Wydział Fizyki i Informatyki Stosowanej
Akademia Górniczo-Hutnicza w Krakowie

Bezpieczeństwo w sieci

„Posłuch w sieci na podstawie programów tcpdump oraz wireshark”

Adrian Kałuziński
Tomasz Juszkiewicz
laboratorium: 02

Kraków, 2012

Spis treści

1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące podsłuchu w sieciach na poziomie warstwy drugiej modelu OSI. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Tematyka laboratorium

Tematem labolatorium jest przedstawienie czym jest przechwytywanie pakietów ethernetowych (lub przepływających w sieci bezprzewodowej) czyli tzw. **sniffing** na poziomie 2 warstwy OSI(ang. *Open System Interconnection*) oraz zademonstrowanie go na przykładach w programach tcpdump oraz Wireshark(dawniej Ethereal). Sniffingiem, nazywamy przechwytywanie niekoniecznie za wiedzą i zgodą użytkowników sieci, oraz ewentualne analizowanie danych przepływających przez sieć. W sieci przewodowej polega ono na odczytywaniu pakietów przepływających we wspólnej domenie kolizyjnej, natomiast w sieciach bezprzewodowych na odczytywaniu informacji na konkretnym kanale (ang. *channel)*. Następnie zostaną wymienione metody na ograniczenie możliwości przechwytywania naszych pakietów przez intruza.

Sniffing wykorzystuje się w:

- Analizie problemów sieci,
- Wykrywaniu prób włamania,
- Wykrywanie przeciążeń i zużycia sieci,
- Inżynierwia wsteczna protokołów,
- Ataki przechwytywania jawnie wysłanych pakietów zawierających istotne dane.

1.2. Zagadnienia do przygotowania

Przed przystąpieniem do realizacji laboratorium należy zapoznać się z zagadnieniami dotyczącymi **opcji Ethernet**:

- Znajomość pojęcia modelu ISO/OSI [1,6]
- Znajomość budowy sieci Ethernet [2,7]
- Pojęcie **adresu MAC** [2, 4]

- Nazwy i przeznaczenie opcji ramki WLAN [3]
- Obsługa programu **ifconfig** i **ping** [5]

Literatura:

[1] Andrew Stuart Tanenbaum *Sieci komputerowe*. Helion 2004.
[2] IEEE (standards.ieee.org) Standard 802.3 Ethernet.
[3] IETF RFC 2369, RFC 791
[4] MAN ifconfig, (http://linux.die.net/man/8/ifconfig)
[5] Jon Erickson *Hacking. Sztuka penetracji*. Helion 2008
[6] Michal Zalewski *Cisza w sieci*, Helion 2005
[7] http://pl.wikipedia.org/wiki/Transmission_Control_Protocol

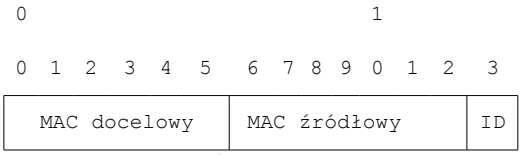
1.3. Opis laboratorium

Labolatorium koncentruje się na drugiej warstwie modelu referencyjnego ISO/OSI – warstwie łącza danych. W oparciu o użycie programów tcpdump oraz wireshark zostanie zaprezentowane działanie drugiej warstwy modelu w prostej oraz skomplikowanej sieci komputerowej, możliwość interpretowania i analizowania przesyłanych pakietów oraz związane z tym potencjalne ryzyko dla użytkowanika.

1.3.1. Warstwa łącza danych

Najniższą widoczną z poziomu użytkownika warstwą jest warstwa łacza danych. Warstwa łącza danych okresla metody adresowania i przesyłania wiadomości do wszystkich osób w danej sieci, sposoby sprawdzania obecności innych węzłów oraz zapewnie podstawową detekcję kolizji CSMA (ang. *Carrier Sense Multiple Access)*. Warstwa ta jest odpowiedzialna za obsługę standardu Ethernet 803.3 i zapewnia użytkownikom system fizycznego adresowania urządzeń. Adresy te znane są jako adresy **MAC** (ang. *Media Access Control*). Każde urządzenie ethernetowe otrzymuje niepowtarzalny adress złożony z 6 oktetów. Zwykle zapisywanych heksadecymalnie w postaci XX:XX:XX:XX:XX:XX. Adresowanie MAC są także metodą adresowania sprzętowego ponieważ każde urządzenie sieciowe posiada unikalny adres, umieszczony w zintegrowanej pamięci urządzenia. Z założenia każde urządzenie powinno mieć globalnie różny adres, co umożliwia jego jedonoznaczą identyfikację.

Nagłówek Ethernetowy ma rozmiar 14 bajtów i zawiera adresy MAC źródłowy oraz docelowy używane do przekazywania pakietów w miejsca docelowe. Standard adresowania Ethernet gwarantuje unikalny adres rozgłaszania, który zawiera wyłącznie jedynki. Każdy pakiet ethernetowy przesłany na ten adres zostanie przekazany do wszystkich przyłączonych urządzeń.



Rys 1. *Nagłówek Ethernet IEEE 803.3 (14 bajtów)*

1.3.2. Warstwa sieci i protokół IP

Zadaniem warstwy sieciowej jest przenoszenie pakietów na całej trasie od źródła do celu. Dotarcie do miejsca przeznaczenia może wymagać wielu przeskoków przez routery na trasie pakietu. W oczywisty sposób różni się ona tym od warstwy łącza danych, której głównym zadaniem jest przesyłanie ramki z jednego końca kabla na drugi. Wartwa sieciowa jest więc najniższą warstwą zajmującą się transmisją punkt-punkt. Aby osiągnąć cele wartstwa sieciowa musi znać topologię podsieci (tzn. zbiór wszystkich routerów) i wybierać odpowiednią trasę poprzez tę podsieć. Musi w taki sposób wybierać trasy aby uniknąć przeciążania części linii komunikacyjnych i routerów, pozostawiając inne bez pracy. Protokół używany do tego celu przez warstwę sieci nosi nazwę **IP** (ang*. Internet Protocol*). W Internecie każdy system ma adres IP, który stanowi grupę czterech bajtów xxx.xxx.xxx.xxx. Nagłówek IP dla pakietów warstwy sieci ma rozmiar 20 bajtów i zawiera różne pola i flagi bitowe.

0 1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	3 1	3 2
Version				IHL				Type of service (ToS)								Total Length (TL)																
Identification																Flags			Fragment Offset												
Time to Live (TTL)								Protocol								Header Checksum																
Source Address																																
Destination Address																																
Options (up to 5 words)																													Padding			
Data																																

Type of Service

1	2	3	4	5	6	7	8
Precedence			Delay	Throughput	Reliability	Reserved	

Flags

1	2	3
Reserved	Don’t Fragment (DF)	More Fragments (MF)

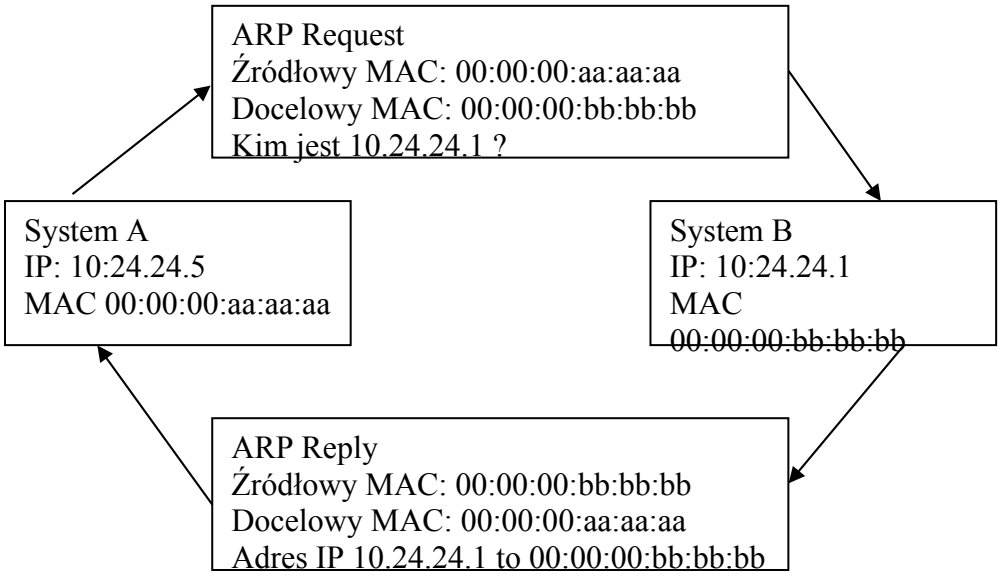
Rys 2. *Nagłówek IP (RFC 791)*

Pojawia się problem jeżeli chcemy przetłumaczyć adresy IP na adresowanie fizyczne. Wówczas, aby powiązać obie metody adresowania, należy skorzystać z dodatkowych protokołów takich jak ARP.

1.3.3. Protokół ARP

Protokół ARP (ang. Adress Resolution Protocol) jest metodą powiązania ze sobą adresowania IP oraz MAC. Istnieją następujące typy komunikatów ARP:

- **ARP Request** – stanowi zapytanie „Kto o numerze MAC xx:xx:xx:xx:xx ma adres IP yyy.yyy.yyy.yyy? „
- **ARP Reply** – Jeżeli w podsieci znajduje się komputer o poszukiwanym adresie IP, rozgłasza on swoją obecność. Odpowiedź ARP jest wysyłana w formacie: „Ja mam adres MAC xx:xx:xx:xx:xx, a mój adres IP to yyy.yyy.yyy.yyy”



1.3.4. Warstwa transportu – protokół TCP

Protokół **TCP** (ang. *Transmission Control Protocol*) to wraz z UDP(*ang. User Datagram Protocol*) podstawowy protokół warstwy transportowej. Jest najczęściej wykorzystywany przez usługi internetowe takie jak telnet, http (ruch WWW), smtp(poczta elektroniczna) i ftp (przesyłanie plików). Tak duża popularność tego

protokołu jest związana z możliwością utworzenia transparentnego i niezawodnego połączenia dwukierunkowego pomiędzy dwoma adresami IP. Dwukierunkowe połączenie TCP można uznać za analog połączenia telefonicznego – po wybraniu numeru nawiązywane jest połączenie poprzez które komunikują się obaj rozmówcy. Charakteryzuje go wysoka niezawodność, która oznacza, że wszystkie dane osiągną miejsce docelowe we właściwej kolejności. Jeżeli część z pakietów dostrze do odbiorcy w złej kolejności, protokół TCP za pomocą odpowiednich flag posortuje je przed przekazaniem danych do wyższej warstwy. W razie zagubiena niektórych pakietów system odbiory zachowa wszystkie odebrane dane, podczas gdy system nadawcy ponownie prześle brakujące (lub uszkodzone) pakiety. Działanie wszystkich funkcji protokołu TCP stało się możliwe po zastosowaniu specjalnego zestawu znaczników (nazywanie znacznikami TCP) oraz wartości śledzenia numerów sekwencyjnych, chararakteryzujące kolejność pakietów. Znaczniki są przechowywane w nagłówku TCP wraz z oczeniami portu zrodlowego i docelowego (RFC 793).

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Source port																Destination Port															
Sequence Number																															
Acknowledgement Number																															
Offset				Reserved				TCP Flags								Window															
Checksum																Urgent Pointer															
Options (0 or more 32-bit words)																															
Data (optional)																															

TCP Flags

1	2	3	4	5	6	7	8
Cogenstion Window Reduced (CWR)	ECN Echo (ECE)	Urgent	Ack	Push	Reset	Syn	Fin

Specyfikacja nagłówka TCP (RFC 793):

Zwięzły opis flag TCP znajduje się pod adresem [7].

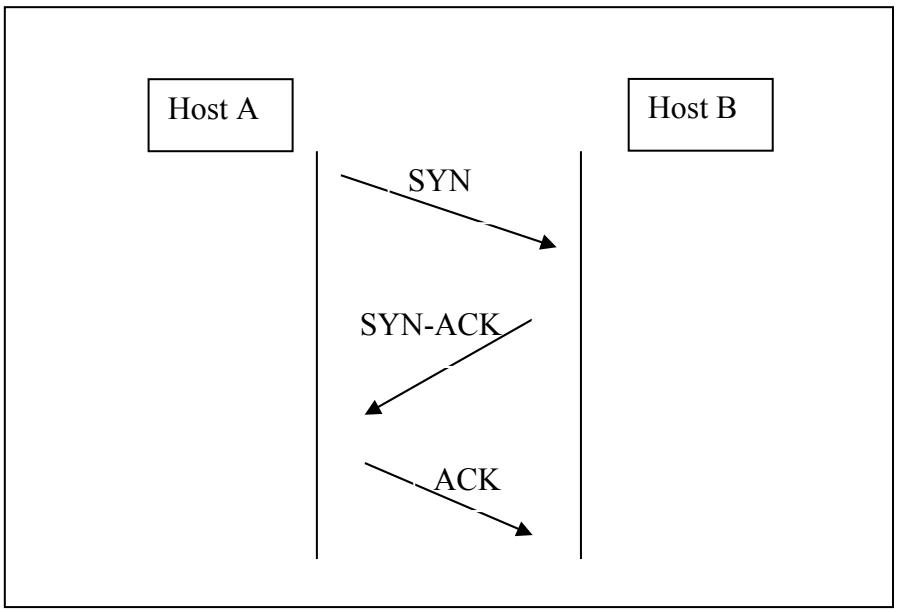
1.3.5. Three-way handshake

W protokole TCP do nawiązania połączenia pomiędzy dwoma hostami wykorzystywana jest procedura nazwana *three-way handshake*. W sytuacji normalnej jest ona rozpoczynana, gdy host A chce nawiązać połączenie z hostem B, procedura wygląda następująco:

1. host A wysyła do hosta B segment SYN wraz z informacją o dolnej wartości numerów sekwencyjnych używanych do numerowania segmentów wysyłanych przez host A a następnie przechodzi w stan SYN-SENT,

2. host B, po otrzymaniu segmentu SYN, przechodzi w stan SYN-RECEIVED i, jeżeli również chce nawiązać połączenie, wysyła hostowi A segment SYN z informacją o dolnej wartości numerów seq używanych do numerowania segmentów wysyłanych przez host B, oraz segment ACK z polem numeru sekwencji ustawionym na wartość o jeden większą, niż wartość pola sekwencji pierwszego segmentu SYN hosta A.
3. host A, po odebraniu segmentów SYN i ACK od hosta B przechodzi w stan ESTABLISHED i wysyła do niego segment ACK potwierdzający odebranie segmentu SYN (numer sekwencji ustawiony na 301)
4. host B odbiera segment ACK i przechodzi w stan ESTABLISHED
5. host A może teraz rozpocząć przesyłanie danych

Jeśli host odbierający połączenie nie chce lub nie może odebrać połączenia, powinien odpowiedzieć pakietem z ustawioną flagą RST (ang. *reset*).



Three-way hanshake

1.3.6. Podsłuchiwanie w sieci

Od warstwy łącza danych zależy czy sieć pracuje w trybie przełączania ramek. W przypadku sieci nieprzełączanej pakiety ethernetowe są przekazywane przez wszystkie urządzenia w sieci, ponieważ podczas konstruowania standardu przyjęto założenie, że urządzenie sieciowe będzie badało tylko pakiety skierowane na jego adres sieciowy. Okazuje się jednak, że znaczną część z dostępnych kart sieciowych można przestawić w tryb odbierania (ang. promiscuous mode), dzięki czemu będzie możliwe badanie wszystkich pakietów, niezależnie od ich adresu docelowego. Programy typu sniffer domyślnie ustawiają taki tryb pracy interfejsu, jednak jeżeli tego nie zrobią można tego dokonać za pomocą polecenia:

```
# ifconfig eth0 promisc
```


Część usług takich jak POP3, http, FTP domyślnie nie szyfruje przesyłanych informacji i właśnie za sprawą tego, w snifferach widzimy zagrożenie bezpieczeństwa. Potencjalny agresor nawet o elementarnej wiedzy na temat narzędzi do przechwytywania pakietów, może stanowić zagrożenie bezpieczeństwa jeżeli znajduje się w tej samej sieci co my, odpowiednio długo monitoruje przesyłane pakiety i poświęci czas na analizę danych. Powinno to ilustrować dlaczego należy przywiązywać uwagę do konstrukcji sieci oraz jej szyfrowania. Infrastruktura rozmieszczenia urządzeń sieciowych i podzielenia jej na domeny kolizyjne, musi nam zagwarantować że nikt niepowołany nie będzie mógł podpiąć się do sieci i analizować naszych informacji. Analogiczne zagrożenie ma miejsce w wypadku sieci bezprzewodowych. Sieć nieszyfrowana za pomocą bezpiecznych protokołów takich jak WPA2 i o prostym do złamania haśle (lub wręcz go pozbawiona) umożliwia dowolnej osobie podpiętej do danego Access Pointa przechwytywanie naszych pakietów, a tym samym uzyskanie poufnych informacji. Aby temu przeciwdziałać należy korzystać wyłącznie z sieci szyfrowanych, a jeżeli nie jest to możliwe, stosować protokół SSL w używanych aplikacjach.

1.4. Cel laboratorium

- Celem labolatorium jest zaprezentowanie metod nasłuchu w sieci za pomocą trzech programów. Zostaną pokazane metody identyfikacji różnorodnych typów pakietów i interpretacji informacji zawartych w tych pakietach. Następnie przedstawimy potencjalne ryzyko związane z nieodpowiedną konstrukcją sieci oparnej na przełączniku oraz niezaszyfrowanego WLAN, aby uświadomić jak łatwo osoba niepowołana może odczytać informacje, które moglibyśmy chcieć ukryć przed światłem dziennym.

2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

2.1. Przygotowanie laboratorium

Aby wykonać zadania w labolatorium należy pobrać programy wireshark oraz tcpdump jeżeli nie są preinstalowane w systemie. W systemach opartych na Debianie robimy to za pomocą: (należy to robić z prawami administratora)

```
# apt-get wireshark
# apt-get tcpdump
```

2.2. Zadanie 1. Prosty sniffer

Pierwszym zadanie polega na skompilowaniu i wykorzystaniu bardzo prostego sniffera. Został on przygotowany na te zajęcia w oparciu o bibliotekę libpcap i ma jedynie możliwość przechwytywania pakietów TCP/UDP/ICMP.

2.2.1. Uruchomienie programu

1. Pobierz niezbędne do kompilacji pakiety:
   ```
   $ sudo su
   # apt-get install libpcap
   # apt-get install libpcap-dev
   ```
2. Skompiluj kod źródłowy za pomocą:
   ```
   # gcc prosty_sniffer.c –lpcap –o sniffer
   ```
3. Uruchom go za pomocą:
   `# ./sniffer [nazwa_interfejsu]` np. `./sniffer eth0`
   Uwaga! Musisz posiadać prawa administratora!
4. Podaj ilość pakietów, jaką chcesz przechwycić. Zalecane jest podanie liczby ok. *100*
5. Podaj nazwę protokołu, który chcesz przechwytywać. Możesz to zrobić za pomocą słów *tcp*, *icmp*, *ip*, *udp.* Na początek podaj *ip* aby monitorować dowolny ruch.

Parametr/opcja	Opis
`ip`	Przechwyć wszystkie pakiety
`tcp`	Wyłącznie pakiety TCP
`tcp port 53`	Wyłącznie pakiety TCP pod portem 53
`ip host 42.32.3.4`	Przechwyć wszystkie pakiety od/do hosta 42.32.3.4

6. Skontroluj czy w sieci w sali komputerowej są przesyłane jakiekolwiek informacje i jeżeli jest, przeanalizuj adresy źródłowe i docelowe.

2.2.2. Odczytanie niezabezpieczonych danych użytkownika

1. Niech pierwszy członek zespołu uruchomi na swoim komputerze sniffer dla dużej liczby pakietów (ok. 1000) i protokołu tcp.
2. Druga osoba z grupy odwiedza dowolną witrynę, o której wiadomo, że nie szyfruje autoryzacji za pomocą SSL np. *http://haxite.org.* Należy podać wymyślone dane w celu próby zalogowania.
3. Gdy już zostanie zarejestowana odpowiednio duża liczba pakietów tcp, aby program zakończył działanie, student na której komputerze uruchomiony jest sniffer, powinien włączyć plik *log.txt.* Plik ten zostanie utworzony w folderze i zawiera przechwyconą treść pakietów tcp w formie znaków ASCII. W pliku tym spróbować znaleźć podane przez kolegę wymyślone dane autoryzacyjne.

2.2.3. Rozbudowa sniffera

Jako zadanie dodatkowe możesz spróbować rozszerzyć możliwości dostarczonego sniffera poprzez wypisywanie na ekran zawartości pakietów innego typu niż tcp, bądź bardziej zaawansowane filtry.

2.3. Zadanie 2. Wireshark.

Wireshark jest najbardziej znanym graficznym programem do analizy ruchu w sieci. Zadanie polega za zbadaniu, w jaki sposób nasz komputer nawiązuje połączenie z innymi komputerami za pomocą prostych usług.

1. Uruchom program wireshark i za pomocą przycisku '*List the available capture interfaces'* wybierz używany interfejs sieciowy i rozpocznij nasłuchiwanie.

```
#wireshark
```

2. Za pomocą programu **ping** uruchom na drugim komputerze wysyłanie pakietów ICMP. Możesz to zrobić np. z interwałem czasowym 0.1s i ilością 100 pakietów na witrynę google'a.

```
#ping –i 0.1 –c 100 173.194.32.183
```

3. Sprawdź budowę i zawartość ramek ICMP Request oraz ICMP Reply
4. Nasłuchuj sieć za pomocą filtru `arp` w poszukiwaniu pakietów oraz ana logicznie do podpunktu 3. przebadaj ich zawartość i porównaj działanie do przedstawionego w wstępie teoretycznym.
5. Podobnie jak w poprzednim zadaniu spróbuj zalogować się na dowolną stronę nie posiadającą szyfrowania i za pomocą filtru http i poszukaj ciągu znaków odpowiadającemu danym autoryzacyjnym.
6. Następnie zaloguj się na stronę posiadającą szyfrowanie np. SSL lub za pomocą tunelu ssh i sprawdź w jakiej formie są przesyłane dane.
7. Za pomocą serwisu moodle AGH prześlij wiadomość do partnera, a następnie skontroluj czy została ona zarejestrowana w postaci jawnej. Jakie zagrożenia stwarza przesyłanie rozmów w postaci otwartego tekstu?
8. Włącz filtr http i ogranicz Połącz się z rozbudowanym serwisem takim jak serwis społecznościowy a następnie połącz się ze swoją własną, prostą witryną i porównaj ilość wymienianych danych. W ilu pakietach HTTP i TCP przyszły odpowiedzi?
9. (dodatkowe) Utwórz swoją witrynę, która przesyła informacje za pomocą HTTP GET i HTTP POST i za pomocą wiresharka porównaj nagłówki.
10. Uruchom program traceroute służący do badania węzłów pośredniczących w komunikacji między hostami i sprawdź w snifferze jak zmienia się pole TTL pakietu IP.

```
#traceroute www.google.com
```


2.4. Zadanie 3. Three-way handshake

Zadanie polega na analizie działania mechanizmu nawiązywania połączenia TCP. Rozpocznij połączenie z dowolnym serwerem, zakończ ją, a następnie:

1. Policz ilość pakietów w jednej sesji
2. Kto inicjuje połaczenie?
3. Jakie flagi znajdziemy w kolejnych pakietach TCP?
4. Kto rozpoczyna proces zerwania połączenia?
5. Jakie flagi są wówczas ustawiane?

2.5. Tcpdump

Tcpdump to działające w trybie tekstowym proste, aczkolwiek potężne narzędzie do nasłuchu sieci, tzw. *sniffer* (*sniff* – ang. węszyć). Program ten wykorzytuje się głównie w następujących celach:

- analiza pakietów, które przepływają przez sieć
- przechwytywanie informacji wysyłanych przez innych użytkowników.

Jak wszystkie sniffery, tcpdump wykorzystuje tryb *promiscuous mode* (ang. tryb mieszany) karty sieciowej, przez co odbiera wszystkie pakiety krążące z sieci, a nie tylko te, które są do niej adresowane. Program domyślnie uruchamia kartę sieciową w trybie *promiscuous,* jednak istnieje możliwość uruchomienia programu bez włączania trybu *promiscuous* – aby to uczynić, wystarczy użyć opcji `-p`.

Program nie jest domyślnie instalowany w systemie Linux Debian – należy pobrać go za pomocą polecenia:

```
#apt-get install tcpdump
```

lub, przypadku awersji do poleceń konsolowych można wykorzystać narzędzie Synaptic Packet Manager. tcpdump do pracy wymaga uprawnień administratora.

Do uruchomienia programu wystarczy komenda `#tcpdump` – domyślnie przyjęte zostaną następujące ustawienia:

- interfejs nasłuchu `eth0`,
- brak wyświetlania zawartości pakietu – wyświetlane zostaną tylko nagłówki,

wyniki nie będą filtrowane ze względu na protokół bądź port.

Aby odfiltrować pakiety danego protokołu należy podać jego nazwę albo port:

np. ARP, ICMP, TCP, UDP: `#tcpdump <parametry> arp|icmp|tcp|udp`
ale np. HTTP: `#tcpdump <parametry> port 80`
FTP: `#tcpdump <parametry> port 21` (lub `#tcpdump <parametry> port 20` – dlaczego?)
Ważne parametry:

Parametr/przełącznik	Opis
`-i <interfejs>`	Ustawia nasłuch na podany interfejs
`-c <liczba>`	Wyłącza nasłuch po odebraniu liczby pakietów
`-q`	Wyświetla nagłówki odebranych pakietów w skóconej formie
`-x`	Wyświetla zawartość pakietów w trybie heksadecymalnym
`-X`	Wyświetla zawartość pakietów w trybie ASCII

Listę wszystkich parametrów zobaczyć można po wykonaniu: `#man tcpdump`

Aby przekierować wyniki do pliku, wystarczy użyć poleceń:

```
#tcpdump <...> > plik
```

lub

```
#tcpdump <...> >> plik
```

Pozwalają one na zapis do pliku, który możemy otworzyć dowolnym edytorem tekstu.

Informacje zwrócone przez tcpdump-a mogą być swobodnie przetwarzane przez programy takie jak `grep`, `awk`, `sort` itp.

Korzystając z podanych informacji wykonaj następujące zadania:

2.6. Przechwytywanie sesji POP3 pomiędzy klientem telnet a serwerem poczty elektronicznej

1. Załóż nowe konto poczty elektronicznej na serwerze wykorzystującym nieszyfrowany SSL-em protokół POP3 np. na *http://poczta.interia.pl/,* lub użyj już istniejącego nie ujawniając danych autoryzacyjnych swojemu partnerowi z zespołu.
2. Wyślij z innego konta poczty elektronicznej na nowo utworzony kilka wiadomości o krótkiej treści.

3. Uruchom program tcpdump tak, aby przechwytywał pakiety protokołu POP3 (port 110).
4. Na drugim komputerze połącz się z serwerem korzystając z programu telnet:

   `#telnet <nazwa_serwera> 110` (w naszym przykładzie <nazwa_serwera> to poczta.interia.pl)

   a następnie zaloguj się wg. poniższego schematu:

   `USER <login>`

   `PASS <hasło>`

   odpowiedź w stylu: `+OK Welcome aboard! You have x messages`. będzie świadczyła o tym, że autoryzacja zakończyła się powodzeniem.

1. Uzyskaj listę wiadomości poleceniem LIST. Otrzymasz listę maili znajdujących się aktualnie na skrzynce w postaci: `<numer> <rozmiar_wiadomości_w_bajtach>`
2. Odczytaj wybranego maila poleceniem `RETR <numer_maila>`
3. Zamknij połączenie wpisując `QUIT`

Przeanalizuj zawartość przechwyconych pakietów pod kątem nazwy użytkownika, hasła, zawartości listy wiadomości oraz treści odczytanych wiadomości.
Jakie niebezpieczeństwa niesie ze sobą nieszyfrowana sesja POP3? Jakie technologie temu zapobiegają?

Zestawianie i analiza sesji FTP

1. Na jednym komputerze uruchom dwa okna terminali, w jednym w nich uruchom program tcpdump nasłuchujący na porcie 20, a drugi na porcie 21.
2. Wykorzystując program konsolowy ftp połącz się na drugiej stacji roboczej z dowolnym serwerem obsługującym ten protokół (przykładowo `ftp.pl.debian.org`), i zaloguj się na niego, np. za pomocą konta anonymous.
3. Poprzemieszczaj się pomiędzy katalogami wykorzystując polecenia `cd` i `ls` oraz pobierz wybrany plik (najlepiej tekstowy) poprzez polecenie `get <nazwa_pliku>`
4. Zamknij połączenie w serwerem wpisując polecenie disconnect.
5. Po zapoznaniu się z zawartością pakietów zebranych z obu portów przeanalizuj poniższe zagadnienia:
   a) Jakimi danymi użytkownik autoryzował dostęp do serwera i jakie pliki pobierał?
   b) Jak wygląda zestawienia połączenia klient-serwer przez protokół FTP?
   c) Dlaczego FTP jest protokołem dwupołączeniowym? Do czego służą poszczególne protokoły?

