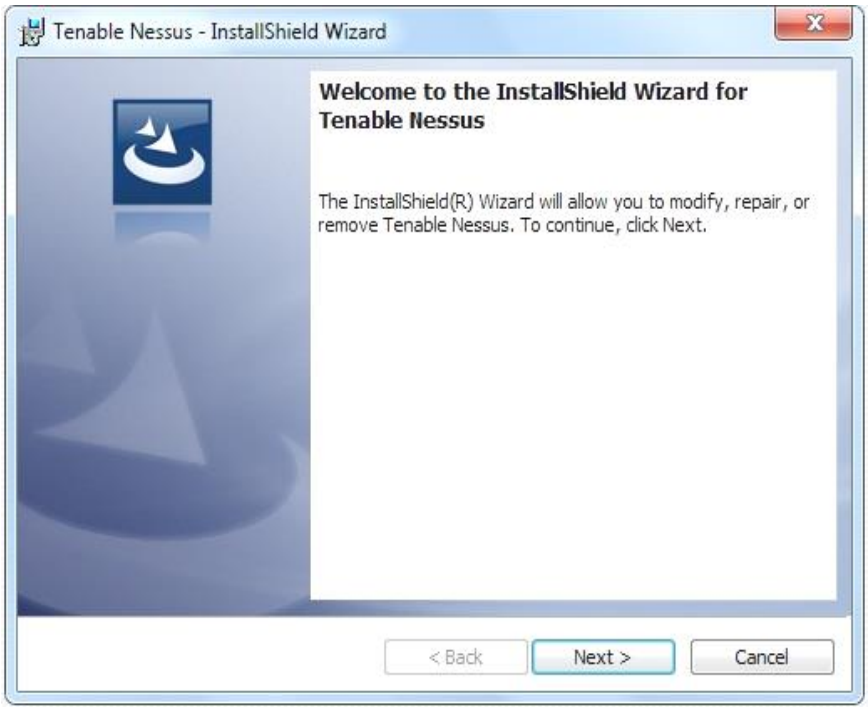


Figura 5: Exemplo de instalação Nessus em Windows**.** (NESSUS, 2014)

Logo após o processo de instalação é possível acessar a interface gráfica de qualquer *browser* para prosseguir à fase de autenticação do cliente, registro e atualização dos *plugins*. Feitos os registros e as atualizações necessárias, faz-se necessário informar o *login* e senha pré-determinados nas fases anteriores a Figura 6 ilustra o acesso ao painel *login* e senha e a Figura 7 a visão da *interface* de comandos do Nessus.

Figura 6: Tela login Nessus. (NESSUS, 2014)

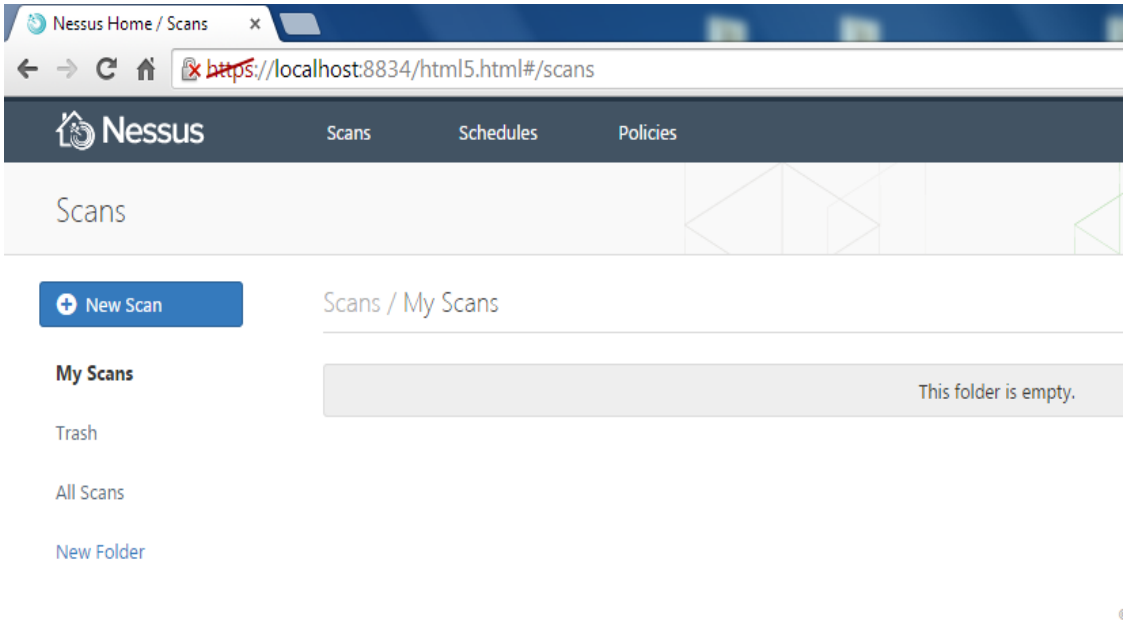


Figura 7: Tela de inicial Nessus. (NESSUS, 2014)

O funcionamento do Nessus segue uma sintaxe que exige a criação de uma política, escolha do alvo e assim iniciar verificação.

- Definir política – São as regras que serão seguidas pelo Nessus no momento de executar a varredura, por exemplo, configurações de parâmetro, *plugins* que são *scripts* responsáveis por detectar vulnerabilidades específicas, entre outros;
- Definir o alvo - Serão definidos os hosts alvos das varreduras, escanear apenas um host, simplesmente informe o IP, se deseja varrer um *range* de várias máquinas, digite o endereço inicial seguido de um traço e depois o endereço IP final.

4.2 *Open Vulnerability Assessment System* (OpenVAS)

Segundo o site oficial do OpenVAS (2014) o *software* OpenVAS é um *scanner* com a estrutura de varredura e gerenciamento de vulnerabilidades, criado a partir de uma derivação de códigos do Nessus, que teve sua licença alterada para comercial. No entanto, a linha de desenvolvimento e disponibilização pública foi continuada, mas com o nome OpenVAS e mantido sob licença *General Public License* (GPL). O projeto mantém uma alimentação pública de testes de vulnerabilidades de rede (NVTs), com uma base de mais de 35.000 testes de vulnerabilidades de acordo com dados até abril de 2014, mantendo atualizações constantes. Para Zúquete (2013) o *scanner* OpenVAS é uma ferramenta inestimável para automatizar a

descoberta de vulnerabilidades numa rede local e orientar a sua eliminação, antes que as mesmas possam ser procuradas e exploradas por atacantes ou por código automatizado.

4.2.1 Arquitetura

Possui uma arquitetura cliente-servidor conforme Figura 8, o cliente requer o inventário sobre uma máquina ou conjunto de máquinas ao servidor e este o realiza, enviando o resultado para o cliente. O resultado é um relatório que indica as vulnerabilidades detectadas. O cliente disponibiliza ao usuário uma interface utilizada para configuração das buscas e acesso aos resultados. O OpenVAS pode analisar simultaneamente várias máquinas, indicadas explicitamente ou através de uma máscara de rede (ZÚQUETE, 2013).

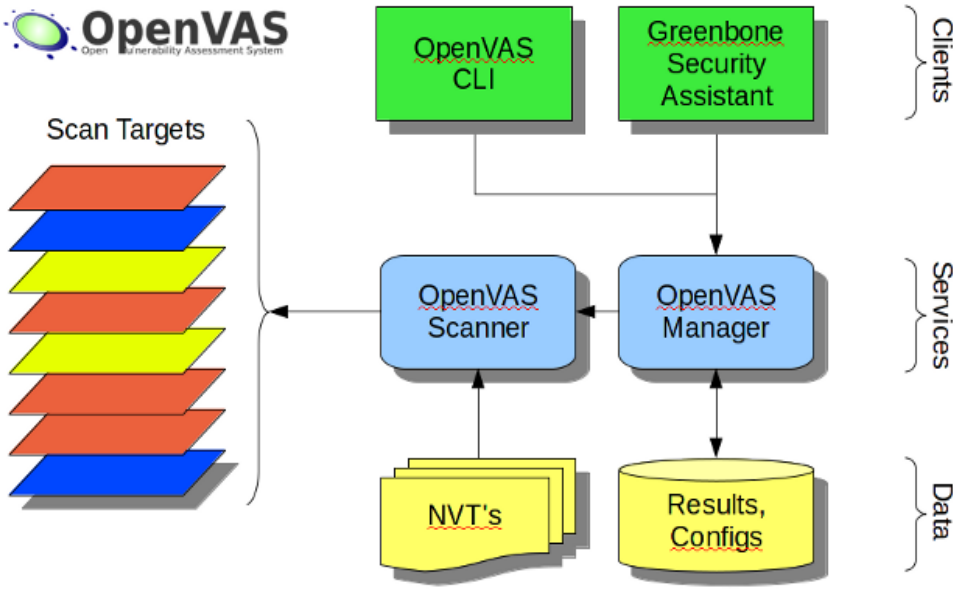


Figura 8: Arquitetura e componentes OpenVAS. (OPENVAS, 2014)

Os componentes da arquitetura são descritos conforme o Greenbone (2014) da seguinte forma:

NVT: *Network Vulnerability Tests* são testes de segurança desenvolvidos na linguagem de script do Nessus, Nessus *Attack Scripting Language* (NASL) uma linguagem inspirada na linguagem C. Tal permite uma fácil atualização da lista de problemas que a ferramenta consegue identificar, o que é fundamental para garantir uma segurança efetiva dos sistemas analisados. Este funcionamento é muito semelhante ao da atualização de assinaturas de sistemas antivírus. Estes testes são disponibilizados diariamente através do serviço openVAS NVT *Feed* que é acessado pelo comando do openvas-nvt-sync, para uma sincronização online, a forma offline requer download de um único arquivo e requer uso de

rotinas de atualização. Atualmente o OpenVAS suporta igualmente interpretação de NVTs escritos na linguagem *Open Vulnerability and Assessment Language* (OVAL). Também é possível desenvolver NVTs próprios, para suprir necessidades específicas, como testar, por exemplo, vulnerabilidades em sistemas próprios que são alterados com frequência:

OpenVAS *Scanner:* Programa que executa nos alvos os testes NVTs, é controlado pelo OpenVAS Manager via protocolo *OpenVAS Tranfer Protocol* (OTP);

OpenVAS *Manager:* Serviço principal do OpenVAS que executa e gerencia as varreduras feitas pelo OpenVAS Scanner. Os resultados das varreduras e das configurações são armazenados pelo *Manager* de forma centralizada em um banco de dados baseado em SQL. Também é responsável pelo controle de acesso e gerenciamento dos clientes aos quais se conectam no *Manager* através do protocolo *OpenVAS Management Protocol* (OMP);

OpenVAS CLI: Cliente OpenVAS com interface de linha de comando;

Greenbone Security Desktop (GSD): Cliente OpenVAS com interface desktop é possível rodar em sistemas operacionais Windows, Linux e Mac OS;

Greenbone Security Assistant (GSA): Cliente OpenVAS com interface Web onde é possível rodar em qualquer browser.

4.2.2 Instalação

Como visto no item 4.2.1 deste Capítulo é baseado na arquitetura cliente-servidor e está disponível na forma de pacotes binários para a maioria das distribuições *Linux* sendo que é possível obtê-lo em repositórios de terceiros ou diretamente do site do *OpenVAS*. O servidor é disponibilizado para plataformas Linux e FreeBSD, o site do desenvolvedor disponibiliza todos os passos da instalação para diferentes distribuições como *CentOS*, *Debian*, *Fedora*, *OpenSUSE* e *Redhat*, a Figura 9 ilustra as diferentes distribuições disponíveis.

Figura 9: Diferentes distribuições para servidor OpenVAS. (OPENVAS, 2014)

Para instalação em ambiente *Linux* é necessário a adição do repositório do OpenVAS no *sourcelist* e após instalar os diferentes pacotes via *apt-get install*, dentre os passos de instalação estão a dos *plugins* ou NVTs, os quais devido ao grande número, mais de 35.000, levam um tempo considerável. Após a conclusão da instalação inicia-se o servidor conforme a o comando descrito na Figura 10.

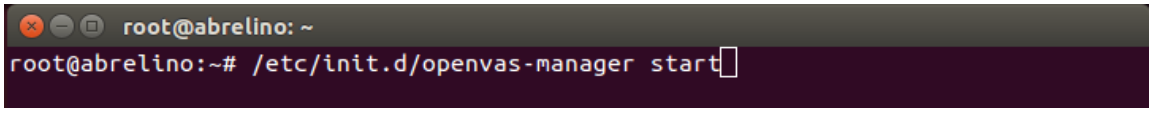


Figura 10: Comando de inicialização do servidor OpenVAS.

A Autoridade de Atribuição de Números da Internet (IANA) atribuiu oficialmente a porta TCP 9392 ao OpenVAS. Como o programa é um projeto voltado para a segurança, a conexão do cliente OpenVAS a essa porta é sempre por SSL para garantir que apenas o usuário determinado acesse os dados gerados. A interface web do cliente OpenVAS pode ser executada de qualquer *browser*, como no exemplo da Figura 11:

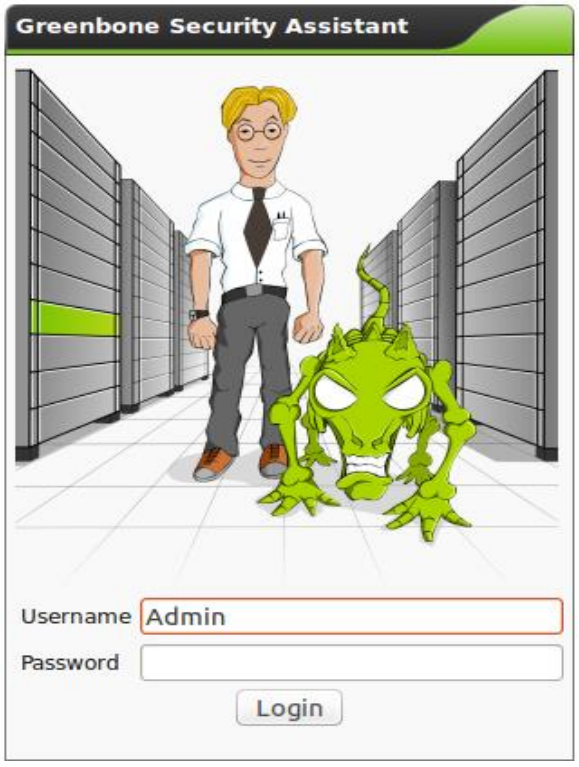


Figura 11: Tela de login do cliente OpenVAS. (OPENVAS, 2014)

Após efetuar o acesso com uma conta pré-configurada *login* e senha é iniciada a janela principal do *OpenVAS* como mostrado na Figura 12. A partir da interface gráfica é possível

elaborar as configurações necessárias para o escaneamento de um alvo. *OpenVAS* segue a mesma sintaxe do Nessus que exige a criação de uma política, escolha do alvo para então iniciar um escaneamento.

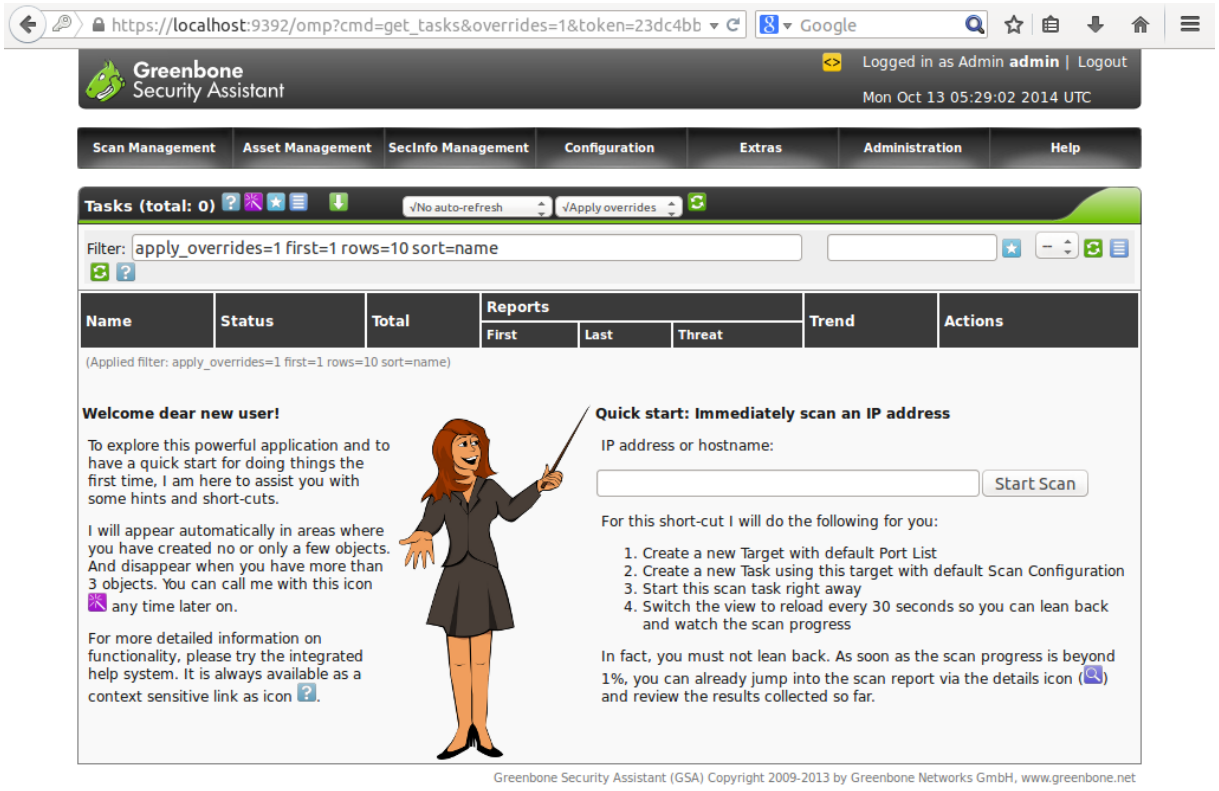


Figura 12: Interface administrativa Web cliente OpenVAS. (OPENVAS, 2014)

4.3 Nexpose

O Nexpose é um *scanner* de vulnerabilidades que visa apoio ao ciclo de gerenciamento de vulnerabilidades desde a verificação, até a análise de impacto. Produto desenvolvido pela empresa Rapid7 no ano de 2009 encontra-se em constante evolução de acordo Nexpose Install Guide (2014) que trás a revisão histórica do *software*:

2009 – Verificado, testado e atualizado seus procedimentos de instalação;

2009 – Requisitos do sistema atualizados;

2010 - Adicionado nota recomendando configuração de 64 bits;

2010 – Ampliada documentação de reinstalação;

2011 – Removidas referências para sistemas Windows 32 bits;

2011 – Corrigidas várias instruções e instalar e remover o *software*;

2012 - Nexpose 5.1 atualizada instruções de pré-instalação do Linux;

2012 - Nexpose 5.2 informações atualizadas acerca dos navegadores suportados;
2012 - Nexpose 5.3 lista atualizada dos navegadores suportados;
2012 – Nexpose 5.4 lista atualizada dos sistemas operacionais suportados;
2013 - Nexpose 5.5 adicionado seções para identificar diferentes fases de instalação;
2013 - Nexpose 5.6 capturas de tela atualizadas para mostrar nova marca;
2014 – Lista atualizada de plataformas suportadas;
2014 – Nexpose 5.11 atualizada a versão do produto.

4.3.1 Arquitetura

De forma semelhante como ocorre em *softwares* mais antigos do gênero, possui uma arquitetura cliente-servidor, permitindo ao usuário acessar sua *interface* administrativa via *browser* em diferentes plataformas. Do lado do servidor a flexibilidade fica entre ambiente Windows, Linux e aplicação a máquinas virtuais.

Figura 13: Visão da arquitetura Nexpose.

4.3.2 Instalação

O *scanner* Nexpose está disponível em diferentes versões em sua maior parte comerciais e voltadas para ambientes organizacionais, porém mantém a versão gratuita visando usuários individuais, para fins de estudos acadêmicos e testes, são elas:

- *Ultimate:* versão de recursos completa, sob licença comercial;
- *Enterprise:* destinado a grandes e médias organizações, sob licença comercial;
- *Consultant:* foco nas organizações de consultoria em segurança, sob licença comercial;

- *Express:* destinado a pequenas organizações, sob licença comercial;
- *Community:* para usuários individuais, com licença gratuita.

Para instalação da versão gratuita segue os passos do site oficial do *software* Nexpose (2014). Como primeira etapa é necessária realizar um registro que dará acesso a chave de licença enviada por *e-mail*, feito o registro será disponibilizada a área de *downloads*, uma vez concluído o *download* executa-se o instalador. Todo processo é de forma simples e intuitiva, tendo como etapa mais extensa a fase de estruturação da base de *plugins*, conforme a Figura 14, que mostra parte do processo executado.

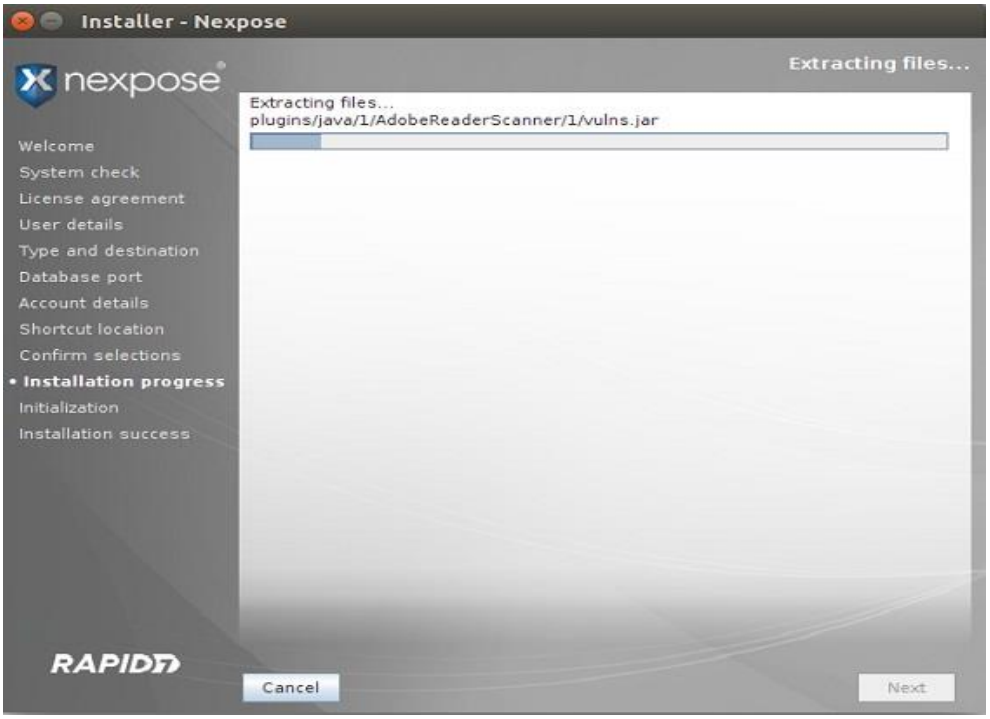


Figura 14: Etapa de instalação Nexpose. (NEXPOSE, 2014)

Após o processo de instalação concluído, o usuário pode acessar interface gráfica via *localhost* com a porta 3780 como no exemplo da Figura 15 e dar seguimento ao processo de ativação do *software*.

Figura 15: Tela de *login* Nexpose. (NEXPOSE, 2014)

A sintaxe de funcionamento segue o padrão dos *scanners* Nessus e OpenVAS visto neste Capítulo, com a configuração de alvos de escaneamento com políticas disponíveis no *software.* A visão da interface Nexpose é mostrada conforme a Figura 16.

Figura 16: Visão interface Nexpose. (NEXPOSE, 2014)

4.4 Ambiente de Instalação

Uma vez conhecidos os *software* a serem utilizados, cria-se a necessidade da elaboração de um ambiente de testes para a utilização dos mesmos. Para este trabalho os *softwares* OpenVAS e Nexpose foram instalados em ambiente Linux Ubuntu 14.04 64 bits e o *scanner* Nessus em Windows 7 Ultimate 32 bits. O *hardware* utilizado foi o mesmo para os três *softwares*, processador Pentium Dual-Core 2.10 GHz e memória RAM 4GB. A Figura 17 faz o resumo do tipo instalação utilizada e as versões utilizadas.

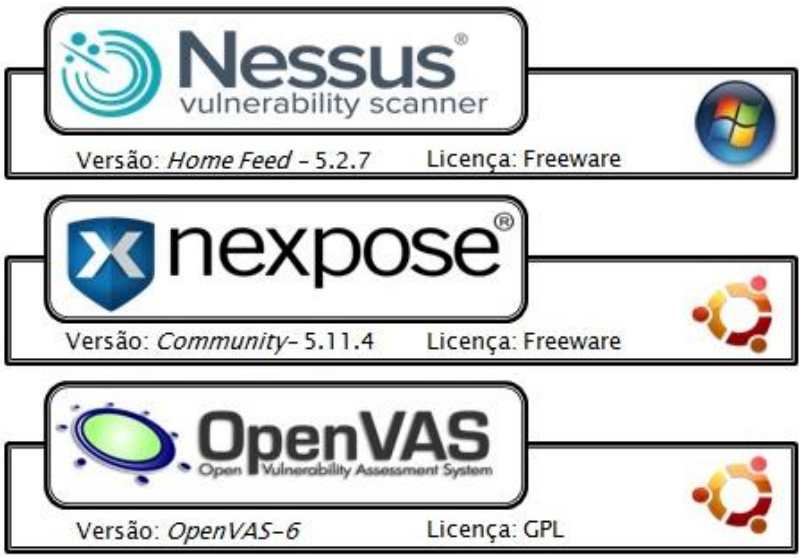


Figura 17: Visão instalação dos *software*.

5 TESTES E RESULTADOS

Neste capítulo foi projetado o cenário no qual serão realizados os testes. Para o cenário de testes foram utilizadas duas máquinas alvos, simulando um ambiente de rede *Local Area Network* (LAN) em âmbito doméstico ao qual não estariam expostos a grandes vulnerabilidades, gerando assim uma expectativa para que tipos de resultados fossem encontrados pelos *scanners* de rede.

Para o alvo 1 esta um *host* com sistema operacional Windows XP, o qual esta desde abril deste ano sem suporte deixando de receber assim as atualizações automáticas que ajudam a proteger o computador, devendo então ser um ponto de alguma notificação de fragilidade por parte dos *scanners*, o alvo 2 com sistema operacional Windows 7 Ultimate, devidamente atualizado leva a uma perspectiva de *host* mais seguro que o anterior. A Figura 18 ilustra o ambiente proposto.

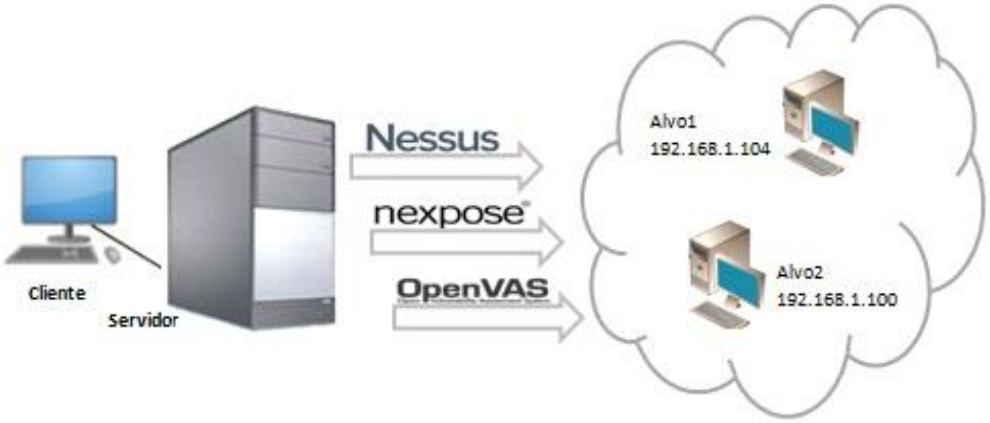


Figura 18: Ambiente simulado de testes.

Para o ambiente proposto foram configuradas políticas padrões de varredura presentes em cada *software* para um maior equilíbrio entre os resultados reportados, a primeira análise feita durante o escaneamento foi o nível de uso do processador como fator de importância para descrever o desempenho.

Conforme os resultados obtidos, observou que o *software* Nexpose exigiu mais do *hardware* com altos picos de uso do recurso durante o processo de escaneamento dos alvos, tornando em alguns momentos o desempenho lento. Verificou-se também que o OpenVAS fez uso moderado de recurso com alguns picos mais severos porém sem comprometer o desenvolvimento durante o escaneamento.

O *scanner* Nessus teve média de uso de recurso normal em todo processo de escaneamento. A Figura 19 ilustra parte processo de escaneamento com o Nexpose. Na sequência a Figura 20 ilustra o processo para o *scanner* OpenVAS assim como a Figura 21 trás para o Nessus.

Figura 19: Uso de recurso com Nexpose.

Figura 20: Uso de recurso com OpenVAS.

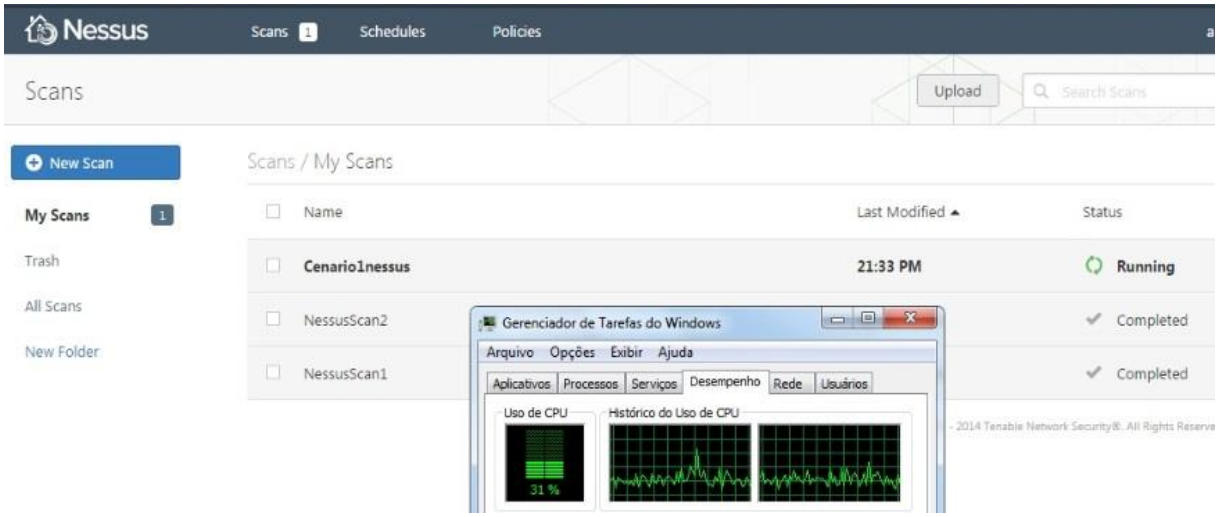


Figura 21: Uso de recurso com Nessus.

Na sequência finalizado o processo de escaneamento, encontram-se os cenários quantitativos dos resultados obtidos por cada *scanner*, cada qual com interpretações e forma próprias de demonstrar. O *software* Nessus não detectou vulnerabilidades no alvo 2, em contra-partida detectou 4 situações de risco no alvo 1, o tempo total de escaneamento no ambiente com dois alvos foi em torno de 15 minutos. A Figura 22 mostra os resultados obtidos em um dos alvos após o escaneamento. A Figura 23 mostra parte do relatório gerado.

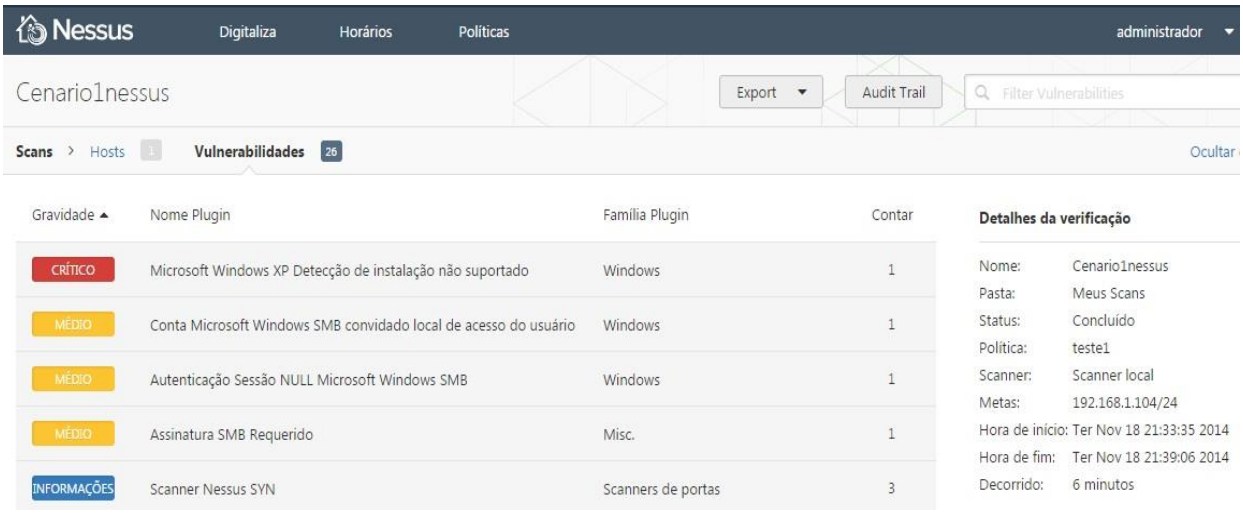


Figura 22: Resumo de escaneamento Nessus.

Table Of Contents

Vulnerabilities By Host

- 192.168.1.104

Vulnerabilities By Host

[-] Collapse All

[+] Expand All

192.168.1.104

Scan Information

Start time:	Tue Nov 18 21:33:36 2014
End time:	Tue Nov 18 21:37:02 2014

Host Information

Netbios Name:	BITENCOURT
IP:	192.168.1.104
MAC Address:	00:11:5b:d5:78:35
OS:	Microsoft Windows XP Service Pack 2, Microsoft Windows XP Service Pack 3

Results Summary

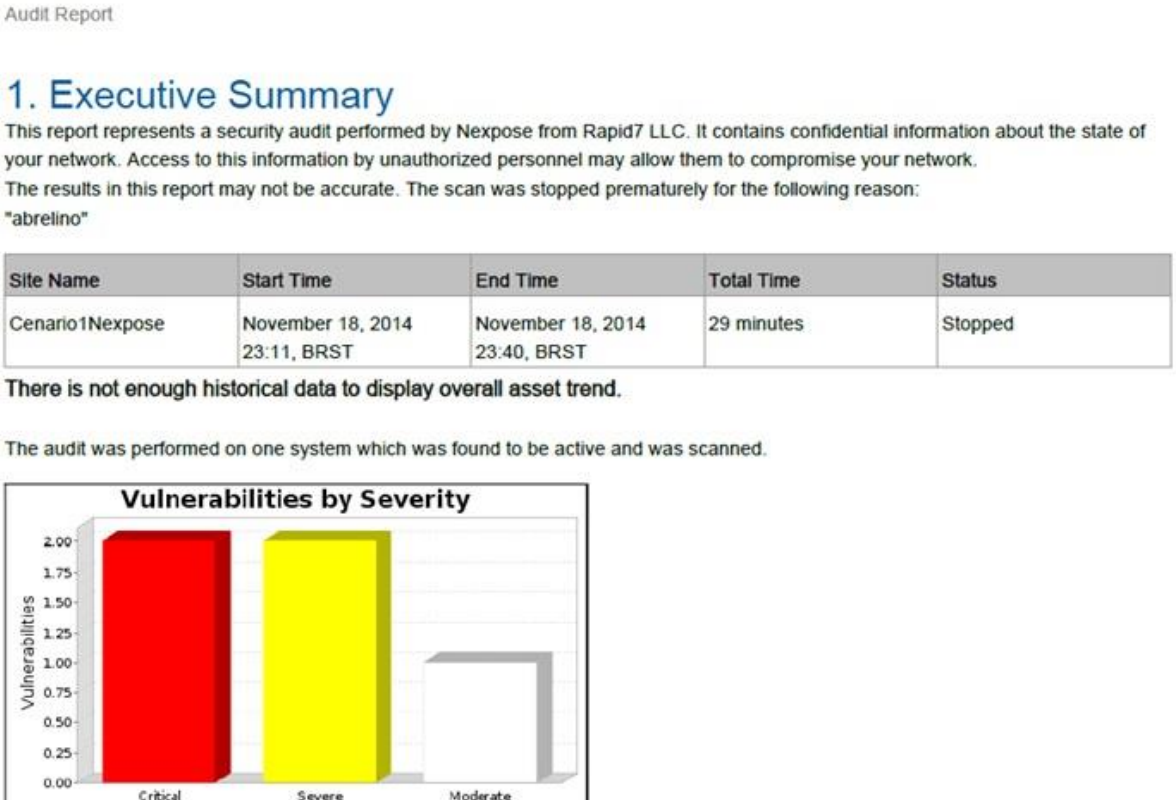
Critical	High	Medium	Low	Info	Total
1	0	3	0	25	29

Figura 23: Exemplo de relatório Nessus.

Para o *software* Nexpose, assim como o *scanner* Nessus, não detectou nenhuma nível risco para o alvo 2 com Windows 7 Ultimate, para o alvo 1 atribui risco alto para duas situações, risco médio para 2 ocorrências e baixo risco em uma, num total de 5 vulnerabilidades. O tempo somado de escaneamento para os dois *hosts* foi de 40 minutos. A Figura 24 mostra o resumo do escaneamento de um alvo 1 com Nexpose.

Figura 24: Resumo de verificação Nexpose.

A Figura 25 é trás um trecho do relatório gerado pelo Nexpose após o escaneamento.



Audit Report

1. Executive Summary

This report represents a security audit performed by Nexpose from Rapid7 LLC. It contains confidential information about the state of your network. Access to this information by unauthorized personnel may allow them to compromise your network.
The results in this report may not be accurate. The scan was stopped prematurely for the following reason:
"abrelino"

Site Name	Start Time	End Time	Total Time	Status
Cenario1Nexpose	November 18, 2014 23:11, BRST	November 18, 2014 23:40, BRST	29 minutes	Stopped

There is not enough historical data to display overall asset trend.

The audit was performed on one system which was found to be active and was scanned.

Figura25: Amostra do relatório Nexpose.

No cenário com o uso do *scanner* OpenVAS, foi o único que identificou na verificação uma vulnerabilidade para o alvo 2 sendo uma de nível médio, já para o alvo 1 reportou 10 notificações porém de baixa gravidade, com tempo total de escaneamento de 21 minutos.

A Figura 26 mostra o resumo final do escaneamento com os detalhes do alvo e resultados encontrados. Na sequência a Figura 27 trás um trecho do relatório gerado pelo escaneamento.

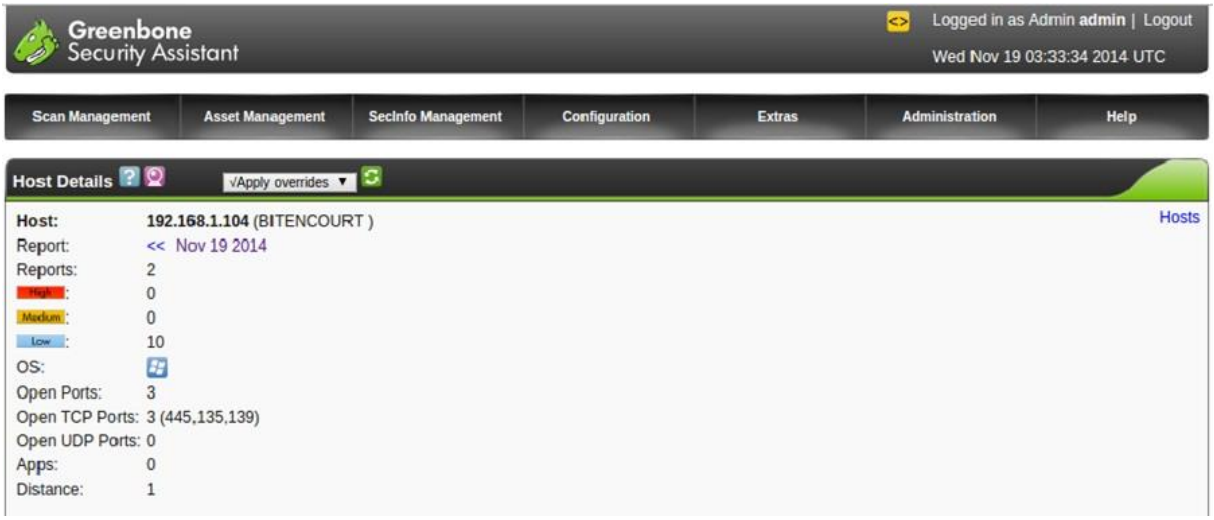
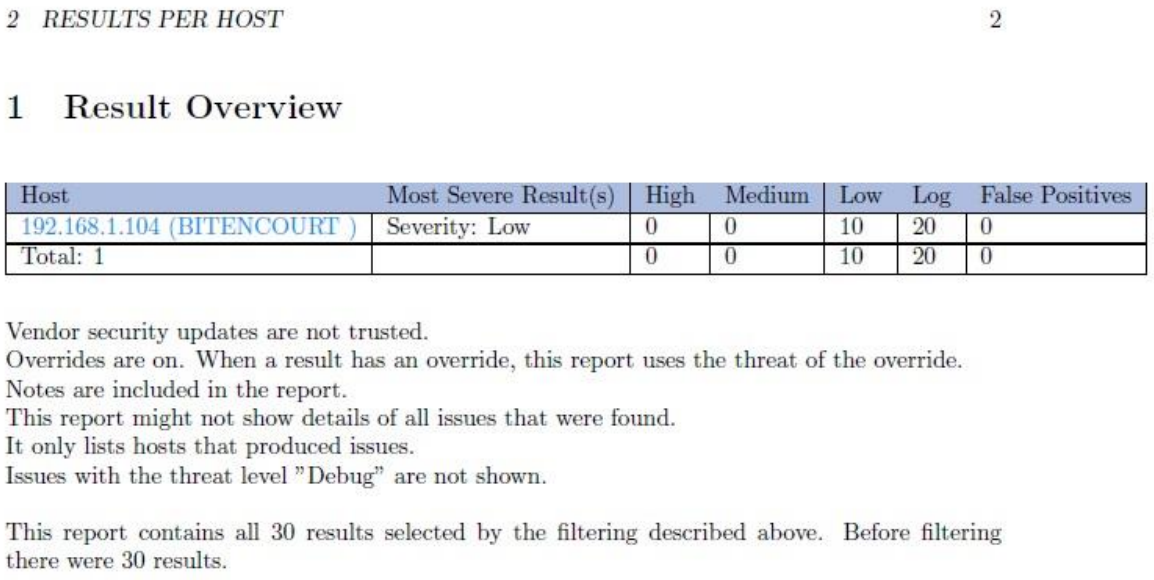


Figura 26: Resumo de verificação OpenVAS.



2 *RESULTS PER HOST* 2

1 Result Overview

Host	Most Severe Result(s)	High	Medium	Low	Log	False Positives
192.168.1.104 (BITENCOURT)	Severity: Low	0	0	10	20	0
Total: 1		0	0	10	20	0

Vendor security updates are not trusted.
Overrides are on. When a result has an override, this report uses the threat of the override.
Notes are included in the report.
This report might not show details of all issues that were found.
It only lists hosts that produced issues.
Issues with the threat level "Debug" are not shown.

This report contains all 30 results selected by the filtering described above. Before filtering there were 30 results.

Figura 27: Amostra relatório de verificação OpenVAS.

Dentro dos resultados aguardados deste cenário de teste previa-se uma menção ao alvo 1 pelo sistema operacional sem suporte a atualizações de segurança, essa notificação foi abordada de forma direta por um *scanner*, o Nessus considerou essa vulnerabilidade de alta gravidade, de acordo com os dados do relatório. O *Common Vulnerability Scoring System* (CVSS) que mede o escore de gravidade da vulnerabilidade atribui nível máximo de 10.0, e a solução apontada foi à atualização do sistema operacional para uma versão com suporte.

O *scanner* OpenVAS notificou o não suporte ao sistema operacional Windows XP como vulnerabilidade de baixo risco. Para o alvo 2 detectou uma vulnerabilidade de nível

médio conforme relatório o CVSS foi de 2.6 com descrição que o *host* remoto respondeu com um *timestamp* TCP. Esta pode ser usada para aproximar o tempo de atividade do *host* remoto, potencialmente ajudando em novos ataques. Além disso, alguns sistemas de operação podem ser impressões digitais com base no comportamento de as datas e horas de TCP.

O *scanner* Nexpose atribuiu nível crítico a ocorrência do serviço *Common Internet File System* (CIFS) com CVSS 8.0, permite que os usuários não autenticados para se conectem ao serviço com acesso limitado. A solução reportada foi à desativação da conta de convidado no painel de controle de usuários. A segunda vulnerabilidade descrita como crítica é uma variação da anterior também envolvendo o serviço CIFS.

A Figura 28 faz um resumo de resultados do cenário de testes, com a relação do número de vulnerabilidades encontradas em todos os níveis de gravidades e o tempo de escaneamento.

Figura 28: Resumo de resultados.

6 CONSIDERAÇÕES FINAIS

Diante da proposta de uma análise de *scanners* de vulnerabilidades, os três exemplares de *software* utilizados, foram capazes de realizar a escaneamento do cenário de testes proposto com êxito. Foram abordadas características desde a instalação, até os resultados reportados. Com relação à instalação propositalmente foram instalados em plataformas diferentes no caso do Nessus em Windows, Nexpose e OpenVAS em Linux, de forma a justificar a flexibilidade dos mesmos se tornando opção de adeptos de diferentes sistemas operacionais. Cabe salientar que as instalações se deram sem complicações, dentre as etapas mais demoradas a em que são verificação da base de dados de *plugins* disponíveis, neste caso o maior tempo foi do Nessus por possuir mais de 54.000.

Em referência a utilização de recursos no processo de escaneamento o desempenho de Nessus e OpenVAS foi de nível satisfatório, em contra-partida o *scanner* Nexpose em vários momentos fez uso extremo de recursos durante a atividade de varredura, mostrando a necessidade de ser instalado em *hardware* que supere os requisitos mínimos de instalação.

Com interfaces de fácil compreensão, a usabilidade é um ponto positivo de todos os *software*, com configuração dos alvos de escaneamento em poucos passos, que faz com que o usuário de qualquer nível de experiência não encontre grandes dificuldades.

Com os resultados obtidos e comparados a visão de que mesmo sendo semelhantes em arquitetura e métodos de varreduras, os *software* possuem características diferentes na forma de interpretar tipos de vulnerabilidades, tanto que não houve nenhuma vulnerabilidade em comum entre os três *scanners*.

O Nessus com o menor tempo de escaneamento detectou 3 vulnerabilidades, sendo uma delas interpretada com de alto risco, o Nexpose apontou maior número de vulnerabilidades de alto risco, porém levou o maior tempo de verificação e o *scanner* OpenVAS fez um tempo total de escaneamento intermediário, não detectou nenhuma vulnerabilidade com risco crítico, sendo ao total 10 de nível baixo e uma de nível médio. Esses resultados devem levar em consideração o uso de versões gratuitas para os *software* proprietários Nessus e Nexpose.

Dentro do escopo deste estudo, os *software* analisados se mostraram aptos ao que se propõe com descobertas de possíveis pontos vulneráveis mesmo em um cenário de teoricamente sem grandes riscos de segurança, ambos assim se credenciando como boas opções de uso para a técnica *scanning* de vulnerabilidades.

Este trabalho teve por objetivo justificar o uso de *scanners* de vulnerabilidades como forma de contribuir para aumentar o nível de segurança de um ambiente computacional, agregando suas funcionalidades aos administradores de rede, facilitando e automatizando testes de segurança.

Visto que tentativas de invasão de segurança e incidentes decorrentes destes são uma constante, realizar estudos e pesquisa sobre métodos de segurança será sempre viável e benéfica. Como sugestão para trabalhos futuros, fica a de uma análise detalhada de vulnerabilidades encontradas, buscando uma ação proativa de correção ou eliminação de problemas.

7 REFERÊNCIAS BIBLIOGRAFICAS

CERT.BR – **Cartilha de segurança para internet – CERT.br**, 2 ed, Comitê Gestor da Internet no Brasil, São Paulo, 2012.

CERT.BR – Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil. **Incidentes Reportados ao CERT.br – de 1999 a Dezembro de 2013**. Disponível em: < http://www.cert.br/stats/incidentes/ > Acesso em: 27 ago. 2014.

CERT.ORG – Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança – Centro de Coordenação – **Análise de vulnerabilidades**. Disponível em: < http://www.cert.org/vulnerability-analysis/> Acesso em: 25 set. 2014.

GREENBONE - **Greenbone Networks**, Disponível em: <http://www.greenbone.net/technology/tool_architecture.html> Acesso em: 14 out. 2014.

HOLM, H.; SOMMESTAD, T.; ALMROTH, J., PERSSON, M. **A quantative Evaluation of Vulnerability Scanning.** Royal Institute of Technology, Sweden, 2011.

ISO/IEC 27002. ABNT NBR ISSO/IEC 27002:2005 – **Tecnologia da Informação – Técnicas de Segurança – Código de prática para a gestão de segurança da informação.** Associação Brasileira de Normas Técnicas – Rio de Janeiro: ABNT, 2005.

KUBOTA, L. C. **Tecnologias da informação e comunicação: competência, políticas e tendências** – Brasília: Ipea, 2012.

KUROSE, J.; ROSS, K. **Redes de Computadores e a Internet: uma abordagem top-down**, 6 ed., São Paulo: Addison Wesley, 2013.

MALERBA, C. **Vulnerabilidades e Exploits: técnicas, detecção e prevenção.** Universidade federal do Rio Grande do Sul – Porto Alegre, 2010.

MARTINELO, C. G.; BELLEZI, M. A. **Análise de Vulnerabilidades com OpenVAS e Nessus**, Revista T.I.S vol. 3, São Carlos, 2014.

MOREIRA, J.; TEIXEIRA, C.; TAVARES, C.; VERBENA, M., QUINTÃO, P. **Scanners de Vulnerabilidades Aplicados a Ambientes Organizacionais**, Revista Eletrônica Faculdade Metodista Granbery, N. 5, 2008.

NAKAMURA, E.; GEUS, P. **Segurança de Redes em ambientes cooperativos** - São Paulo : Novatec Editora, 2007.

NESSUS - **Tenable Network Security.** Disponível em:
< http://www.tenable.com/products/nessus/> Acesso em: 09 out. 2014.

NEXPOSE – **Rapid7 – IT Security & Analytics.** Disponível em:
< http://www.rapid7.com/products/nexpose/> Acesso em 03 nov. 2014.

NEXPOSE INSTALL GUIDE – **Nexpose, Sofware Installation & Quick-Start Guide**, Disponível em: < https://community.rapid7.com/docs/DOC-1385> Acesso em 03 nov. 2014.

OPENVAS - OpenVAS *Open Vulnerability Assessment System* - Disponível em:
< http://www.openvas.org/> Acesso em: 14 out. 2014.

PASA, T.; **Avaliação de Ferramentas de Análise de Segurança: Nessus – OpenVAS**, Faculdade de Tecnologia Senac, Pelotas, 2013.

STALLINGS, W. **Network Security Essentials: Applications and Standards**. 4 ed., Estados Unidos: Prentice Hall, 2010.

SKOUDIS, E.; LISTON, T. **Counter Hack Reloaded: A Step-by-step Guide to Computer Attcaks and Efffective Defenses**, 2 ed., Englewood Cliffs: Printice Hall, 2006.

TANENBAUM, A.; WETHEREALL, D. **Redes de Computadores,** 5 ed. Rio de Janeiro: Pearson Education Brasil, 2011.

ZÚQUETE, A.; **Segurança em Redes Informáticas** – 4 ed.: Editora FCA, Lisboa, 2013.