Sieci komputerowe

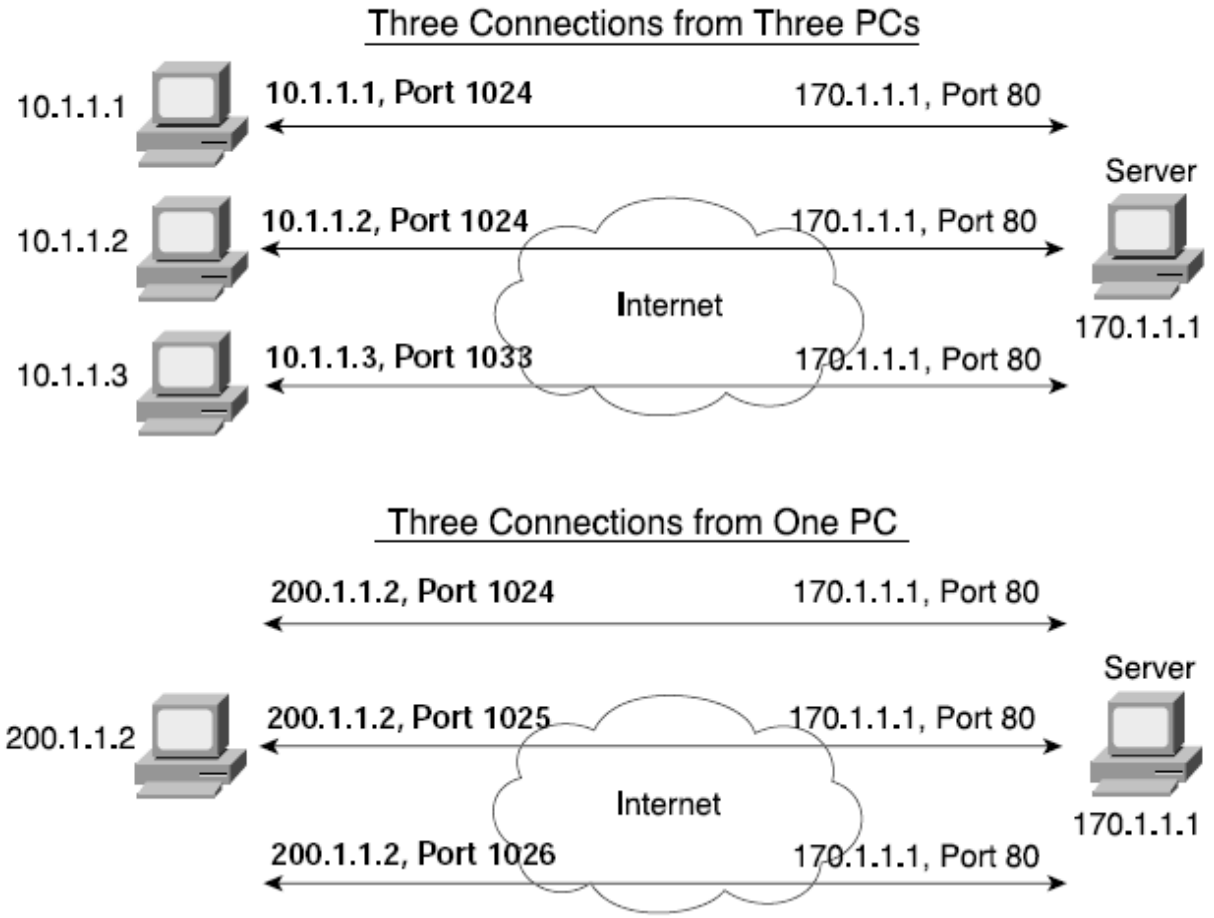
Wykład 10
Bezpieczeństwo w sieciach komputerowych

Translacja adresów (NAT)

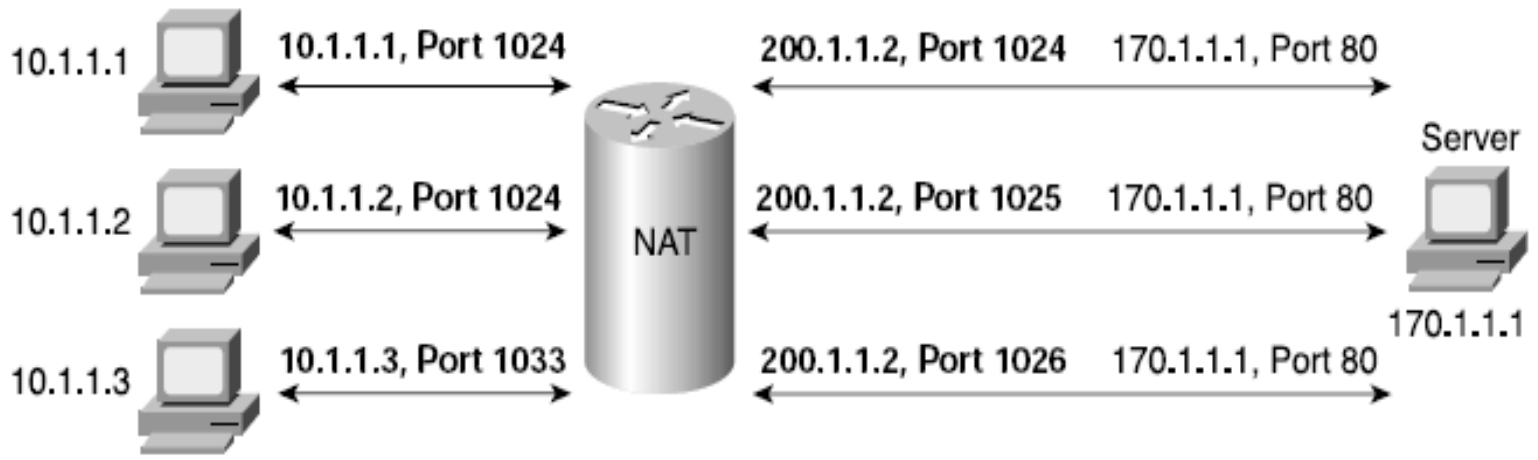
- NAT (ang. Network Address Translation) umożliwia używanie adresów nierutowalnych (niepublicznych)
- Polega na „maskowaniu” połączeń z wielu adresów nierutowalnych jednym publicznym adresem IP
- Nie można wykonywać połączeń bezpośrednio do adresów niepublicznych, wpływa to na zwiększenie bezpieczeństwa
- NAT został wprowadzony ze względu na brak adresów IP

Połączenia bez użycia NAT

Three TCP Connections: From Three Different Hosts, and from One Host



Połączenia z użyciem NAT



Dynamic NAT Table, With Overloading

Inside Local	Inside Global
10.1.1.1:1024	200.1.1.2:1024
10.1.1.2:1024	200.1.1.2:1025
10.1.1.3:1033	200.1.1.2:1026

- Zmieniany jest adres IP i/lub port
- Powyższy rysunek ilustruje technikę SNAT (zmiana adresu źródłowego)
- Stosuje się także DNAT (zmiana adresu docelowego)

Serwery proxy

- Serwer proxy (pośredniczący) wykonuje połączenia w imieniu użytkownika
- Użytkownik „zleca” wykonanie połączenia za pomocą oprogramowania klienckiego – zwykle przeglądarki internetowej
- Użytkownik nie musi posiadać publicznego adresu IP
- Proxy wykona za niego połączenia do sieci zewnętrznych i przekaże odpowiedź
- Proxy działa w warstwie aplikacji – wadą jego jest obsługa niewielu protokołów (głównie FTP i HTTP)

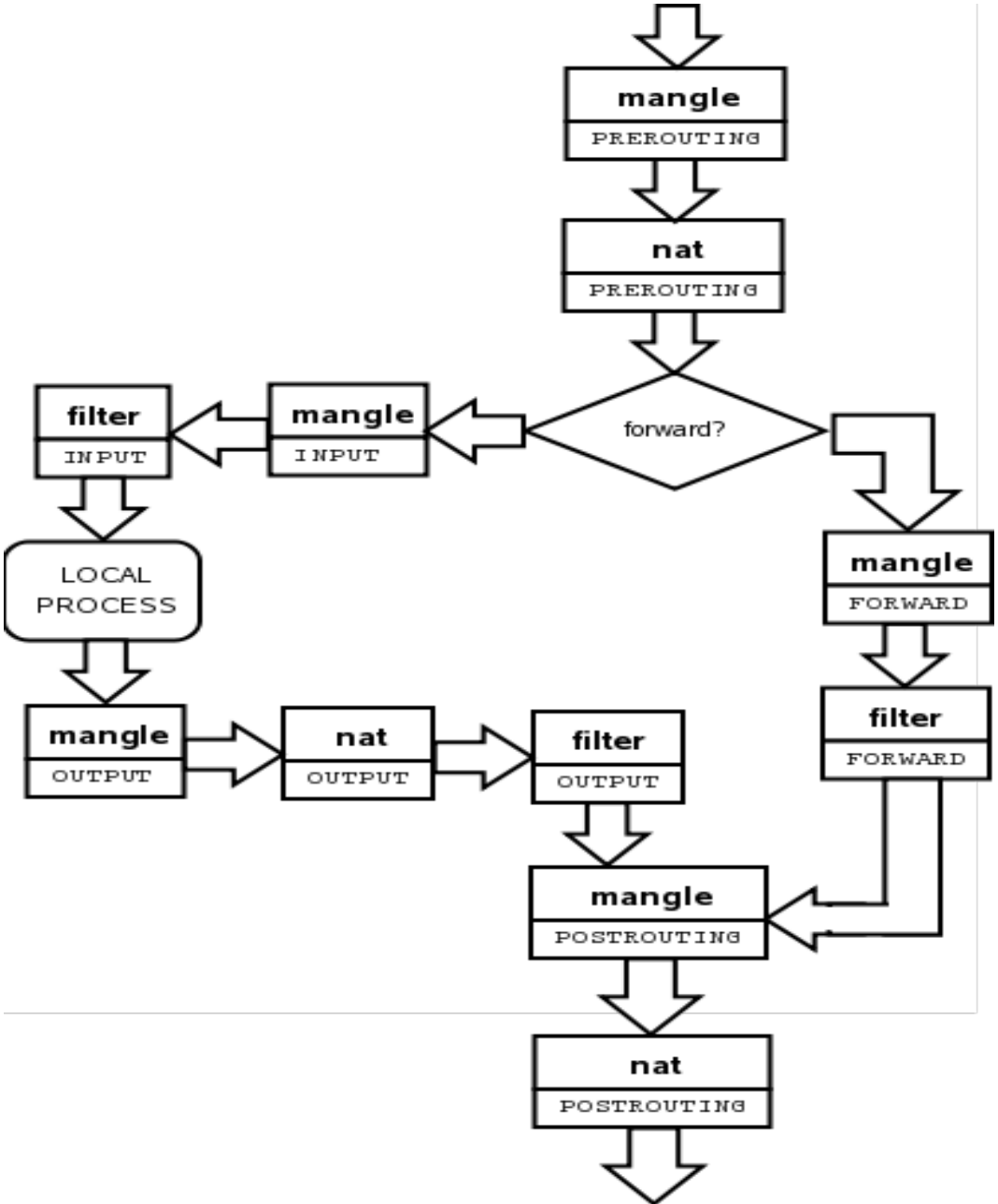
Ściany ogniowe

- Ściany ogniowe (ang. firewall) filtrują adresy
- Mogą też obserwować stany połączeń
  - firewall stanowy
  - firewall bezstanowy (nie śledzi stanu połączeń)
- Należy utworzyć reguły

Iptables

- Iptables to program do obsługi filtru (firewalla) wbudowanego w jądro systemu operacyjnego Linux
- http://www.netfilter.org

Iptables – diagram przepływu



Iptables

- Dodawanie reguły
  - iptables [-t tablica] komenda [wzorzec] [akcja]

Iptables - przykład

```
# Adres IP hosta
ip=”10.1.1.134”

#policy na DROP
iptables -P INPUT DROP
iptables -F INPUT
iptables -P OUTPUT DROP
iptables -F OUTPUT
iptables -P FORWARD DROP
iptables -F FORWARD
iptables -F -t nat

iptables -X
iptables -Z

# Wpuszczamy jedynie nawiązane połączenia
iptables -A INPUT -i eth0 -s 0.0.0.0/0 -d $ip -m state --state \
ESTABLISHED,RELATED -j ACCEPT

# Wypuszczamy wszystko z naszego hosta
iptables -A OUTPUT -o eth0 -s $ip -d 0.0.0.0/0 -j ACCEPT
```

Iptables – przykład c.d.

```
iptables -A INPUT -i eth0 -s 0.0.0.0/0 -d 0.0.0.0/0 -j DROP
iptables -A OUTPUT -o eth0 -s 0.0.0.0/0  -d 0.0.0.0/0 -j DROP

iptables -A INPUT -i eth0 -s 0.0.0.0/0 -d 0.0.0.0/0 -j REJECT
iptables -A OUTPUT -o eth0 -s 0.0.0.0/0  -d 0.0.0.0/0 -j REJECT\
--reject-with icmp-host-unreachable
```

Iptables – przykłady c.d.

```
#limit liczby połączeń:
iptables -A INPUT -m limit --limit 3/min -j ACCEPT

#limit wielkości pakietu:
iptables -A INPUT -m length --length 1000: -j DROP

#sprytny limit
iptables -A INPUT -p tcp --dport 22 -m hashlimit --hashlimit
1/min --hashlimit-mode srcip --hashlimit-name ssh -m state \
--state NEW -j ACCEPT

# zezwól na 2 połączenia na klienta
iptables  -A INPUT -p tcp --syn --dport 23 -m connlimit \
--connlimit-above 2 -j REJECT

# j.w.
iptables -A INPUT -p tcp --syn --dport 23 -m connlimit \
! --connlimit-above 2 -j ACCEPT

#lista adresów IP w /proc/net/ipt_recent/www
iptables -A INPUT -p tcp –dport 80 -m recent –name www \
--set -j ACCEPT
```

Iptables, NAT

```
iptables -t nat -A PREROUTING -s 10.0.0.0/16 -j SNAT \
--to 193.0.96.15
iptables -t nat -A POSTROUTING -d 193.0.96.200 -j DNAT \
--to 10.1.1.20
iptables -t nat -A POSTROUTING -p tcp -d 193.0.96.201 \
--dport 222 -j DNAT -to 10.1.1.212:22
```

Komendy iptables (podsumowanie)

* -P łańcuch polityka - ustawienie domyślnej polityki dla łańcucha. Domyślna polityka stosowana jest dopiero wówczas, gdy pakiet nie pasuje do żadnej reguły łańcucha,
* -A łańcuch - dodanie reguły do określonego łańcucha,
* -I łańcuch [nr reguły] - wstawienie reguły do określonego łańcucha, jeśli zostanie podany nr reguły wtedy reguła zostanie wpisane w to miejsce zmieniając kolejność pozostałych,
* -L [łańcuch] - wyświetlenie wszystkich reguł łańcucha (lub wszystkich łańcuchów w danej tablicy); często używane z opcjami -n i -v,
* -F [łańcuch] - usunięcie wszystkich reguł łańcucha (lub ze wszystkich łańcuchów danej tablicy),
* -D łańcuch [nr reguły] - usunięcie konkretnej reguły w określonym łańcuchu, jeśli zostanie podany nr reguły wtedy zostanie usunięta reguła o tym numerze,
* -R łańcuch nr_reguły - zastąpienie reguły nume

Opcje wzorca

**
* -s [!] ip[/netmask] - adres IP źródłowy (może być uogólniony do adresu sieci),
* -d [!] ip[/netmask] - adres IP docelowy (może być uogólniony do adresu sieci),
* -p [!] protokół - wybór protokołu: tcp, udp, icmp lub all (wszystkie protokoły stosu TCP/IP),
* -i [!] interfejs wejściowy
* -o [!] interfejs wyjściowy
* --sport [!] port:[port] - port źródłowy,
* --dport [!] port:[port] - port docelowy,
* -m moduł - załadowanie modułu rozszerzającego, dzięki temu można wykorzystać kolejne opcje danego modułu.**

Akcje

* -j ACCEPT - przepuszczenie dopasowanych pakietów,
* -j DROP - usunięcie dopasowanych pakietów,
* -j REJECT - odrzucenie dopasowanych pakietów,
* -j LOG - logowanie dopasowanych pakietów
* -j SNAT - translacja adresów źródłowych,
* -j DNAT - translacja adresów docelowych,
* -j MASQUERADE - translacja adresów źródłowych na adres dynamicznie przyznawany na interfejsie.

Kryptografia w sieciach komputerowych

- Stosuje się następujące algorytmy:
  - symetryczny
  - niesymetryczny
- Należy pamiętać, że jeśli nie zastosowano szyfrowania, hasła są przesyłane przez sieć otwartym tekstem
- Dlatego, jeśli korzystamy z autoryzacji, należy używać szyfrowanej wersji protokołów:
  - HTTPs
  - POP3s
  - SMTPs

Szyfrowanie symetryczne

- Polega na użyciu tylko jednego klucza
- Ten sam klucz musi być w posiadaniu nadawcy i odbiorcy, gdyż nim się szyfruje i deszyfruje wiadomości
- Istnieje poważny problem z przekazywaniem klucza

Szyfrowanie symetryczne

- Polega na użyciu pary kluczy - publicznego i prywatnego
- Należy chronić jedynie klucz prywatny
- Można upublicznić klucz publiczny
- Nie posiada wady algorytmu symetrycznego

Wiadomość zaszyfrowana kluczem prywatnym może być odszyfrowana jedynie odpowiadającym mu kluczem publicznym. I analogicznie, wiadomość zaszyfrowana kluczem publicznym, może być odszyfrowana tylko odpowiadającym mu kluczem prywatnym. Jeśli chcemy zaszyfrować wiadomość do naszego korespondenta, musi on przysłać nam swój klucz publiczny. Klucz nie musi jednak być przysłany w bezpieczny sposób. Jeśli ktoś przechwyci klucz publiczny nie będzie mógł odszyfrować wiadomości nim zaszyfrowanych. Para kluczy publiczny i prywatny może też być używana do podpisywania dokumentów i weryfikacji podpisów.

Przebieg procesu szyfrowania

- Korespondenci wymieniają się kluczami publicznymi
- Nadawca szyfruje wiadomość dla odbiorcy kluczem publicznym odbiorcy
- Tylko odbiorca może ją odczytać, gdyż tylko on ma swój klucz prywatny

Przebieg procesu podpisywania cyfrowego

- Korespondenci wymieniają się kluczami publicznymi
- Nadawca podpisuje wiadomość dla odbiorcy swoim kluczem prywatnym
- Odbiorca weryfikuje podpis za pomocą klucza publicznego nadawcy

Mechanizm podpisywania elektronicznego

- Tworzony jest tzw. skrót wiadomości, która ma być podpisana. Należy wyjaśnić, iż skrót jest utworzony w taki sposób, aby był unikalny dla danej wiadomości. Dba o to oprogramowanie szyfrujące
- Skrót jest następnie szyfrowany kluczem prywatnym twórcy wiadomości i załączany do oryginalnej wiadomości
- Wiadomość oraz jej zaszyfrowany skrót (który jest właśnie podpisem elektronicznym) są przesyłane do odbiorcy

Mechanizm weryfikowania podpisu

- Tworzony jest skrót otrzymanej wiadomości
- Skrót otrzymany od nadawcy jest odszyfrowywany kluczem publicznym nadawcy (odbiorca musi ten klucz posiadać)
- Jeśli dało się odszyfrować skrót przysłany przez nadawcę jego kluczem publicznym, to oznacza, że nadawca zaszyfrował skrót swoim kluczem prywatnym, więc można mieć pewność, że to nadawca jest faktycznym autorem wiadomości. Tylko nadawca ma dostęp do swojego klucza prywatnego, czyli to nadawca złożył podpis.
- Jeśli odszyfrowany skrót, jest taki sam jak utworzony skrót, to dodatkowo istnieje pewność, że wiadomość nie została po drodze do odbiorcy zmodyfikowana. Pamiętamy przecież, że skrót jest unikalny dla danej wiadomości.

Szyfrowanie wiadomości w komunikacji klient-serwer np. https

- Serwer wysyła klientowi tzw. certyfikat, który zawiera klucz publiczny serwera
- Klient generuje klucz sesji i szyfruje go kluczem publicznym serwera, następnie przesyła do serwera
- Dalszy ciąg komunikacji odbywa się przez kanał szyfrowany kluczem sesji (ale już z użyciem algorytmu symetrycznego – dla uzyskania większej wydajności)

Ustalanie tożsamości serwera

- Serwer A przesyła klientowi (przeglądarce) certyfikat zawierający klucz publiczny
- Skąd wiemy, że to jest certyfikat prawdziwego serwera?
- Przeglądarka potrafi sprawdzić autentyczność certyfikatu serwera
- Używa do tego mechanizmu podpisu elektronicznego

Public Key Infrastructure (PKI)

- PKI to powszechny kryptosystem
- Certyfikaty serwerów są podpisywane kluczami prywatnymi organizacji CA (ang. Certification Authority) – CA to tzw. zaufana strona trzecia
- CA zanim podpisze cokolwiek swoim kluczem prywatnym, sprawdza tożsamość osoby lub instytucji
- Przeglądarki internetowe posiadają listę certyfikatów znanych CA, mogą więc sprawdzać podpisy na certyfikatach serwerów z którymi się łączą

PKI

- PKI podlega standaryzacji
- Wystawiane certyfikaty muszą być w standardzie X.509

Informacje zawarte w certyfikacie X.509

- Wersja standardu X.509
- Numer seryjny certyfikatu
- Nazwa organizacji do której należy certyfikat (O)
- Nazwa jednostki organizacyjnej i nazwa pospolita (OU, CN)
- Okres ważności certyfikatu
- Dane o wystawcy certyfikatu (czyli o CA, które złożyło podpis na certyfikacie)
- Klucz publiczny
- Podpis wszystkich ww. informacji dokonany kluczem prywatnym CA

Oprogramowanie

- GnuPG
  - Implementacja standardu OpenPGP
- OpenSSL
  - Implementacja SSL v2, v3 (Netscape) i TLS v1 (IETF)
  - /usr/lib/ssl/misc/CA.sh

Własne CA

- CA.sh --newca
- CA.sh --newreq
- CA.sh --sign

SSH (ang. Secure Shell)

- Umożliwia połączenia interaktywne
- Można też zestawić tunel:
  - ssh -L 8081:absurd.mimuw.edu.pl:80 login@students.mimuw.edu.pl

IDS

- IDS (Intrusion Detection System) to oprogramowanie mające za zadanie automatyczne wykrywanie włamań
- Przykładem dobrego oprogramowania IDS Open Source jest *snort*