Narzędzia do podsłuchu sieciowego.

Chociaż posłuch sieci może kojarzyć się jednoznacznie (negatywnie) to jest on, a przynajmniej powinien być, jedną z metod do oceny funkcjonowania naszej sieci. Ponadto narzędzia pozwalają na zorientowanie się w słabych punktach naszej sieci i umożliwiają na ich zabezpieczenie/wyeliminowanie.

Obecnie można znaleźć sporo narzędzi do badania oraz wykonywania podsłuchu sieci. Programy te nie posiadają przeważnie graficznego interfejsu, aczkolwiek nie jest to regułą. Wiele narzędzi przypisane jest do konkretnego systemu operacyjnego (Linux, Unix bądź Windows). Dzieje się tak dlatego, że każdy z dostępnych systemów posiada inną implementację sieciową. Najbardziej zbliżone są ze sobą systemu Linux i Unix – niemal każde narzędzie można pomiędzy nimi wymieniać. Dla tychże systemów została opracowana, pierwotnie na potrzeby analizatora tcpdump, biblioteka libpcap (nazwa pochodzi od library packet capture). Zawarte w niej rozwiązania pozwalają na przechwytywanie całego ruchu jaki generowany jest w sieci lokalnej dzięki wykorzystaniu pracy kart sieciowych w trybie mieszanym/nasłuchującym (promiscuous) oraz gniazd sieciowych typu surowego (RAW) – pozwalających na zebranie przesyłanych danych bajt po bajcie (bez rozróżnienia na struktury czy sekwencję). Bez względu na pierwotny zamysł dzisiaj opisywana biblioteka stanowi zbiór gotowych funkcji API (Application Programming Interface) wykorzystywany przez niemal wszystkie narzędzia podsłuchujące/testujące działanie sieci.

Ponieważ biblioteka ta dobrze sprawdziła się w systemach Unix/Unix-Like, stworzono jej wersję specjalnie dla systemu Windows; nosi ona w nim nazwę Winpcap. Ponieważ systemy z rodziny Windows nie są typowo sieciowymi systemami (jądro nie posiada implementacji gniazd sieciowych; gniazda sieciowe nie są podnoszone wraz ze startem systemu a na „życzenie” aplikacji sieciowych) toteż projekt na Windows musiał wykorzystać mechanizm sieciowy dostępny dla Windows o nazwie NDIS (Network Driver Interface Specification), stanowiący pomost (API) pomiędzy wszystkimi programami sieciowymi a niskopoziomowymi rozkazami kart sieciowych (sterownik). Obecna wersja sterownika NDIS to 10 (dostępna w systemie Windows 10).

Narzędzia do testowania działania sieci:

a) Wireshark (wieloplatformowy) – potężny program pozwalający na przechwycenie każdego pakietu wędrującego w naszej sieci. Dzięki rozbudowanym filtrom oraz intuicyjnemu interfejsowi można podsłuchać każdy pakiet w sieci adresowany do nas, grupy (multicast) bądź do całej podsieci (broadcast). Pełny zakres działania programu (wraz z zadaniami dodatkowymi) znajdują się w osobnym dokumencie o nazwie wireshark.pdf)

b) ngrep – narzędzie konsolowe. Pozwala na przechwytywanie wszystkich pakietów ruchu sieciowego bądź przechwytywanie tylko tych, które spełnią nasze kryteria (np. wyrażenie regularne). W przeciwieństwie do większości programów szpiegujących ngrep podaje wyniki w postaci czytelnej dla człowieka, przeważnie w postaci chronologicznej. Samo wykorzystanie narzędzia jest bardzo proste i sprowadza się np. do wpisania takiego polecenia (w przypadku systemu Linux):

ngrep

W przypadku systemu Windows, ponieważ folder programu nie będzie się prawdopodobnie znajdował w zmiennej środowiskowej PATH, należy przejść do odpowiedniego katalogu i dopiero wtedy go uruchomić.

NIEZALEŻNIE OD SYSTEMU program należy uruchamiać z uprawnieniami administratora bądź

nadać mu odpowiednie uprawnienia do korzystania z komunikacji sieciowej z prawami administratora (systemu Unix/Unix-like).

Powyższy przykład będzie działał poprawnie o ile:
- chcemy wyłapywać WSZYSTKIE pakiety
- działamy na domyślnym interfejsie sieciowym (przeważnie pierwsza karta LAN)
- nie chcemy zapisywać wyników ani nie interesuje nas żadne formatowanie wyników

Jeżeli natomiast chcemy w jakikolwiek sposób poprawić sobie czytelność wyświetlanych wyników to możemy zastosować np. parametr -W. Pozwala on na formatowanie wyjścia programu (wyświetlanych komunikatów). Przykładowo użycie

ngrep -W byline

spowoduje, że znaki nowej linii będą odpowiednio interpretowane i np. nagłówki HTTP staną się bardziej czytelne

Innym, równie ciekawym przełącznikiem jest możliwość wybrania protokołu, który nasłuchujemy. Przeważnie będziemy wybierać pomiędzy tcp a udp

ngrep -W byline tcp

Od tego momentu wszystko co będzie działo się w komunikacji na innych protokołach nie będzie wyświetlane

Kolejnym zawężeniem poszukiwań może być podanie portu, na którym nasuchujemy

ngrep -W byline tcp and port 80

Teraz tylko to, co zostanie przesłane przez port 80 będzie dla nas interesujące.

Jeżeli interesować nas będzie tylko to, jaka przeglądarka zadała zapytanie to można użyć wyrażenia regularnego:

ngrep -W byline 'User-Agent: ' tcp and port 80

Jak widać używanie aplikacji ngrep jest łatwe i daje sporo cennych informacji o ruchu sieciowym.

c) tcpdump/ssldump – tcpdump to narzędzie pozwalające na odbieranie i wyświetlanie zawartości wszystkich pakietów dochodzących/wychodzących do/z naszej sieci. To pod nie napisana została biblioteka pcap. ssldump z kolei napisany został z myślą o przechwytywaniu/sprawdzaniu słabych punktów protokołów zabezpieczonych (np. HTTPS). W przypadku przechwycenia klucza możliwe jest nawet dekodowanie przesyłanych informacji. W przypadku braku klucza narzędzie najczęściej pokaże nam wszelkie jawne dane oraz sklasyfikuje dane zakodowane (który fragment to nagłówek, który to dane aplikacji itp.)

PROSZĘ PAMIĘTAĆ że tcpdump, tak jak większość przedstawianych programów, wymaga podwyższonych uprawnień do zakładania gniazd typu surowego/przechodzenia karty w tryb mieszany.
Najprostsze użycie aplikacji to wpisanie w linii poleceń:

tcpdump

Jeżeli nasłuchujemy na karcie, która jest naszym domyślnym dostępem do sieci to powinniśmy otrzymywać informacje o każdym przewijającym się przez nasz interfejs pakiecie. Jeżeli nasłuch chcemy użyć na innym urządzeniu sieciowym (np. WLAN zamiast LAN) to narzędzie trzeba uruchomić z taką opcją:

tcpdump -i wlan0

co spowoduje przechwytywanie ruchu z karty bezprzewodowej.

Jeżeli interesuje nas komunikacja na wybranym porcie:

tcpdump -i wlan0 port 80

od tego momentu będziemy informowani jedynie o pakietach przesyłanych na porcie 80.

Jeżeli chcemy podsłuchać pakiety na porcie 80 i dodatkowo jedynie z protokołu tcp:

tcpdump -i wlan0 tcp and port 80

Proszę zauważyć, że w dokumentacji (do starszej wersji) wskazane jest by używać dodatkowych opcji zapisanych w cudzysłowu/apostrofach. Oznacza to, że równie poprawnym zapisem byłaby postać:

tcpdump -i wlan0 'tcp and port 80'

Użycie ssldump w zasadnie nie różni się od używania tcpdump. Tutaj jednak mamy możliwość używania np. wcześniej przechwyconych kluczy SSL (bądź przez nas skopiowanych), dzięki którym możemy podejrzeć co jest transmitowane pomiędzy nadajnikami.
Przykład użycia:

ssldump -d -i wlan0 'tcp and port 443'

Jak widać doszedł parametr -d (jeżeli będzie możliwe, dane zostaną odszyfrowane) oraz zmieniony został port nasłuchiwania (teraz to 443).

d) nmap – jedno z potężniejszych narzędzi do skanowania i tworzenia audytów bezpieczeństwa zarówno sieci lokalnej jak i sieci rozległej. W przeciwieństwie do przedstawionych powyżej programów narzędzie nie bazuje na bibliotece pcap. Wykorzystuje on wszystkie dostępne opcje niskopoziomowych pakietów IP (m. in. narzędzia ICMP; w grę wchodzą również inne, specyficzne pakiety dla konkretnych rozwiązań/systemów) celem rozeznania czy dane urządzenie jest w sieci (jednak np. nie odpowiada na pakiet ping), jaki system operacyjny/jakie urządzenie znajduje się pod danym adresem, jakie oprogramowanie jest na nim uruchomione (np. jaka wykorzystywana jest zapora sieciowa) oraz ile urządzenie sieciowe działa w sieci.

Przykładowe skanowanie:

nmap -v scanme.nmap.org

Nastąpi przeskanowanie wszystkich zarezerwowanych portów TCP pod wskazaną nazwą domenową (nmap zadba o przetłumaczenie jej na adres IP). Parameter -v mówi, że nmap ma być bardziej „wygadany” tj. informować użytkownika o bieżących postępach (brak tego parametru

powoduje, że nmap milczy przez cały proces skanowania wyświetlając dopiero raport końcowy).

nmap -sS -O -v -T4 scanme.nmap.org/24

Powyższa komenda przebada nasz potencjalny cel oraz wszystkie adresy należące do jego puli w zakresie 24 bit (sieć zawierająca do 254 maszyn). Ponadto przy wykrywaniu potencjalnie otwartych portów nmap ma użyć metody ukrycia pakietu synchronizacji (Stealth SYN, opcja -sS). Takie podejście pozwala na rozpoznanie czy port jest otwarty, zamknięty bądź filtrowany. Parametr -O nakazuje nmap wykrycie systemu operacyjnego obsługującego nasz cel (niekoniecznie musi się powieść). -T4 oznacza, że skanowanie ma odbyć się pobieżnie (możliwie najszybciej).

nmap -Pn -A -v scanme.nmap.org

Powyższe polecenie uruchomi nmap w trybie skanowania docelowego adresu bez pingowania go (opcja -Pn) – w przypadku, gdy administrator wyłączył odpowiedzi na pakiet ping nmap domyślnie uznaje, że wskazana maszyna jest wyłączona. Gdy użyta zostanie wskazana opcja nmap zamiast pingowania od razu przechodzi do wysłania serii zapytań o otwartość portów. Jeżeli chociaż jedna odpowiedź będzie pozytywna oznaczać to będzie, że serwer jest aktywny. Parametr -A jest podobny do -O z tą różnicą, że prócz wykrycia samego systemu próbuje także ustalić nazwy działających na nim usług.

ZADANIA:
1. Jak działa protokół ICMP. Jaka jest różnica pomiędzy nim a protokołem IGMP.
2. W jaki sposób najłatwiej jest podsłuchać ruch w sieci lokalnej. Co powoduje, że podsłuch jest utrudniony.
3. Sprawdzić jakie narzędzia są dostępne wyłącznie na system Windows; czy ich funkcjonalność można w jakikolwiek sposób odtworzyć w systemie Linux?
4. W jaki sposób poprzez tcpdump/ssldump uzyskać nasłuch na wielu portach równocześnie?
5. Proszę podać 5 przykładów filtrów dla tcpdump/ssldump, dzięki którym można będzie wytropić konkretne poszukiwane dane (np. zawartości wskazanych ciasteczek).
6. Jakie nieprawidłowości może wykryć dla nas nmap? Proszę podać przynajmniej 5 potencjalnych zagrożeń, składni jaka pozwala je wykryć oraz jakie można zastosować środki zaradcze.
7. Jakie inne opcje skanowania przewiduje nmap? Czy któraś z nich jest najskuteczniejsza? Należy podać wyniki skanowań (minimum 5 technik).
8. W jaki sposób przeskanować poprzez nmap zakres portów z wybranego protokołu (np. udp)?
9. Czy istnieją metody oszukiwania zapory sieciowej poprzez program nmap?
10. W jaki sposób możliwe jest podsłuchanie pakietów, które nie są do nas adresowane przy wykorzystaniu systemu Linux?
11. Proszę podać i scharakteryzować przynajmniej 5 najużyteczniejszych narzędzi sieciowych dostępnych w dystrybucji Kali Linux. Mile widziane przykłady użyteczności (polecenia/zrzuty ekranu).

ZADANIE DODATKOWE:
W niniejszym materiale nie został poruszony temat popularnego dla systemu Windows narzędzia Cain and Abel. Należy zapoznać się z jego możliwościami (i ich potencjalnym zastosowaniem) oraz podać zamienniki w systemie Linux (pod względem funkcjonalności).

https://wiki.wireshark.org/CaptureSetup/Ethernet
http://viralzones.blogspot.com/2013/08/man-in-middle-mitm-attack-using.html
http://www.nextgenupdate.com/forums/computer-programming/348518-how-hack-cain-wireshark.html

http://www.netresec.com/?page=Blog&month=2014-02&post=HowTo-install-NetworkMiner-in-Ubuntu-Fedora-and-Arch-Linux
https://sourceforge.net/projects/networkminer/
http://linux.die.net/man/1/ssldump
https://nmap.org/man/pl/
https://nmap.org/bennieston-tutorial/

http://www.kalitutorials.net/2013/08/kali-linux.html
http://www.kalitutorials.net/2014/02/penetration-testing-hacking-xp.html
http://www.kalitutorials.net/2014/04/hack-wpawpa2-wps-reaver-kali-linux.html
http://www.kalitutorials.net/2013/08/wifi-hacking-wep-kali-linux-aircrack-ng.html
http://www.hacking-tutorial.com/hacking-tutorial/kali-linux-man-middle-attack/#sthash.XxT4p0r0.dpbs
http://www.hacking-tutorial.com/hacking-tutorial/kali-linux-man-middle-attack/#sthash.uUkDyDc4.dpbs