Automatização da identificação de vulnerabilidades

André Vaz de Jesus, 2018071303
Iscac17286@alumni.iscac.pt
Pedro David de Jesus Almeida, 2018069272
Iscac17282@alumni.iscac.pt

Coimbra Business School – ISCAC
Licenciatura em Informática de Gestão
Segurança Informática

Resumo

Nos dias que correm, as tecnologias de informação representam um dos principais fatores de sucesso de qualquer organização, não só facilitando a comunicação, como também implementando a grande maioria dos principais processos responsáveis pelo funcionamento da sua atividade em si. No entanto, ao confiar os seus dados a certos softwares, a sua segurança poderá revelar-se comprometida se não for realizada previamente uma devida auditoria e análise ao software utilizado. Assim, esta apresenta-se como sendo a principal motivação para o presente trabalho, pretendendo testar softwares de identificação de vulnerabilidades em servidores web, como forma de garantir a segurança da informação partilhada.

Para o alcance deste objetivo, testaremos uma máquina virtual Metasploitable versão 2, intencionalmente vulnerável, com recurso às ferramentas OpenVAS e Nessus para a devida identificação e correção de vulnerabilidades propícias a falhas de segurança.

Os resultados obtidos servirão como referência nos conhecimentos associados à segurança informática, podendo certamente vir a ser requisitados no mercado de trabalho futuro.

1 Introdução

As tecnologias de informação sem o qual a nossa sociedade já não consegue sobreviver. Estas permitem-nos comunicar e partilhar todo o tipo de dados de forma simples e dinâmica com apenas um clique. No entanto, o facto dos nossos dados serem partilhados online pode traduzir-se em novos riscos de segurança, que devem ser imediatamente identificados e devidamente mitigados. Por essa mesma razão, paralelo à evolução das tecnologias de informação, tem-se observado também uma notável evolução no desenvolvimento de estratégias no âmbito da segurança, numa luta constante contra os riscos associados a estas tecnologias.

Para atingir o objetivo proposto, este trabalho de investigação baseará o seu trabalho de auditoria na análise e comparação de duas das principais ferramentas de análise automatizada de vulnerabilidades, OpenVAS e Nessus, tendo por base a distribuição Kali Linux do sistema operativo Linux. Centrado neste ambiente, o principal foco será a análise de uma máquina virtual intencionalmente vulnerável, com o objetivo de perceber quais as vulnerabilidades encontradas no mesmo, observar a hierarquização das mesmas e entender as soluções propostas pelo software.

Assim, um dos principais resultados a obter neste trabalho centra-se na obtenção de uma classificação de segurança para o ambiente testado, podendo assim perceber quais os maiores riscos associados a servidores web, e como os combater.

Este documento tem como propósito servir de relatório final da disciplina de Segurança Informática, ministrada no terceiro ano da licenciatura em Informática de Gestão no Instituto Superior de Contabilidade e Administração de Coimbra, abordando a temática de automatização da identificação de vulnerabilidades num servidor web.

2 Conceitos fundamentais

Neste capítulo serão detalhados alguns dos conceitos fundamentais associados a uma plena compreensão da auditoria de vulnerabilidades efetuada.

2.1 Kali Linux

Kali Linux é uma distribuição GNU/Linux baseada no sistema operativo Debian. Esta distribuição é direcionada essencialmente para a auditoria e segurança informática. Lançado oficialmente a 13 de março de 2013 pela *Offensive Security Ltd*, o Kali Linux é uma distribuição "rolling-release", o que significa que se encontra em constante desenvolvimento através de updates esporádicos.

Esta conceituada distribuição dispõe de inúmeros softwares pré-instalados, incluindo o Nmap, Wireshark, John the Ripper, Aircraft-ng e, para futura análise, o OpenVAS.

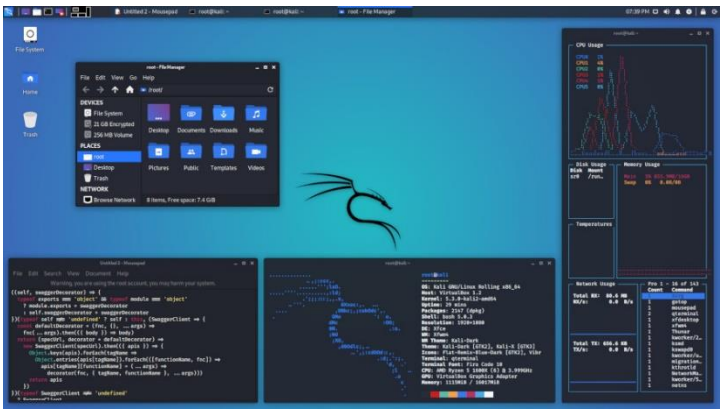

Figura 1 - Sistema Operativo Kali Linux

2.2 OpenVAS

OpenVAS significa *Open Vulnerability Assessment System* que pode ser traduzido para Sistema Aberto de Avaliação de Vulnerabilidade, e corresponde a uma framework de vários serviços e ferramentas que oferece uma solução de gestão e limpeza de vulnerabilidades.

a) Arquitetura

O núcleo de sua arquitetura orientada a serviços com segurança SSL é o Scanner OpenVAS. O scanner executa os testes de vulnerabilidade de rede, que são servidos com atualizações diárias, fornecidas pelo Feed NVT OpenVAS ou através de um serviço de feed comercial.

O Manager OpenVAS é o serviço central que consolida a limpeza de vulnerabilidades numa solução completa de gestão de vulnerabilidades. O manager controla o scanner através do OTP (OpenVAS Transfer

Protocol) e oferece o OMP (OpenaVAS Management Protocol), um protocolo sem estado baseado em XML.

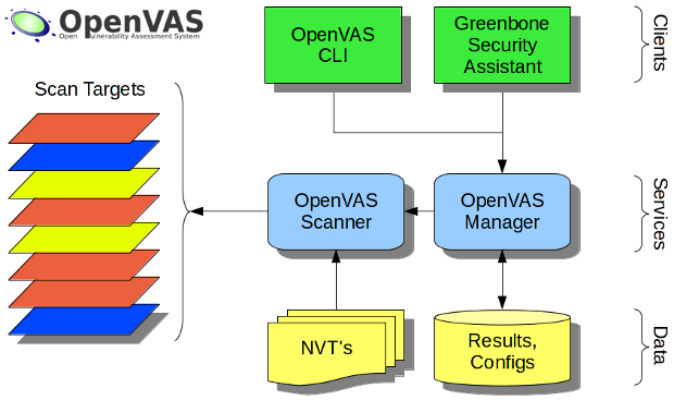

Figura 2 - Arquitetura OpenVAS

2.3 Nessus

O software Nessus é uma ferramenta de análise de vulnerabilidades desenvolvida pela organização Tenable, Inc. O projeto Nessus teve o seu início em 1998, pelas mãos de Renaud Deraison, com o objetivo de proporcionar à comunidade de utilizadores da internet um scanner de segurança remoto, totalmente grátis. Mais tarde, a 5 de Outubro de 2005, a empresa Tenable Network Security, co-fundada por Renaud Deraison, o Nessus 3 para um software de licença proprietária (closed-source).

A partir do projeto Nessus, surgiram mais recentes projetos *open-source* baseados no mesmo conceito, tais como o OpenVAS e o porjeto Greenbone Sustainable Resilience.

a) Arquitetura

Relativamente à arquitetura do software, o Nessus é baseado no modelo Cliente-Servidor. O servidor Nessus, nessusd, é o responsável pela realização dos testes de vulnerabilidade propriamente ditos. O servidor está à escuta para possíveis conexões clientes que configurarão e darão início aos scans desejados. Para poderem realizar scans de vulnerabilidades, os clientes terão primeiro de se autenticar no servidor, facilitando assim a administração das instalações do software.

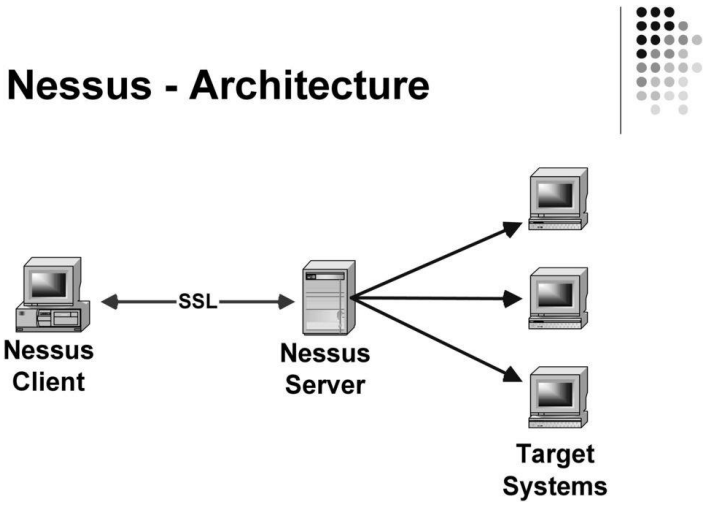

Figura 3 - Arquitetura Nessus

2.4 Vulnerabilidades

Uma vulnerabilidade consiste numa fraqueza informática que permite um hacker reduza a garantia da informação de um sistema. Desta forma, podemos assumir que uma vulnerabilidade é a interseção de três ocorrências: uma falha do sistema, o acesso de um hacker à falha existente, e a capacidade do mesmo de explorar essa falha. Estas ditas falhas podem originadas de diversas formas, de entre as quais:

- **Físicas:** acesso a ativos por pessoas não autorizadas, devido à falta de controlo de acesso.
- **Hardware:** falhas no hardware que resultam em indisponibilidade no sistema ou perda de dados.
- **Naturais:** desastres naturais que comprometem a segurança dos dados armazenados
- **Humanas:** uso errado de uma função por um operador de sistema que resulta no mau funcionamento do mesmo ou perda de informações
- **Software:** falha de programação, deixando brechas suscetíveis a serem exploradas

Muitas vulnerabilidades são exploradas ou criadas através de softwares desenvolvidos com essa única finalidade. A estes softwares chamamos de Malwares, ou malicious software, um programa com efeitos indesejados, desenhado para causar dano no ambiente onde se encontra.

Podemos analisar os vários tipos de malwares atuais, bem como as suas causas e efeitos na Tabela 1:

Códigos Maliciosos	*Virus*	*Worm*	*Bot*	*Trojan*	*Spyware*	*Backdoor*	*Rootkit*
Como é obtido:							
Recebido automaticamente pela rede		□	□				
Recebido por *e-mail*	□	□	□	□	□		
Baixado de *sites* na Internet	□	□	□	□	□		
Compartilhamento de arquivos	□	□	□	□	□		
Uso de mídias removíveis infectadas	□	□	□	□	□		
Redes sociais	□	□	□	□	□		
Mensagens instantâneas	□	□	□	□	□		
Inserido por um invasor		□	□	□	□	□	□
Ação de outro código malicioso		□	□	□	□	□	□
Como ocorre a instalação:							
Execução de um arquivo infectado	□						
Execução explícita do código malicioso		□	□	□	□		
Via execução de outro código malicioso						□	□
Exploração de vulnerabilidades		□	□			□	□
Como se propaga:							
Insere cópia de si próprio em arquivos	□						
Envia cópia de si próprio automaticamente pela rede		□	□				
Envia cópia de si próprio automaticamente por *e-mail*		□	□				
Não se propaga				□	□	□	□
Ações maliciosas mais comuns:							
Altera e/ou remove arquivos	□			□			□
Consome grande quantidade de recursos		□	□				
Furta informações sensíveis			□	□	□		
Instala outros códigos maliciosos		□	□	□			□
Possibilita o retorno do invasor						□	□
Envia *spam* e *phishing*			□				
Desfere ataques na Internet		□	□				
Procura se manter escondido	□				□	□	□

Tabela 1 - Exemplos de Malwares

Metasploitable

O software Metasploitable consiste numa máquina virtual Linux com sistema operacional Ubuntu, intencionalmente vulnerável, utilizada para testar ferramentas de segurança, ou simplesmente treinar a prática de penetração de segurança.

No nosso projeto utilizámos a versão 2 do software que serviu de alvo num processo de enumeração, que consiste num scan de portos, onde o servidor é examinado em busca de portos TCP e UDP abertos, e num processo de fingerprinting, onde são identificados os serviços conectados aos portos encontrados. Para a realização deste processo, utilizámos a ferramenta NMap (Network Mapper). Além deste processo, esta máquina foi igualmente usada como alvo de scan para os softwares de identificação de vulnerabilidades.

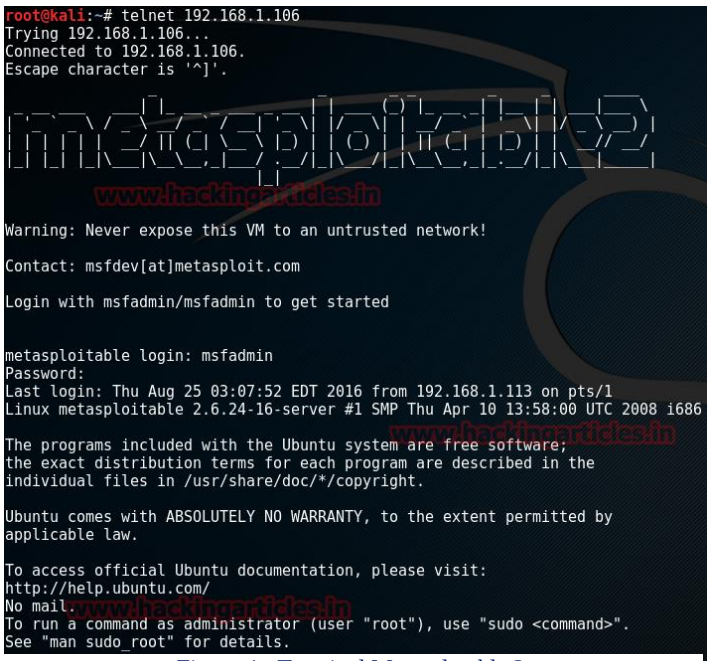

Figura 4 - Terminal Metasplotable 2

3 Arquitetura de testes

Para realização deste projeto surgiu a necessidade de instalar a máquina virtual Metasploitable 2 que contém a máquina host vulnerável, host este que será o alvo dos nossos varrimentos. Antes de começarmos a configurar as análises de vulnerabilidades é necessário instalar os softwares dedicados ao mesmo, nomeadamente o OpenVAS e Nessus

a) OpenVAS

Para instalar OpenVAS e as suas dependências corremos o seguinte comando:

```
sudo apt-get update && apt-get install openvas
```

O próximo passo foi executar o processo de configuração que esculpiu o OpenVAS e transferiu um grande número de testes de vulnerabilidade de rede (NVTs), processo este efetivamente longo devido ao elevado número de NVTs (+50.000):

```
sudo gvm-setup
```

Após término deste procedimento, a ferramenta ficou então disponível para usufruição dos seus serviços. Para inicializar estes serviços recorreu-se ao seguinte comando:

```
sudo gvm-start
```

Todos os processos necessários para arranque do OpenVAS são iniciados e a interface da web é disponibilizada para correr localmente no porto 9392, sendo possível acessar à mesma através do link https://localhost:9392

b) Nessus

Para instalar o Nessus realizámos a transferência do ficheiro para Debian/Kali Linux, o ambiente de trabalho em questão, a partir da web. De seguida recorremos ao seguinte comando para instalar o respetivo software:

```
sudo dpkg -i Nessus-8.14.0-debian6_amd64.deb
```

Depois deste procedimento finalizar, foi necessário iniciar o serviço:

```
sudo systemctl start nessusd
```

Seguidamente recorremos à interface web disponibilizada para correr localmente, conforme o OpenVAS, mas desta vez no porto 8834 (https://localhost:8834), para finalizar a instalação da ferramenta e efetuar a respetiva ativação.

Antes de realizar quaisquer testes à nossa máquina virtual, decidimos primeiro examinar as suas características, procurando possíveis vulnerabilidades que pudessem ser mais tarde confirmadas pelas nossas ferramentas de identificação automática de vulnerabilidades.

c) NMap

Através da ferramenta NMap conduzimos um processo de enumeração onde identificámos quais os portos TCP e UDP abertos, bem como os serviços associados a cada porto.

Para isto, efetuámos inicialmente um TCP SYN scan para descobrir os portos TCP abertos. Neste scan, tal como em qualquer outro scan de portos, caso não seja especificado o número de portos a testar, irá examinar os primeiros 1.000 portos, considerados os mais importantes, em vez de verificar todos os 65.535 portos (para realizar o scan a todas as portas deverá ser usado a opção -p- no respetivo comando). Com este intuito, utilizámos então o seguinte comando:

- nmap -sS -sV -O 192.168.21.128

Sendo 192.168.21.128 o IP address da máquina virtual em análise (Metasploitable 2), obtido através do comando ifconfig na própria máquina.

```
┌──(root💀kali)-[~]
└─# nmap -sS -sV -O 192.168.21.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-12 11:30 WEST
Nmap scan report for 192.168.21.128
Host is up (0.0012s latency).
Not shown: 977 filtered ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: bridge|general purpose|switch
Running (JUST GUESSING): Oracle Virtualbox (98%), QEMU (93%), Bay Networks em
bedded (88%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:baynetworks:baystack
_450
Aggressive OS guesses: Oracle Virtualbox (98%), QEMU user mode network gatewa
y (93%), Bay Networks BayStack 450 switch (software version 3.1.0.22) (88%)
No exact OS matches for host (test conditions non-ideal).
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs
: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.79 seconds
```

Figura 5 - NMap para mapeamento de portos abertos TCP

A partir do NMap foi possível reunir várias informações. Sabemos, agora, que o alvo inspecionado está a correr Linux como sistema operacional, com a distribuição Ubuntu. O host executa um serviço SSH usando OpenSSH, um serviço telnet, um servidor web Apache 2.2.8, 2 servidores SQL e mais alguns serviços. Sumariamente:

- Vsftpd 2.3.4 no porto 21
- OpenSSH 4.7p1 Debian 8ubuntu 1 (protocol 2.0) no porto 22
- Linux telnetd no porto 23
- Postfix smtpd no porto 25
- ISC BIND 9.4.2 no porto 53
- Apache httpd 2.2.8 Ubuntu DAV/2 no porto 80
- A RPCbind no porto 111
- Samba smbd 3.X no porto 139 and 445
- 3 r s no porto 512, 513 and 514
- GNU Classpath grmiregistry no porto 1099
- Metasploitable root shell no porto 1524
- A NFS no porto 2049
- ProFTPD 1.3.1 no porto 2121
- MySQL 5.0.51a-3ubuntu5 no porto 3306
- PostgreSQL DB 8.3.0 – 8.3.7 no porto 5432
- VNC protocol v1.3 no porto 5900
- X11 no porto 6000
- Unreal ircd no porto 6667
- Apache Jserv protocol 1.3 no porto 8009
- Apache Tomcat/Coyote JSP engine 1.1 no porto 8180

Nesta fase, temos então listados todos os portos TCP abertos, no entanto, faltam ainda listar os portos UDP. Para este objetivo, corremos o seguinte comando:

- nmap -sU 192.168.21.128

```
┌──(root💀kali)-[~]
└─# nmap -sU 192.168.21.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-12 11:44 WEST
Nmap scan report for 192.168.21.128
Host is up (0.0016s latency).
Not shown: 999 open|filtered ports
PORT   STATE SERVICE
53/udp open  domain

Nmap done: 1 IP address (1 host up) scanned in 5.07 seconds
```

Figura 6 - NMap para mapeamento de portos abertos UDP

NMap apenas retornou o serviço domain a correr no porto 53.

É de referir que num scan de portos UDP podem surgir vários "falsos positivos". Este fenómeno acontece porque, quando um porto UDP se encontra fechado, o sistema responde com uma mensagem ICMP de porto indisponível, no entanto, quando há presença de uma firewall, a mensagem ICMP é bloqueada pela mesma, o que dará a entender que esses portos se encontram abertos, quando na realidade não estão.

A maioria dos serviços em execução verificados pelo Nmap provavelmente estarão vulneráveis. Obviamente sabemos que a máquina virtual Metasploitable 2 é intencionalmente vulnerável, portanto, só podemos suspeitar que a maioria, senão todos, os serviços contêm vulnerabilidades, backdoors, etc.. A próxima etapa é recorrer aos softwares de identificação de vulnerabilidades e descobrir quais os serviços se encontram efetivamente vulneráveis e coletar informações sobre como estas podem ser resolvidas.

4 Testes e análise de resultados

Para realizar os testes de exploração de vulnerabilidades ao Metasploitable 2, foram utilizados os já mencionados softwares Nessus 8.14.0 e o OpenVas 21.4.0, ambos instalados numa máquina virtual com distribuição Kali Linux. Em ambos os softwares foram utilizados as configurações padrões de análise.

	OpenVas	**Nessus**
Tempo de inspeção	28min e 43s	8min e 20s
Vulnerabilidades com fator de risco crítico/alto	14	11+8
Vulnerabilidades com fator de risco médio	33	26
Vulnerabilidades com fator de risco baixo	1	4

Tabela 2 - Comparação dos scans

Considerando que os dois softwares estavam instalados no mesmo ambiente, denotou-se uma grande diferença no tempo de procura. O OpenVAS demorou 3 vezes mais que o tempo utilizado pelo Nessus ao analisar o alvo para descoberta de vulnerabilidades. Nos resultados exibidos por ambos não consta a quantidade de testes realizados por cada software. Se esta métrica fosse disponibilizada seria possível analisar a diferença no tempo de inspeção.

Outra diferença consta no número de vulnerabilidades conforme o fator de risco. Realçar que enquanto o OpenVas divide o risco em três categorias (alto, médio, baixo), o Nessus, por sua vez, faz essa distribuição por quatro (crítico, alto, médio, baixo). Para esta análise iremos proceder à união da categoria de risco crítico com a categoria de risco alto. A quantidade de vulnerabilidades com fator de risco alto é de 14 e 19, respetivamente. É necessária atenção imediata aquando da constatação de vulnerabilidades "altas". Estas são comparativamente simples de explorar para os invasores e a através delas estes podem adquirir controlo completo dos sistemas impactados. Vulnerabilidades "médias" também foram encontradas. São frequentemente mais difíceis de explorar pelos invasores e podem não fornecer igual acesso aos sistemas afetados. Além destas, também vulnerabilidades com fator de risco baixo foram identificadas. Geralmente fornecem dados aos invasores que podem ajudá-los a orquestrar ataques subsequentes à respetiva rede. Estas deveriam ser corrigidos oportunamente também, mas não são tão gritantes quanto as outras vulnerabilidades.

Um dos fatores que contribui para a diferença no número de vulnerabilidades detetadas foi a configuração padrão das ferramentas, onde o OpenVas apenas tem ativo a análise a portos TCP enquanto o Nessus também faz a inspeção em portos UDP. Outro item que contribuiu para a diferença é a classificação de cada vulnerabilidade por cada software. O uso de cifras SSL que oferecem criptografia de nível médio é avaliado pelo Nessus como uma vulnerabilidade com fator de risco alto, no entanto o OpenVAS classifica esta vulnerabilidade com fator de risco médio.

A identificação de senhas de acesso fracas é uma característica que apenas foi observada no relatório do OpenVas. O Metasploitable 2 possui utilizadores e senhas de acesso padrões para o MySQL/MariaDB (utilizador e senha root) e para o Postgres (utilizador e senhas postgres) sendo que a ferramenta as considerou como vulnerabilidades com fator de risco alto.

Embora ambos os softwares tenham revelado bastantes e severas vulnerabilidades, não podemos confiar inteiramente nos resultados da verificação (falsos positivos, vulnerabilidades despercebidas, etc.) e, portanto, também é importante realizar testes manuais em combinação com a verificação automatizada.

Conclusões

Após o estudo efetuado cobrindo os vários tipos de vulnerabilidades, as suas razões e consequências, podemos facilmente concluir que a análise de vulnerabilidades é uma necessidade para qualquer tipo de sistema.

A utilização de ferramentas próprias de análise de vulnerabilidades, tais como o OpenVAS e o Nessus, facilita bastante o processo de descoberta de vulnerabilidades em sistemas. Para além desta benesse, estes softwares auxiliam ainda a solucionar as vulnerabilidades encontradas, uma vez que os relatórios gerados incluem possíveis correções ou soluções provisórias.

Através das ferramentas OpenVAS e Nessus, conseguimos eficazmente detetar as vulnerabilidades existentes no ambiente em que nos inserimos. Assim, com estes softwares torna-se possível aumentar a segurança da informação através da correção das vulnerabilidades antes que estas possam ser exploradas pelos atacantes

5 Referências

ADDEE, Tudo o que precisa de saber sobre análise de vulnerabilidades. Disponível em: <https://addee.com.br/blog/analise-de-vulnerabilidade-2/>

WELIVESECURITY, Como usar o OpenVAS para avaliação de vulnerabilidades. Disponível em: <https://www.welivesecurity.com/br/2019/07/24/como-usar-o-openvas-para-avaliacao-de-vulnerabilidades/>

RAPID7, Metasploit – Metasploitable2. Disponível em: <https://docs.rapid7.com/metasploit/metasploitable-2/>

HACKINGTUTORIALS, Metasploit and Metasploitable2 Installation Guide. Disponível em: <https://www.hackingtutorials.org/metasploit-tutorials/metasploit-metasploitable-2-installation/>

HACKINGTUTORIALS, Vulnerability Scanning with OpenVAS 9 part 1: Vulnerability Scanning. Disponível em: <https://www.hackingtutorials.org/scanning-tutorials/vulnerability-scanning-openvas-9-0-part-1/>

HACKINGTUTORIALS, Vulnerability Scanning with OpenVAS 9 part 2: Vulnerability Scanning. Disponível em: https://www.hackingtutorials.org/scanning-tutorials/vulnerability-scanning-openvas-9-0-part-2/

HACKINGTUTORIALS, Metasploitable 2 enumeration. Disponível em: <https://www.hackingtutorials.org/metasploit-tutorials/metasploitable-2-enumeration/>

Bruno Franco Oliveira – Universidade Federal de Uberlândia. Vulnerabilidades em Sistemas Web. Repositório UFU, 2019.

Nuno Miguel da Silva Monteiro – Instituto Superior de Engenharia do Porto. Estudo de vulnerabilidades em aplicações web e o seu reflexo em domínios portugueses. p 25-42, nov. 2015.

OPENVAS, OpenVAS – Open Vulnerability Assessment Scanner. Disponível em: <https://www.openvas.org/>

GREENBONE NETWORKS, Vulnerability Management. Disponível em: <https://www.greenbone.net/en/vulnerability-management/>

TENABLE, Products - The Nessus Family. Disponível em: <https://www.tenable.com/products/nessus>

TENABLE, Documentation – Configure Nessus. Disponível em: <https://docs.tenable.com/nessus/8_4/Content/ConfigureNessus.htm>

OFFENSIVESECURITY, Metasploit Unleashed Requirements. Disponível em: <https://www.offensive-security.com/metasploit-unleashed/requirements/>

KALI, Get Kali Linux – Virtual Machines – Kali Inside VirtualBox. Disponível em: <https://www.kali.org/docs/virtualization/install-virtualbox-guest-vm/>