Bezpieczeństwo sieci

Sieci komputerowe

Wykład 12

Marcin Bieńkowski

Założenia

Atakujący kontroluje pewną część sieci:

- komputery,
- routery / przełączniki,
- nośniki (kable, fale radiowe).

Co można zepsuć?

- **Poufność**
  - Atakujący czyta nasze dane.
- **Integralność**
  - Atakujący podszywa się pod nas.
  - Atakujący modyfikuje nasze wiadomości.
- **Dostępność**
  - Atakujący uniemożliwia nam komunikację.

Podsłuchiwanie i podszywanie się

Podsłuchiwanie we współdzielonym kanale

- **Nieprzełączany Ethernet (z koncentratorami).**
  - Wszyscy słyszą wszystkie ramki → tryb nasłuchu (*promiscous mode*).

- **Sieci WLAN ze współdzielonym kluczem (WPA Personal)**
  - Klucze sesji przesyłane przy wiązaniu klienta z punktem dostępowym (AP).
  - Wystarczy je podsłuchać i możemy deszyfrować całą komunikację między AP i klientem.

- **Przełączany Ethernet**
  - Przełącznik ma sprzętową tablicę haszującą (CAM = *content addressable memory*) z wpisami „adres MAC → port“.
  - Zmieniając często adres MAC można zalać CAM nowymi wpisami → przełącznik przejdzie w tryb uczenia się.

Ataki „man-in-the-middle”: sieci lokalne

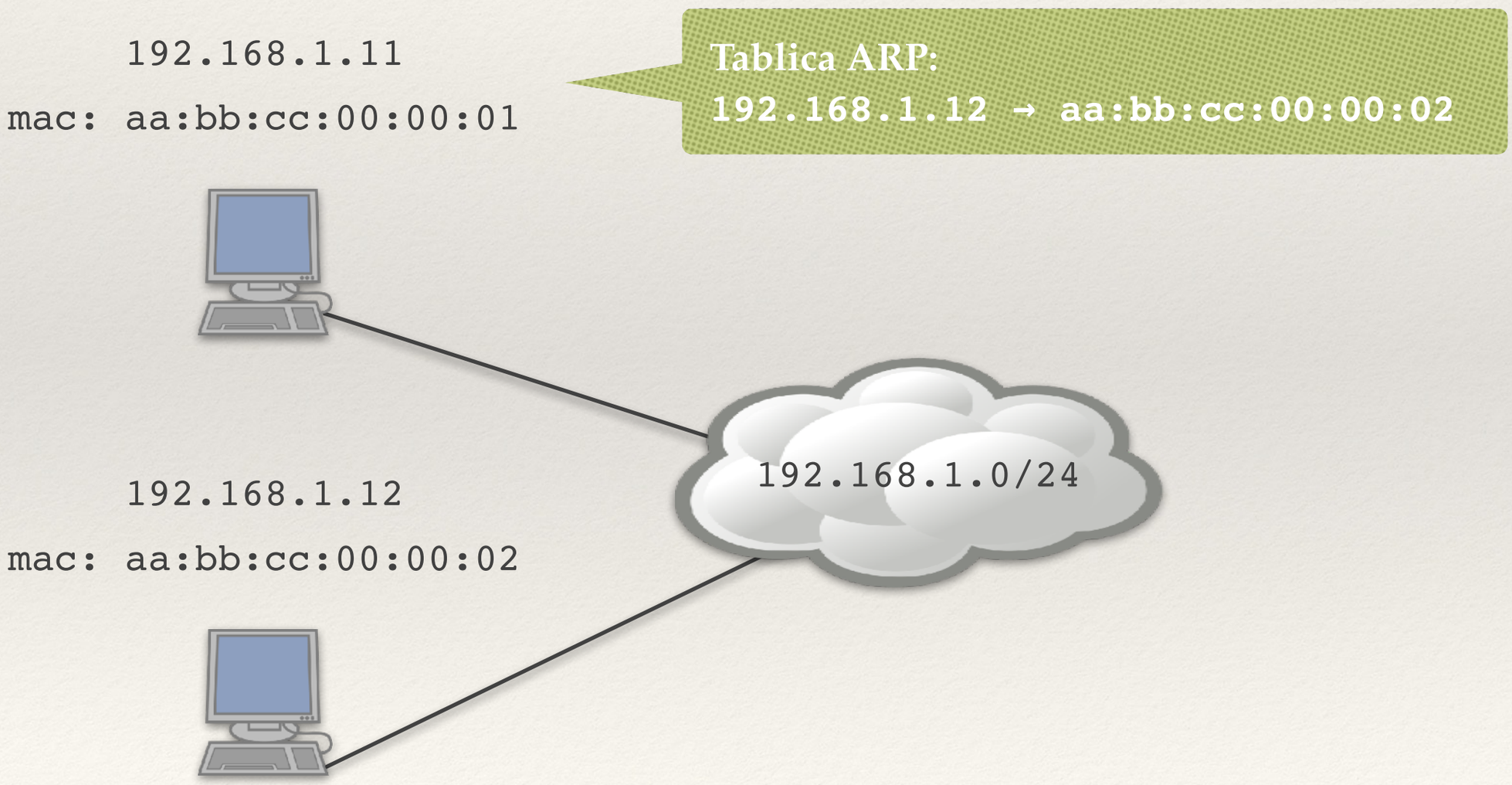
Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Zatruwanie pamięci podręcznej ARP.**

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

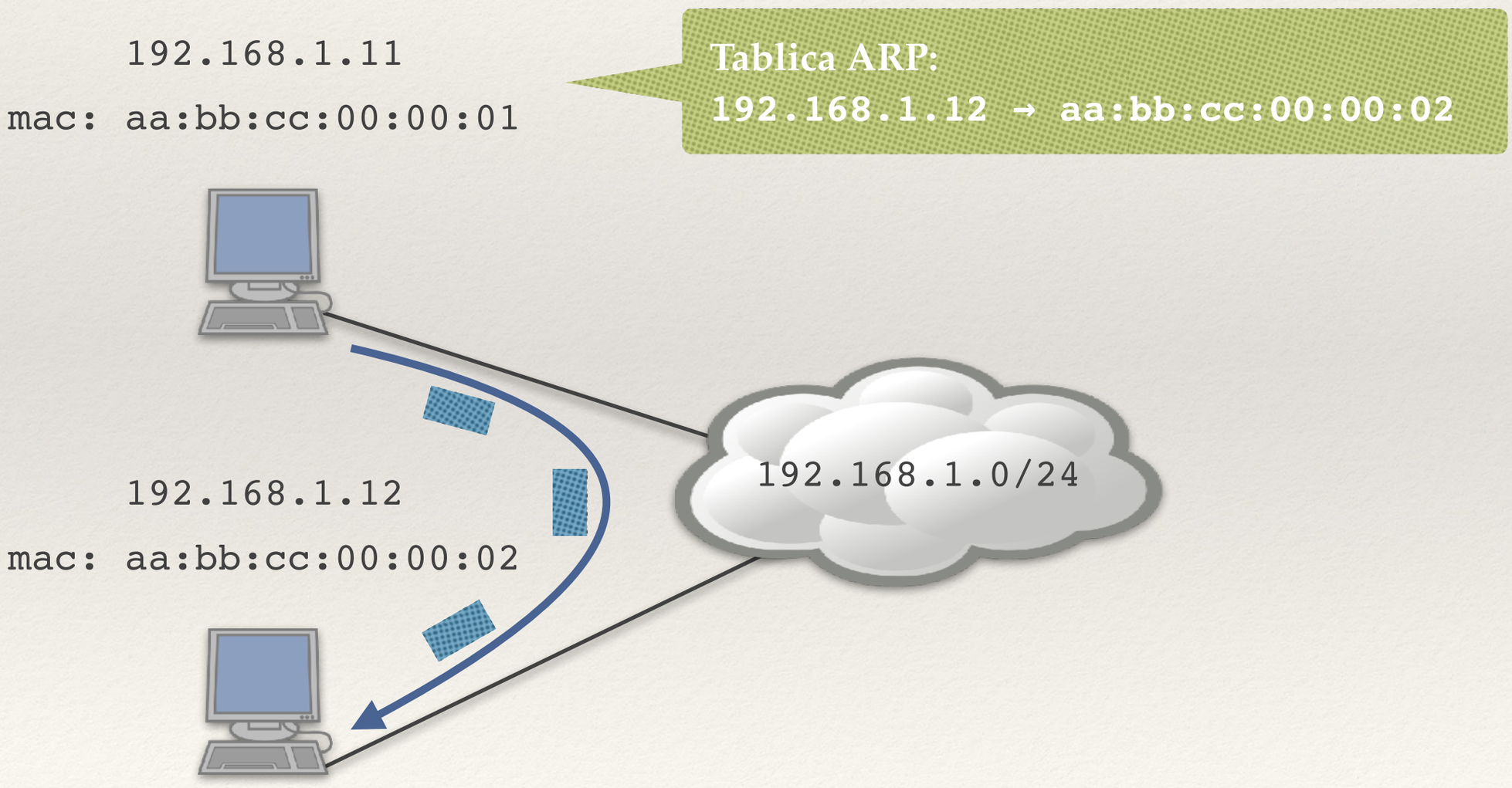
- **Zatruwanie pamięci podręcznej ARP.**



Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

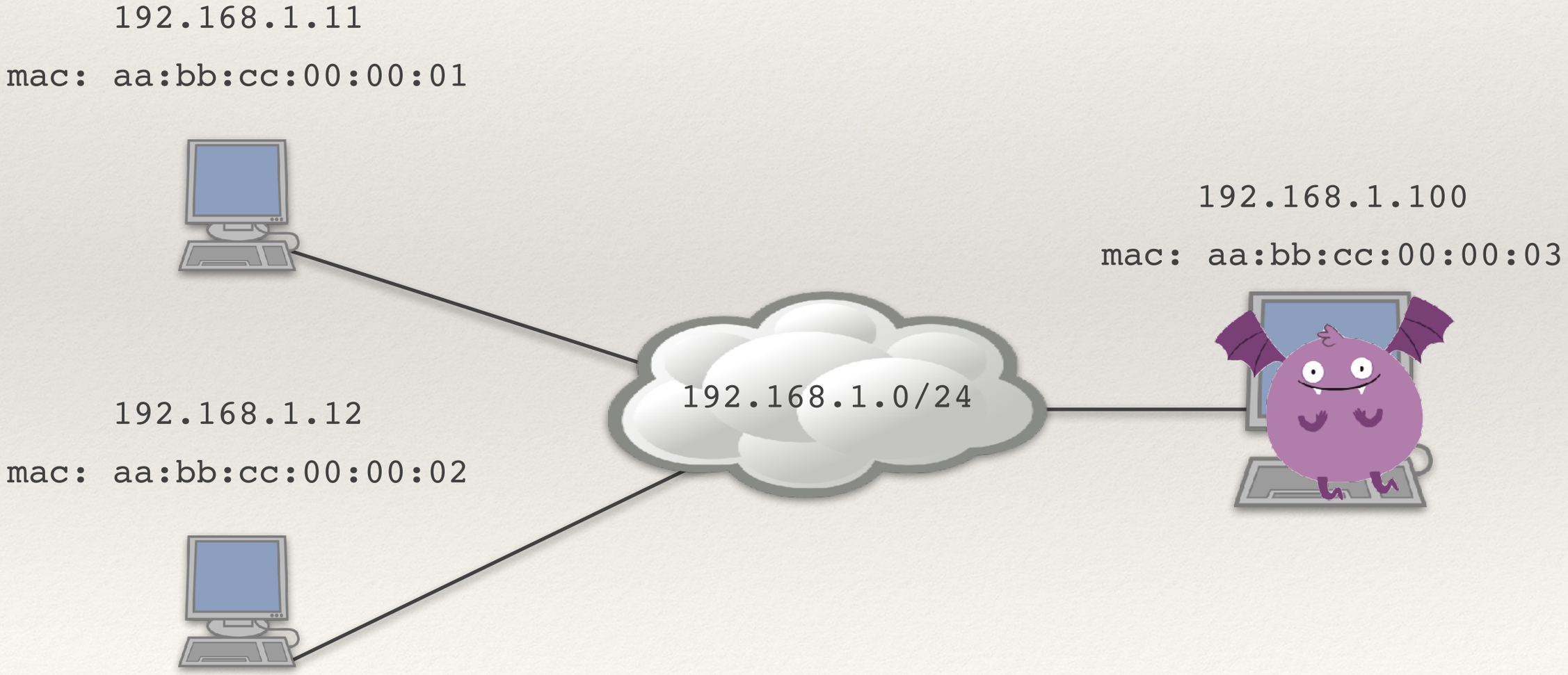
- **Zatruwanie pamięci podręcznej ARP.**



Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Zatruwanie pamięci podręcznej ARP.**

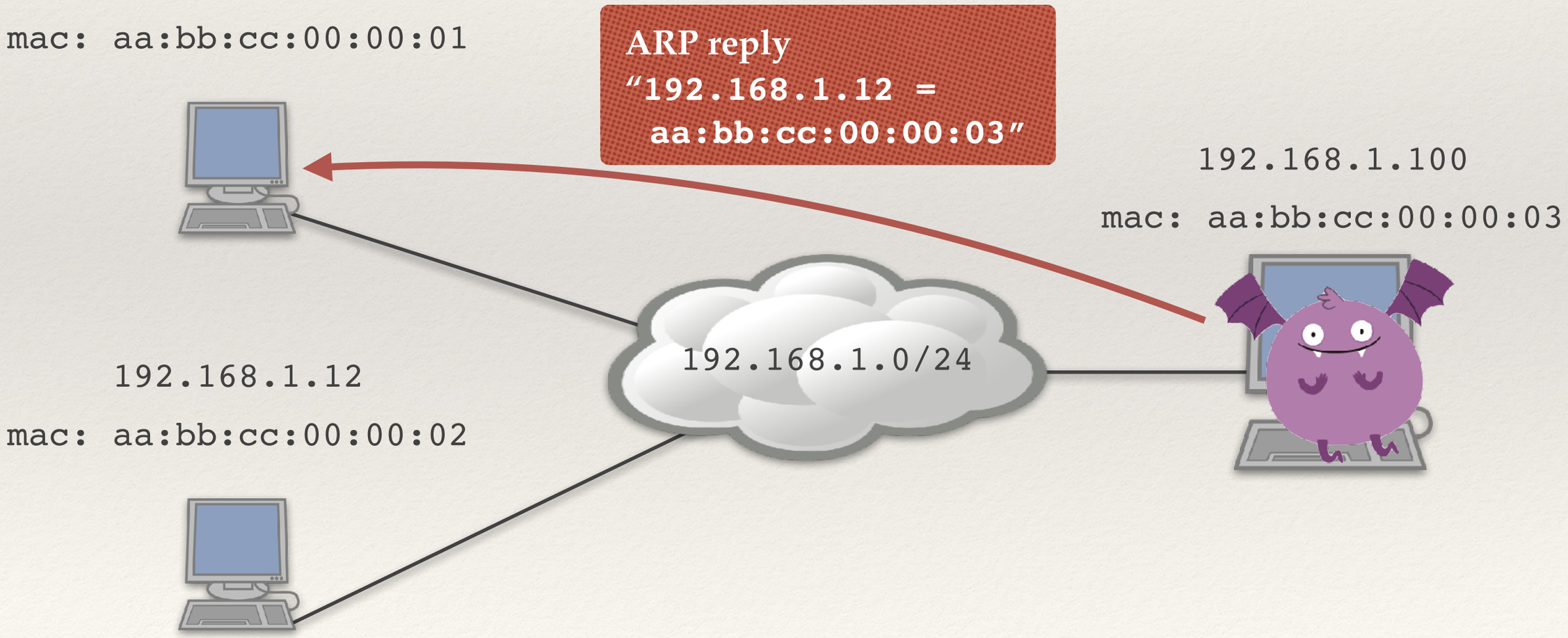


People and hand vectors created by www.freepik.com

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Zatruwanie pamięci podręcznej ARP.**

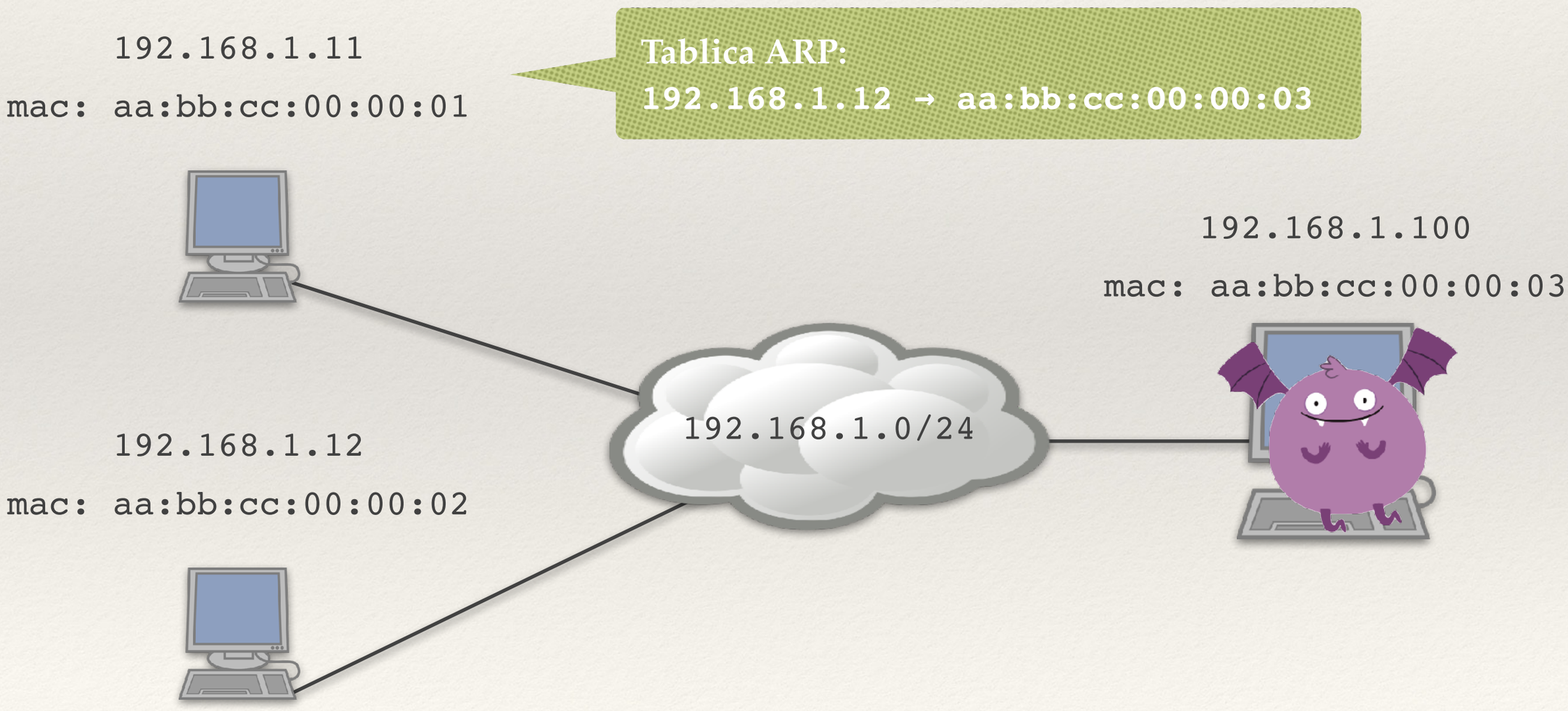


People and hand vectors created by www.freepik.com

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Zatruwanie pamięci podręcznej ARP.**

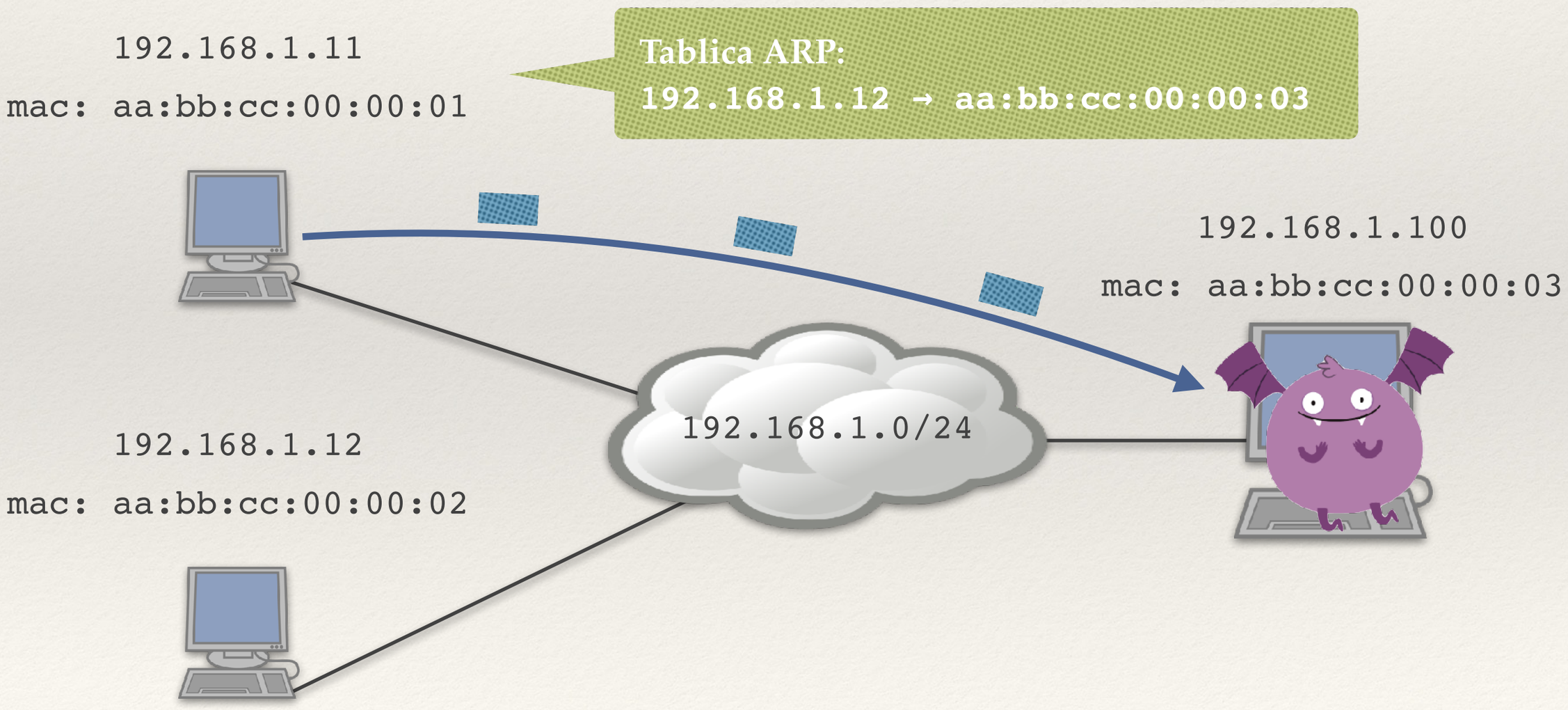


People and hand vectors created by www.freepik.com

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Zatruwanie pamięci podręcznej ARP.**



People and hand vectors created by www.freepik.com

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Własny serwer DHCP.**
  - Atakujący rozgłasza swój adres IP jako bramę domyślną.
  - Dobre przełączniki mogą filtrować takie odpowiedzi.

Ataki „man-in-the-middle”: sieci lokalne

Podsłuchujemy przez wstawienie swojego urządzenia „w środku“ ścieżki komunikacji.

- **Własny serwer DHCP.**
  - Atakujący rozgłasza swój adres IP jako bramę domyślną.
  - Dobre przełączniki mogą filtrować takie odpowiedzi.
- Własny punkt dostępowy nazywający się *Free City Internet*
  - Kto sprawdza chociaż adres MAC punktu dostępowego?
  - WPA Personal nie uwierzytelnia punktu dostępowego.

Ataki „man-in-the-middle”: routing

- **RIP spoofing**
  - RIPv1 nie ma uwierzytelniania
  - Wystarczy rozgłaszać trasę do różnych sieci o małym koszcie.

Ataki „man-in-the-middle”: routing

- **RIP spoofing**
  - RIPv1 nie ma uwierzytelniania
  - Wystarczy rozgłaszać trasę do różnych sieci o małym koszcie.
- **Nadużycia BGP**
  - Są uwierzytelnienia, ale przez wiele lat nie sprawdzano, czy dany zakres adresów faktycznie należy do danego AS.
  - 1997: ISP w Virginii, USA, ogłosza, że cały Internet należy do niego.
  - 2008: ISP z Pakistanu ogłasza, że ma zakres adresów należący do YouTube.
  - 2020: ISP z Rosji ogłasza, że ma zakres adresów należący do Google, Facebook, Akamai, Cloudflare, i Amazona.

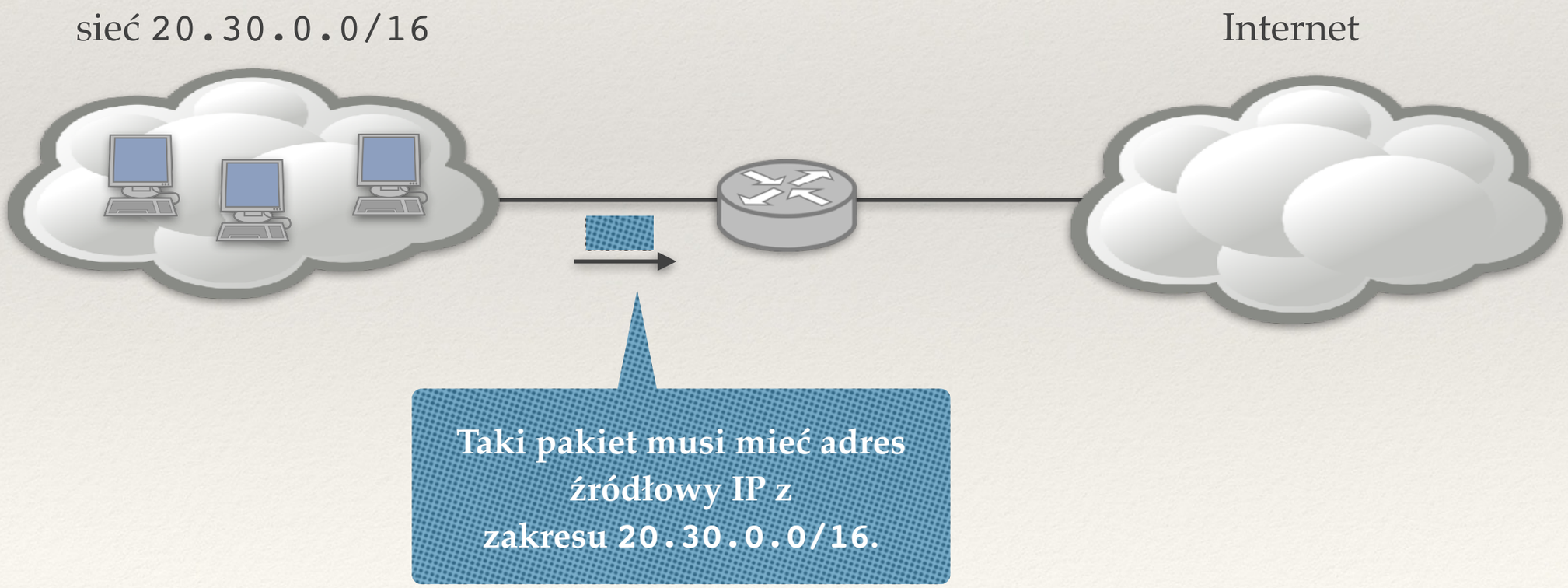
IP spoofing

- **Fałszowanie adresu źródłowego IP.**
  - Unikanie odpowiedzialności za atak.
  - Uzyskiwanie dostępu do niektórych usług.

IP spoofing

- **Fałszowanie adresu źródłowego IP.**
  - Unikanie odpowiedzialności za atak.
  - Uzyskiwanie dostępu do niektórych usług.

- **Rozwiązanie:** ***ingress filtering / martian address filtering***
  - *S*kuteczne jeśli router jest blisko nadawcy.



Zatruwanie pamięci podręcznej DNS (1)

- Atakujący kontroluje serwer o adresie IP = `11.22.33.44`.
- Resolver DNS (*R*) ma pamięć cache, w której pamięta niedawno odpytywane domeny.
- Atakujący chce, żeby dostał się tam fałszywy wpis `amazon.com` → `11.22.33.44`.

Zatruwanie pamięci podręcznej DNS (1)

- Atakujący kontroluje serwer o adresie IP = `11.22.33.44`.
- Resolver DNS (*R*) ma pamięć cache, w której pamięta niedawno odpytywane domeny.
- Atakujący chce, żeby dostał się tam fałszywy wpis `amazon.com` → `11.22.33.44`.

- **Stara wersja ataku:**
  - Atakujący wysyła do *R* zapytanie o domenę `xyz.com`, którą posiada.
  - *R* pyta o `xyz.com` serwer nazw kontrolowany przez atakującego.
  - W odpowiedzi na zapytanie o `xyz.com` atakujący odpowiada dodatkowo rekordem `amazon.com` → `11.22.33.44`.
  - Współcześnie taki dodatkowy rekord zostanie zignorowany.

Zatruwanie pamięci podręcznej DNS (2)

Nowa wersja ataku:

- Atakujący wysyła do *R* zapytanie o `amazon.com`
- *R* wysyła zapytanie (przez UDP) o `amazon.com` do odpowiedzialnego serwera DNS (o adresie IP = *X*)
- Atakujący wysyła odpowiedzi DNS (datagramy UDP) podszywając się pod *X*.
- *R* sprawdza, czy w odpowiedzi jest taki sam 16-bitowy ID jak w zapytaniu…
- … wystarczy, że atakujący wyśle 2^{16} odpowiedzi ze wszystkimi możliwymi ID.

Zapobieganie:

- DNSSEC (kryptograficzne uwierzytelnianie pakietów DNS) → stosowane m.in. przez serwery główne DNS.
- Ataki na DNS są mniej współcześnie mniej skuteczne ze względu na uwierzytelnianie punktów końcowych połączenia (w TLS).

Kryptografia na ratunek

TLS (1)

- Następca SSL (Secure Socket Layer).
- Warstwa pośrednicząca pomiędzy warstwą transportową i warstwą aplikacji.
- Odpowiada za szyfrowanie i uwierzytelnianie.
- Większość tradycyjnych usług ma swoje szyfrowane warianty:
  - HTTPS = HTTP + TLS
  - POP3 + TLS
  - SMTP + TLS

TLS (2)

Hello!

Mój klucz publiczny to C.

Symetryczny klucz sesji K zaszyfrowany C.

Komunikacja szyfrowana K.

domena www.example.com, klucz publiczny *C* i prywatny *c*

TLS (3)

Uwierzytelnianie serwera: za pomocą certyfikatu.

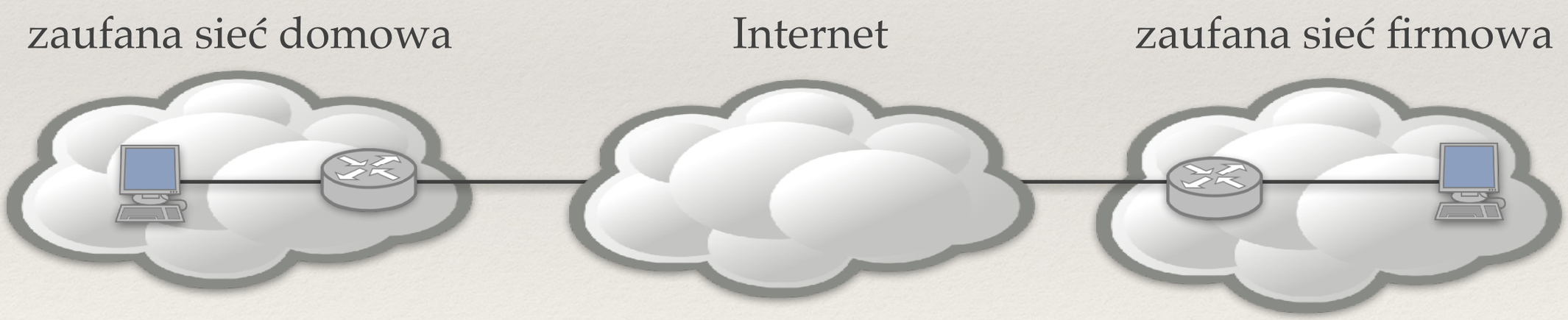
- Klucz publiczny serwera jest podpisany przez centrum uwierzytelniające (CA).
- Możemy zweryfikować autentyczność tego podpisu bo mamy klucz publiczny centrum (w przeglądarce).

Uwierzytelnianie klienta: za pomocą loginu i hasła.

VPN

VPN = Wirtualna sieć prywatna

- Mamy dwie sieci połączone Internetem i chcemy zrobić z nich jedną logiczną sieć.
- Transmisja wewnątrz każdej z nich jest bezpieczna, ale transmisja w Internecie już nie.



Tunelowanie

Model warstwowy

- Protokół warstwy *i+1* jest przesyłany jako *dane* protokołu warstwy *i*.

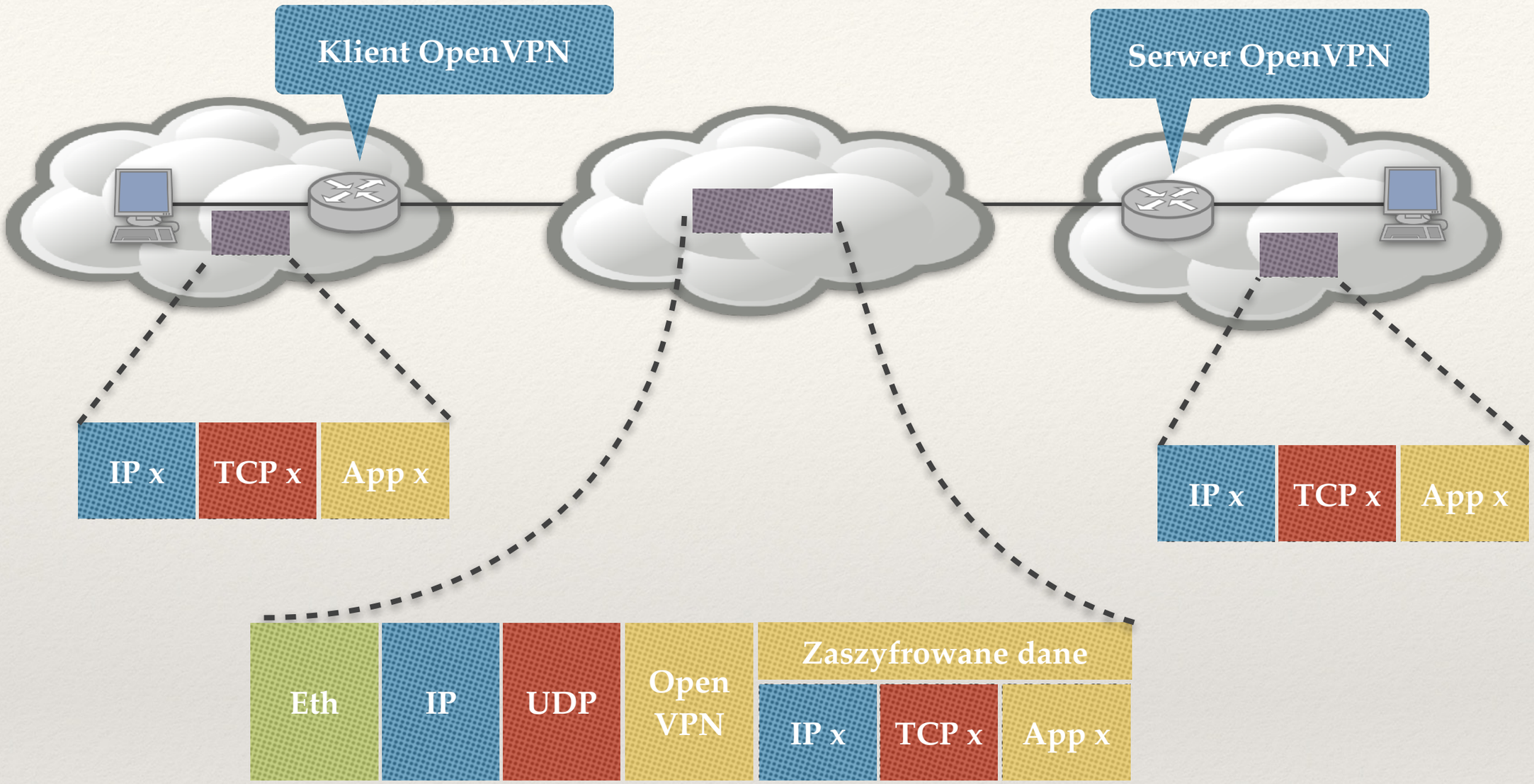
Tunelowanie

- Przesyłanie pewnych usług sieciowych za pomocą innych usług sieciowych w sposób łamiący standardowy model warstwowy.
- Cel: zestawianie wirtualnego (często szyfrowanego) połączenia.
- Poznaliśmy już tunelowanie pakietów IPv6 w pakietach IPv4.

VPN, dwa popularne podejścia:

- **IPSec**: tunelowanie (szyfrowanych) pakietów IP w danych pakietów IP.
- **OpenVPN lub WireGuard** : tunelowanie (szyfrowanych) pakietów IP w danych protokołu UDP.

OpenVPN lub WireGuard



- Logiczne działanie tak jakby klient i serwer VPN stanowiły jedno urządzenie (router).
- W przypadku OpenVPN możliwe jest też tunelowanie ramek.

SSH (Secure SHell)

Standardowe narzędzie pracy zdalnej (w terminalu tekstowym).

- Następca nieszyfrowanego `telneta`.
- Mechanizmy szyfrowania jak przy TLS.
- Co z uwierzytelnianiem serwera?

SSH: uwierzytelnianie serwera

Znowu nie wiemy, czy łączymy się z dobrym serwerem!

- Przy pierwszym połączeniu serwer przesyła nam klucz publiczny, program klienta oblicza funkcje skrótu klucza.

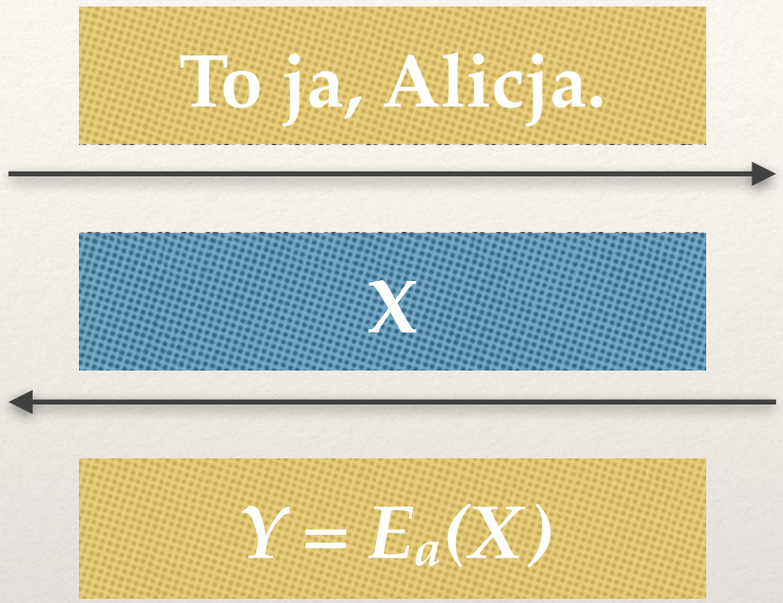
```
The authenticity of host 'ssh-server.example.com
(12.18.49.21)' can't be established. RSA key
fingerprint is
98:2e:d7:e0:de:9f:ac:67:28:c2:42:2d:37:16:58:4d. Are
you sure you want to continue connecting?
```

- Warto skomunikować się z opiekunem serwera i zweryfikować poprawność funkcji skrótu.
- Po zaakceptowaniu klucz publiczny serwera zostaje zapisany lokalnie.

SSH: uwierzytelnianie klienta

- Klient może po prostu podać hasło do konta. Albo:

klucz publiczny A
i prywatny a



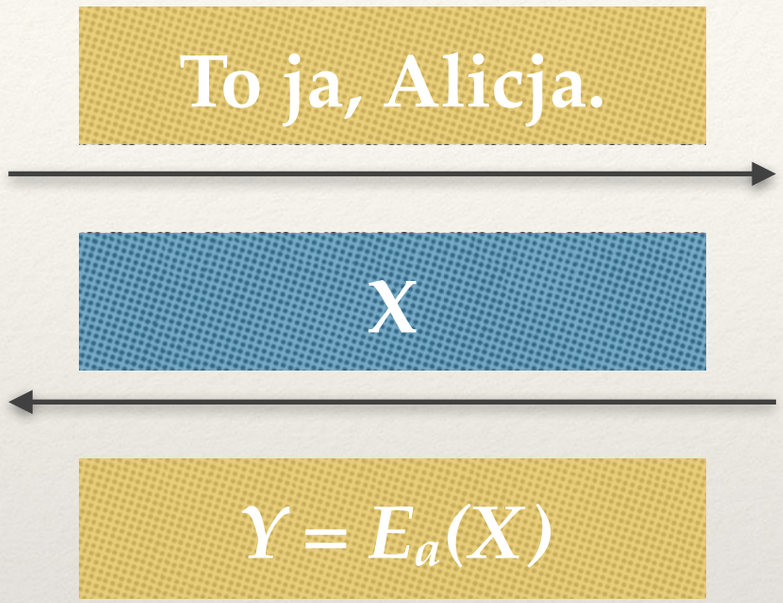
Zna A. Sprawdza,
czy $E_A(Y) = X$.

- Serwer zna klucz publiczny A, bo Alicja zapisała go uprzednio w pliku `authorized_keys` na serwerze.

SSH: uwierzytelnianie klienta

- Klient może po prostu podać hasło do konta. Albo:

klucz publiczny A
i prywatny a



Zna A. Sprawdza,
czy $E_A(Y) = X$.

- Serwer zna klucz publiczny A, bo Alicja zapisała go uprzednio w pliku `authorized_keys` na serwerze.
- Lepsze niż hasło, bo podsłuchanie takiej komunikacji (nawet rozszyfrowanej) nie pozwoli na ponowne zalogowanie się.

Tunelowanie TCP w SSH

- Jeśli mamy konto na zdalnej maszynie, możemy szyfrować połączenia z tymi usługami za pomocą `ssh`.
- `ssh -L 4025:localhost:25 user@zdalny-serwer`

  przekazuje segmenty TCP skierowane do portu 4025 lokalnego komputera w (zaszyfrowanych) danych SSH do `zdalny-serwer` (do portu 22) a następnie do portu 25 `zdalny-serwer`.

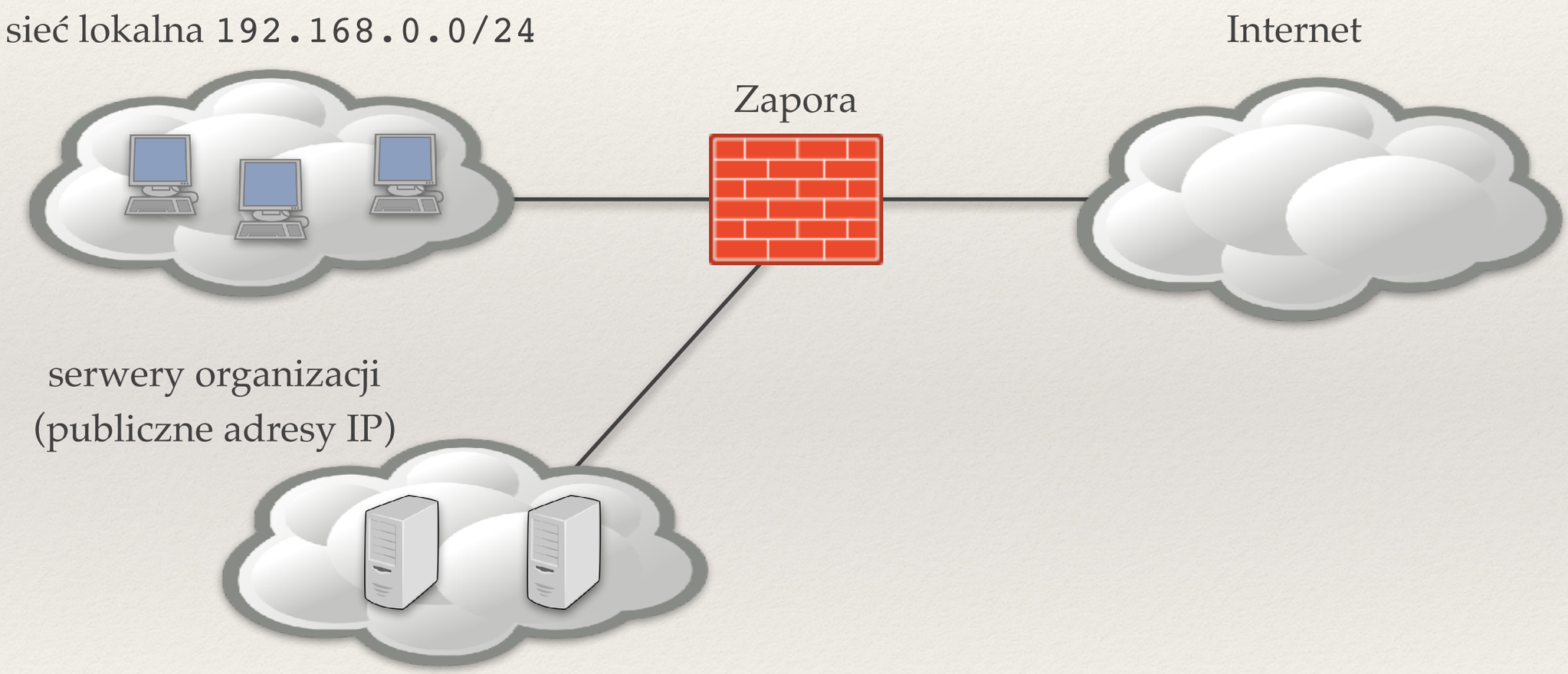
Zapora: kontrolowanie dostępu

Po co?

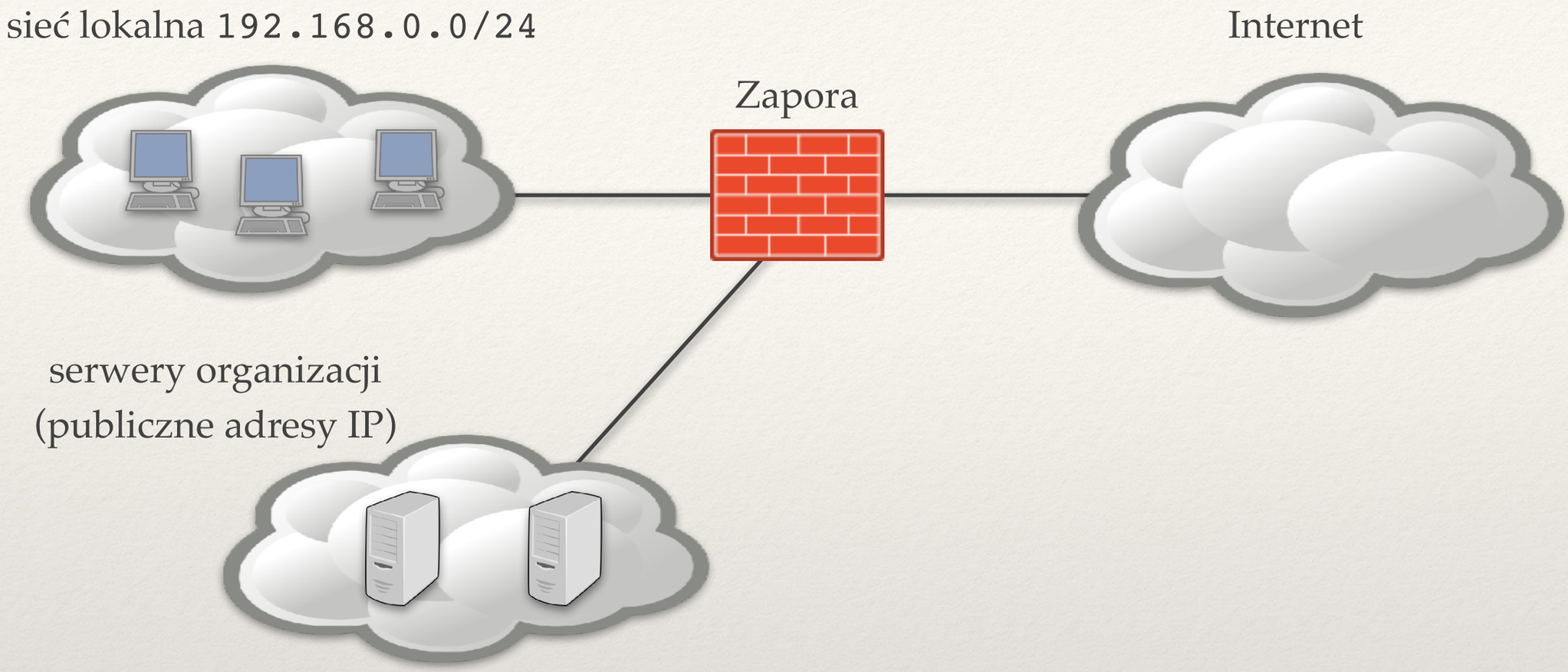
- Pierwsza linia obrony.
- Rejestrowanie i kontrolowanie dostępu do usług.

Gdzie?

- Często zapora jest osobnym urządzeniem, pełniącym też funkcję routera:



Polityka



Zapora realizuje pewną politykę bezpieczeństwa. Przykładowo:

- Wpuszcza pakiety do serwerów organizacji (do określonych portów).
- Wpuszcza pakiety do portu SSH serwerów tylko z sieci wewnętrznej.
- Nie wpuszcza pakietów do sieci wewnętrznej.

Klasyfikacja filtrów pakietów

Filtry proste (warstwa sieciowa)

- Analizują tylko nagłówki IP (+ porty).
- Szybkie, bardzo nieprecyzyjne.

Filtry stanowe (warstwa transportowa)

- Analizują nagłówki IP i TCP.
- Śledzą nawiązywania połączenia TCP, pamiętają stan połączenia.

Filtry działające w warstwie aplikacji

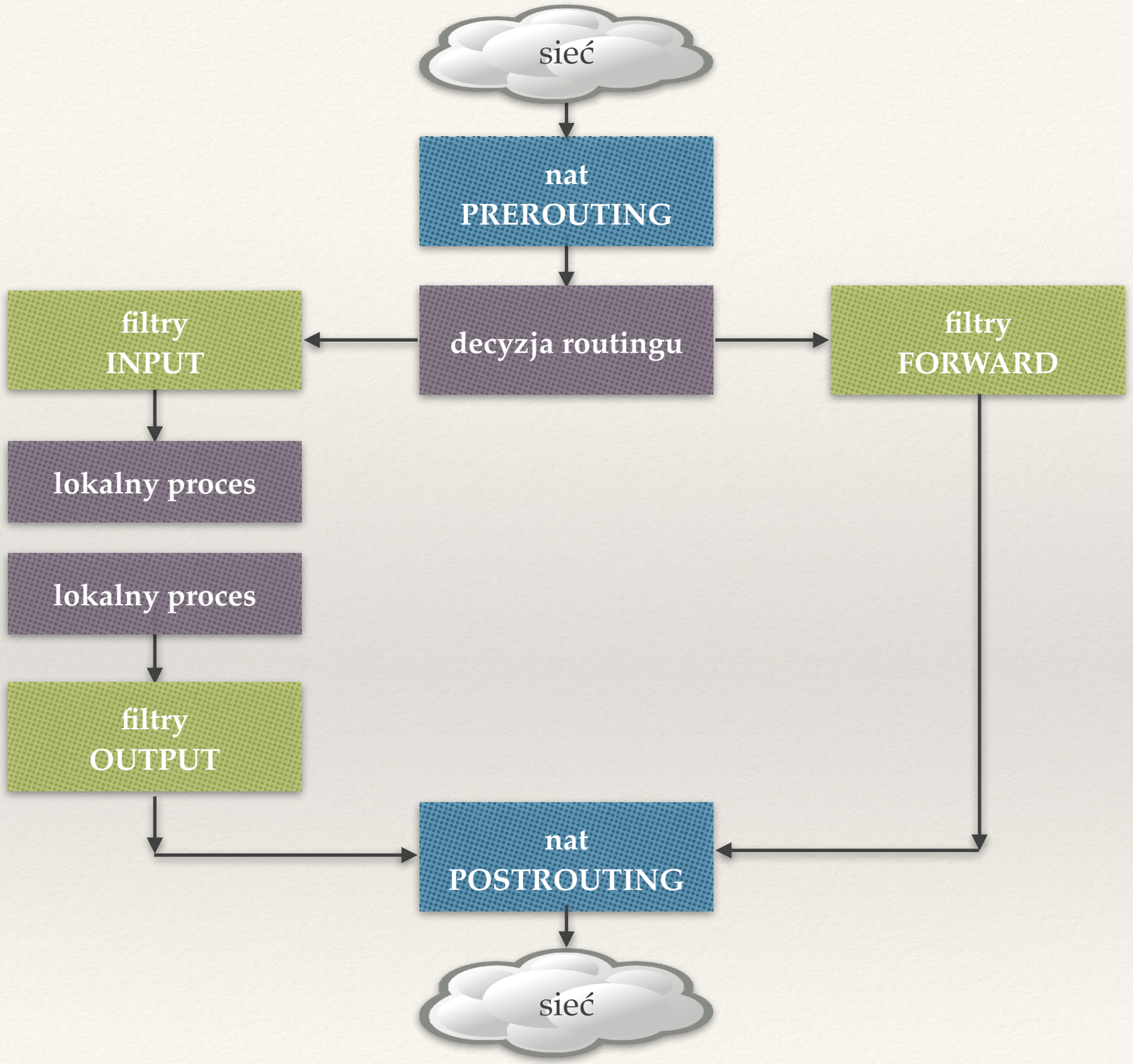
- Analizują zawartość segmentów i datagramów
- Potrafią otwierać dodatkowe porty.
- To co innego niż “zapory aplikacji”, które analizują *wywołania systemowe* aplikacji.

Netfilter/nftables

Filtr pakietów w Linuksie

- Filtr stanowy.
- Elementy filtra działającego w warstwie aplikacji (możliwość śledzenia połączeń niektórych protokołów).

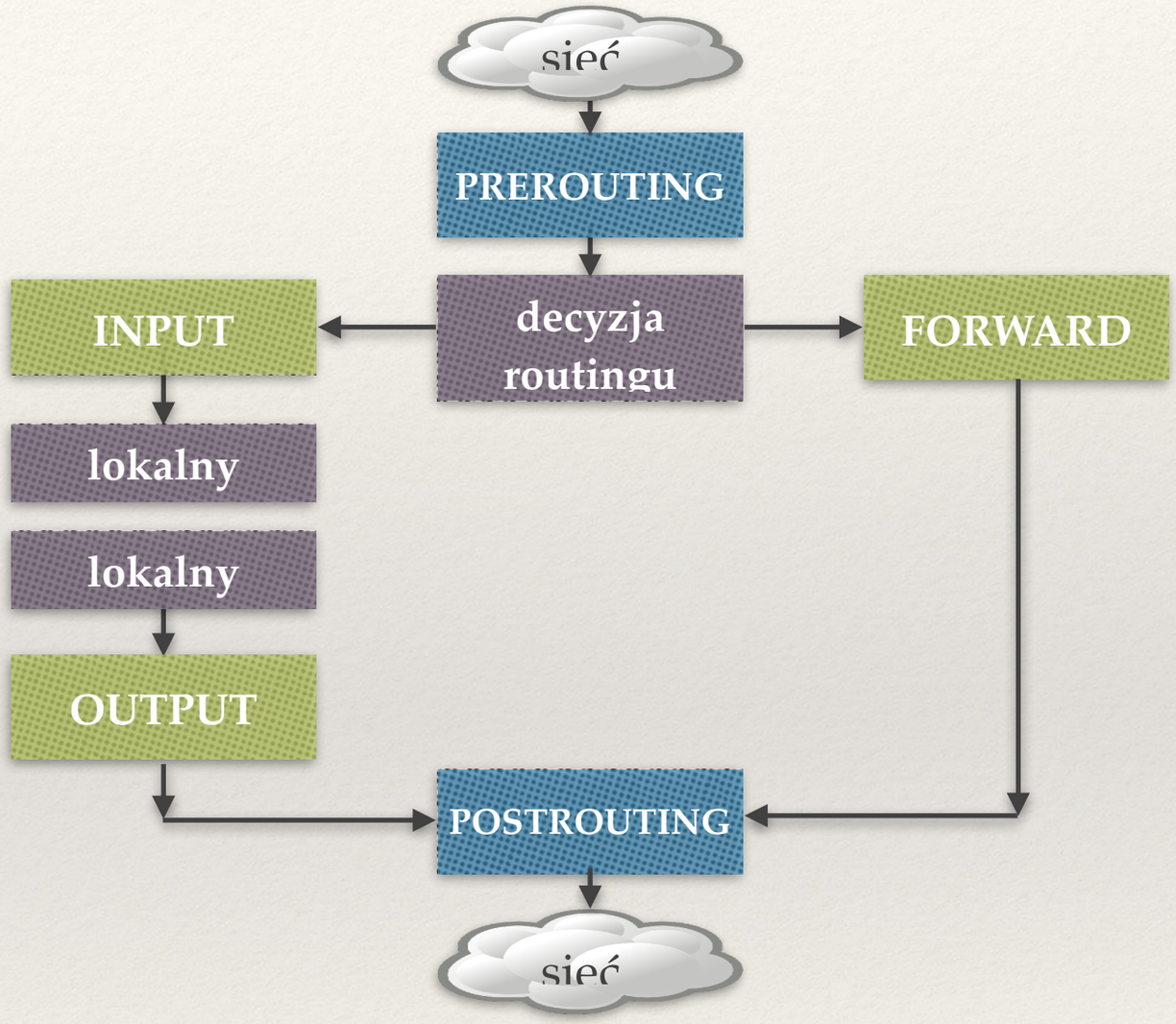
Przetwarzanie pakietów (1)



Przetwarzanie pakietów (2)

Idea

- Tworzymy łańcuch reguł.
- Reguły z łańcucha są uruchamiane w odpowiednim miejscu diagramu.
- Reguły w łańcuchach przetwarzane po kolei do pierwszej pasującej reguły.



Przykładowy łańcuch (pakiety wychodzące)

```
table inet my_table {
  chain my_chain {
    type filter hook output priority filter;
    accept
  }
}
```

- Akceptuj wszystkie pakiety wychodzące

Przykładowy łańcuch (pakiety przychodzące)

```
table inet my_table {
  chain my_chain {
    type filter hook input priority filter;
    tcp dport 22 accept
    reject
  }
}
```

- Akceptuj wszystkie pakiety przychodzące do portu SSH.
- Odrzuć resztę.

Czy to wystarczy?

- Pakiety wychodzące np. do portu 80 bedą wypuszczane, ale odpowiedzi na te pakiety bedą odrzucane.

Stan połączeń

Nieprecyzyjna heurystyka (stosowana w prostych filtrach)

- Wpuść pakiety przychodzące do portów ≥ 32678.

Filtry stanowe

- Możemy wpuszczać pakiety związane wysłanymi pakietami.
- W szczególności segmenty TCP już nawiązanego połączenia.

Przykładowy łańcuch (pakiety przychodzące)

```
table inet my_table {
  chain my_chain {
    type filter hook input priority filter;
    ct state established accept
    ct state new tcp dport 22 accept
    reject
  }
}
```

- Akceptuj wszystkie pakiety przychodzące do portu SSH.
- Akceptuj odpowiedzi na już wysłane pakiety.
- Odrzuć resztę.

Odrzucanie pakietów

Zgodnie ze standardem:

- Przez `reject`.
- Zapora odpowie komunikatem ICMP *port-unreachable*.

Wyrzucenie pakietu bez żadnego komunikatu:

- Przez `drop`.
- Utrudnia pracę skanerów portów → muszą czekać na timeout dla połączenia.

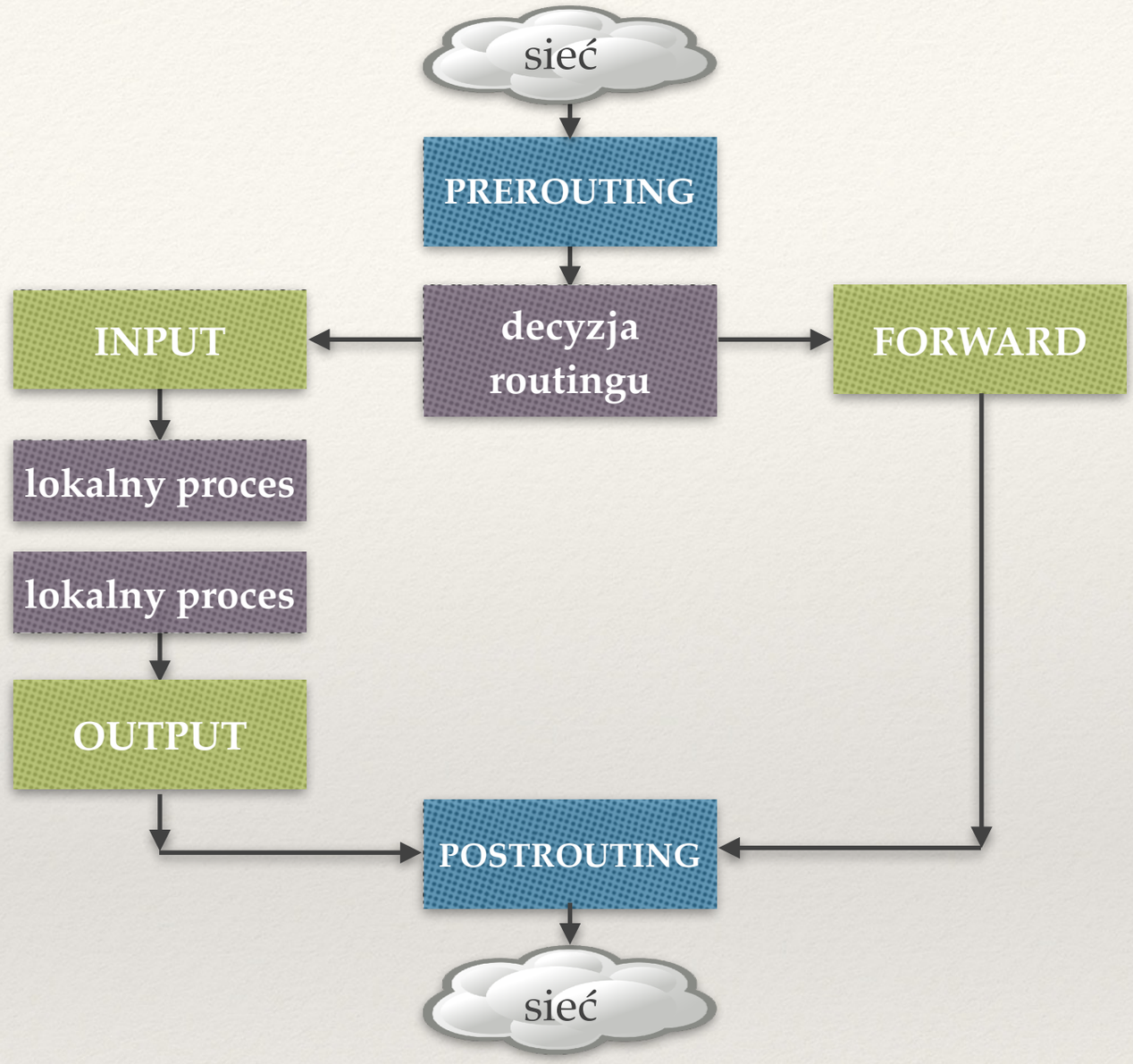
NAT

POSTROUTING:

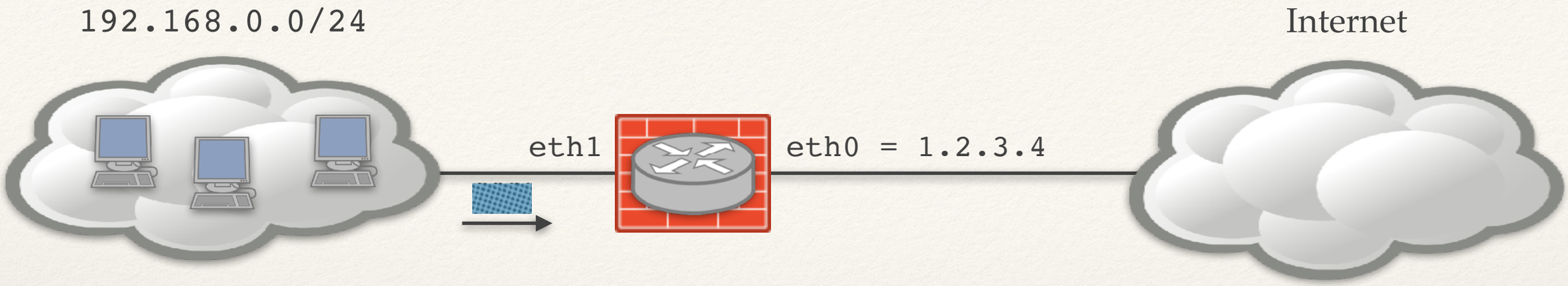
- Podmiana źródłowych adresów IP.
- Ostatnia czynność przed wysłaniem pakietu.

PREROUTING:

- Podmiana docelowych adresów IP
- Przykładowo: przekierowanie do innego komputera.



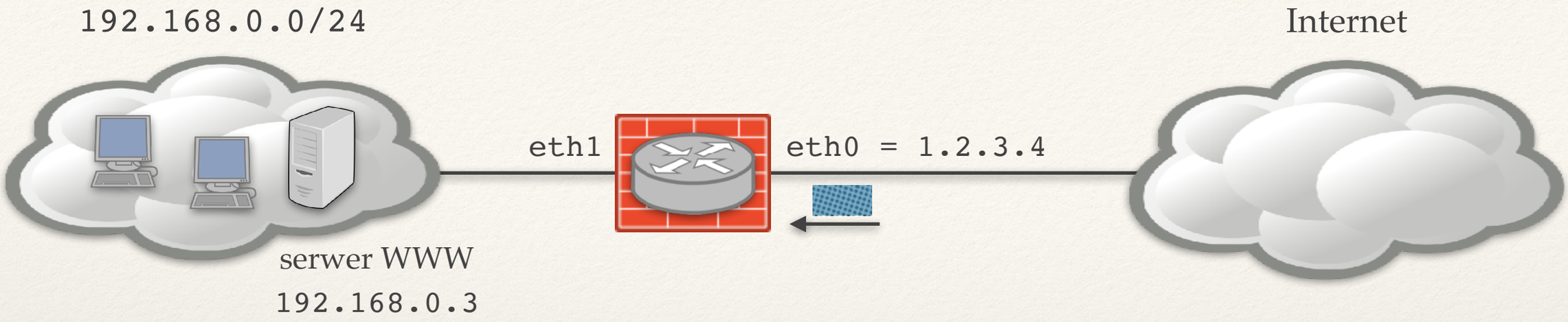
Źródłowy NAT (SNAT)



Podmiana adresu źródłowego pakietu wychodzącego z LAN:

```
table ip my_nat_table {
  chain my_chain {
    type nat hook postrouting priority srcnat;
    ip sadddr 192.168.0.0/24 oif eth0 snat 1.2.3.4
  }
}
```

Docelowy NAT (DNAT)



Podmiana adresu docelowego pakietu przychodzącego do LAN:

```
table ip my_nat_tables {
  chain my_chain {
    type nat hook prerouting priority dstnat;
    iif eth0 tcp port 80 dnat 192.168.0.3:80
  }
}
```

Źródła podatności na ataki

Źródła podatności na ataki

- **Źle zaprojektowane protokoły.**

- **Błędy implementacji.**
  - Najczęściej: brak weryfikacji poprawności wprowadzanych danych.
  - Atakujący zmusza aplikację do wykonania nieprzewidzianych operacji.

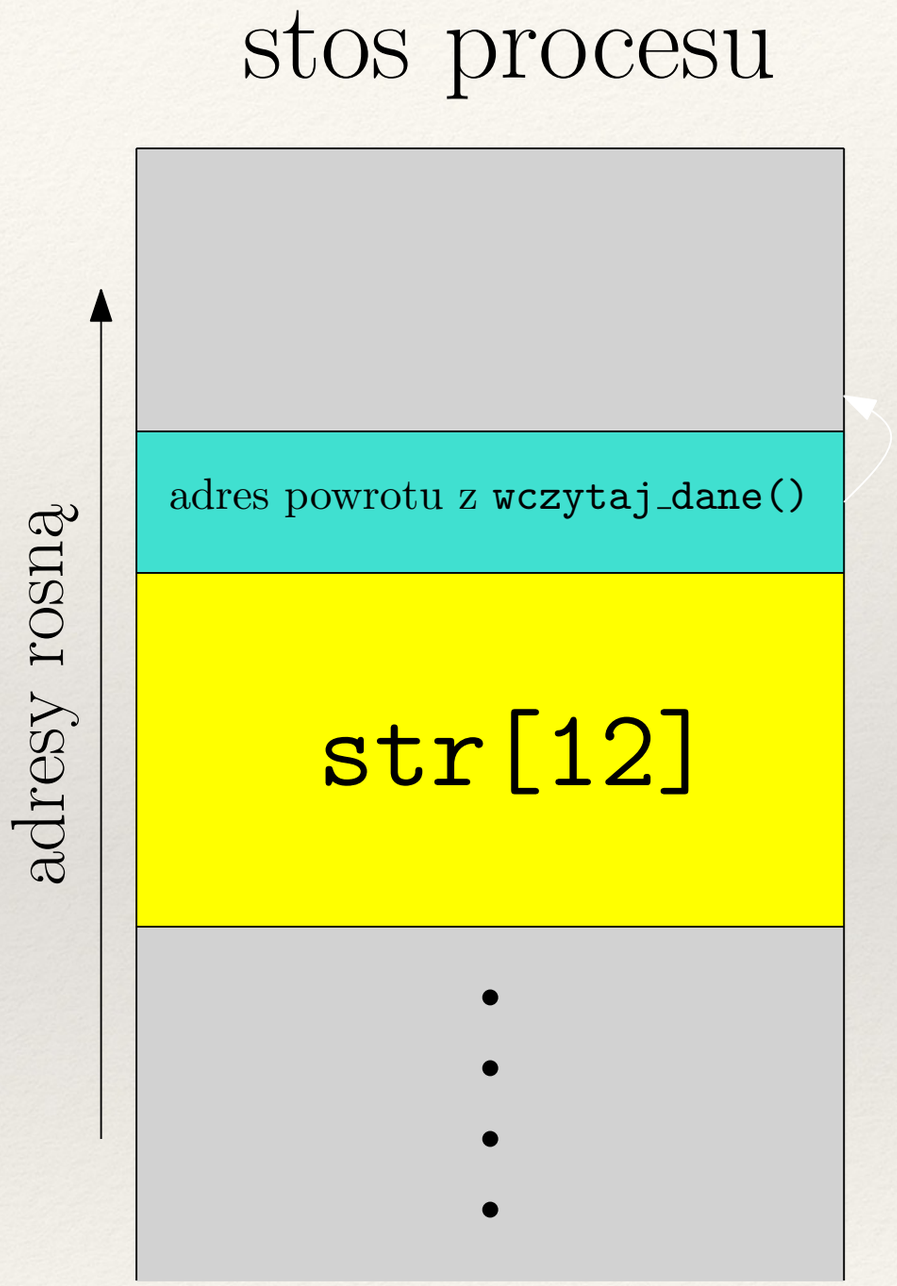
- **Czynnik ludzki**

Błędy implementacji: przepełnienie bufora

```
void wczytaj_dane() {
   char str[12];
   scanf (“%s“, str);
}
```

Atakujący wpisuje:

- jakieś 12 znaków,
- odpowiedni adres powrotu,
- kod maszynowy do uruchomienia.

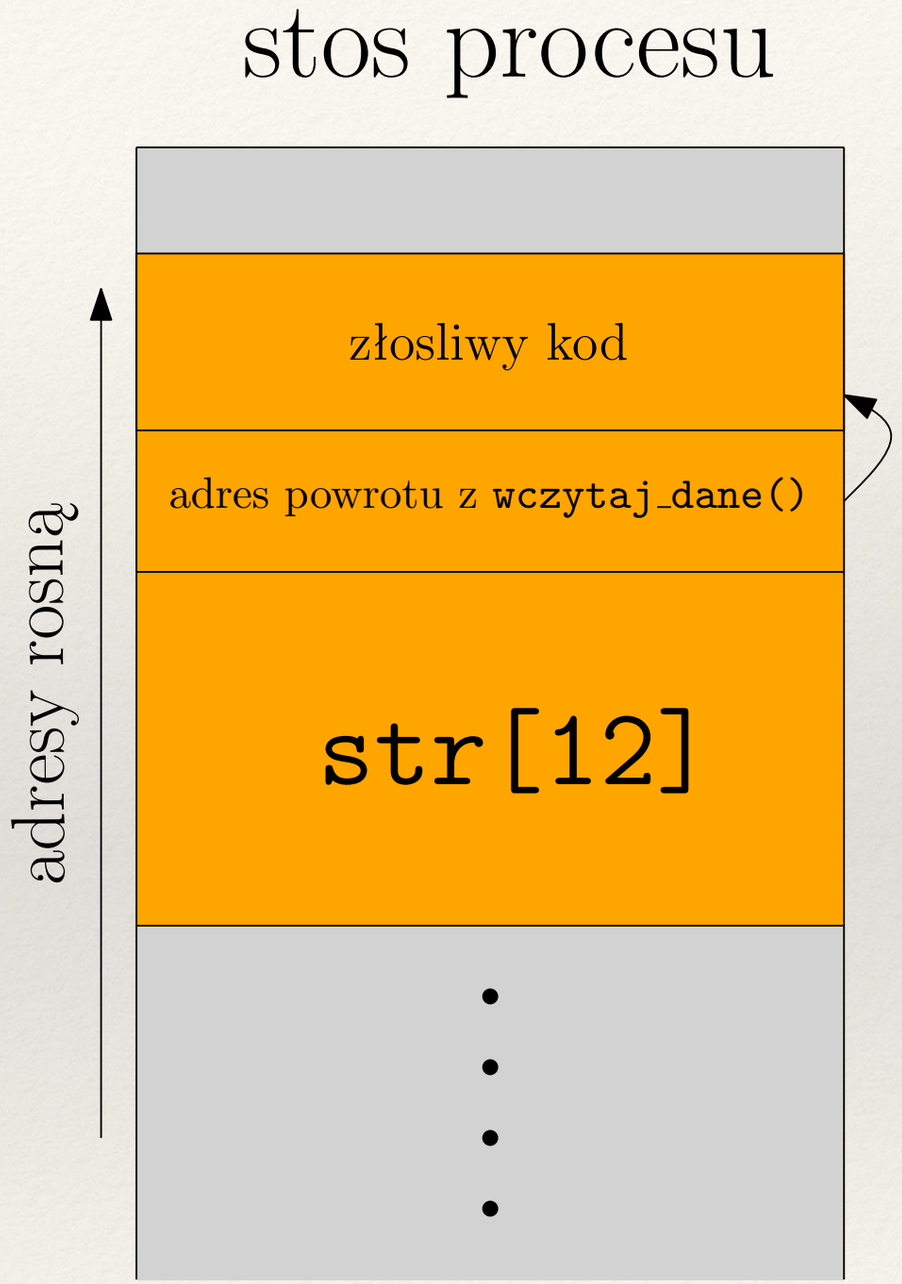


Błędy implementacji: przepełnienie bufora

```
void wczytaj_dane() {
   char str[12];
   scanf (“%s“, str);
}
```

Atakujący wpisuje:

- jakieś 12 znaków,
- odpowiedni adres powrotu,
- kod maszynowy do uruchomienia.



Błędy implementacji: atak typu ../

Pomocniczy skrypt WWW

- `skrypt` jest w katalogu `/var/www/`.
- `skrypt` jest programem wyświetlającym zawartość pliku.
- Przykładowo `http://example.com/skrypt?plik=test` wyświetla zawartość pliku `/var/www/test`.

Błędy implementacji: atak typu ../

Pomocniczy skrypt WWW

- `skrypt` jest w katalogu `/var/www/`.
- `skrypt` jest programem wyświetlającym zawartość pliku.
- Przykładowo `http://example.com/skrypt?plik=test` wyświetla zawartość pliku `/var/www/test`.

- Co otrzymamy po wejściu na stronę `http://example.com/skrypt?plik=../../etc/passwd`?

Błędy implementacji: atak SQL injection

Aplikacja WWW odwołująca się do bazy danych

- Skrypt dostaje przez formularz argument `$user` i wykonuje polecenie
  - `mysql_query("SELECT * FROM tab WHERE user = '$user'");`

Błędy implementacji: atak SQL injection

Aplikacja WWW odwołująca się do bazy danych

- Skrypt dostaje przez formularz argument `$user` i wykonuje polecenie
  - `mysql_query("SELECT * FROM tab WHERE user = '$user'");`

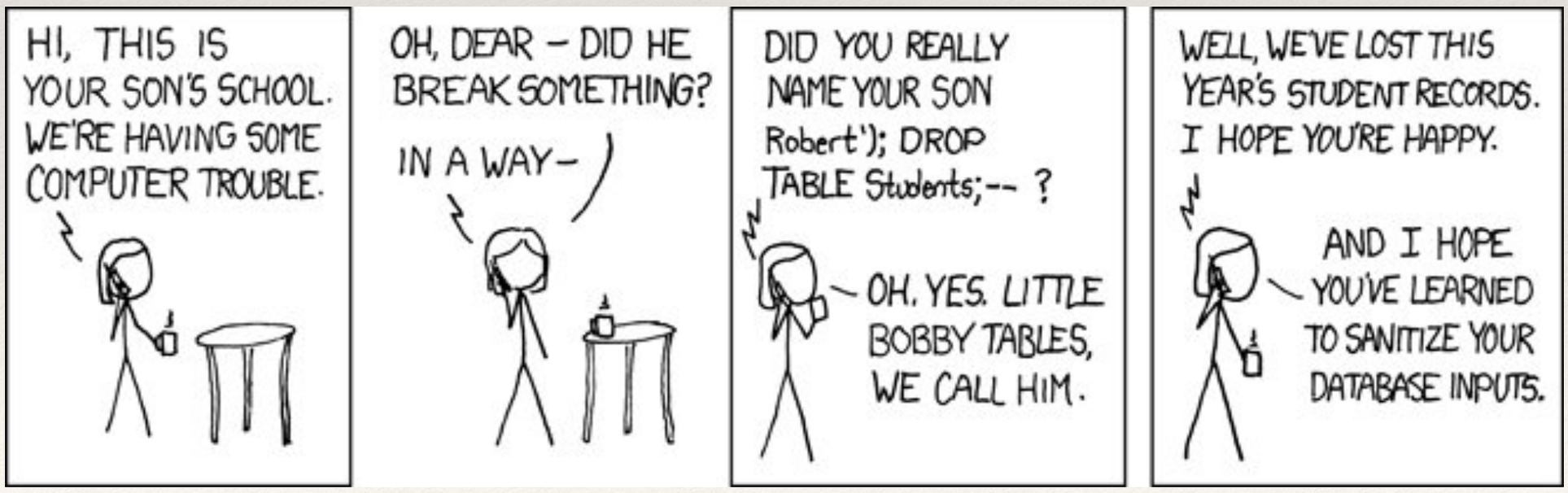
- Podajemy w tym polu: `x' OR '1'='1`

Błędy implementacji: atak SQL injection

Aplikacja WWW odwołująca się do bazy danych

- Skrypt dostaje przez formularz argument `$user` i wykonuje polecenie
  - `mysql_query("SELECT * FROM tab WHERE user = '$user'");`

- Podajemy w tym polu: `x' OR '1'='1`
- Albo lepiej: `x'; DROP TABLE tab; --`



Obrazek ze strony `https://xkcd.com/327/`

Błędy implementacji: wysyłanie części RAM procesu

- **Protokół: otrzymujemy od serwera fragment przechowywanej przez niego tablicy.**
  - Klient prosi go o więcej niż rozmiar tablicy + serwer nie sprawdza tego rozmiaru.
  - Serwer odsyła część pamięci procesu.
  - Przykład: *SSL heartbleed*.

Błędy implementacji

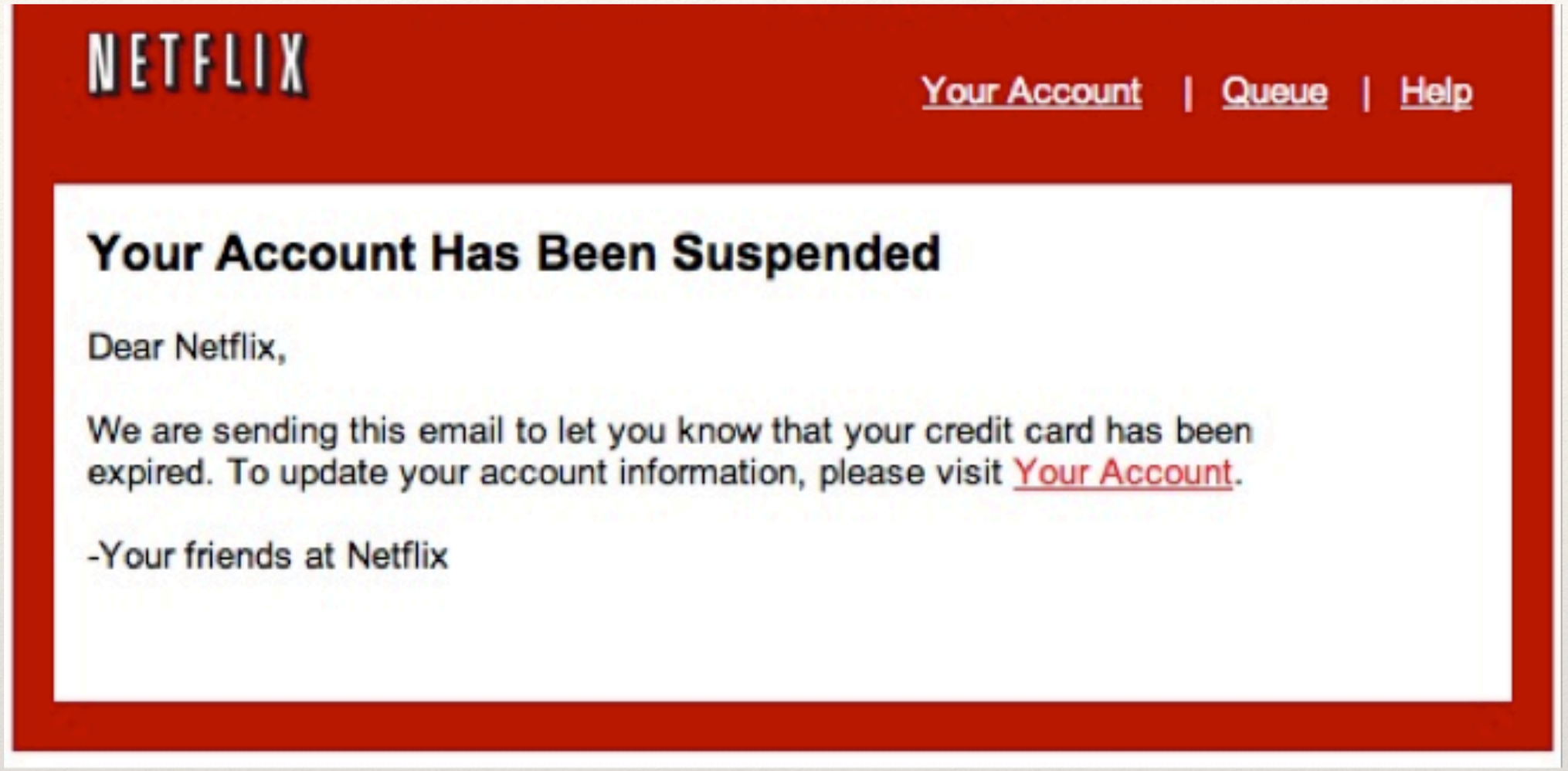
Co można zrobić?

- Zawsze sprawdzać poprawność danych od użytkownika
  - Sprawdzanie danych po stronie klienta nie jest bezpieczne: np. nie można zakładać, że użytkownik używa przepisowego klienta usługi.

- Ogólne techniki na poziomie jądra: np. zabronienie wykonywania programów na stosie.

- Zasada minimalnych uprawnień:
  - program A działający z uprawnieniami B ma lukę;
  - atakujący może wykonać to, na co pozwalają uprawnienia B.

Czynnik ludzki: phishing



Obrazek ze strony `http://resources.infosecinstitute.com/category/enterprise/phishing/`

Odnośnik „Your Account“ prowadzi na stronę atakującego `http://www.netflix.domena.com/`.

- Wygląda bardzo podobnie do `http://www.netflix.com/`.
- Atakujący może mieć dla niej również poprawny certyfikat HTTPS.

Skanowanie portów

Wyszukiwanie otwartych portów

- `nmap`
- Przykładowe rodzaje:
  - connect scan,
  - SYN scan,
  - ustawianie różnych dziwnych flag (np. RST+PSH).
- Najczęściej wykorzystywane narzędzie w filmach: `https://nmap.org/movies/`

Blokowanie dostępu

Denial of Service (DoS) = Blokowanie dostępu do usług

Po co?

- Wymuszenia okupu od dochodowych stron, instytucji finansowych itp.
- Blokowanie serwerów umożliwiających pobranie aktualizacji.

DoS: wyczerpywanie zasobów serwera (1)

Ataki zajmujące CPU

- Wymaganie *proof-of-work:*
  - Nie powinno być operacji tanich dla klienta i drogich dla serwera (np. ataki powodujące kolizje funkcji haszujących).
  - Przykładowo: CDN mają mechanizmy sprzętowego przyspieszania nawiązywania połączenia TLS (po stronie serwera).

- Filtrowanie pakietów na poziomie sterowników kart sieciowych (przed stosem jądra / nftables).

DoS: wyczerpywanie zasobów serwera (2)

Ataki zajmujące inne ograniczone zasoby.

- **Przykład 1. Limit jednoczesnych połączeń.**
  - Nawiązujemy połączenie TCP i nic nie wysyłamy.
  - Opcjonalnie: wiele protokołów na początku wysyła długość komunikatu: wysyłamy długość a potem już nic.

- **Przykład 2. Dużo komunikatów TCP SYN.**
  - Jądro utrzymuje tworzoną przez `listen()` kolejkę połączeń TCP oczekujących na nawiązanie.
  - Kolejka ma ograniczoną wielkość.

Ataki wolumetryczne (zalewanie łącza pakietami)

Ataki wolumetryczne (zalewanie łącza pakietami)

- Zazwyczaj pakietami UDP lub ICMP.

Ataki wolumetryczne (zalewanie łącza pakietami)

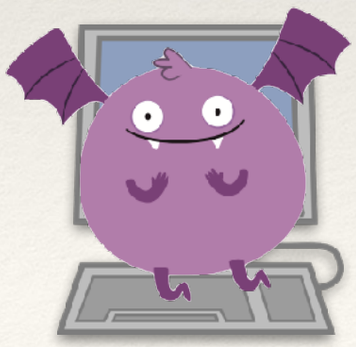
- Zazwyczaj pakietami UDP lub ICMP.
- Problem dla atakującego: musi być w stanie wysyłać szybciej komunikaty niż ofiara odbierać.
- Odbity DoS + amplifikacja rozmiaru odpowiedzi.
- DDos + botnety.

Odbity (reflected) DoS

- Zapytania z (fałszywym) adresem źródłowym równym adresowi ofiary.

ofiara

`IP = 12.34.56.78`



serwer

DNS

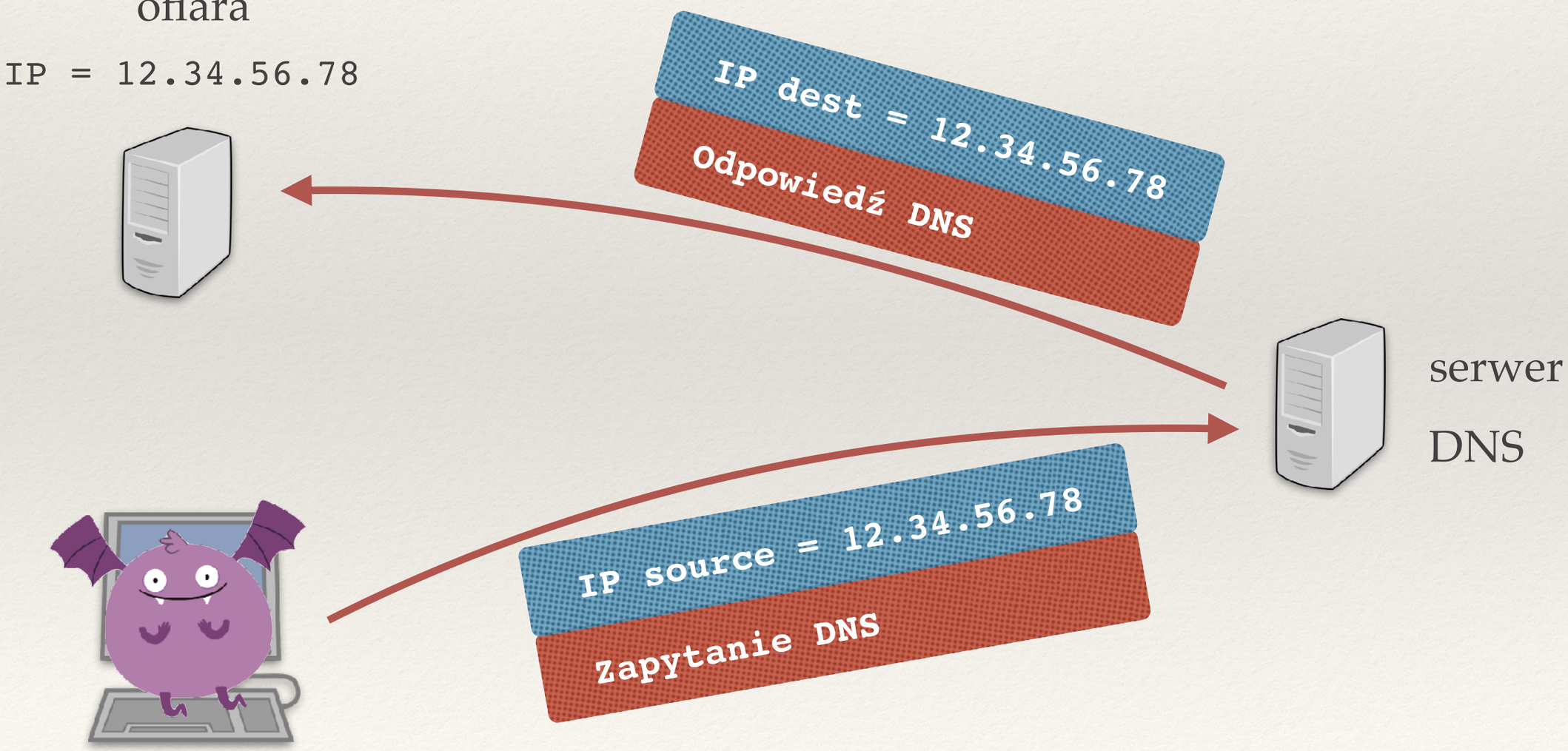
Odbity (reflected) DoS

- Zapytania z (fałszywym) adresem źródłowym równym adresowi ofiary.

People and hand vectors created by www.freepik.com

Odbity (reflected) DoS

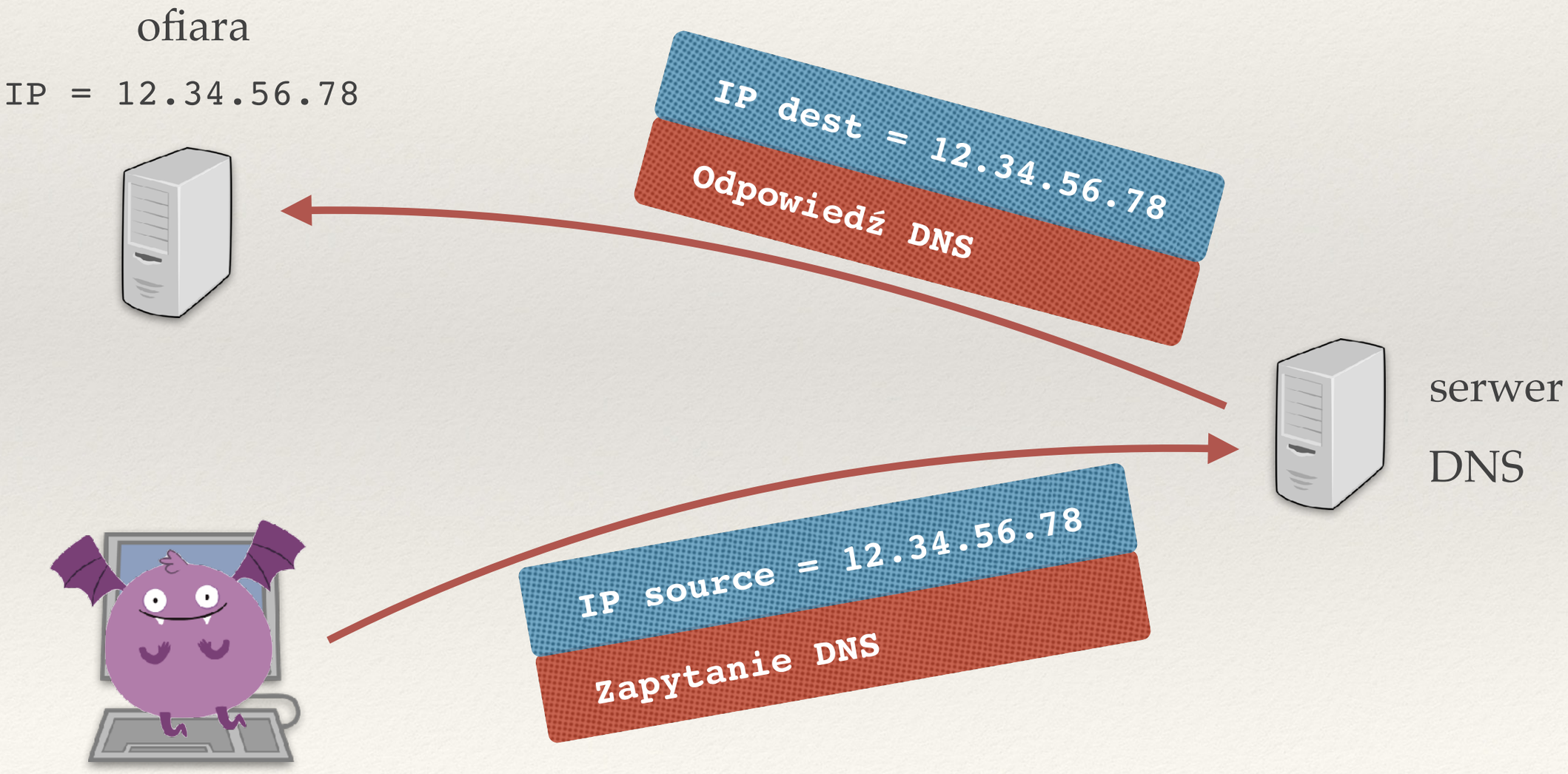
- Zapytania z (fałszywym) adresem źródłowym równym adresowi ofiary.



People and hand vectors created by www.freepik.com

Odbity (reflected) DoS

- Zapytania z (fałszywym) adresem źródłowym równym adresowi ofiary.
- Ofierze trudniej wyśledzić atakującego.
- Odpowiedź może być większa niż zapytanie (DNS do 70:1, NTP do 20:1).



People and hand vectors created by www.freepik.com

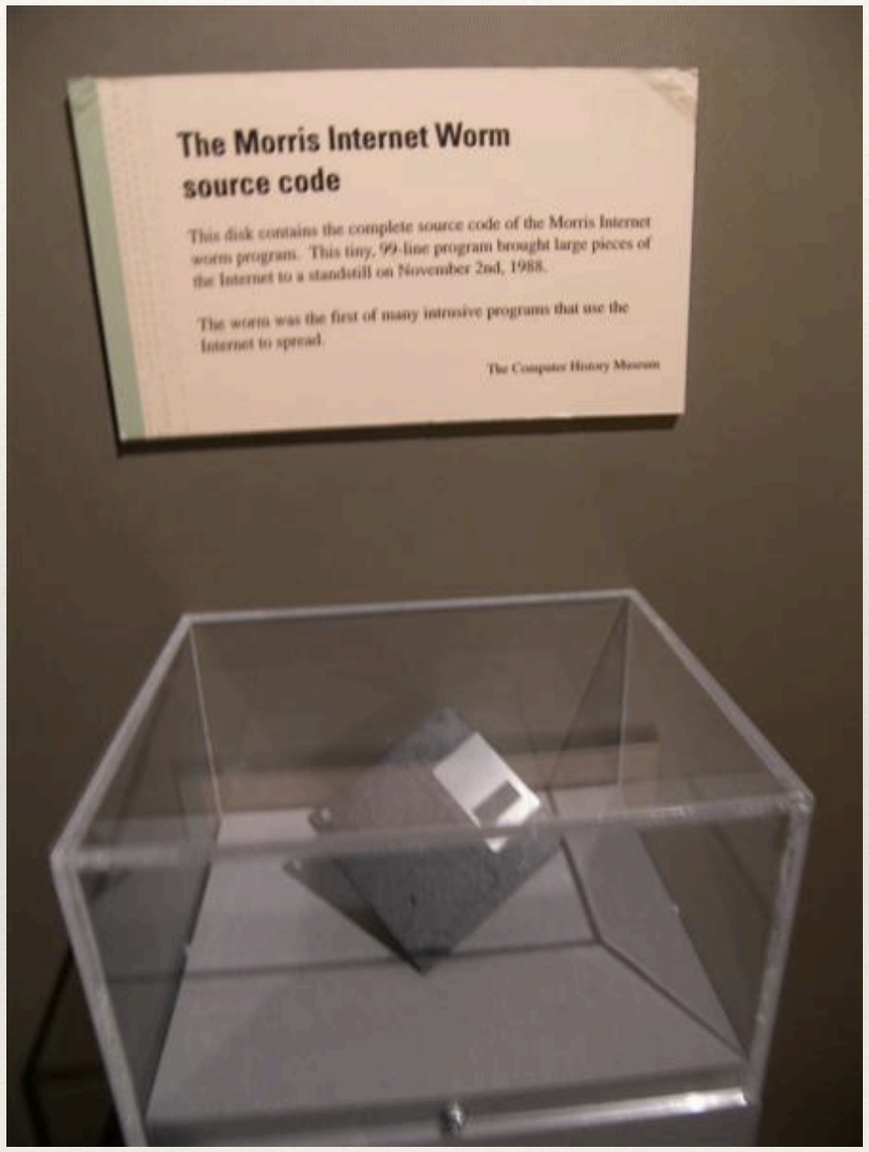
Wariant odbitego DoS: smurf attack

- Wykorzystuje ICMP *echo* (ping).
- Rozmiar odpowiedzi podobny do rozmiaru zapytania…
- … ale możemy wysłać *jeden* pakiet do przełącznika na adres broadcast!
- Przełącznik go zwielokrotni i wyśle do wszystkich uczestników sieci lokalnej.
- Wszyscy z LAN odpowiedzą ofierze.

Rozproszony DoS (DDoS)

DDoS = Distributed DoS

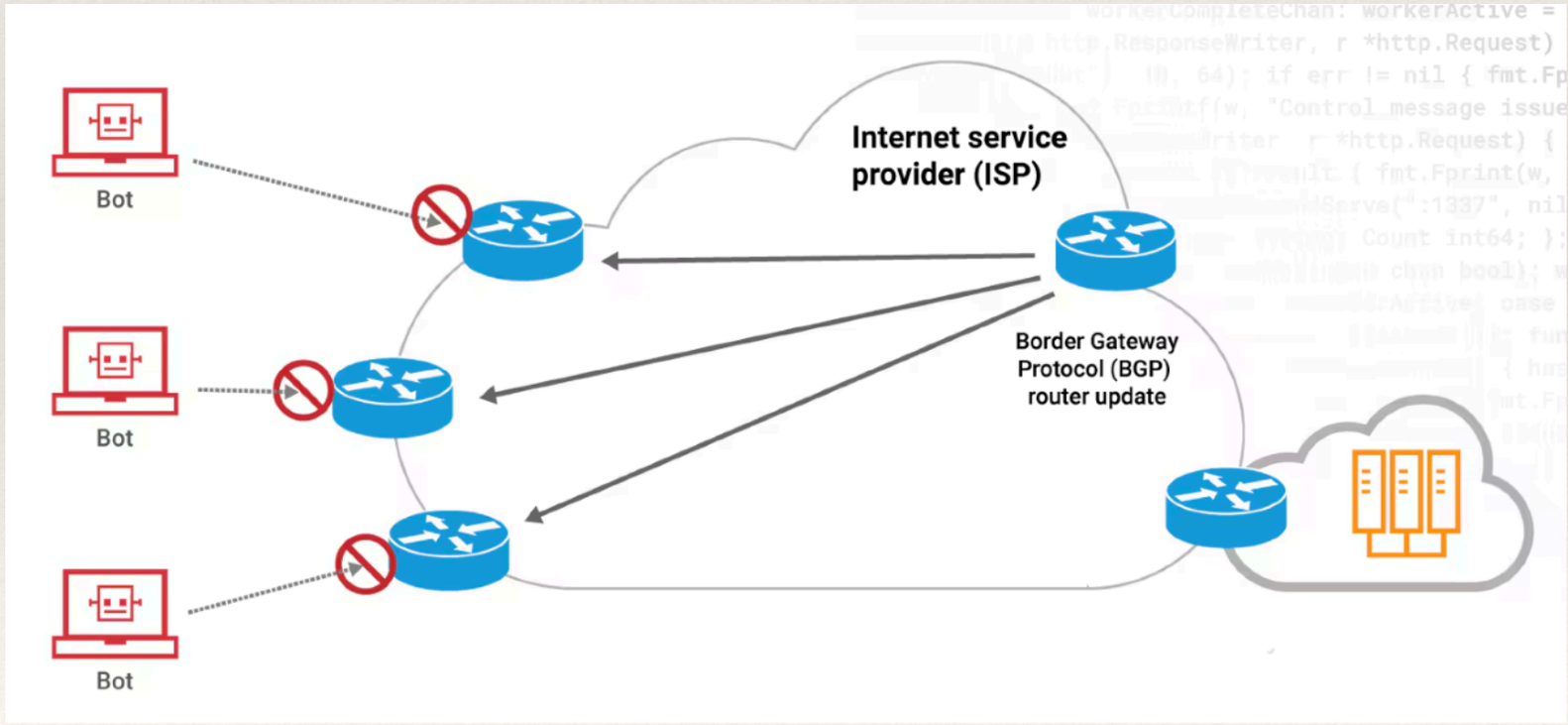
- Wykonywany z wielu komputerów pod kontrolą atakującego (botnet).
- Komputery uprzednio zainfekowane:
  - wirusem (wykorzystanie czynnika ludzkiego) lub
  - robakiem internetowym (wykorzystanie luk w bezpieczeństwie).



Obrazek ze strony
`https://en.wikipedia.org/wiki/Morris_worm`

Ataki wolumetryczne: obrona

- Anycast + sieci CDN: atakowany adres IP istnieje w wielu kopiach na świecie.
- BGP blackhole routing: zablokowanie całego ruchu do atakowanego adresu IP zanim w ogóle dotrze do naszej sieci.



Obrazek ze strony `https://www.akamai.com/glossary/what-is-blackhole-routing`

Lektura dodatkowa

- Kurose & Ross: rozdział 8.
- Tanenbaum: rozdział 8.
- Nftables HOWTO: `https://wiki.nftables.org/`

Zagadnienia

- Co to jest pamięć CAM i jak stosuje się ją w przełącznikach? Jak można ją przepełnić?
- Opisz atak typu ARP spoofing.
- Co oznacza termin IP spoofing? Na czym polega metoda weryfikacji tak zmodyfikowanych pakietów (ingress filtering)?
- Na czym polega atak RIP spoofing?
- Na czym polega zatruwanie pamięci podręcznej serwera DNS?
- Jak wygląda uwierzytelnianie serwera SSH?
- Na czym polega uwierzytelnianie użytkownika przez SSH z wykorzystaniem kluczy RSA?
- Podaj przykłady tunelowania.
- Rozwiń skrót VPN. Do czego służy?
- Porównaj wady i zalety filtrów pakietów: prostych, stanowych i działających w warstwie aplikacji.
- Do czego służą moduły `input`, `output`, `forward` w filtrze Netfilter/nftables?
- W jakich łańcuchach zapory Linuksa wykonywany jest źródłowy a w jakich docelowy NAT?
- Przedstaw przykładowe ataki wykorzystujące brak sprawdzania poprawności wprowadzanych danych.
- Wyjaśnij pojęcia: robak internetowy, botnet.
- Na czym polega phishing?
- Co to jest skanowanie portów? Po co się je wykonuje?
- Co to są ataki DoS i DDoS?
- Na czym polega atak typu odbity (reflected) DoS?