Eduardo Chedid

WriteUP: EasyCTF

1) Enumeração com nmap
   Executei o comando: **nmap -sCV -sS 10.49.0.14**
   O que retornou as seguintes informações (resumidamente):

PORTA	SERVIÇO	VERSÃO
21	vsFTPd	3.0.3
80	openemr	5.0.1.3
2222	openSSH	7.2p2

   - Aqui havia a resposta para a primeira e a segunda perguntas:
   - Quantos serviços há abaixo da porta 1000: 2
   - Qual o serviço da porta mais alta: ssh

2) Abri o navegador e digitei o IP (para acessar a porta 80)
   http://10.49.0.14:80 / é equivalente a http://10.49.0.14/
   Mas retornou a página padrão do Apache.

3) Decidi, então, buscar por exploits dos outros serviços usando o ExploitDB. Encontrei alguns para o OpenSSH e para o openEMR.

   Então copiei o CVE deles e tentei entrar como resposta no TryHackMe (THM – Daqui para frente). Mas nenhum deu certo. Por isso, optei por não insistir nesses exploits.

4) Voltei para a página WEB e decidi rodar o GoBuster no modo de descoberta de diretórios. Para isso usei o seguinte comando:
   sudo gobuster dir -u http://10.49.9.14/ -w /usr/share/wordlists/dirbuster/directory0list-2.3-medium.txt

   Demorou um pouco, mas retornou um resultado para /simple

   (Caso não tenha as wordlists: sudo apt install wordlists)

5) Então abri no navegador o endereço http://10.49.0.14/simple e comecei a analisar a página.

   Percebi que havia um campo de buscas, o que indicava uma possibilidade de SQL Injection (Além do mais, estudamos para isso no grupo, tinha que ser usado para alguma coisa).

   Analisei também o código fonte da página, mas não vi nada que me chamasse a atenção.

Então, voltei a analisar a página, e no rodapé havia a versão do software que estava rodando CMSSimple 2.2.8 (se não me falha a memória)

6) Voltei para o ExploitDB, pesquisei por “CMS Simple 2.2.8” e nada. Resolvi então jogar no Google “Exploit CMS Simple 2.2.8” e o primeiro resultado foi: “CMS Made Simple <2.2.10 – SQL Injection”

   Então abri, copiei o CVE e colei no THM. Deu certo

7) Então copiei o código da vulnerabilidade. E salvei em um arquivo. Tentei executar com Python3, deu erro nos “prints” sem parênteses -principal característica de códigos em Python2. Tentei executar em python2.7 mas reclamou da falta de uma biblioteca.
   Então tentei instalar a biblioteca de algumas formas, reinstalei o pip para o python2 mas caiu em outros problemas. Nesse momento, como a biblioteca que estava dando problema era apenas para colorir o terminal, resolvi editar o código para não precisar mais dela.
   Funcionou.

   Rodando o exploit, obtive as seguintes informações que foram relevantes:
   Nome do usuário: **mitch**
   Senha: **secret**

   Respostas de outras perguntas:

   - Qual o nome do usuário: mitch
   - Senha: secret
   - Em qual serviço esses dados podem ser utilizados: ssh

8) Com as informações descobertas no item anterior era possível tentar o acesso via SSH:
   ssh **mitch**@10.49.0.14 -p 2222
   Ao pedir a senha: **secret**
   Voilá, temos acesso.

9) Ao entrar, executei o **ls** e percebi o arquivo user.txt, olhei o conteúdo usando: cat user.txt, e obtive a primeira flag

10) Então precisava descobrir a flag de root, para a qual, provavelmente precisaria de mais privilégios.

Então rodei: **sudo -l**
Para descobrir quais executáveis possuíam permissões de sudo sem precisar de senha. Resposta: vim

11) Então joguei no Google: GTFOBins vim
E lá estava como escalar privilégios com vim.

Executei: **sudo vim -c ':!/bin/sh'**
E consegui acesso de root
Então executei: cd /root
Para navegar até a pasta root

E: ls
Para olhar o que havia dentro da pasta
Resultado: root.txt
Usando o **cat** novamente, foi possível olha o conteúdo e capturar a última flag.