

# ***PODSŁUCH W SIECIACH ETHERNET***

## ***SSL – PRZECIWDZIAŁANIE PODSŁUCHOWI***

# Przyczyny

Najpowszechniej używane protokoły sieciowe powstały gdy:

- w Internecie nie było tylu zagrożeń ile jest dziś
- komputery nie były tak szybkie,  
by szyfrować dane on-line

Dlatego też dziś występuje problem podsłuchiwania. Można podsłuchać większość z komunikacji sieciowej, a narzędzia umożliwiające taki podsłuch są ogólnodostępne.

# Protokoły zagrożone

- **HTTP** - serwisy bankowe itp.
- **FTP** - przesyłanie poufnych plików
- **Telnet** - praca ze zdalnym systemem
- **POP3, IMAP** - protokoły poczty elektronicznej

## Dodatkowo w sieciach lokalnych:

- **SMB** - protokół udostępniania plików i drukarek
- **NFS** - sieciowy system plików
- komunikację klientów z serwerami baz danych  
(np. połączenia ODBC)

# Rodzaje metod podsłuchu sieci

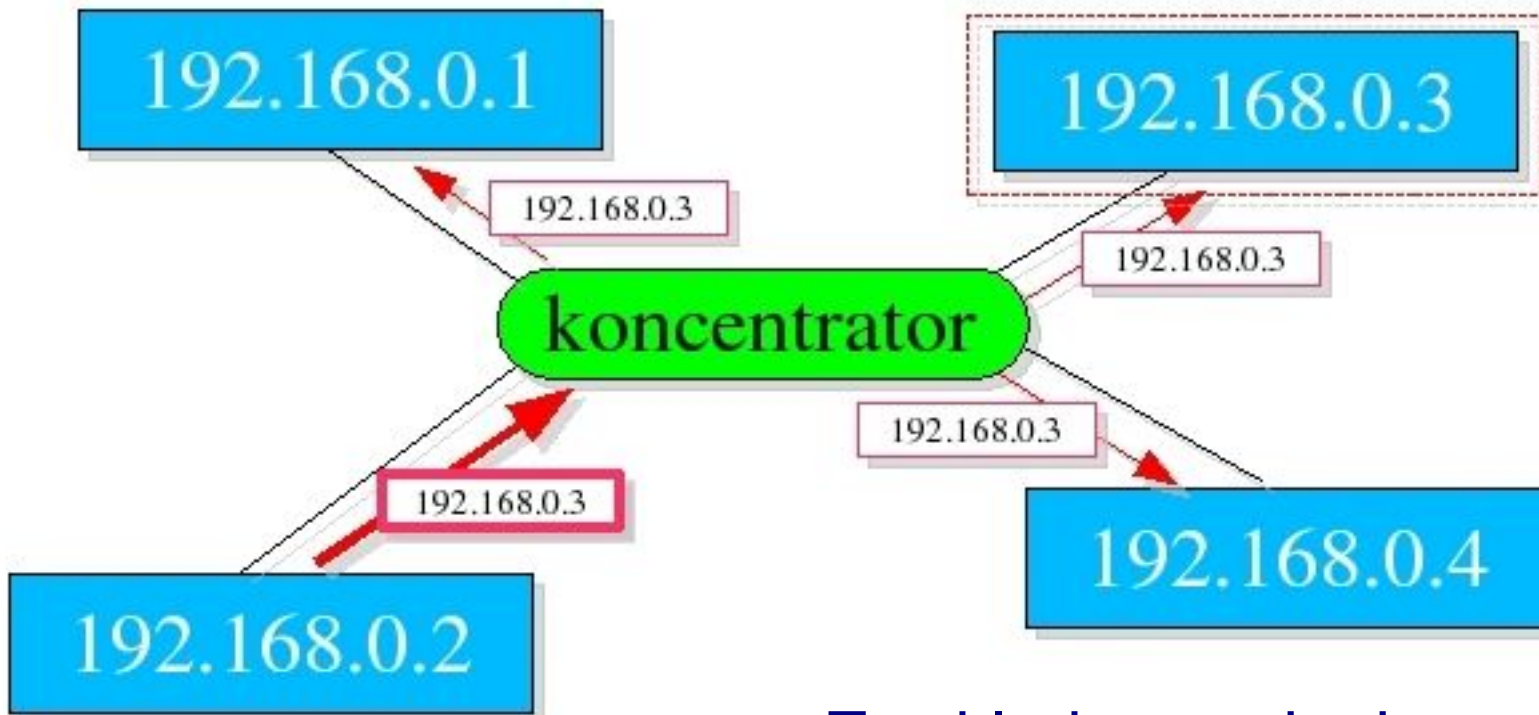
- podsłuch pasywny
- podsłuch aktywny

## Inne metody podsłuchu:

- podsłuch sieci bezprzewodowych
- podsłuch monitorów CRT
- podsłuch klawiatury

# Dostęp pasywny - SNIFFING

- Proste sieci – topologia gwiazdy
- Pakiety docierają do wszystkich komputerów



Zwykle karta sieciowa ignoruje pakiety skierowane do innych komputerów, ale można to zmienić przełączając ją w tryb nasłuchiwania.

# Tryb nasłuchiwania (*promiscuous*)

- adresowanie oparte jest na fizycznych adresach kart sieciowych (MAC)
- zwykle pakiety obce są ignorowane
- w trybie *nasłuchiwania* karta czyta również pakiety do zaadresowane do innych kart

## Uruchamianie trybu *nasłuchiwania*:

- ręcznie – *ifconfig* (parametr *promisc*)
- automatycznie przy uruchamianiu narzędzi

# Narzędzia do podstuchu (*sniffery*)

W większości oprogramowanie Open Source. Głównie pochodzi ze środowisk unix'owych.

## Funkcjonalności:

- rejestracja samych nagłówek pakietów
- ekstrakcja najważniejszych danych
- podstuch w czasie rzeczywistym

## Przydatne przy:

- wykrywaniu nieprawidłowości w działaniu sieci
- wykonywaniu pomiarów i analizie działania sieci

# tcpdump

Pozwala na przechwycenie ruchu sieciowego (nie analizuje)

```
wget http://rainbow.mimuw.edu.pl/SO/PUBLIC-SO/2004-05/Zadania/zadanie1.txt
```

```
15:58:07.792957 83.28.137.149.32931 > 193.0.96.128.http: P 1:149(148) ack 1 win 5808 (DF)
```

0x0000	4500	00c8	2e57	4000	4006	0da7	531c	8995
0x0010	c100	6080	80a3	0050	02f7	01af	6385	222a
0x0020	8018	16b0	a2a4	0000	0101	080a	0013	2fff
0x0030	b420	abde	4745	5420	2f53	4f2f	5055	424c
0x0040	4943	2d53	4f2f	3230	3034	2d30	352f	5a61
0x0050	6461	6e69	612f	7a61	6461	6e69	6531	2e74
0x0060	7874	2048	5454	502f	312e	300d	0a55	7365
0x0070	722d	4167	656e	743a	2057	6765	742f	312e
0x0080	382e	320d	0a48	6f73	743a	2072	6169	6e62
0x0090	6f77	2e6d	696d	7577	2e65	6475	2e70	6c0d
0x00a0	0a41	6363	6570	743a	202a	2f2a	0d0a	436f
0x00b0	6e6e	6563	7469	6f6e	3a20	4b65	6570	2d41
0x00c0	6c69	7665	0d0a	0d0a				

```
E....W@.@...S...
..`....P....c."*
...../
...GET./SO/PUBL
IC-SO/2004-05/Za
dania/zadanie1.t
xt.HTTP/1.0..Use
r-Agent:.Wget/1.
8.2..Host:.rainb
ow.mimuw.edu.pl.
.Accept:.*/*..Co
nnection:.Keep-A
live....
```



# tcpdump

```
15:58:07.968289 193.0.96.128.http > 83.28.137.149.32931: . 317:1757(1440)
ack 149 win
5792 (DF)
```

```
0x0000 4500 05d4 8e26 4000 3906 afcb c100 6080
0x0010 531c 8995 0050 80a3 6385 2366 02f7 0243
0x0020 8010 16a0 90e9 0000 0101 080a b420 ac27
0x0030 0013 2fff 5a61 6461 6e69 6520 3120 2873
0x0040 6b72 7970 7429 0a2d 2d2d 2d2d 2d2d 2d2d
0x0050 2d2d 2d2d 2d2d 2d2d 2d0a 0a57 737a 7973
0x0060 746b 6965 207a 6164 616e 6961 2064 6f20
0x0070 7465 6a20 706f 7279 2028 7072 7a79 6e61
0x0080 6a6d 6e69 656a 2077 2074 656f 7269 6929
0x0090 2062 796c 7920 7072 7a65 7379 6c61 6e65
```

```
E....&@.9.....`.
S...P..c.#f...C
.....'
../.Zadanie.1.(s
krypt).-----
-----..Wszys
tkie.zadania.do.
tej.pory.(przyna
jmniej.w.teorii)
.byly.przesylane
```

# Ethereal

- narzędzie graficzne (m.in. Linux, Windows)
- dekodowanie pakietów na żywo (*live decoding*)
- analiza danych wczytanych z pliku
- rekonstrukcja całej sesji TCP (*Follow TCP Stream*)

The screenshot displays the Ethereal network protocol analyzer interface. The main window is titled "<capture> - Ethereal" and contains a menu bar (File, Edit, Capture, Display, Tools, Help) and a list of captured packets. The packet list is as follows:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
2	0.000310	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
3	0.000376	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
4	0.000437	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
5	0.000498	192.168.0.1	192.168.0.1	TELNET	Telnet Data ...
6	0.000559	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
7	0.000642	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
8	0.000650	127.0.0.1	127.0.0.1	TCP	32920 > telnet [ACK] Seq=412
9	0.001044	192.168.0.1	192.168.0.1	TELNET	Telnet Data ...
10	0.001058	192.168.0.1	192.168.0.1	TCP	32919 > telnet [ACK] Seq=402
11	0.001134	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...

Below the packet list, the details for the selected packet (Frame 3) are shown:

- Frame 3 (68 bytes on wire, 68 bytes captured)
- Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
- Internet Protocol, Src Addr: 127.0.0.1 (127.0.0.1), Dst Addr: 127.0.0.1 (127.0.0.1)
- Transmission Control Protocol, Src Port: 32917 (32917), Dst Port: telnet (23), Seq: 3732916266,
- Telnet

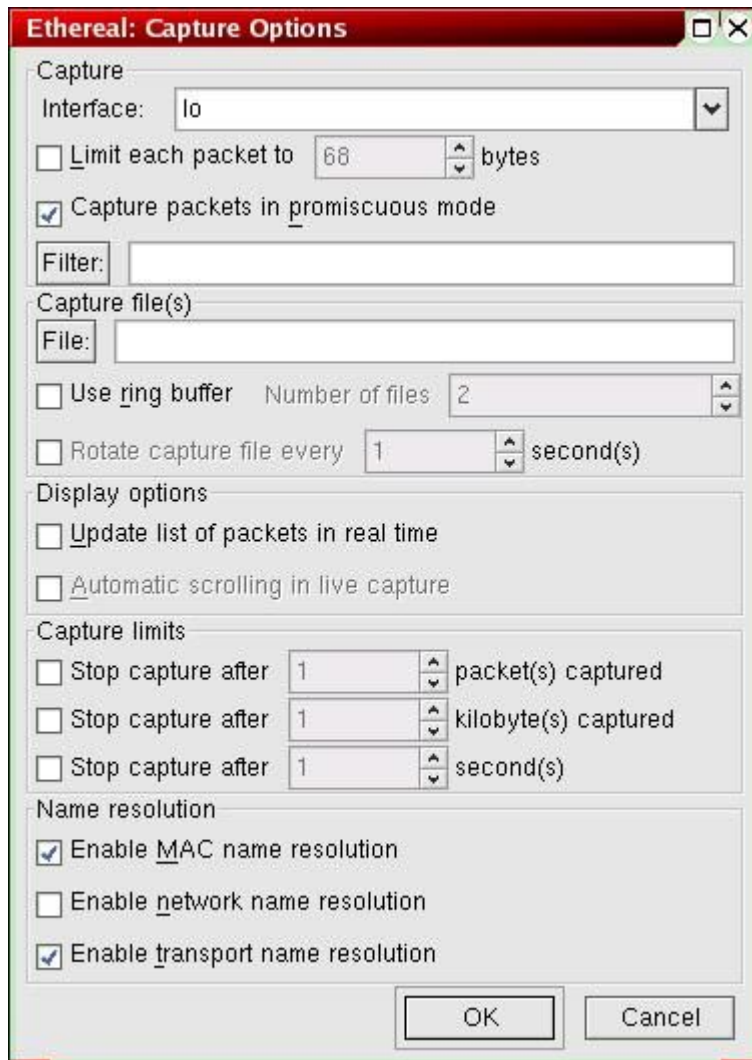
The bottom section of the interface shows the raw data in hexadecimal and ASCII:

```
0000 00 00 00 00 00 00 00 00 00 00 00 08 00 45 10  .....E.
0010 00 36 83 19 40 00 40 06 b9 96 7f 00 00 01 7f 00  .6..@.0. 1.....
0020 00 01 80 95 00 17 de 7f c8 2a de bd db fc 80 18  .....P. ẽ*PÜ..
0030 7f ff 63 54 00 00 01 01 08 0a 00 13 d4 35 00 13  .ÿcT.... ....õ5..
0040 d1 ef 0d 00                                     ñi..
```

The interface also includes a filter field, a "Filter:" label, and buttons for "Reset" and "Apply". The status bar at the bottom indicates "File: <capture> Drops: 0".

*Podstuchana sesja Telnet*

# Ethereal



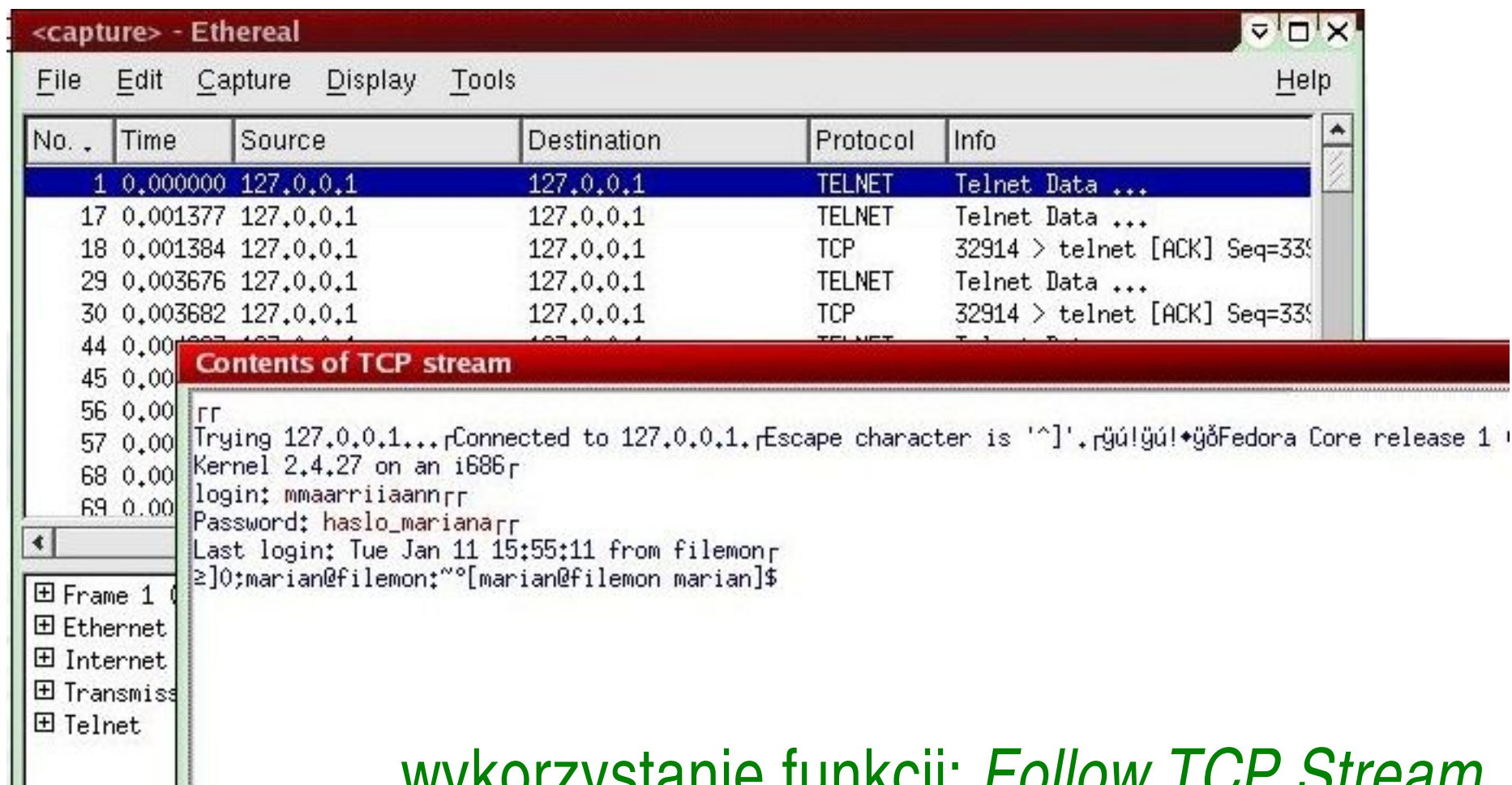
... zanim zaczniemy podstuchiwanie, wybieramy kilka parametrów, m. in.:

- interfejs sieciowy
- filtr
- czy tłumaczyć adresy IP na nazwy
- ...

*Podstuchana sesja Telnet*

# Ethereal

## Odtworzona sesja Telnet



The screenshot shows the Ethereal interface with a captured Telnet session. The main window displays a list of packets, and a 'Contents of TCP stream' window is open, showing the raw data of a Telnet connection, including the login prompt and user input.

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
17	0.001377	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
18	0.001384	127.0.0.1	127.0.0.1	TCP	32914 > telnet [ACK] Seq=339
29	0.003676	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
30	0.003682	127.0.0.1	127.0.0.1	TCP	32914 > telnet [ACK] Seq=339
44	0.003687	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
45	0.003693	127.0.0.1	127.0.0.1	TCP	32914 > telnet [ACK] Seq=339
56	0.003699	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
57	0.003705	127.0.0.1	127.0.0.1	TCP	32914 > telnet [ACK] Seq=339
68	0.003711	127.0.0.1	127.0.0.1	TELNET	Telnet Data ...
69	0.003717	127.0.0.1	127.0.0.1	TCP	32914 > telnet [ACK] Seq=339

**Contents of TCP stream**

```
rr
Trying 127.0.0.1... Connected to 127.0.0.1. Escape character is '^'.
Fedora Core release 1
Kernel 2.4.27 on an i686
login: mmaarriiaannrr
Password: haslo_marianarr
Last login: Tue Jan 11 15:55:11 from filemonr
[~]0;marian@filemon:~°[marian@filemon marian]$
```

wykorzystanie funkcji: *Follow TCP Stream*

# Pakiet *dsniff*

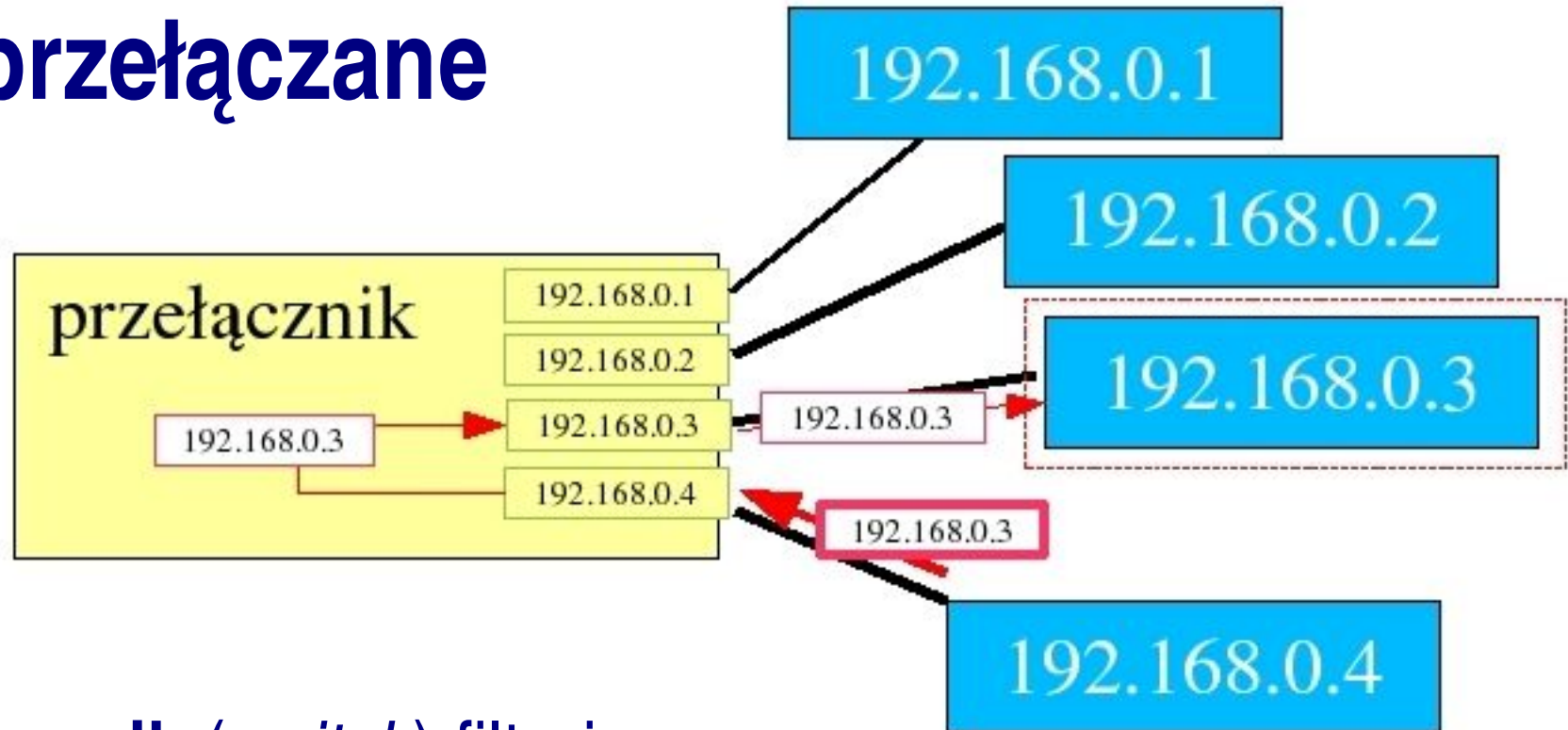
Program *dsniff* z pakietu o tej samej nazwie obsługuje wiele protokołów. Wykorzystuje fakt, że poprzez sieć hasła są przesyłane po określonym ciągu znaków (np. USER, PASS):

*Podstuchane hasła  
Telnet i FTP:*

```
marcin@filemon:~  
[root@filemon root]# /usr/local/sbin/dsniff -i lo  
dsniff: listening on lo  
-----  
01/11/05 16:24:38 tcp filemon.talala.32941 -> filemon.talala.21 (ftp)  
USER marian  
PASS haslo_mariana  
-----  
01/11/05 16:26:01 tcp filemon.talala.32945 -> filemon.talala.21 (ftp)  
USER marian  
PASS haslo_mariana  
-----  
01/11/05 16:26:21 tcp filemon.talala.32944 -> filemon.talala.23 (telnet)  
[marian]  
haslo_mariana  
exit
```

# **PODSŁUCH W SIECIACH PRZEŁĄCZANYCH – DOSTĘP AKTYWNY**

# Sieci przełączane



- **przełącznik** (*switch*) filtruje pakiety względem adresów przeznaczenia
- zmniejszone obciążenie sieci
- wzrasta bezpieczeństwo
- ... ale podstuch nadal możliwy ...

## Atak *MAC FLOODING*

### Cel:

- zapełnienie tablicy adresów służącej do translacji adresów MAC na adresy IP (przełącznik działa jak koncentrator)

### Narzędzia:

- pakiet *dsniff* (do generowania losowych adresów MAC i wysyłania ich do sieci służy polecenie *macof*)

### Realizacja:

- przepelniamy tablicę adresów przy pomocy *macof* (polecenie uruchomione na jednym komputerze)
- podstuchujemy sieć na innym komputerze



## Atak ARP SPOOFING

- fałszowanie adresu komputera podsłuchującego

### Narzędzia:

- pakiet *dsniff* (program *arpspoof*)
- **Windoze Connection Inspector** (Windows)
- **ARPTools** (Windows)

## Atak ARP SPOOFING - realizacja

Założmy, że chcemy podsłuchać komunikacje między komputerami o adresach IP1 i IP2 mając do dyspozycji komputer o adresie IP3:

- włączamy na naszym komputerze przekazywanie pakietów:  
`echo 1 > /proc/sys/net/ipv4/ip_forward`
- przekonujemy komputer IP1, że nasz komputer ma adres IP2: `arp spoof -t IP1 IP2`
- analogicznie dla IP2: `arp spoof -t IP2 IP1`
- Teraz wszystkie pakiety przechodzą przez komputer IP3.

### Uwaga:

Niniejsza metoda podsłuchu jest skuteczna, ale ingeruje w trasę pakietów więc jest również łatwa do wykrycia (wystarczy użyć polecenia ping)

## Atak DNS SPOOFING

- Komputer śledzony wysyła do serwera DNS zapytanie o adres jakiegoś serwera (np. pocztowego)
- zanim serwer DNS odpowie, komputer podsłuchujący wysyła zafałszowaną odpowiedź i
- dzięki temu komputer podsłuchiwany zamiast z prawidłowym serwerem, komunikuje się z komputerem podsłuchującym.
- komputer podsłuchujący uzyskuje w ten sposób dane do uwierzytelniania

# Przeciwdziałanie podsłuchowi

## Używanie bezpiecznych usług sieciowych:

- **SSL/TLS** - poczta i transfer plików (zamiast POP3 i FTP)
- **SSH2** - zdalna praca w sieci (zamiast TELNET)
- stosowanie jednokrotnych haseł, których ewentualne podsłuchanie następuje w momencie ich dezaktualizacji

## UWAGA:

Trzeba sobie zdawać sprawę, że narzędzia (metody) zabezpieczające powinny być łączone (np. szyfrowanie danych podczas transmisji nie wystarczy, jeśli już na serwerze do którego te dane wysyłamy, nie istnieje zabezpieczenie przed niepowołanym dostępem).

# Wykrywanie podsłuchu - IDS

Do wykrywania podsłuchu często wykorzystuje się narzędzia zbiorczo oznaczane skrótem **IDS** (*Intrusion Detection System*).

## Działanie:

- monitorowanie ruchu sieciowego i porównywanie go z wewnętrzną bazą danych typowych ataków
- powiadamianie administratora o nieprawidłowościach, przerwanie zagrożonego połączenia, bądź wykonanie innych czynności (łącznie z kontratakami)

## Uwaga:

Szyfrowanie utrudnia działanie IDS'ów.

# Wykrywanie – sieci z koncentratorami

Lokalne sprawdzenie czy karta sieciowa działa w trybie nasłuchiwania:

- **Windows** - program **PromiscDetect** (<http://ntsecurity.nu/toolbox/promiscdetect>)
- **Linux** - polecenie **ifconfig** (flaga **PROMISC**)

## Sprawdzanie zdalne:

komputer z systemem Linux i kartą sieciową przełączoną w ten tryb często odpowiada na wszystkie pakiety

## Ale:

- Tryb promiscuous wcale nie musi oznaczać podsłuchu, np. używają go emulatory VMware.
- Istnieją łaty na jądro, które utrudniają wykrycie ([www.sOftpj.org/tools/aasniff.tgz](http://www.sOftpj.org/tools/aasniff.tgz), <http://downloads.securityfocus.com/tools/aass.c>)

# Wykrywanie – sieci z przełącznikami

## Narzędzia wykrywające:

- **AntiSniff** (Windows) ([www.securityfocus.com/tools/1004](http://www.securityfocus.com/tools/1004))
- **Sentinel** (Linux) ([www.packetfactory.net/projects/sentinel](http://www.packetfactory.net/projects/sentinel))
- **ARPwatch** - na bieżąco sprawdza poprawność tablic ARP  
([www.securityfocus.com/tools/142](http://www.securityfocus.com/tools/142))
- **ANASIL** - komercyjne narzędzie przeznaczone dla dużych sieci korporacyjnych

# Inne metody wykrywania podsłuchu

## Prowokacje:

- wysyłanie pakietów do nie istniejącego hosta komputer podsłuchujący będzie chciał sprawdzić
- wykorzystywanie fałszywych danych uwierzytelniających i nasłuchiwanie czy ktoś używa ich ponownie



# **SZYFROWANIE DANYCH - PROTOKÓŁ SSL**

# SSL

W celu wyeliminowania możliwości uzyskania przez osoby trzecie cennych informacji z podsłuchanej transmisji sieciowej **należy stosować szyfrowanie poufnych danych.**

**SSL** - Secure Sockets Layer - protokół służący do szyfrowanej komunikacji sieciowej.

## Umożliwia:

- identyfikację serwera
- szyfrowanie przesyłanych danych
- w wersji 3 SSL umożliwia także identyfikację użytkownika

# Certyfikat

Certyfikat elektroniczny jest mechanizmem pozwalającym skutecznie zidentyfikować osobę lub serwer w sieci Internet.

## Zawiera:

- informację, która pozwala stwierdzić osobie komunikującej się z właścicielem tego certyfikatu, że ów właściciel jest tym, za kogo się podaje
- publiczny klucz szyfrujący

Certyfikaty są nadawane przez zaufane instytucje trzecie, tzw. centra certyfikacji.

# Certyfikat serwera

Serwer wykorzystujący SSL posiada swój certyfikat, który umożliwia udowodnienie tożsamości.

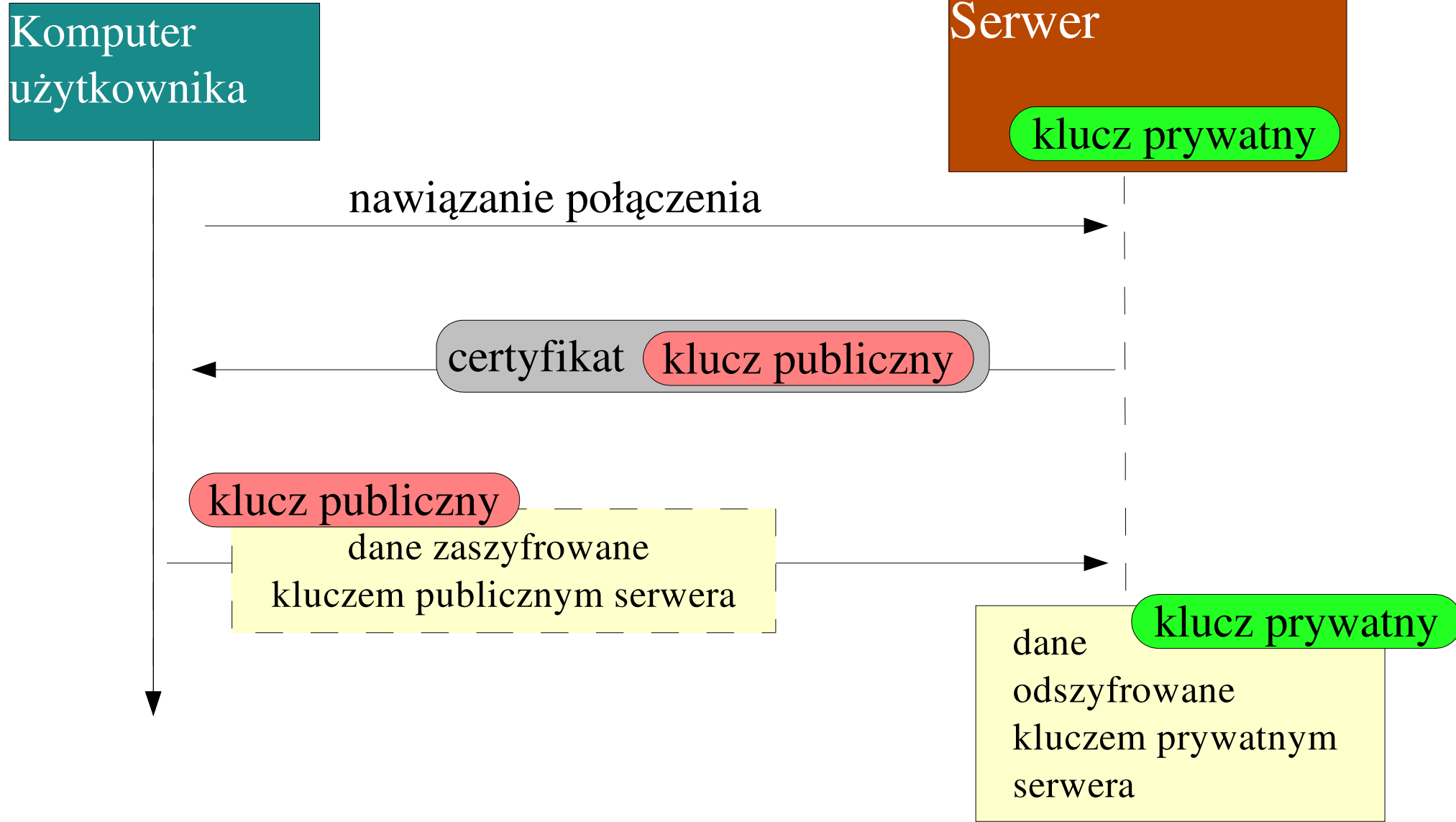
## Certyfikat serwera zawiera m.in.:

- **nazwę domeny serwera**; jeśli użytkownik łączy się z serwerem i okazuje się, że nazwa rzeczywista różni się od nazwy z certyfikatu, to użytkownik zostaje o tym poinformowany

## Działa dobrze gdy:

- użytkownik wykorzysta powyższą informację
- użytkownik jest świadomy, że serwer posiada certyfikat

# Certyfikat serwera



# Certyfikat użytkownika

- udostępniony w SSL w wersji 3
- udowadnia tożsamość użytkownika – nie trzeba przesyłać danych uwierzytelniających

## **W praktyce okazuje się, że:**

- Instytucje wydające certyfikaty nie przykładają zbytnej wagi do weryfikowania danych przy tworzeniu.
- Możliwość ataku Brute-force poprzez próbowanie stworzyć certyfikatów dla różnych użytkowników.
- Kradzież certyfikatów przy pomocy "koni trojańskich".

Z powyższych względów podpis elektroniczny ważnych dokumentów wykonywany jest przy pomocy specjalnych dodatkowych urządzeń i kart, które wykluczają możliwość skopiowania klucza prywatnego danej osoby.

# Sesja SSH - przykład

The screenshot shows a network capture in Wireshark. The main pane displays a list of packets with the following columns: No., Time, Source, Destination, Protocol, and Info. Packet 7 is highlighted, showing an SSHv2 connection from 127.0.0.1 to 127.0.0.1. The info field for packet 7 reads: "Server Protocol: SSH-1.99-OpenSSH\_3.6.1p2".

A secondary pane titled "Contents of TCP stream" is open, showing the raw data of the selected packet. The data is displayed in hexadecimal and ASCII. The ASCII portion shows the SSH protocol version negotiation:

```

SSH-1.99-OpenSSH_3.6.1p2
SSH-2.0-OpenSSH_3.6.1p2

```

The interface also shows a packet list on the left with filters like "Frame 7 (91 B)", "Ethernet II", "Internet Protocol", "Transmission Control Protocol", and "SSH Protocol". The bottom pane shows the hex and ASCII data for the selected packet.

---

## *Podśluch w sieciach Ethernet*