Wprowadzenie do systemów unixowych

Unix, GNU, Linux

Grzegorz J. Nalepa

AGH w Krakowie

wiosna 2015

Wykład: Sieci TCP/IP i ich obsługa w systemie GNU/Linux

Plan wykładu I

Wstęp do TCP/IP

Konfigurowanie TCP/IP

Demony sieciowe

Monitorowanie TCP/IP

Motto I

You live and learn.
Or you don't live long.
– Robert Heinlein

Plan punktu: Wstęp do TCP/IP

Wstęp do TCP/IP

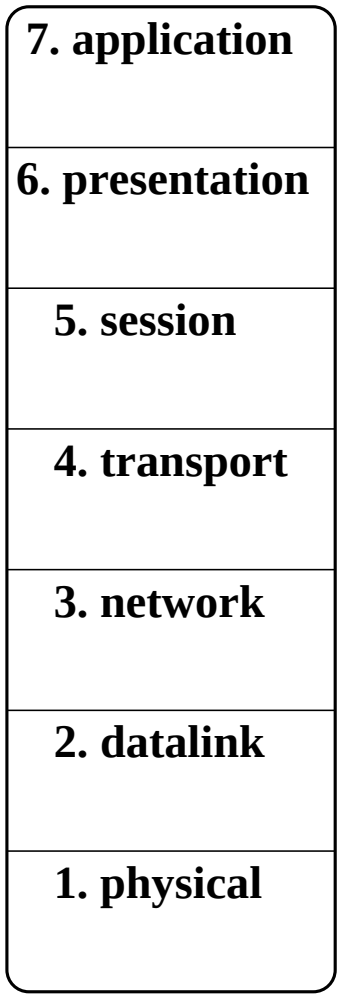
Struktura sieci otwartych I

- W latach 70. *International Standard Organization* określiła Standard: *Open System Interconnection*, znany później jako *OSI / ISO*.
- Wskazuje on na metody tworzenia i łączenia sieci otwartych.
- Opiera się na 7 niezależnych warstwach przepływu informacji.

Model OSI/ISO I

1. *Physical* – Fizyczna,
2. *Data Link* – Łącza danych,
3. *Network* – Sieci,
4. *Transport* – Transportowa,
5. *Session* – Sesji,
6. *Presentation* – Prezentacji,
7. *Application* – Aplikacji

Model OSI/ISO I



Pojęcia Ethernet I

- interfejs sieciowy
- ramka
- adres MAC
- łącze fizyczne, typy: 10/0/0Base2/T/TX/SX/LX
- *hub*,
- *switch*,
- r/ARP
- */etc/ethers*, adresy MAC i IP

Podstawowe pojęcia TCP/IP I

- protokoły: IP, ICMP, TCP, UDP
- pojęcia IP: adres, klasa adresów, maska
- pojęcie IP: routing, (r)DNS, (r)ARP
- pojęcia TCP: port, sesja

Sieci TCP/IP są sieciami z *komutacją pakietów*.

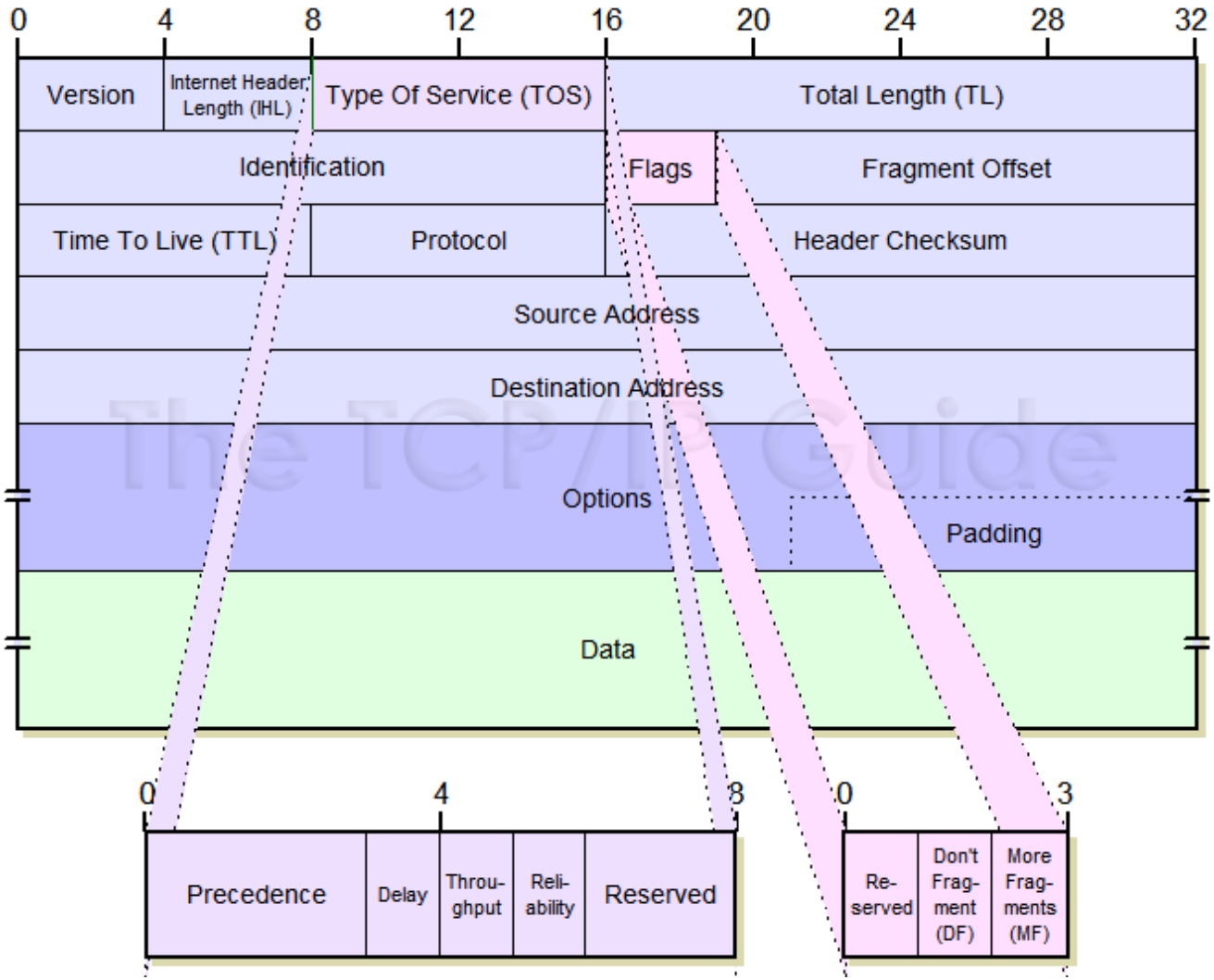
Protokoły TCP/IP I

- TCP: *Transmission Control Protocol*
- UDP: *User Datagram Protocol*
- IP: *Internet Protocol*
- ICMP: *Internet Control Message Protocol*

Pojęcia IP I

- pakiet (składa się z szeregu pól-części)
- adres/y (zawarte w pakiecie)
- routing – przekazywanie pakietów w sieci od nadawcy do adresata
- wersja protokołu: IPv4, IPv6
- r/DNS

Pakiet IP I



www.tcpipguide.com

Adresacja IP I

- adres interfejsu sieciowego w IPv4 jest liczbą 32b
- adresy w notacji szesnastkowej:
  0x00000000 – 0xFFFFFFFF
- adresy w notacji *kropkowo-dziesiętnej*:
  0.0.0.0 – 255.255.255.255
- w wersji IPv6 adres ma 6 bajtów
- (maska sieci, klasy adresów)

Klasy, sieci, Maski I

- każdy adres IP dzieli się na 2 cześci: adres sieci (*network*) i interfejsu (*host*)
- adres sieci jest opisywany przez ciąg starszych bitów (MSB) („lewych"),
- adres interfejsu jest opisywany przez ciąg młodszych bitów (LSB) („prawych),
- 3 MSB określają *klasę* adresu: A, B, C, D
- długość adresu sieci określa *maska sieci* (koniunkcja z adresem IP)
- adres sieci i *IP broadcast*
- sieci mogą być dzielone na podsieci wykorzystując kolejne bity z adresu hosta na adres podsieci.

Przykład I

```
addr: 149.156.199.59
Mask: 255.255.255.0
Net:  149.156.199.0
Bcast:149.156.199.255
```

Tabela adresów sieci I

Kl.	MSB	Zakres	Maska	Prywatne
A	0	0-127	255.0.0.0	10.0.0.0/8
B	10	128-191	255.255.0.0	172.16–31.0.0/16
C	110	192-223	255.255.255.0	192.168.0.0/24
D	1110	224-254	255.255.255.0	–

Routing IP I

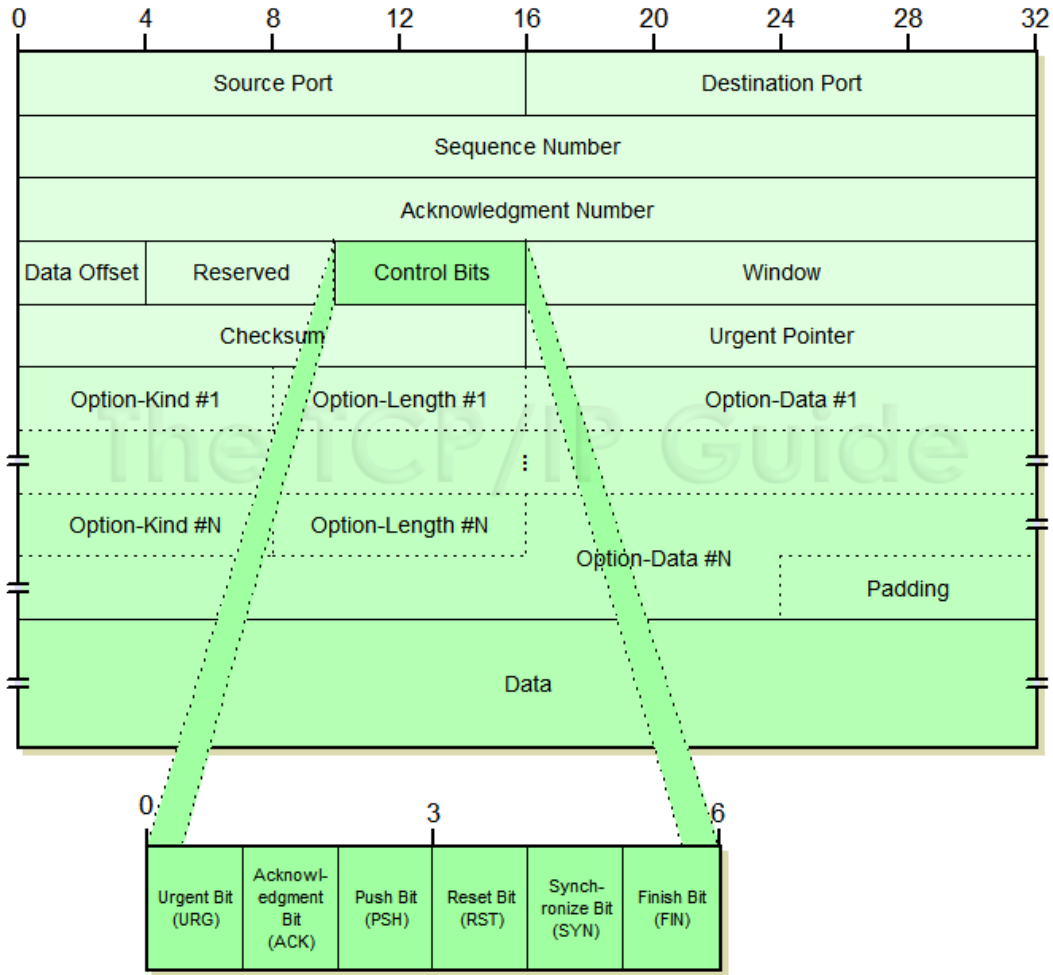
Podstawowe pojęcie w sieciach IP.

- *routing* – określanie, odnajdywanie, wyznaczanie trasy pakietów IP w sieci,
- routing może być *statyczny* (w tym *domyślny*), *dynamiczny*,
- w przypadku łączenia różnych sieci IP konieczny jest router,
- każdy system pracujący w sieci musi mieć w podstawowy sposób skonfigurowany routing,

Pojęcia TCP I

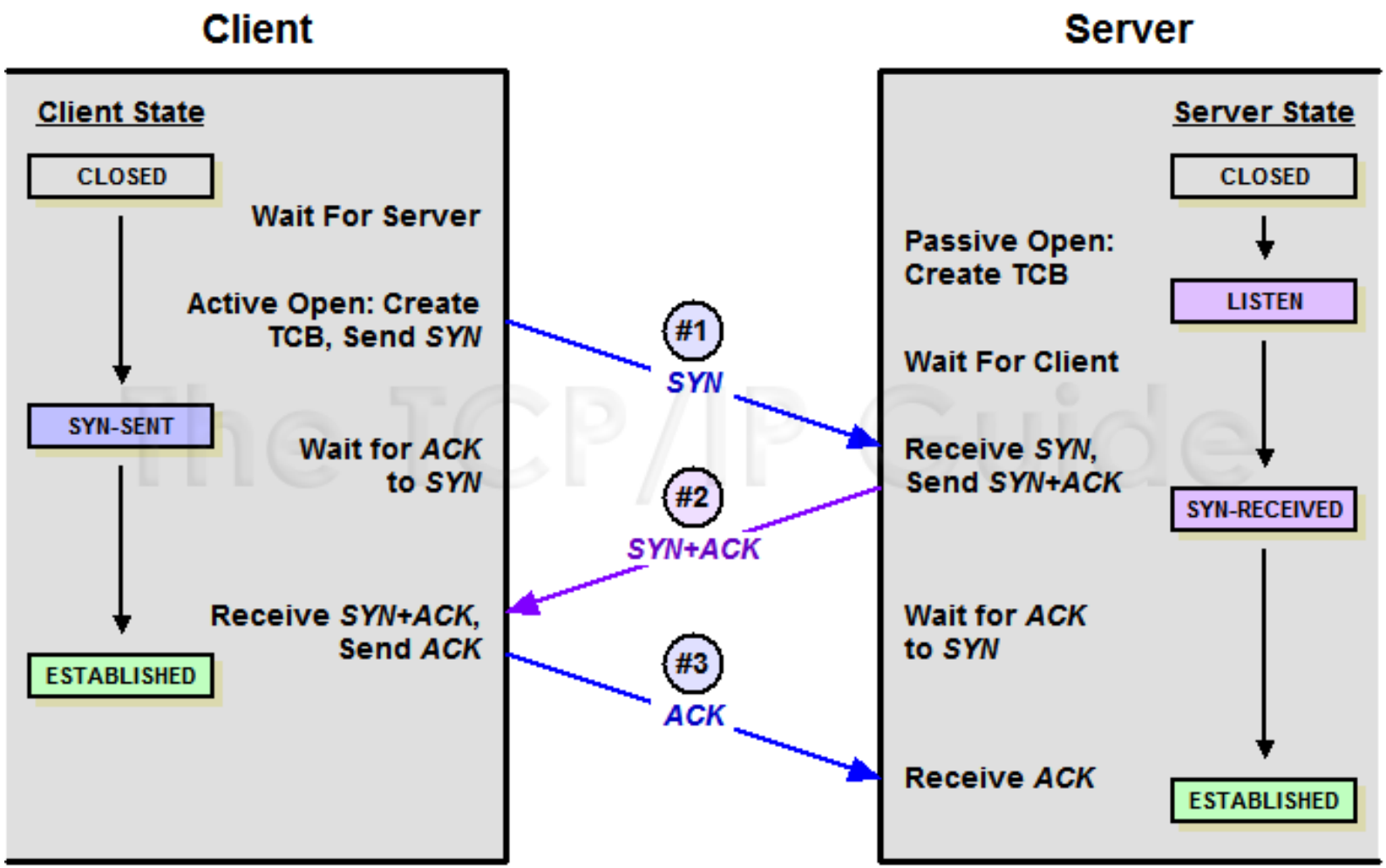
- pakiet
- port
- sesja

Pakiet TCP I



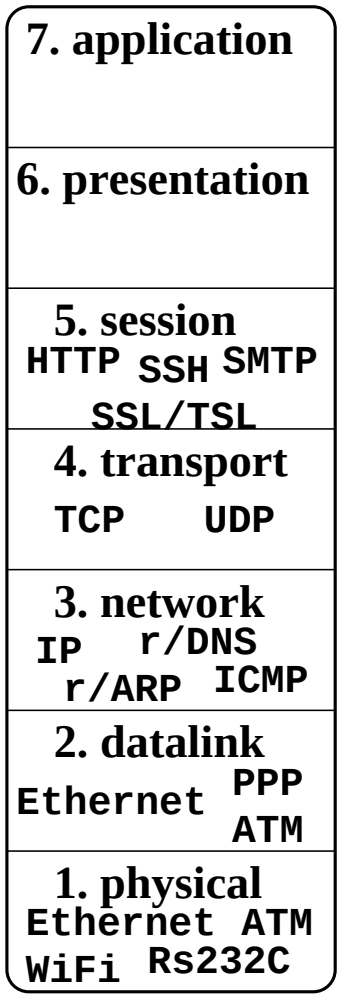
www.tcpipguide.com

Sesja TCP I



www.tcpipguide.com

Protokoły OSI/ISO I



Plan punktu: Konfigurowanie TCP/IP

Konfigurowanie TCP/IP

Obsługa protokołów sieciowych I

Linux obsługuje wiele protokołów sieciowych, np:

- Gigabit/Fast/Ethernet,
- ATM,
- TokenRing,
- FrameRelay,
- ISDN,
- FibreChannel.

Większość istniejących...

Obsługa kart sieciowych I

- Przy pomocy uniwersalnych sterowników Linux obsługuje wiele typów i modeli kart sieciowych.
- Większość sterowników konfiguruje automatycznie urządzenia sieciowe.
- Istnieją też możliwości ręcznego konfigurowania sterowników poprzez opcje modułów jądra systemu.

Konfigurownie interfejsów I

Konfigurowanie interfejsu sieciowego można podzielić na kilka etapów.

- określenie parametrów (IP, maska, itp.),
- konfigurownie przy pomocy **ifconfig**,
- testowanie przy pomocy **ping**,
- zapis konfiguracji w plikach dystrybucji systemu.

Parametry interfejsu I

Przed przystąpieniem do konfigurowania interfejsu trzeba ustalić jego podstawowe parametry. Są to:

- adres IP,
- maska sieci IP,
- broadcast IP,
- adres MAC, jeżli wykorzystuje się DHCP.

Program **ifconfig** I

ifconfig służy do konfigurowanie interfejsu sieciowego.

- Podstawowy sposób wywołania to:
  `ifconfig` *if* `address` *adres* `netmask` *maska*
- Włączanie, wyłącznie interfejsu:
  `ifconfig` *if* `up,down`
- Wyświetlanie informacji o interfejsie:
  `ifconfig` *if*

Testowanie **ping** I

ping jest uniwersalnym programem będącym częścią narzędzi dla sieci TCP/IP. Program:

- służy do testowania połączeń w sieci IP,
- wykorzystuje protokół ICMP (pakiety *echo replay/requst*),
- występuje w każdym systemie wykorzystującym protokół IP,
- bez konfiguracji routingu może być używany jedynie w tej samej podsieci IP.

Interfejs *loopback* I

Jest to wewnętrzny interfejs sieciowy:

- pełni rolę interfejsu zwrotnego,
- ma zawsze adres `127.0.0.1`,
- jest niezbędny do poprawnego działania aplikacji opartych o sieć TCP/IP.

Konfiguracja specyficzna dla dystrybucji I

Dystrybucje mogą dostarczać:

- programów typu **ifup**, **ifdown**,
- plików opisujących konfigurację interfejsów */etc/network*, */etc/sysconfig/network*,
- dodatkowych narzędzi konfiguracyjnych typu `ControlPanel`.

Konfigurowanie routingu I

Przed przystąpieniem do konfigurowania tablicy routingu należy określić podstawowe parametry sieci do której włącza się interfejsy. Są to:

- adresy interfejsów,
- adresy sieci IP,
- ewentualnie inne sieci do których określa się bezpośrednio dostęp,
- adres routera przekazującego pakiety poza sieć, *default route*,

Program **route** I

Podstawowym programem do konfigurowania routingu jest program **route**. Program służy do:

- zarządzania tablicą routingu,
- tablica określa sposób przekazywania pakietów przez stos TCP/IP,
- sposób użycia to:
  `route -v -n {add|del} {-host|-net}` *adres* `gw` *adres* `netmask` *maska interfejs*

Przykład route I


```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Iface
as1.cyf-kr.edu. *               255.255.255.255 UH    0      ppp0
192.168.1.0     *               255.255.255.0   U     0      eth0
default         as1.cyf-kr.edu. 0.0.0.0         UG    0      ppp0
```

Testowanie routingu **traceroute** I

Do testowania routingu można użyć programu **traceroute**.

- służy do testowania routingu w sieci IP,
- wykorzystuje pole TTL pakietów,
- występuje w każdym systemie wykorzystującym protokół IP,
- wyświetla kolejne routery na trasie pakietu.

Przykład traceroute I


```
$ traceroute ftp.icm.edu.pl
traceroute to sunsite.icm.edu.pl (193.219.28.2), 30 hops max, 40 byte pack
 1  ucirtr.agh.edu.pl (149.156.96.4)  0.938 ms  0.796 ms  0.784 ms
 2  manrtr.cyf-kr.edu.pl (149.156.6.217)  0.926 ms  0.726 ms  0.594 ms
 3  gwk.cyfronet.krakow.pl (195.150.0.247)  0.667 ms  0.633 ms  0.614 ms
 4  149.156.9.233 (149.156.9.233)  1.525 ms  1.400 ms  3.543 ms
 5  z-krakowa-oc3.lodz.pol34.pl (150.254.213.109)  6.265 ms  6.686 ms  6.1
 6  c7-icm-atm0-9.icm.edu.pl (193.219.28.249)  15.554 ms  15.192 ms  15.77
 7  c7-icm-atm2-0-1.icm.edu.pl (212.87.0.1)  15.520 ms  15.962 ms  21.597
 8  SunSITE.icm.edu.pl (193.219.28.2)  18.168 ms * *
```

Pakiet IProute I

Pakiet `IProute`:

- jest zaawansowanym oprogramowaniem mającym zastąpić standardowe narzędzia do konfigurowania interfejsów sieciowych i routingu w systemie Linux,
- wykorzystuje protokół IPv6 (standardowe narzędzia również),
- zapewnia *QoS*, *load balancing*, *traffic shaping*,
- współpracuje z jądrem w wersji od 2.2.

Konfigurowanie *resolvera* I

- *Resolver* jest biblioteką będącą częścią biblioteki systemowej. Zajmuje się odnajdywaniem nazw maszyn pracujących w sieci. Jego konfiguracja składa się z kilku plików.
- Nazwę maszyny w sieci IP ustawia się przy pomocy polecenia **hostname**.
- Nazwę najczęściej zapisuje się na stałę w pliku */etc/hostname* czytanym w trakcie startu systemu.

Plik *host.conf* I

Jest jednym z podstawowych elementów konfiguracji:

- zawiera część konfiguracji *resolvera*,
- opcje to:
  - `order {hosts,bind,nis}` – porządek odwoływania się do DNS i *hosts*,
  - `trim on/off` – usuwanie nazw domen
  - `multi on/off` – wielokrotne nazwy
  - `nospoof,spoofalert,reorder on/off`

Plik *resolv.conf* I

Zapisuje się w nim:

- adresy serwerów DNS: `nameserver addr`,
- nazwę lokalnej domeny: `domain`,
- sposób przeszukiwania domen: `search`.

Przykład resolv.conf I

```
domain uci.agh.edu.pl
nameserver  149.156.96.9
nameserver  149.156.96.12
search uci.agh.edu.pl agh.edu.pl
```

Plik *hosts* I

Plik */etc/hosts* zawiera informacje istotne dla funkcjonowania *resolvera*. Znajdują się w nim między innymi:

- adresy IP lokalnych interfejsów (w tym musi być *loopback*),
- nazwa FQDN maszyny,
- adresy sieci podłączonych do interfejsów lokalnych,
- adresy innych maszyn i sieci.

/etc/networks zawiera adresy znanych sieci IP.

Testowanie resolvera **host** I

Do testowania można użyć:

- **ping** wraz z nazwą symboliczną (domenową),
- narzędzi do pracy z DNS: **host**, **nslookup**, **dig**
- te programy pozwalają też na ogólne testowanie DNS.

Przykład host I


```
$ host ftp.icm.edu.pl
ftp.icm.edu.pl is a nickname for sunsite.icm.edu.pl
sunsite.icm.edu.pl has address 193.219.28.2
sunsite.icm.edu.pl mail is handled (pri=40) by gw.icm.edu.pl
sunsite.icm.edu.pl mail is handled (pri=10) by sunsite.icm.edu.pl
```

Pliki usług I

/etc/services

```
ftp     21/tcp
ssh     22/tcp
ssh     22/udp
smtp    25/tcp
www     80/tcp
```

/etc/protocols

```
ip      0   IP
icmp    1   CMP
igmp    2   IGMP
tcp     6   TCP
udp    17   UDP
```

Wybrane serwisy sieciowe I

- Telnet
- Ssh
- Nfs
- Samba
- Ftp
- Smtp
- Imap (POP)
- Http/s
- Xdm

Plan punktu: Demony sieciowe

Demony sieciowe

Koncepcja demona I

- komunikacja w sieci TCP odbywa się najczęściej w architekturze klient/serwer
- serwery sieciowe nasłuchują na określonych portach, związanych z usługami TCP
- klienty łączą się z serwerami (otwierają do tego lokalne porty)
- z logicznego punktu widzenia serwer realizuje dla klienta pewną *usługę*
- w unixie serwery sieciowe pracują jako demony, tzn. w trybie nieinteraktywnym, w tle

Uruchamianie usług I

Demony, w tym sieciowe najczęściej uruchamia sie tak:

```
# /etc/init.d/postfix
Usage: /etc/init.d/postfix {start|stop|restart|reload|force-reload}

# /etc/init.d/postfix restart
```

Najważniejsze demony sieciowe I

- inetd
- arpd
- routed
- httpd
- smtpd
- sshd
- dhcpd
- nfsd
- portmap
- bind
- ...

Demon InetD I

- uniwersalny demon TCP
- „Internet Super-Server”
- może nasłuchiwać na dowolnym podanym porcie TCP
- dla zdefiniowanych portów uruchamia dowolne usługi sieciowe
- obsługuje też niektóre trywialne protokoły sieciowe
- po nawiązaniu połączenia tworzy nową instancję, która je obsługuje

Konfigurowanie Inetd I

Cała konfiguracja jest w pliku */etc/inetd.conf*

```
# <servname> <socktype> <proto> <flags> <user> <server_path> <args>
#echo   stream tcp   nowait root    internal
#echo   dgram  udp   wait   root    internal
#time   stream tcp   nowait root    internal
#time   dgram  udp   wait   root    internal
ident   stream tcp   nowait identd /usr/sbin/ident2 ident2 -i -n
#smtp   stream tcp   nowait mail    /usr/sbin/exim exim -bs
tftp    dgram  udp   wait   nobody /usr/sbin/tcpd /usr/sbin/in.tftpd  /tftpb
```

Słabe punkty I

Ogólne problemy z konfigurowaniem i zabezpieczaniem usług sieciowych w Unixie:

- sam Inet ma ograniczenia, (wydajność)
- poza Inet, każda usługa może mieć własne pliki konfiguracyjne i własną specyfikę,
- istnieje wiele implementacji danego serwisu,
- różnice dla różnych wersji Unixa,
- nie wszystkie usługi mają wbudowane mechanizmy bezpieczeństwa,
- większość standardowych usług nie zachowuje poufności (brak szyfrowania).

Plan punktu: Monitorowanie TCP/IP

Monitorowanie TCP/IP

Monitorowanie ifconfig, Linux 2.6 I

```
eth0 Link encap:Ethernet  HWaddr 00:02:B3:EA:4F:38
     inet addr:149.156.xxx.xxx  Bcast:149.156.xxx.xxx  Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:50973150 errors:0 dropped:0 overruns:0 frame:0
     TX packets:33191012 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:3471235083 (3.2 GiB)  TX bytes:3166500094 (2.9 GiB)
     Base address:0x7440 Memory:fc320000-fc340000

lo   Link encap:Local Loopback
     inet addr:127.0.0.1  Mask:255.0.0.0
     UP LOOPBACK RUNNING  MTU:16436  Metric:1
     RX packets:980611 errors:0 dropped:0 overruns:0 frame:0
     TX packets:980611 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:848365118 (809.0 MiB)  TX bytes:848365118 (809.0 MiB)
```

Monitorowanie ifconfig, FreeBSD I


```
xl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     options=9<RXCSUM,VLAN_MTU>
     inet6 fe80::204:75ff:fec0:5037%xl0 prefixlen 64 scopeid 0x1
     inet 149.156.xxx.xxx netmask 0xffffff00 broadcast 149.156.xxx.xxx
     ether 00:04:75:c0:50:37
     media: Ethernet autoselect (100baseTX <full-duplex>)
     status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
     inet 127.0.0.1 netmask 0xff000000
     inet6 ::1 prefixlen 128
     inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
```

netstat (Linux) I

- a all
- l listening
- A inet, unix, ipx, ax25, netrom, ddp
- p pid
- n numeric
- s statistics
- r route
- i iface

netstat - przykłady I


```
# netstat -apnv
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address     State    PID/Program name
tcp    0   0 0.0.0.0:928  0.0.0.0:*   LISTEN   21103/amd
tcp    0   0 0.0.0.0:2049 0.0.0.0:*   LISTEN   -
tcp    0   0 0.0.0.0:642  0.0.0.0:*   LISTEN   21394/rpc.rquotad
tcp    0   0 0.0.0.0:836  0.0.0.0:*   LISTEN   2350/rpc.statd
tcp    0   0 0.0.0.0:676  0.0.0.0:*   LISTEN   3133/rpc.mountd
tcp    0   0 0.0.0.0:389  0.0.0.0:*   LISTEN   16479/slapd
tcp    0   0 0.0.0.0:44650 0.0.0.0:*   LISTEN   8452/xdm
tcp    0   0 0.0.0.0:139  0.0.0.0:*   LISTEN   17627/smbd
tcp    0   0 0.0.0.0:42411 0.0.0.0:*   LISTEN   -
tcp    0   0 0.0.0.0:111  0.0.0.0:*   LISTEN   12957/portmap
tcp    0   0 0.0.0.0:80  0.0.0.0:*   LISTEN   11980/apache
tcp    0   0 0.0.0.0:113  0.0.0.0:*   LISTEN   31371/inetd
tcp    0   0 0.0.0.0:22  0.0.0.0:*   LISTEN   1860/sshd
tcp    0   0 0.0.0.0:5432 0.0.0.0:*   LISTEN   803/postmaster
tcp    0   0 0.0.0.0:25  0.0.0.0:*   LISTEN   15604/master
tcp    0   0 127.0.0.1:6010     0.0.0.0:*   LISTEN   817/0
tcp    0   0 127.0.0.1:6011     0.0.0.0:*   LISTEN   14355/1
tcp    0   0 0.0.0.0:443  0.0.0.0:*   LISTEN   11980/apache
tcp    0   0 0.0.0.0:445  0.0.0.0:*   LISTEN   17627/smbd
tcp    0   0 0.0.0.0:3551 0.0.0.0:*   LISTEN   29085/apcupsd
tcp    0   0 127.0.0.1:389 127.0.0.1:42498     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:34511     127.0.0.1:389 ESTABLISHED5687/pickup
tcp    0   0 127.0.0.1:33305     127.0.0.1:389 ESTABLISHED28520/sshd: gjn [pr
tcp    0   0 127.0.0.1:35834     127.0.0.1:389 ESTABLISHED27775/apache
tcp    0   0 127.0.0.1:389 127.0.0.1:49175     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:389 127.0.0.1:33305     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:389 127.0.0.1:55066     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:39849     127.0.0.1:389 ESTABLISHED12517/snort
tcp    0   0 127.0.0.1:42498     127.0.0.1:389 ESTABLISHED5445/apache
tcp    0   0 127.0.0.1:42734     127.0.0.1:389 ESTABLISHED14355/1
tcp    0   0 127.0.0.1:45559     127.0.0.1:389 ESTABLISHED7892/qmgr
tcp    0   0 127.0.0.1:50289     127.0.0.1:389 ESTABLISHED23592/apache
tcp    0   0 127.0.0.1:49175     127.0.0.1:389 ESTABLISHED10418/sshd: gjn [pr
```

netstat - przykłady II

```
tcp    0   0 127.0.0.1:52625     127.0.0.1:389 ESTABLISHED817/0
tcp    0   0 127.0.0.1:53124     127.0.0.1:389 ESTABLISHED29601/apache
tcp    0   0 127.0.0.1:389 127.0.0.1:55377     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:55066     127.0.0.1:389 ESTABLISHED17747/apache
tcp    0   0 127.0.0.1:55377     127.0.0.1:389 ESTABLISHED11570/nscd
tcp    0   0 127.0.0.1:60093     127.0.0.1:389 TIME_WAIT -
tcp    0   0 127.0.0.1:389 127.0.0.1:50289     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:389 127.0.0.1:53124     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:389 127.0.0.1:52625     ESTABLISHED16479/slapd
tcp    0   0 127.0.0.1:389 127.0.0.1:39849     ESTABLISHED16479/slapd
tcp    0   32 149.156.199.59:22    149.156.96.9:47748   ESTABLISHED10418/sshd: gjn [pr
tcp    0   0 149.156.199.59:22    149.156.96.9:47856   ESTABLISHED28520/sshd: gjn [pr
tcp    0   0 149.156.199.59:56719  149.156.127.1:1998   TIME_WAIT -
tcp    0   0 149.156.199.59:58231  149.156.127.1:1998   TIME_WAIT -
tcp    0   0 149.156.199.59:36135  149.156.127.1:1998   TIME_WAIT -
tcp    0   0 149.156.199.59:42447  149.156.127.1:1998   TIME_WAIT -
tcp    0   0 149.156.199.59:47763  149.156.127.1:1998   TIME_WAIT -
tcp    0   0 149.156.199.59:389   149.156.127.1:58491   ESTABLISHED16479/slapd
udp    0   0 0.0.0.0:32768 0.0.0.0:*   -
udp    0   0 0.0.0.0:2049 0.0.0.0:*   -
udp    0   0 127.0.0.1:32770     127.0.0.1:32770     ESTABLISHED803/postmaster
udp    0   0 149.156.199.59:137   0.0.0.0:*   9431/nmbd
udp    0   0 0.0.0.0:137  0.0.0.0:*   9431/nmbd
udp    0   0 149.156.199.59:138   0.0.0.0:*   9431/nmbd
udp    0   0 0.0.0.0:138  0.0.0.0:*   9431/nmbd
udp    0   0 0.0.0.0:798  0.0.0.0:*   -
udp    0   0 0.0.0.0:799  0.0.0.0:*   -
udp    0   0 0.0.0.0:800  0.0.0.0:*   -
udp    0   0 0.0.0.0:929  0.0.0.0:*   21103/amd
udp    0   0 0.0.0.0:673  0.0.0.0:*   3133/rpc.mountd
udp    0   0 0.0.0.0:673  0.0.0.0:*   3133/rpc.mountd
udp    0   0 0.0.0.0:177  0.0.0.0:*   8452/xdm
udp    0   0 0.0.0.0:830  0.0.0.0:*   2350/rpc.statd
udp    0   0 0.0.0.0:833  0.0.0.0:*   2350/rpc.statd
udp    0   0 0.0.0.0:67  0.0.0.0:*   15507/dhcpd3
udp    0   0 0.0.0.0:69  0.0.0.0:*   31371/inetd
```

netstat - przykłady III

```
udp    0   0 0.0.0.0:10080 0.0.0.0:*   31371/inetd
udp    0   0 0.0.0.0:111   0.0.0.0:*   12957/portmap
udp    0   0 149.156.199.59:123   0.0.0.0:*   30210/ntpd
udp    0   0 127.0.0.1:123 0.0.0.0:*   30210/ntpd
udp    0   0 0.0.0.0:123   0.0.0.0:*   30210/ntpd
udp    0   0 0.0.0.0:1022 0.0.0.0:*   21103/amd
udp    0   0 0.0.0.0:1023 0.0.0.0:*   21103/amd
udp    0   0 0.0.0.0:639   0.0.0.0:*   21394/rpc.rquotad
raw    0   0 0.0.0.0:1   0.0.0.0:*   7     15507/dhcpd3
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags Type State  I-Node PID/Program name  Path
unix  17  [ ]   DGRAM     1122  4287/syslogd  /dev/log
unix  2   [ ACC ]  STREAM  LISTENING  2999  15604/master  public/cleanup
unix  2   [ ACC ]  STREAM  LISTENING  3025  15604/master  public/flush
unix  2   [ ACC ]  STREAM  LISTENING  3041  15604/master  public/showq
unix  2   [ ACC ]  STREAM  LISTENING  3006  15604/master  private/rewrite
unix  2   [ ACC ]  STREAM  LISTENING  3017  15604/master  private/bounce
unix  2   [ ACC ]  STREAM  LISTENING  3029  15604/master  private/proxymap
unix  2   [ ACC ]  STREAM  LISTENING  3033  15604/master  private/smtp
unix  2   [ ACC ]  STREAM  LISTENING  2644761  14355/1    /tmp/ssh-fpcmS14355/ag
ent.14355
unix  2   [ ACC ]  STREAM  LISTENING  3037  15604/master  private/relay
unix  2   [ ACC ]  STREAM  LISTENING  3045  15604/master  private/error
unix  2   [ ACC ]  STREAM  LISTENING  3049  15604/master  private/local
unix  2   [ ACC ]  STREAM  LISTENING  3053  15604/master  private/virtual
unix  2   [ ACC ]  STREAM  LISTENING  3057  15604/master  private/lmtp
unix  2   [ ACC ]  STREAM  LISTENING  3061  15604/master  private/maildrop
unix  2   [ ACC ]  STREAM  LISTENING  3065  15604/master  private/cyrus
unix  2   [ ACC ]  STREAM  LISTENING  3069  15604/master  private/uucp
unix  2   [ ACC ]  STREAM  LISTENING  3073  15604/master  private/ifmail
unix  2   [ ACC ]  STREAM  LISTENING  3077  15604/master  private/bsmtp
unix  2   [ ACC ]  STREAM  LISTENING  3081  15604/master  private/scalemail-back
end
unix  2   [ ACC ]  STREAM  LISTENING  3085  15604/master  private/trace
unix  2   [ ACC ]  STREAM  LISTENING  3089  15604/master  private/verify
unix  2   [ ACC ]  STREAM  LISTENING  3184  803/postmaster   /var/run/postgresql/.s
```

netstat - przykłady IV

```
.PGSQL.5432
unix  2    [ ]   DGRAM     2644694  14355/1
unix  3    [ ]   STREAM  CONNECTED  2644691  28520/sshd: gjn [pr
unix  3    [ ]   STREAM  CONNECTED  2644690  14355/1
unix  2    [ ]   DGRAM     2644284  817/0
unix  3    [ ]   STREAM  CONNECTED  2644281  10418/sshd: gjn [pr
unix  3    [ ]   STREAM  CONNECTED  2644280  817/0
unix  2    [ ]   DGRAM     2643456  5687/pickup
unix  2    [ ]   DGRAM     2549466  12517/snort
unix  2    [ ]   DGRAM     781309   15229/uptimed
unix  2    [ ]   DGRAM     117294   3133/rpc.mountd
unix  2    [ ]   DGRAM     4221  29085/apcupsd
unix  2    [ ]   DGRAM     3898  30210/ntpd
unix  2    [ ]   DGRAM     3447  2350/rpc.statd
unix  2    [ ]   DGRAM     3181  803/postmaster
unix  2    [ ]   DGRAM     3126  7892/qmgr
unix  3    [ ]   STREAM  CONNECTED  3092  15604/master
unix  3    [ ]   STREAM  CONNECTED  3091  15604/master
unix  2    [ ]   DGRAM     1317  15507/dhcpd3
unix  2    [ ]   DGRAM     1303  16479/slapd
unix  2    [ ]   DGRAM     1134  14543/klogd
```

arp I

```
# arp -vn
Address HWtype  HWaddress     Flags Mask   Iface
149.156.yyy.xxx    ether   08:00:20:82:B9:DE   C eth0
149.156.yyy.xxx    ether   00:0D:BD:F1:BE:00   C eth0
Entries: 2   Skipped: 0   Found: 2
```

lsof I


```
COMMAND  PID  USER   FD   TYPE  DEVICE   SIZE    NODE NAME
rpc.statd  2350  root  cwd DIR  8,7   4096  107299 /var/lib/nfs
rpc.statd  2350  root  rtd DIR  8,1   1024    2 /
rpc.statd  2350  root  txt REG  8,1  36664  108601 /sbin/rpc.statd
rpc.statd  2350  root  mem REG  8,1  90088  100371 /lib/ld-2.3.2.so
rpc.statd  2350  root  mem REG  8,1  28688   37118 /lib/libwrap.so.0.7.6
rpc.statd  2350  root  mem REG  8,1  73304  106566 /lib/tls/libnsl-2.3.2.so
rpc.statd  2350  root  mem REG  8,1   1253924  106518 /lib/tls/libc-2.3.2.so
rpc.statd  2350  root  mem REG  8,1  15316   37119 /lib/libnss_db-2.2.so
rpc.statd  2350  root  mem REG  8,1  34748  106627 /lib/tls/libnss_files-2.3.2.so
rpc.statd  2350  root  mem REG  8,5 692456  303010 /usr/lib/libdb3.so.3.0.2
rpc.statd  2350  root 0u   CHR  1,3    19853 /dev/null
rpc.statd  2350  root 1u   CHR  1,3    19853 /dev/null
rpc.statd  2350  root 2u   CHR  1,3    19853 /dev/null
rpc.statd  2350  root 3u  unix 0xf6e83900     3447 socket
rpc.statd  2350  root 4u  IPv4    3457 UDP *:833
rpc.statd  2350  root 5u  IPv4    3448 UDP *:830
rpc.statd  2350  root 6u  IPv4    3460 TCP *:836 (LISTEN)
rpc.statd  2350  root 7w   REG  8,7   5   46336 /var/run/rpc.statd.pid
```

fuser I

```
# fuser -auv /var/lib/nfs

      USER PID ACCESS COMMAND
/var/lib/nfs   root   2350 ..c..  rpc.statd
      root   3133 ..c..  rpc.mountd
```

Monitorowanie innych usług I

- każda usługa może mieć własne dodatkowe narzędzia, np.:
- NFS: nfsstat
- Samba: smbstatus
- Apache: apachectl
- etc.

Wykład: Praktyczne narzędzia kryptograficzne w systemach Unix/GNU/Linux

Plan wykładu I

Motto I

Trust No One!
– The X-Files

Tematyka wykładu I

- rozumienie terminu „bezpieczeństwo”
- rola technologii kryptograficznych
- podstawowe pojęcia
- praktyczne zastosowania
- szyfrowanie plików
- szyfrowanie sesji
- szyfrowanie poczty
- tunelowanie transmisji
- podpis elektroniczny

Plan punktu: Wstęp

Wstęp

Różne rozumienia bezpieczeństwa I

Z bezpieczeństwem systemu komputerowego wiąże się:

- działanie systemu zgodne z oczekiwaniami, specyfikacją
- niezawodność i dostępność systemu,
- działanie w sposób bezbłędny i przewidywalny (zaufanie),
- kontrola dostępu, (w tym ograniczenie, uniemożliwienie),
- monitorowanie pracy,
- ochrona przechowywanych i przetwarzanych danych,
- utrzymywanie pełnej kontroli nad systemem.

Ochrona danych I

- udostępnianie danych (np. nieprzerwane, szybkie),
- utrzymywanie spójności i nienaruszalności danych,
- zapewnianie poufności,
- udostępnianie historii zmian, poprzednich wersji,
- przechowywnie kopii zapasowych,
- podtrzymywanie udostępniania w wypadku awarii.

Sytuacja bezpieczeństwa I

1. system komputerowy
   - 1.1 moduły przetwarzające dane
   - 1.2 moduły przechowujące dane
   - 1.3 połączenia, np. sieć komputerowa
     - sprzęt
     - oprogramowanie: systemowe + aplikacje
2. użytkownicy systemu
3. administratorzy systemu

Obserwacje I

Bezpieczeństwo komputerowe jest zbiorem technicznych rozwiązań nietechnicznych problemów...

za: Garfinkel, Spafford

security is a process, not a product...

za: ?

Punkt widzenia użytkownika I

Każdy użytkownik może i powinien dbać o bezpieczeństwo *swoich* danych, poprzez:

- ochronę dostępu do konta: hasła, bezpieczna autoryzacja (ssh, rhosts)
- ochronę dostępu do plików: prawa dostępu
- zachowywnie poufności danych: uwierzytelnianie, szyfrowanie

Rola narzędzi kryptograficznych I

Przy utrzymywaniu bezpieczeństwa istotną rolę odgrywają narzędzia wykorzystujące technologie kryptograficzne. Używa się ich przy:

- uwierzytelnianiu użytkowników i systemów,
- szyfrowaniu danych,
- kontrolowaniu spójności danych.

Plan punktu: Wprowadzenie do Kryptografii

Wprowadzenie do Kryptografii

Podstawowe pojęcia I

- kryptografia
- krypoanaliza
- kryptologia
- szyfr (ang. *cipher*)
- tekst jawny,
- kryptogram,
- klucz (ang. *key*)
- hash, skrót (ang. *hash*, *digest*)

Podstawowe pojęcia I

- szyfrowanie bez klucza: ROT13 (ROT3 - Szyfr C.I.Cezara)
- szyfrowanie z kluczem symetrycznym, np.: Crypt, DES, 3DES, Idea, Blowfish, Twofish, AES
- szyfrowanie z kluczem asymetrycznym (publicznym), np: RSA, DSA, ElGamal
- algorytmy generowania skrótu, np: MD5, SHA
- podpis elektroniczny

Szyfrowanie z kluczem symetrycznym I

Nadawca i odbiorca mają ten sam klucz.
Nadawca:

- przygotowuje tekst wiadomości,
- szyfruje całość wiadomości kluczem,
- wysyła szyfrogram do odbiorcy.

Odbiorca:

- rozszyfrowuje szyfrogram przy pom. klucza
- otrzymuje tekst wiadomości

Oprócz szyfrogramu musi być przekazany klucz!

Szyfrowanie z kluczem asymetrycznym I

N. i O. mają pary własnych kluczy (Pub/Prv). Wymieniają się *publicznymi*.
Nadawca:

- przygotowuje tekst wiadomości,
- szyfruje całość wiadomości kluczem *publicznym* odbiorcy,
- wysyła szyfrogram do odbiorcy.

Odbiorca:

- rozszyfrowuje szyfrogram przy pomocy swojego klucza *prywatnego*,
- otrzymuje tekst wiadomości.

Podpis elektroniczny: Nadawca I

Nadawca i odbiorca mają pary kluczy (Pub/Prv), Wymieniają się *publicznymi*.
Nadawca:

- przygotowuje tekst wiadomości,
- wylicza skrót wiadomości (np. MD5),
- szyfruje skrót przy pomocy swojego klucza *prywatnego*,
- wysyła szyfrogram do odbiorcy.

Podpis elektroniczny: Odbiorca I

- wylicza skrót wiadomości,
- rozszyfrowuje skrót przy pomocy klucza *publicznego* nadawcy,
- porównuje wyliczony skrót z rozszyfrowanym,
- jeżeli są zgodne, potwierdzone zostają tożsamość nadawcy i spójność przesłanych informacji.

Obie części klucza są komplementarne.
W praktyce do podpisywania stosuje się specjalne algorytmy.

Podpis i szyfrowanie I

Przebiega analogicznie do podpisywania, poza tym treść wiadomości szyfrowana jest kluczem publicznym odbiorcy, który rozszyfrowuje ją swoim prywatnym.

Plan punktu: Przegląd narzędzi

Przegląd narzędzi

Zastosowania narzędzi kryptograficznych I

Najważniejsze współczesne zastosowania to:

- szyfrowanie plików
- szyfrowanie poczty (specjalny przypadek powyższego)
- szyfrowanie transmisji
- a także podpis elektroniczny – potwierdzenie autentyczności
- zapewnianie spójności danych
- autoryzacja użytkowników i systemów

Szyfrowanie plików i transmisji I

- może być realizowane przez programy w przestrzeni użytkownika (np. *GnuPG*),
- może być realizowane na poziomie protokołu sieciowego:
  - *SSH* (warstwa 5 OSI/ISO),
  - *SSL* (warstwa 4 OSI/ISO),
  - *IPSec* (warstwa 3 OSI/ISO)),
- na poziomie lokalnego systemu plików, np. *CFS*
- w tym sieciowego systemu plików, np. *SFS*

Potrzeba szyfrowania I

- Należy pamiętać, iż standardowe protokoły używane w TCP/IP *NIE* szyfrują danych.
- Dotyczy to protokołów dostępu np. Telnet, Rsh/cmd czy pocztowych Pop/Imap.
- Oznacza to w praktyce możliwość podsłuchania (ang. *sniffing*) transmisji.
- Szyfrowanie zapobiega nie tyle podsłuchiwaniu, ile możliwości odczytania podsłuchanych danych.
- Protokoły takie jak SSH wzmacniają również mechanizmy uwierzytelniania użytkownika.

Szyfrowanie sesji I

- umożliwia utajnienie sesji w tym przebiegu uwierzytelniania i transferu danych,
- zwiększa bezpieczeństwo pracy i zmniejsza prawdopodobieństwo włamania przy pomocy złamania hasła lub przejęcia sesji,
- podstawowa technologia to SSH,
- *Secure SHell* – protokół bezpiecznego dostępu do sesji,
- SSH to protokół i oprogramowanie
- wykorzystuje różne techniki szyfrowania i uwierzytelniania.

Plan punktu: SSH

SSH
- Nawiązywanie połączenia
- Uwierzytelnianie pbk
- Open SSH

SSH I

- *Secure SHell* – protokół bezpiecznego dostępu do sesji,
- SSH to protokół i oprogramowanie
- wykorzystuje różne techniki szyfrowania i uwierzytelniania.
- jest to oprogramowanie zastępujące usługę telnet i usługi typu R...,
- może zastąpić FTP (SFTP),
- pozwala na tunelowanie portów TCP,
- obecne „oficjalne” wersje są komercyjne
- najważniejsza implementacja to *OpenSSH* (`http://www.openssh.org`).

Nawiązywanie połączenia I

- serwer SSH ma parę kluczy (publiczny i prywatny),
- klient rozpoczyna transmisję szyfrowaną przy pomocy klucza publicznego serwera (algorytm asymetryczny, np. DSA/RSA), (metoda hybrydowa, *challenge/response*)
- podejmowana jest próba uwierzytelnienia użytkownika, (jedną z wybranych metod: klucz, hasło, *OTP*, itp.), najczęściej:
- jeżeli to możliwe, przeprowadzane jest uwierzytelnienie przy pomocy klucza publicznego użytkownika,
- w przypadku pomyślnego uwierzytelnienia reszta sesji jest szyfrowana przy pomocy algorytmu symetrycznego (3DES, Blowfish) z kluczem losowanym na nowo dla każdej sesji,
- transmisja może być opcjonalnie kompresowana,
- zamiast uruchomienia powłoki interaktywnej można uruchomić dowolny program.

Uwierzytelnianie pbk I

- Wymaga wygenerowania pary kluczy na maszynie *z której* się logujemy.
- Przeniesienia kluczy publicznych na maszyny *na które* się logujemy.
- Klucze są zabezpieczane *passfrazami* (szyfrowane).
- Zawierają również informacje o koncie na którym zostały stworzone.
- Pozwala to na uwierzytelnienie: użytkownika i maszyny (miejsca) z którego się loguje.
- Uwierzytelnienie jest dwustopniowe: hasło (passfraza) i token (klucz).
- Jest to wspomagane przez **ssh-agent** który może przechowywać w obrębie sesji passfrazy i przekazywać tę informację dalej, do zdalnych maszyn.

Przykład logowania SSH I

```
OpenSSH_3.8.1p1 Debian-8.sarge.4, OpenSSL 0.9.7e 25 Oct 2004
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Connecting to ---- [W.X.Y.Z] port 22.
debug1: Connection established.
debug1: identity file /home/gjn/.ssh/identity type -1
debug1: identity file /home/gjn/.ssh/id_rsa type -1
debug1: identity file /home/gjn/.ssh/id_dsa type 2
debug1: Remote protocol version 2.0, remote software version OpenSSH_3.8.1p1 FreeBS
debug1: match: OpenSSH_3.8.1p1 FreeBSD-20040419 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_3.8.1p1 Debian-8.sarge.4
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-cbc hmac-md5 none
debug1: kex: client->server aes128-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '----' is known and matches the DSA host key.
debug1: Found key in /home/gjn/.ssh/known_hosts:12
debug1: ssh_dss_verify: signature correct
```

Przykład logowania SSH II

```
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,keyboard-interactive
debug1: Next authentication method: publickey
debug1: Offering public key: /home/gjn/.ssh/id_dsa
debug1: Server accepts key: pkalg ssh-dss blen 433
debug1: Authentication succeeded (publickey).
debug1: channel 0: new [client-session]
debug1: Entering interactive session.
```

Open SSH I

- w związku z komercjalizacją SSH pojawiła się potrzeba opracowania swobodnie dostępnej wersji
- *OpenSSH* powstało w ramach projektu *OpenBSD*, bezpiecznego systemu operacyjnego
- OpenSSH jest rozwijane poza USA, przez programistów z klika krajów,
- rozpowszechniane na licencji BSD
- standardowo dostępne w każdej wersji systemu BSD, GNU/Linux
- dostępne wersje na inne platformy, np. *putty* dla Windows

Plan punktu: PGP

PGP
- Historia
- Wersje
- Możliwości
- GNUPG
- S/MIME

PGP I

- spostrzeżenie: bezpieczeństwo e-maila jest w praktyce niższe niż kartki pocztowej (nieokreślona droga przekazywania, brak utrudnień w odczytaniu)
- twórca: Philip Zimmermann, USA
- *Pretty Good Privacy*, „niezłe szyfrowanie dla każdego”
- uniezależenienie się od monopolu państwa, korporacji

Historia I

- pierwsza wersja na początku lat 90., swobodnie dostępny kod w systemach BBS, Usenecie, Internecie
- niemożliwość uzyskania darmowego patentu RSA (brak możliwości zarabiania)
- 1991, w USA bardzo silna inwigilacja producentów technologi kryptograficznych (pełen wgląd rządu w transmisję)
- rezygnacja z DESa na rzecz IDEI, wykorzystanie RSA na zasadach niekomercyjnych
- PGP przez Internet dociera do „wrogów USA", dochodzenie przeciw P.Z.
- *RSA Security* twierdzi, że P.Z. naruszył licencję i ograniczenia eksportowe USA, proces
- pozwy, śledztwa zostały pozytywnie oddalone, zamknięte, na korzyść P.Z dopiero w 1996 roku
- zaczynają być rozwijane oficjalne wersje komercyjne pakietu

Wersje I

- PGP 2.6.3 i wersja „i”, 1. standard PGP, ograniczenia w zarządzaniu kluczami
- w.w. wersja wyjechała z USA w postaci wydruku w książce, wyjątkowo łatwego do skanowania :)
- komercyjne PGP 5.0, tylko na rynek amerykański
- powstaje standard OpenPGP (RFC2440)
- skomplikowane „losy” PGP, różne firmy i korporacje, z/bez P.Z.
- obecnie *PGP Corporation*, jednym z doradców technicznych jest P.Z.
- niezależnie od 1999 rozwijany jest zgodny z OpenPGP *GNUPG*
- PGP to jeden z najważniejszych *przełomów* w kryptografii...

Możliwości I

- generowanie kluczy RSA, IDEA
- dystrybucja i podstawowe zarządzanie kluczami (*Web of Trust*, omawiane przy GNUPG)
- szyfrowanie i deszyfrowanie
- podpisy elektroniczne
- wersje z linii poleceń i zintegrowane z GUI

GNUPG I

- *GNU Privacy Guard*
- dostępny na licencji GNU GPL
- pracuje na wszystkich platformach (Unix, Linux, Windows, Mac, itd.)
- nie używa żadnych opatentowanych algorytmów: ElGamal, CAST5, Triple DES, AES, Blowfish
- zaawansowane możliwości zarządzania kluczami
- zgodny z OpenPGP i większością wersji PGP
- zaawansowane narzędzie obsługiwane z linii poleceń (opcjonalne interfejsy graficzne)

Web of trust I

- mechanizm ustalania autentyczności i zaufania do kluczy
- zdecentralizowana sieć osób znających innych bez/pośrednio i darzących się zaufaniem
- użytkownicy wzajemnie podpisują elektronicznie swoje klucze publiczne z określonym poziomem zaufania
- mechanizm elastyczny i niezależny od firm komercyjnych, czy centrów autoryzacji, alternatywa dla PKI

Narzędzia dla GNUPG I

GPA	standardowe narzędzie graficzne dla GPG
Enigmail	wtyczka dla Mozilli
WinPT	integracja z Win i MS Outlook

Wiele, wiele innych, w tym gotowe zintegrowane MUA: Evolution, Mutt, KMail, Syplheed.

S/MIME

- nowsza wersja PEM, *Privacy Enhanced Email*
- alternatywa dla PGP
- standard rozwinięty przez konsorcjum amerykańskich korporacji prywatnych, (np. RSA, MS)
- używa PKI w przeciwieństwie do *web of trust*
- najczęściej wykorzystuje wybrane algorytmy amerykańskie
- S/MIME miewa problemy z poufnością (dołącza do przesyłki dodatkowe informacje, np. o nadawcy) i integracją z innymi technologiami (np. systemy antywirusowe)

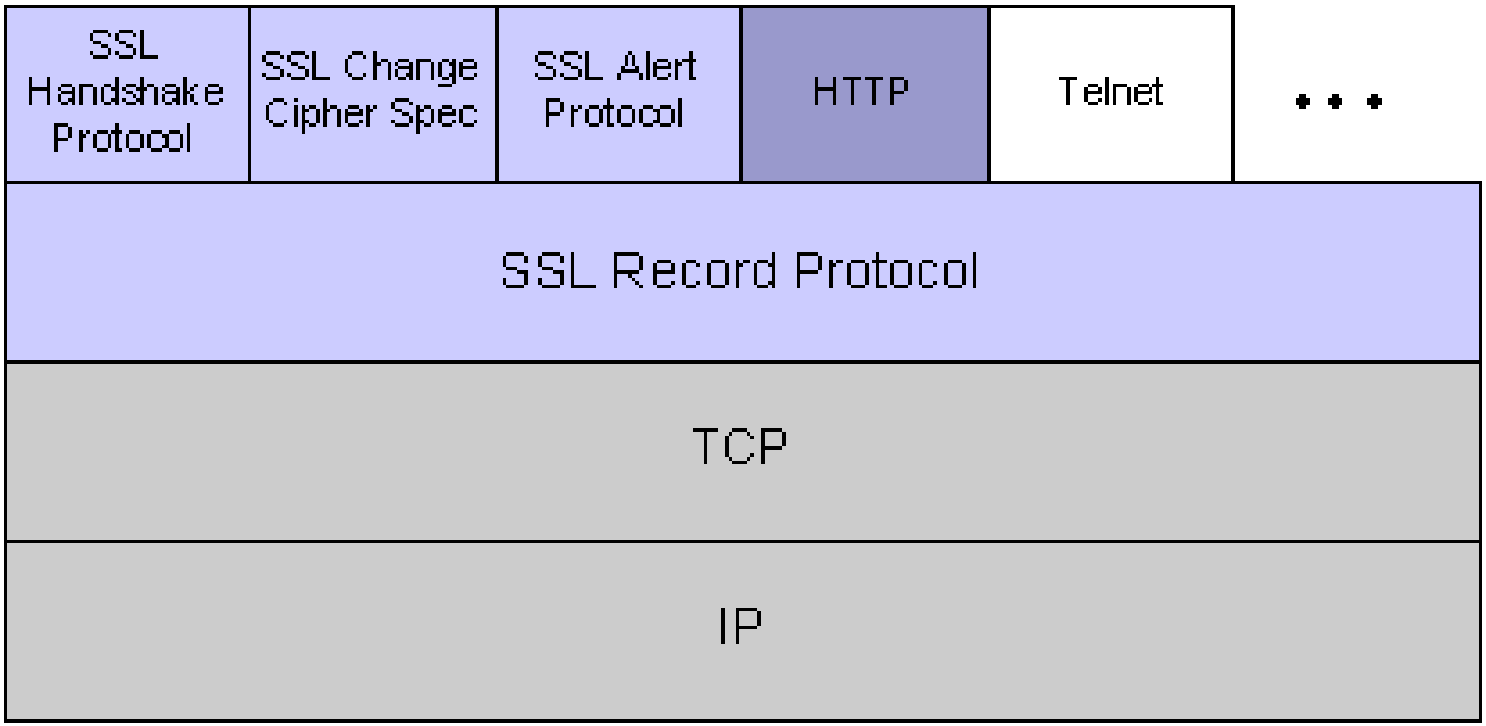
Plan punktu: SSL

SSL
- Techniki kryptograficzne
- Certyfikaty
- Sesja SSL
- Zastosowania

SSL I

- *Secure Sockets Layer*
- protokół szyfrujący transmisję sieci,
- stanowi uzupełnienie TCP (warstwa „4.5”), który nie zapewnia szyfrowania
- dostarcza metod uwierzytelniania klienta i serwera, szyfrowania i cyfrowego podpisywania transmisji
- bazuje na zróżnicowanych metodach kryptograficznych (a/symetrycznych, skrótach)
- jest otwartym standardem, pierwotnie stworzonym przez Netscape Corp., SSL v2.0
- obecna wersja to TLS v1.0, *Transport Layer Security*

SSL a OSI/ISO I



(za modssl.org)

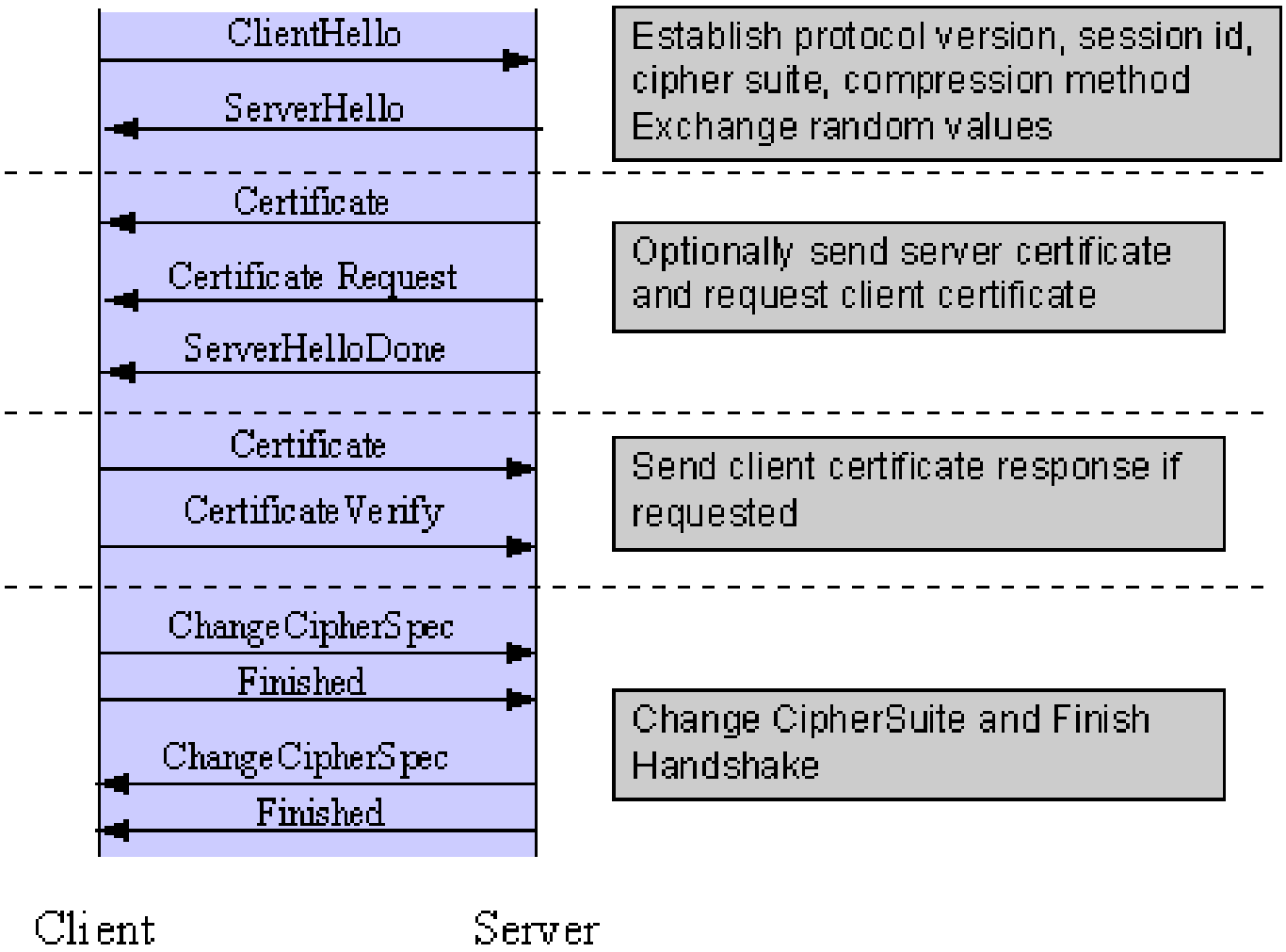
Techniki kryptograficzne I

- algorytmy symetryczne: RC2, RC4, IDEA, DES, Triple DES, AES
- algorytmy asymetryczne: RSA, Diffie-Hellman, DSA, Fortezza
- funkcje skrótu: MD5, SHA-1
- podpisy cyfrowe
- certyfikaty elektroniczne, PKI

Certyfikaty I

- certyfikat jest dokumentem elektronicznym potwierdzającym tożsamość strony biorącej udział w komunikacji
- certyfikat jest wystawiany przez 3. stronę, *Certificate Authority*, CA
- CA wystawia certyfikat dla strony, użytkownika
- certyfikat zawiera między innymi dane o podmiocie, jego klucz publiczny, podpisane przez CA
- dla potwierdzenia ważności certyfikatu inni użytkownicy muszą mieć klucze publiczne CA (np. w przeglądarce WWW)

Otwieranie sesji I



(za modssl.org)

Zastosowania

- uniwersalne szyfrowanie dla TCP
- szereg implementacji, w tym otwarta *OpenSSL*
- podstawowy mechanizm zabezpieczania transmisji HTTP, (HTTPS)
- używany w połączeniu z innymi protokołami, np.: SMTP, IMAP
- jedna z podstaw handlu elektronicznego

Plan punktu: Zagrożnia w sieci

Zagrożnia w sieci

Identyfikacja zagrożeń lokalnych I

- nieuprawniony dostęp do kont użytkowników,
- nieuprawniony dostęp do konta root,
- wykorzystanie luk w dostępie do systemu plików,
- naruszenie bezpieczeństwa przez programy SUID,
- wyczerpanie zasobów systemu,
- naruszenie spójności systemu plików.

Przeciwdziałanie zagrożeniom lokalnym I

Przeciwdziałać powyższym zagrożeniom można:

- chroniąc i ograniczając dostęp do kont,
- stosując specjalne procedury dostępu do konta root,
- kontrolując prawa dostępu,
- ograniczając działanie programów SUID,
- zabezpieczając przydzielanie zasobów użytkownikom,
- monitorując spójność systemu plików,
- uaktualniając dystrybucję.

Identyfikacja zagrożeń sieciowych I

- *skaning* – działanie mające na celu identyfikację komputerów w sieci oraz określenie typu i trybu pracy usług sieciowych,
- *denial of service (DOS)* – ataki mające na celu zakłócenie pracy usług sieciowych komputerów,
- *distributed denial of service (DDOS)* – podobne do powyższych, lecz wykorzystujące całe sieci komputerowe.
- *spamming* – atak polegający na wysyłaniu dużej ilości bezwartościowych wiadomości e-mail na konta użytkowników.
- *spoofing* – ataki polegające na podszywaniu się napastnika pod inne komputery w sieci i przejmowanie danych, które są do nich adresowane.
- *sniffing* – podsłuchiwanie transmisji w sieci.
- *session hijacking* – atak wykorzystujący między innymi dwa powyższe, pozwalający na przechwytywanie całej sesji pracy z programem w sieci,
- *host infection* – szeroko rozumiane włamanie do systemu operacyjnego.

Przeciwdziałanie zagrożeniom sieciowym I

- *skaning* – systemy firewall,
- *denial of service (DOS)* – konfiguracja usług, ogranicznie dostępu, systemy firewall,
- *spamming* – filtrowanie poczty (np. RBL),
- *spoofing* – zaawansowane metody autoryzacji, weryfikowanie transmisji, szyfrowanie,
- *sniffing* – szyfrowanie, fragmentacja sieci,
- *session hijacking* – zaawansowana autentykacja, weryfikowanie transmisji, szyfrowanie,
- *infekcja hosta* – różne metody, w tym kontrola dostępu i odpowiednia konfiguracja.

Źródła... I

- Simon Garfinkel, Gene Spafford, *Practical Unix and Internet Security*, O'Reilly and Associates, 3rd Ed.
- polskie wydanie: *Bezpieczeństwo w Unixie i Internecie*, wyd. RM.
- R. Wobst, *Kryptografia. Budowa i łamanie zabezpieczeń*, Wydawnictwo RM, Warszawa, 2002.
- Neal Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, 1994.
- polskie wydanie: *Wykład z teorii liczb i kryptografii*, WNT, 1995.
- Kevin Fenzi, Dave Wreski, *Linux Security HOWTO*, LDP, 2002.

Koniec

Czy są jakieś pytania?