Relatorio NMap

João Fiuza de Alencastro 15/0131933

***Abstract*—Relatório da matéria Seguranca de Redes com o objetivo de descobrir vulnerabilidades em sistemas da rede local utilizando-se da ferramenta NMap de código aberto.**

***Index Terms*—Seguranca, redes, NMap, ports, TCP, UDP, vulnerabilidade.**

I. Introduction

ANALISAR os dados trafegados ao realizar um ”ataque” de varredura inteligente de portas de determinados *hosts* da rede, a fim de descobrir possíveis vulnerabilidades nos sistemas.

A exploracão, ou *”Information Gathering”*, da rede é a primeira etapa do processo de testes de penetracão (ou, pentest). É nesta etapa em que o possível atacante junta todas as informacões possíveis sobre a(s) vítima(s), para que, eventualmente, tenha todos os aparatos certos para que seu ataque seja bem sucedido.

A. Scripting

Fazer uma varredura NMap é algo simples, já que a ferramenta está disponível para qualquer um. Porém, a metodologia de *”Information Gathering”* vai além disso. Ao utilizarmos o Python temos a possibilidade de fazer scripts condicionais e inteligentes que possam escanear sistemas da forma mais perfomática possível, armazenando tudo isso e apresentando esse resultado de uma forma amigável ao usuário.

1) nmap.py:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

########################################
#       Autor: Joao Alencastro         #
########################################

import nmap, os

nm = nmap.PortScanner() # instantiate nmap.
    PortScanner object

ip_list = ['127.0.0.1','172.16.5.91']
ports = '21,443,8000'

for ip in ip_list:
  nm.scan(ip, arguments="-O -p "+ports)
  print('----------------------------------------------------'
    )
  print('\nFor IP address: ', ip)

  #procurando SO remoto
  if not nm[ip]['osmatch']:
    pass
  else:
    print(nm[ip]['osmatch'][0]['name'])

  if not nm[ip].all_tcp():
    #lista de portas TCP esta vazia
    print('There are no TCP open ports')
  else:
    #para todas as portas TCP da lista
    print('TCP open ports are:\n')
    for port in ports.split(','):
      if nm[ip]['tcp'][int(port)]['state'] != 'open'
    :
        pass
      else:
        print( '[+] Port', port, ' is open')
        print('-----------------------')



if not nm[ip].all_udp():
  #lista de portas UDP esta vazia
  print('There are no UDP open ports')
else:
  #para todas as portas UDP da lista
  print('UDP open ports are:\n')
  for port in ports.split(','):
    if nm[ip]['udp'][int(port)]['state'] != 'open'
    :
        pass
      else:
        print( '[+] Port', port, ' is open')
        print('-----------------------')
```

B. Varreduras NMap

1) Varredura TCP (TCP SYN): Esta é a primeira varredura estudada no curso, considerada sendo a mais simples delas. Ela é baseada na tentativa de conexão TCP com as portas especificadas na varredura. O método utiliza da flag SYN, flag de sincronizacão do protocolo TCP que é essencial para o ”three-way-handshake”. Porém, nesse caso, a conexão não é feita por completa, o que torna a varredura rápida de ser feita em múltiplas portas.

Abaixo está um exemplo de resposta que representa uma porta fechada, especificamente a porta 443. Podemos verificar isso, pois em suas flags TCP, o RST (reset) está setado e o SYN não.

Em seguida, vemos um exemplo de varredura TCP SYN em que a resposta foi positiva, passando a certeza que a porta 8000 está aberta, alvo de possíveis vulnerabilidades. A flag SYN/ACK certifica um possível atacante da abertura da porta, pois pelo protocolo TCP, esta resposta significa possibilidade de conexão.

2) Varredura TCP (TCP FIN): Segundo a documentacão oficial do IETF, RFC 793, página 65, ”Se o estado da porta é FECHADA ...um segmento de entrada que não contém o RST causa que seja enviado um RST em resposta.”. Este comportamento sistemático do protocolo permite explorar vulnerabilidades enviando diferentes pacotes TCP com diferentes

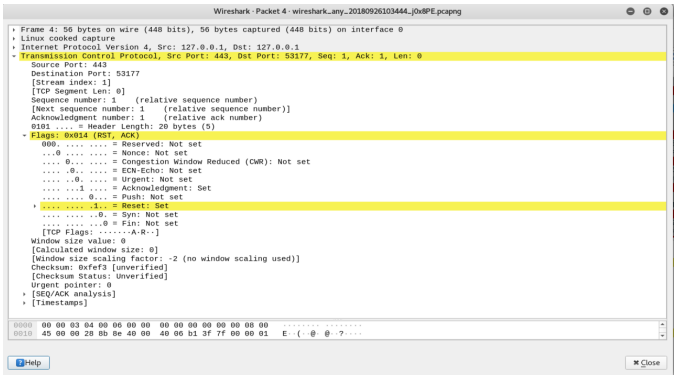

Fig. 1. Resposta de varredura de porta falha.

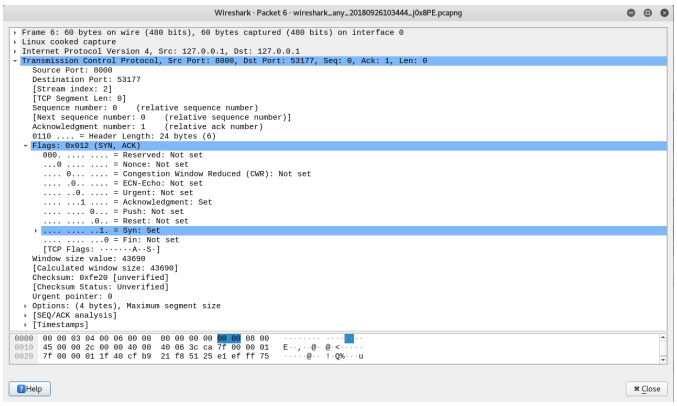

Fig. 2. Resposta de varredura de porta bem sucedida.

flags 'ativadas'. Porém, se enviarmos ativas flags, como, ACK, SYN ou RST, não teremos o resultado que precisamos.

Uma das possibilidades de pacote a ser enviado na varredura é com o bit da flag FIN ativado. Se a resposta é um RST, a porta está fechada, caso a resposta seja um pacote qualquer ou nada, a porta estará aberta.

3) Varredura TCP (TCP Null): Outra possibilidade dessa falha no protocolo TCP mencionada a cima é relizar a varredura com pacotes TCP Null. São pacotes que não 'setam' nenhum bit, ou seja, a parte de flags do cabecalho TCP é toda 0.

4) Varredura TCP (TCP Xmas): A última possibilidade analizada de varredura pelo protocolo TCP é o Xmas, batizado com esse nome por sua característica de setar todas as possíveis flags FIN, PSH e URG, deixando o pacote todo "aceso" como uma árvore de natal.

Tanto a varredura TCP Null, quanto a TCP Xmas funcionam como o TCP FIN, mesma lógica, caso seja recebido como resposta um pacote RST, a porta está, por definicão do RFC, **fechada**. Caso a resposta seja nada ou um outro pacote qualquer, a porta está **aberta**.

5) Varredura UDP: A varredura UDP, certamente, funciona a partir de pacotes UDP. A partir de um envio desse pacote para determinada porta, se o retorno for um pacote ICMP de porta inalcançável, ou seja, erro tipo 3, significa que a porta está fechada. Porém, se o pacote ICMP for de outros tipos, a resposta do nmap será 'filtered', que significa que a porta pode estar aberta sendo filtrada por algum intermediador. Caso o servico responda ao pedido com um pacote UDP, sabe-se que a porta está aberta e preparada para a conexão UDP. Como, no exemplo abaixo, o servidor em questão respondeu ao pedido DNS, sabe-se que a porta 53 está aberta, já a porta 636, não, pois foi retornado um pacote ICMP do tipo porta inalcançável.

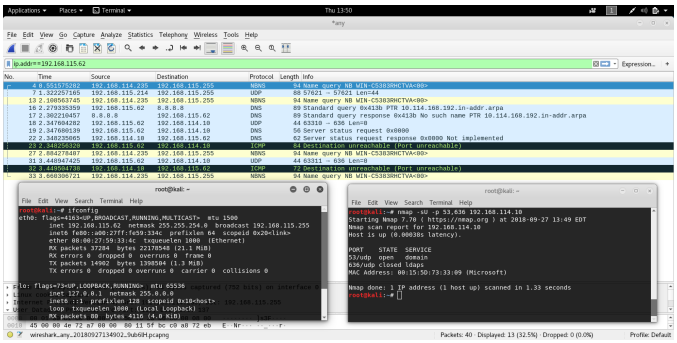

Fig. 3. Varredura UDP na porta 53 e 636.

II. Conclusion

As varreduras de redes e *hosts* são muito importantes para descobrir vulnerabilidades em sistemas internos a ambientes pessoais ou empresariais, porém são ferramentas que devem ser utilizadas com cuidado, pois são extremamente invasivos. Ninguém quer compartilhar informacões valiosas como essas para pessoas desconhecidas. Portanto, devem ser utilizadas sempre juntamente da ética. Além disso, o NMap oferece todas essas opcões, e mais, para que o usuário, com muito conhecimento técnico, possa diversificar seu leque de possibilidades. Conhencendo as flags do protocoloco TCP, por exemplo, dá ao usuário formas diferentes de fazer uma varredura de reconhecimento sem ser percebido. Tarefa essencial também para o hacker ético, pois para simular possíveis ameacas, ele deve se disfarcar de instruso, e um intruso nunca quer ser encontrado, muito menos localizado.

References

[1] https://nmap.org/book/man-port-scanning-techniques.html
[2] https://tools.ietf.org/html/rfc793