

BSK lab 1 – czyli trochę o nmapie i skanowaniu portów

Skanowanie portów służy generalnie do ustalenia jakie usługi są dostępne na jakimś urządzeniu w sieci. To przy okazji może pomóc w znalezieniu wektora ataku (wiadomo w jaki sposób dalej próbować się z urządzeniem dogadać) i zbyt dużo na to nie poradzimy – wiadomo, jeżeli serwer do czegoś służy, choćby do obsługi strony WWW, to jakiś widoczny sposób komunikacji z nim musi być. Istotna jest inna rzecz, żeby na urządzeniu otwarte były tylko te porty, które rzeczywiście muszą być – albo żeby porty, które potrzebne są tylko w obrębie naszej sieci, były jakoś zabezpieczone przed ruchem z zewnątrz (np. za pomocą firewalla). I tu nmap przyda się do sprawdzenia czy rzeczywiście tak jest.

Na porcie mogą być używane dwa protokoły, TCP i UDP. My przede wszystkim będziemy zajmować się skanowaniem TCP. Sposoby na skanowanie UDP też są, ale ponieważ to protokół bezpołączeniowy – jest to sporo trudniejsze i daje mniej dokładne wyniki (natomiast od czasu do czasu takie skanowanie też może być potrzebne).

Sama obsługa nmapa jest dość prosta, podstawowe wywołanie polecenia to:

```
nmap <cel skanowania>
```

gdzie celem może być konkretny host (np. 156.17.193.186 lub student.pwr.edu.pl) albo cała sieć (np. 192.168.1.0/24). Przed celem możemy dopisać dodatkowo parametry, np. wybrać metodę skanowania - trochę ich jest i tutaj warto byłoby wspomnieć o kilku istotniejszych.

Pierwsze dwie polegają na próbie nawiązania sesji TCP na danym porcie:

-sS – skanowanie SYN, metoda domyślna (używana jak nic innego nie wpisujemy) – polega na wysłaniu pakietu TCP z flagą SYN, jeżeli urządzenie odpowie pakietem RST, to port jest zamknięty, jeżeli pakietem SYN-ACK, to port jest otwarty i w tym momencie nmap od razu wysyła pakiet RST przerywający nawiązywanie połączenia – dzięki temu sposób szybki i generalnie niezawodny

-sT – skanowanie TCP connect – podobnie, ale polega na pełnym nawiązaniu połączenia na porcie za pomocą funkcji systemu operacyjnego (sekwencja SYN, SYN-ACK, ACK) i zerwania go za pomocą RST dopiero wtedy – sposób trochę mniej dokładny i wolniejszy, ale czasami przydatny, w przeciwieństwie do innych nie wymagający bezpośredniego dostępu nmapa do sprzętu (np. brak uprawnień roota na Linuksie)

Następne trzy są związane ze sposobem reakcji na wysłane tak zupełnie "z czapy" pakiety nie służące do nawiązania ani zerwania sesji TCP:

-sN – skanowanie NULL – pakiet bez ustawionych żadnych flag

-sF – skanowanie FIN – pakiet z ustawioną samą flagą FIN

-sX – skanowanie metodą choinkową (Xmas) - pakiet z ustawionymi wszystkimi 3 możliwymi w tej sytuacji flagami (FIN, PSH, URG), "rozświetlony" jak lampki na choince - stąd nazwa :-D

Jeżeli trzymać się RFC 793 opisującego protokół TCP, w tych sytuacjach na zamkniętym porcie w odpowiedzi powinien zostać wysłany pakiet RST, a na otwartym taki pakiet powinien zostać zignorowany i nie powinno być żadnej odpowiedzi. Sposób nie całkiem dokładny (nie odróżni się choćby, czy odpowiedzi nie było bo port jest otwarty, czy nie było bo po drodze jest firewall z ustawioną regułą drop), ale czasami może się przydać jak jakiś inny nie działa. Natomiast w praktyce jest trochę systemów operacyjnych, które RFC się nie trzymają i zawsze odeślą wtedy RST – z tego co pamiętam np. Cisco IOS, spróbujcie metodą choinkową przeskanować na labkach router z telnetem (port 23) – najpewniej wszystkie porty zostaną uznane za zamknięte, 23 też. Czyli porównując wyniki różnych sposobów skanowania można jeszcze próbować wnioskować coś o systemie operacyjnym – chociaż na to jest skuteczniejszy sposób.

Kilka innych parametrów, które czasami mogą się przydać:

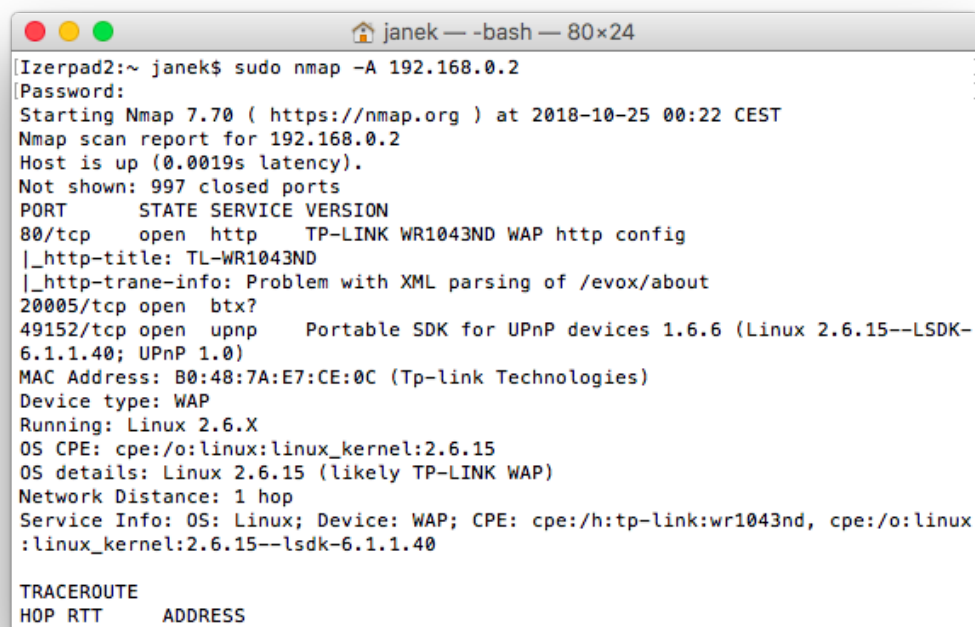
-sA – skanowanie ACK – wysyłanie pakietu z samą flagą ACK na co w obu przypadkach (zamknięty i otwarty port) odpowiedzią będzie RST, ale można sprawdzić, czy po drodze jakiś firewall nie filtruje pakietów

-sU – skanowanie UDP – tzn. wysłanie próbnego pakietu UDP na port i założenie, że brak odpowiedzi = port otwarty (ale równie dobrze może być filtrowany firewallem), odpowiedź ICMP o nieosiągalności = port zamknięty – przy czym odpowiedzi ICMP dla jednego IP mogą być limitowane do jednej na jakiś czas, więc skanowanie może trwać bardzo długo

-A – skanowanie z kompletem kilku ciekawych dodatkowych rzeczy, jak rozpoznawanie systemu operacyjnego na podstawie zachowania urządzenia (-O), wykrywanie wersji uruchomionych usług (-sV) i nie tylko – bardzo polecam sprawdzić samemu, zdziwicie się ile ciekawych rzeczy można się dowiedzieć ;-)

Na zajęciach trzeba zrobić kilka (jakieś 3-4) różne skany nmapem – różnymi sposobami/różne cele i przechwytywać w tym samym czasie w Wiresharku pakiety skanujące, a potem pokazać różnice. I np. możecie zobaczyć jak wyglądają pakiety przy skanowaniu metodą choinkową, a jak przy skanowaniu metodą SYN, jakie odpowiedzi są wysyłane w różnych sytuacjach (metoda/zamknięty albo otwarty port), itp. Generalnie jak pokażecie, że coś takiego jak właśnie opisywałem ma rzeczywiście miejsce – będzie dobrze :-)

Na koniec taki przykład skanowania z parametrem -A routera TP-Link TL-WR1043ND:

A screenshot of a terminal window titled 'janek — -bash — 80x24'. The terminal shows the command 'sudo nmap -A 192.168.0.2' being executed. The output of the nmap scan is displayed, showing that the host is up and providing detailed information about the device, including its MAC address, device type (WAP), and OS details (Linux 2.6.15).

```
[Izerpad2:~ janek$ sudo nmap -A 192.168.0.2
[Password:
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-25 00:22 CEST
Nmap scan report for 192.168.0.2
Host is up (0.0019s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    TP-LINK WR1043ND WAP http config
|_http-title: TL-WR1043ND
|_http-trane-info: Problem with XML parsing of /evox/about
20005/tcp open  btx?
49152/tcp open  upnp    Portable SDK for UPnP devices 1.6.6 (Linux 2.6.15--LSDK-6.1.1.40; UPnP 1.0)
MAC Address: B0:48:7A:E7:CE:0C (Tp-link Technologies)
Device type: WAP
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.15
OS details: Linux 2.6.15 (likely TP-LINK WAP)
Network Distance: 1 hop
Service Info: OS: Linux; Device: WAP; CPE: cpe:/h:tp-link:wr1043nd, cpe:/o:linux:linux_kernel:2.6.15--lsdk-6.1.1.40

TRACEROUTE
HOP RTT      ADDRESS
```

I widać od razu co pod spodem ma soft TP-Linka ;-)

Jan Potocki

Wrocław, 24.10.2018

pisałem głównie na podstawie *man nmap* (ale nie tylko)