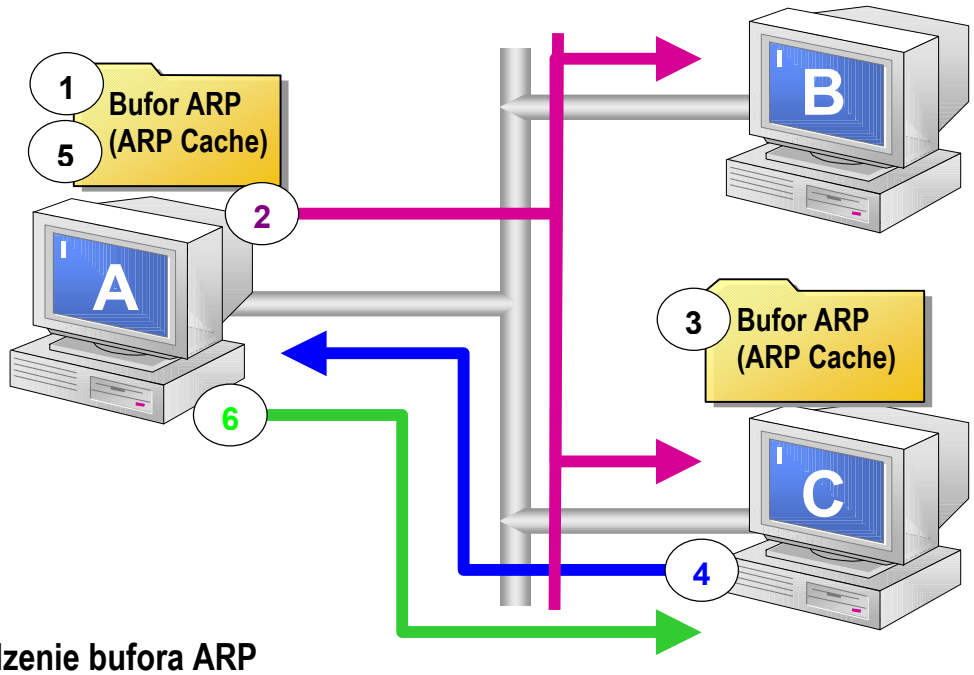
Address Resolution Protocol (ARP)



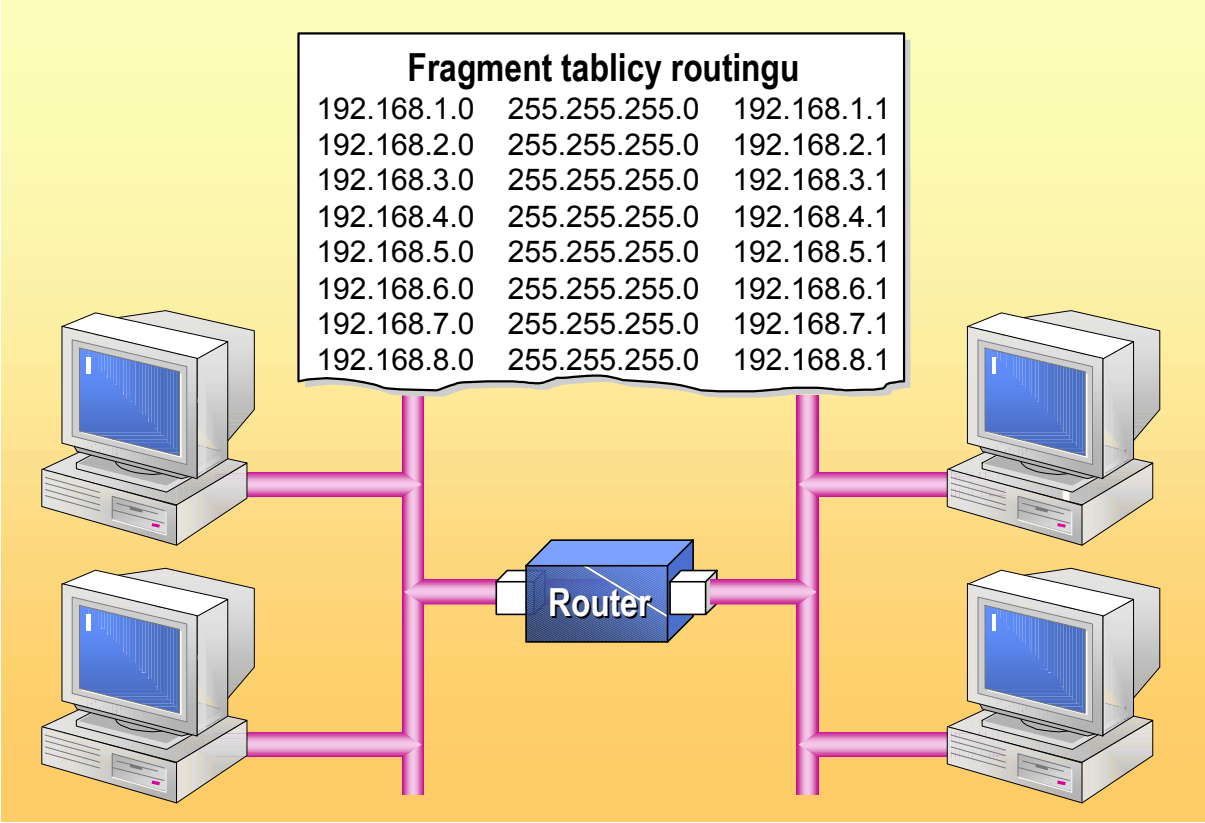
1. Sprawdzenie bufora ARP
2. Wysłanie pytania ARP (jaki jest adres sprzętowy komputera o adresie IP 192.168.1.5)
3. Dodanie pozycji ARP do bufora ARP odbiorcy
4. Wysłanie odpowiedzi ARP
5. Dodanie pozycji ARP do bufora ARP nadawcy
6. Wysłanie pakietu IP

Ochrona przed *spoofingiem* ARP

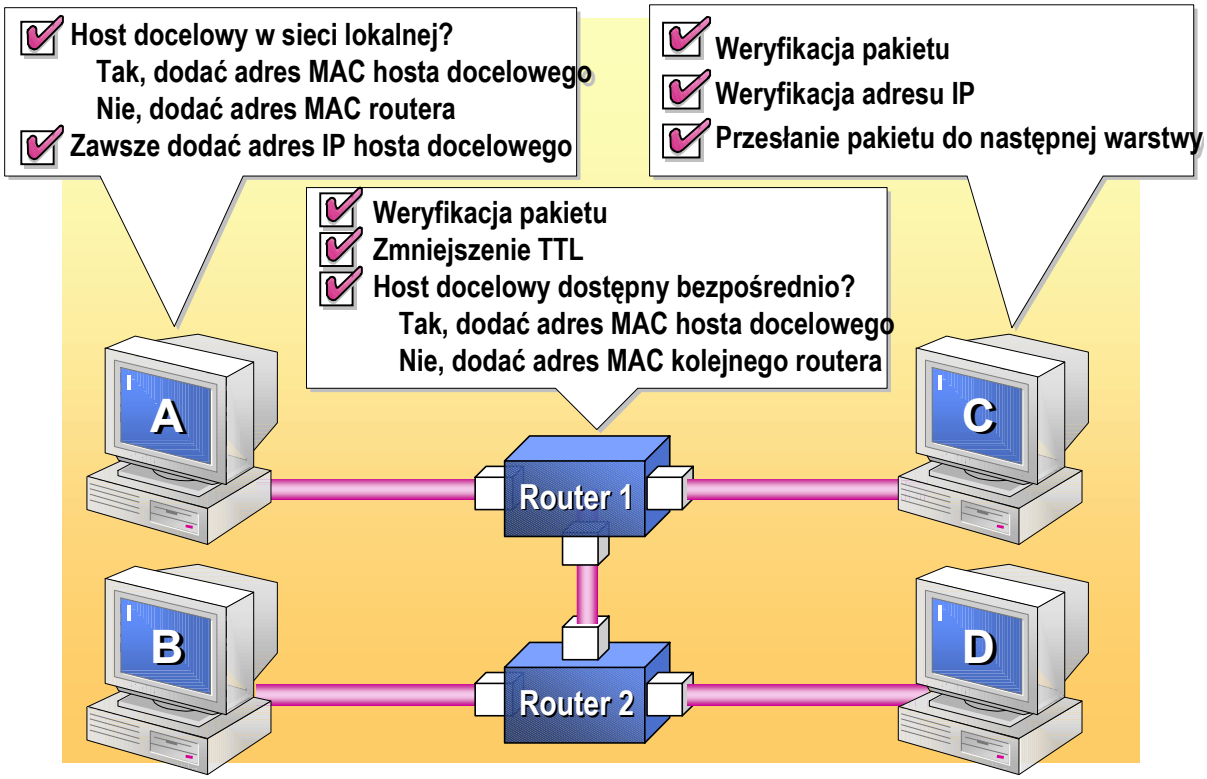
(*Address Resolution Protocol*)

- **Zaprzestanie używania ARP**
- **Bariery sprzętowe (*routery*)**
- **Wykrywanie spoofingu ARP:**
  - **Pasywna detekcja na poziomie hosta**
  - **Aktywna detekcja na poziomie hosta**
  - **Detekcja na poziomie serwera**
  - **Detekcja na poziomie sieci przez okresowe kontrole**
  - **Detekcja na poziomie sieci przez ciągłe monitorowanie**

Routing IP


Fragment tablicy routingu
192.168.1.0 255.255.255.0 192.168.1.1
192.168.2.0 255.255.255.0 192.168.2.1
192.168.3.0 255.255.255.0 192.168.3.1
192.168.4.0 255.255.255.0 192.168.4.1
192.168.5.0 255.255.255.0 192.168.5.1
192.168.6.0 255.255.255.0 192.168.6.1
192.168.7.0 255.255.255.0 192.168.7.1
192.168.8.0 255.255.255.0 192.168.8.1
Router

Transfer danych przez *routery*

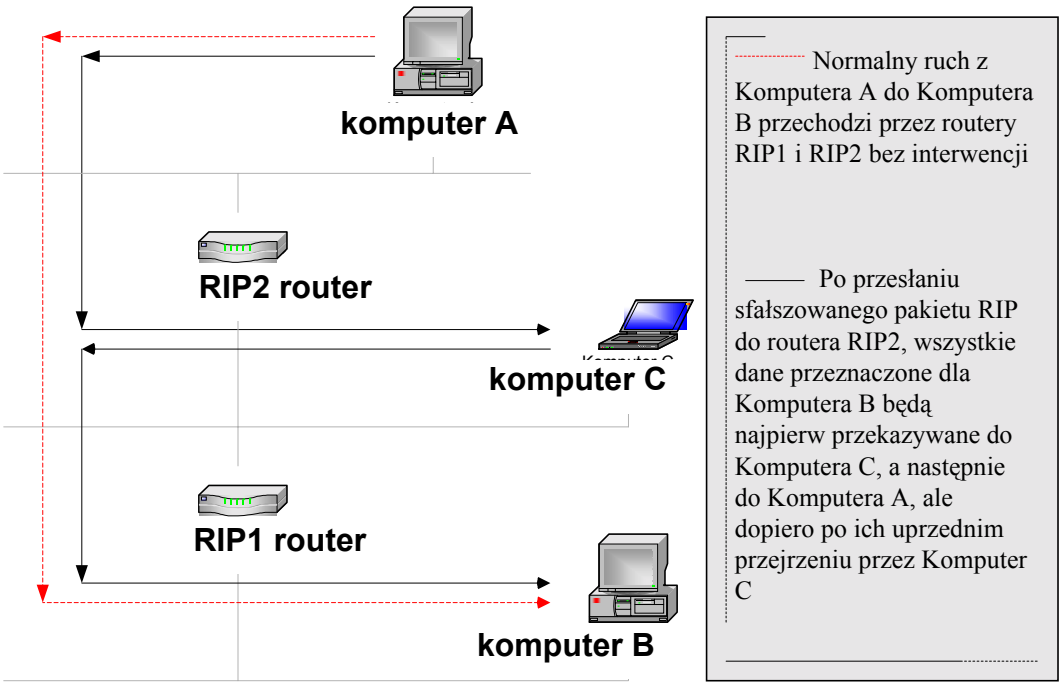


Spoofing routingu - wykorzystanie protokołu ICMP
- permanentne zapisy ARP dotyczące *routerów*

Spoofing routingu opartego na RIP

Spoofing routingu opartego na RIP

- **RIP wykorzystuje port 520 UDP, co oznacza, że nie wymagane jest połączenie, a pakiety przyjmowane są od każdego.**
- **RIP v1 nie posiada żadnego mechanizmu autoryzacji.**
- **W RIP v2 zastosowana jest prosta forma autoryzacji używająca nieszyfrowanych haseł.**



DNS - podstawowe zagadnienia

DNS jest systemem rozproszonej bazy danych udostępniającym usługę translacji nazw na adresy w sieci IP. Jest to system hierarchiczny. Dane umożliwiające translację nazw na adresy są przechowywane w plikach strefowych na serwerze DNS.

Domena prosta - zawiera rekordy, które umożliwiają translację nazw na adresy IP, czyli umożliwia odpowiadania na proste pytania DNS.

Domena odwrotna - zawiera rekordy, które umożliwiają translację adresów IP na nazwy, czyli umożliwia odpowiadania na odwrotne pytania DNS.

Serwer autorytatywny - serwer odpowiedzialny za utrzymywanie dokładnej i pewnej informacji o domenie

Serwer pierwotny - serwer autorytatywny stanowiący pierwotne źródło informacji o domenie

Serwer wtórny - serwer autorytatywny, który okresowo pobiera plik strefowy z serwera głównego.

Serwer *Caching-Only* - serwer nieautorytatywny, który otrzymuje odpowiedzi od innych serwerów, zapamiętuje je i jest wobec tego w stanie udzielać odpowiedzi klientom.

Pytania iteracyjne - jeżeli serwer nie potrafi odpowiedzieć, to zwraca adres serwera autorytatywnego, który powinien znać odpowiedź.

Pytania rekursywne - jeżeli serwer nie potrafi odpowiedzieć, to sam poszukuje pełnej odpowiedzi na zadane pytanie i zwraca odpowiedź klientowi.

Spoofing oparty na DNS

- **weryfikacja odpowiedzi serwera**
- **pytania iteracyjne zamiast rekursywnych**
- **test na autorytatywność**
- **nie używać DNS ?**
- **wykorzystanie ICMP (RFC 1788)**
- **DNS Security**

Wykrywanie

- **porównanie odpowiedzi z różnych serwerów**
- **porównywanie odpowiedzi na pytania proste i odwrotne**

Spoofing IP i TCP

1. **Napastnik zmienia adres IP własnego komputera, tak aby był zgodny z adresem prawdziwego komputera - klienta.**
2. **Następnie napastnik tworzy ścieżkę źródłową do serwera, podającą bezpośrednią trasę, którą pakiety IP powinny przechodzić do serwera i z powrotem do komputera napastnika, używając prawdziwego klienta jako ostatniego etapu na drodze do serwera.**
3. **Napastnik wysyła żądanie komputera - klienta do serwera, korzystając ze ścieżki źródłowej.**
4. **Serwer akceptuje żądanie, tak jakby pochodziło bezpośrednio od prawdziwego klienta, a następnie wysyła do niego odpowiedź.**
5. **Prawdziwy klient, korzystając ze ścieżki źródłowej, przesyła pakiet do napastnika.**

Spoofing na oślep (*blind spoofing***)**

Spoofing z podglądem (*not blind spoofing***)**

Metody zapobiegania s*poofingowi* IP i TCP

- **Ściany ogniowe**
- **Kerberos**
- **Szyfrowanie sesji IP (protokoły)**
- **Opuszczanie wszystkich sesji terminalowych wtedy, kiedy stają się one nieaktywne i uruchamianie ich tylko wtedy, gdy są potrzebne.**
- **Konfiguracja sieci, na poziomie routera, w taki sposób, aby nie przyjmowała pakietów z Internetu podających się za pakiety z sieci lokalnej.**
- **Szyfrowanie sesji na poziomie routera.**
- **Blokowanie przyjmowania TCP na poziomie zapory sieciowej, i korzystanie z protokołu IPX wewnątrz sieci.**
- **Uważne monitorowanie sieci.**
- **Badanie integralności w plikach i katalogach na podstawie zbioru reguł określonych przez administratora.**

Hijacking – przechwycenie sesji

SVR_SEQ – numer sekwencyjny następnego bajtu, który zostanie wysłany przez serwer

SVR_ACK – numer potwierdzenia następnego bajtu, który serwer spodziewa się otrzymać = numer ostatniego otrzymanego bajtu + 1

SVR_WIND – wielkość okna odbiorczego serwera

CLT_SEQ – numer sekwencyjny następnego bajtu, który zostanie wysłany przez klienta

CLT_ACK – numer potwierdzenia następnego bajtu, który klient spodziewa się otrzymać

CLT_WIND – wielkość okna odbiorczego klienta

Na początku połączenia:

$$SVR\_SEQ = CLT\_ACK \quad \text{oraz} \quad CLT\_SEQ = SVR\_ACK$$

W trakcie połączenia

$$SVR\_ACK <= CLT\_SEQ <= SVR\_ACK + SVR\_WIND$$

oraz

$$CLT\_ACK <= SVR\_SEQ <= CLT\_ACK + CLT\_WIND$$

Przesunięcia numerów sekwencyjnych:

$$CLT\_TO\_SVR\_OFFSET = SVR\_ACK - CLT\_SEQ$$

$$SRV\_TO\_CLT\_OFFSET = CLT\_ACK - SVR\_SEQ$$

Wartości wprowadzane przez atakującego:

$$CLT\_SEQ^{new} = CLT\_SEQ + CLT\_TO\_SVR\_OFFSET$$

$$CLT\_ACK^{new} = CLT\_ACK + SRV\_TO\_CLT\_OFFSET$$

Burza pakietów ACK

Hijacking <u>wczesna desynchronizacja</u>

1. **Atakujący nasłuchuje pakietów SYN/ACK zaadresowanych od serwera do klienta.**
2. **Po wykryciu takiego pakietu atakujący wysyła do serwera pakiet RST zamykając połączenia. Następnie generuje pakiet SYN ze sfałszowanym adresem źródła wskazującym na klienta oraz takim samym numerem portu.**
3. **Serwer zamknie połączenie od klienta, po czym po otrzymaniu pakietu SYN otworzy na tym samym porcie drugie połączenie wysyłając do klienta pakiet SYN/ACK.**
4. **Atakujący wykryje pakiet SYN/ACK od serwera i potwierdzi go wysyłając pakiet ACK. W tym momencie serwer przejdzie do stanu stabilnego.**

Hijacking desynchronizacja za pomocą pustych danych

1. **Atakujący przygląda się sesji bez ingerowania w nią**
2. **W wybranym momencie atakujący wysyła dużą ilość pustych danych do serwera. W przypadku sesji *telnet* może to być ATK_SVR_OFFSET liczba bajtów zawierających sekwencje poleceń IAC NOP IAC NOP . Każde dwa bajty IAC NOP zostaną zinterpretowane przez demona telnet i usunięte ze strumienia bez widocznych dla użytkownika efektów. Po przetworzeniu przesłanych przez atakującego danych serwer posiadać będzie numer potwierdzenia równy:**

   SVR_ACK = CLT_SEQ + ATK_SVR_OFFSET

3. **Atakujący postępuje w ten sam sposób z klientem.**

Hijacking wykrywanie

- **Wykrywanie stanu rozsynchronizowanego – porównanie numerów sekwencyjnych po obu stronach połączenia. Potrzebny jest jednak osobny mechanizm dokonujący tego porównania, który zabezpieczony byłby przed możliwością ingerencji przez atakującego**
- **Wykrywanie burzy pakietów ACK. Normalne połączenie *telnet* w sieci lokalnej generuje około 45% pakietów z flagą ACK w stosunku do liczby wszystkich pakietów. W momencie burzy ACK nieomal wszystkie pakiety *telnet* zawierają tą flagę.**
- **Wykrywanie większej liczby zagubionych pakietów oraz retransmisji dla konkretnego połączenia. Spowodowane jest to przeciążeniem sieci pakietami ACK oraz czasami nie przechwytywaniem przez atakującego wszystkich pakietów.**
- **Zrywane połączenia. Porywanie sesji TCP zawiera kilka słabych punktów, których powodzenie zależy od wielu czynników. Błąd w którejś fazie porwania może doprowadzić do zerwania połączenia.**

Programy atakujące

Juggernaut ***HUNT*** ***ETTERCAP***

Ataki typu *Denial Of Service*

1. **Zużycie limitowanych lub nie odnawialnych zasobów**
   - **Blokowanie interfejsu**
   - **Wykorzystanie zasobów serwera przeciwko niemu samemu**
   - **Zużycie przepustowości sieci**
   - **Zużycie innych zasobów**

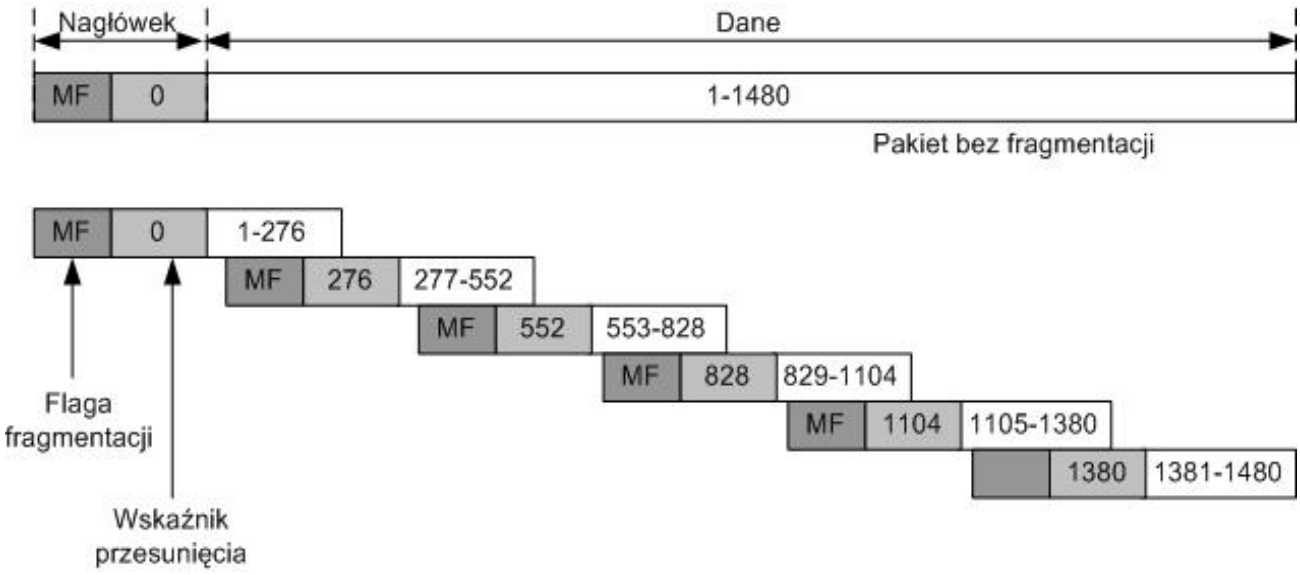
     ```
     cat   /dev/zero  >  /tmp/to_będzie_duży_plik
     main()  { for(;;)  fork(); }
     ```

2. **Zniszczenie lub zmiana informacji konfiguracyjnej**
3. **Fizyczne zniszczenie lub zmiana sprzętu**

Sieciowe ataki DOS:

- **ataki mające na celu zablokowanie konkretnej usługi**
- **ataki nastawione na zablokowanie całego systemu**

Fragmentacja pakietów



Każdy fragment niesie w sobie następujące informacje:

- **Identyfikator pakietu, który uległ fragmentacji (fragment ID)**
- **Wielkość przenoszonych danych**
- **Wskaźnik przesunięcia fragmentacji (*offset*) – umiejscowienie danych z tego fragmentu w pełnym datagramie**
- **Flagę MF (*More Fragments*) określającą czy dany fragment jest ostatnim, czy następują po nim kolejne**

Pakiet *ICMP echo request* o wielkości 4028 bajtów

92.168.1.2 > 192.168.2.43: icmp: echo request (frag 33465:1480@0+)

92.168.1.2 > 192.168.2.43: (frag 33465:1480@1480+)

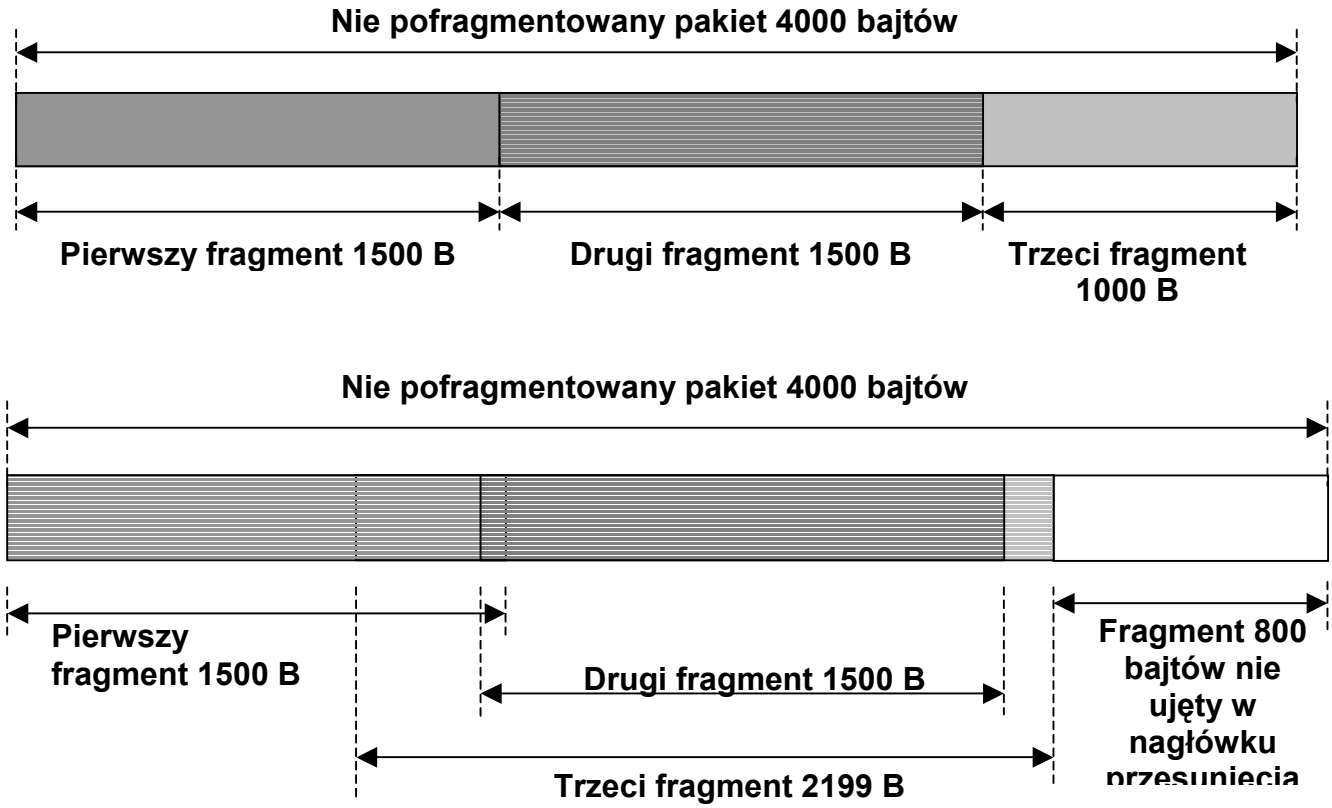
92.168.1.2 > 192.168.2.43: (frag 33465:1048@2960)

Ataki DOS

Ping of Death

wysyłany sfragmentowany datagram *ICMP Echo request* o łącznym rozmiarze przekraczającym 65535 bajtów

Teardrop



Nakładanie fragmentów (*Fragment Overlapping*)

Próba nadpisania części nagłówka TCP z pierwszego fragmentu. Nagłówek ten może zawierać dane, które są zgodne z polityką bezpieczeństwa zaimplementowaną na zaporze przez co nie jest przez nią odrzucany. Drugi fragment poprzez wykorzystanie wskaźnika przesunięcia fragmentacji stara się nadpisać część nagłówka z pierwszego datagramu zmieniając profil całego połączenia.

Jolt2

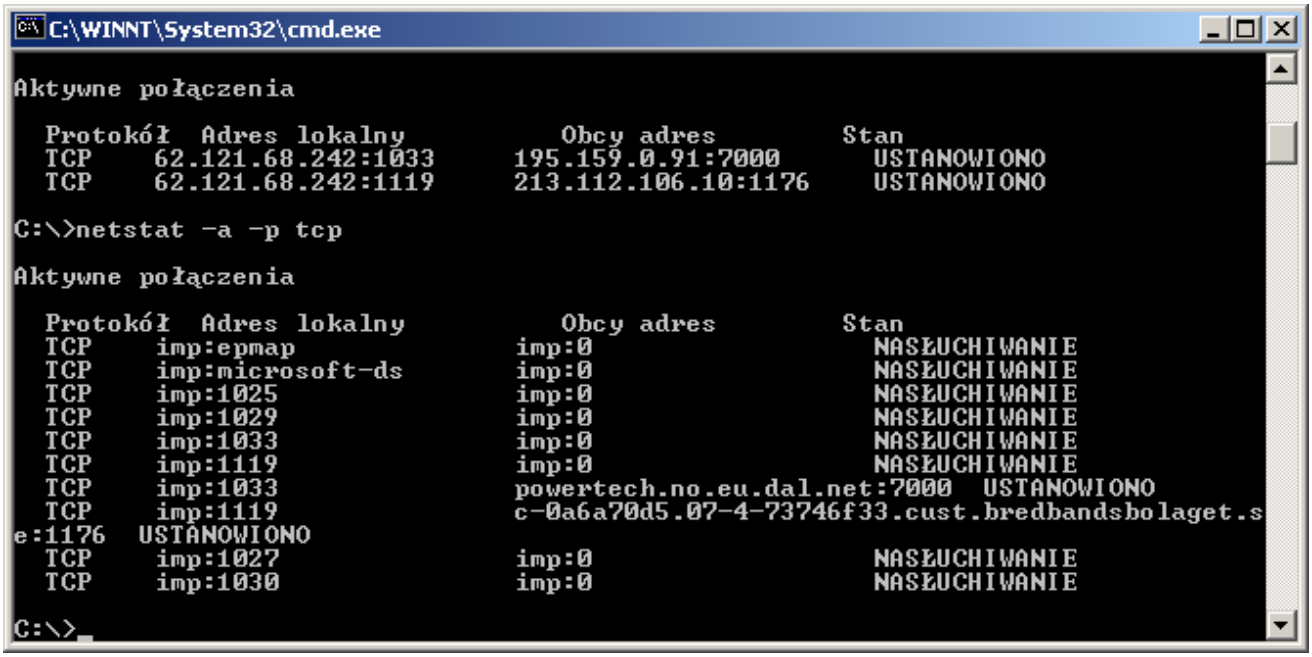
W strumieniu wysyłanych datagramów każdy posiada:

- **wskaźnik przesunięcia fragmentacji ustawiony na 65520**
- **wyłączoną flagę MF (*More Fragments*) – oznacza to, że jest to ostatni fragment**
- **niepoprawną długość pakietu podaną w nagłówku IP – 68 bajtów, kiedy w rzeczywistości jest ich 29 bajtów**

Skutki nie są przewidywalne

Zalew UDP (*UDP Flooding*)

Zalew pakietów SYN (*SYN Flooding*)



LAND – odmiana *SYN Flooding*

Adres nadawcy jak i źródła ustawiany jest na adres atakowanego hosta. Tworzy to nieskończoną pętle, w którą wpada zaatakowany host próbujący sam sobie odpowiadać na otrzymane pakiety

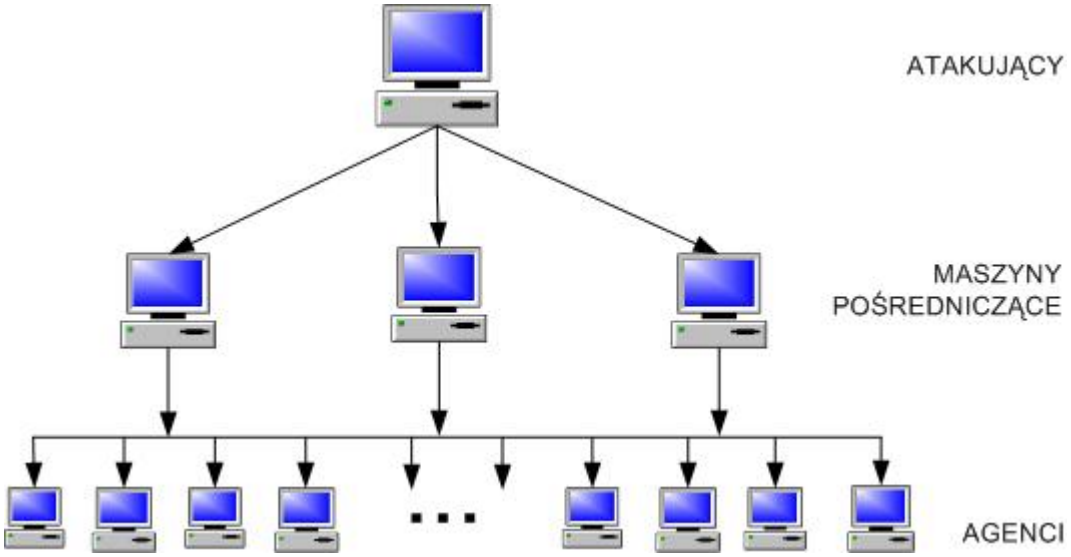
SMURF

„Adres nadawcy w komunikacie z żądaniem echa będzie adresem odbiorcy w odpowiedzi. Aby zbudować komunikat z odpowiedzią należy zamienić miejscami adres nadawcy i odbiorcy, zmienić typ komunikatu na „odpowiedź i obliczyć na nowo sumę kontrolną”".

Wysyłając pakiet ICMP żądania echa ze sfałszowanym adresem źródła na adres rozgłoszeniowy (*broadcast address*) sieci, można spowodować:

1. **duży ruch, często kończący się sztormem kolizyjnym i chwilowym spadkiem wydajności sieci**
2. **komputer ofiary, który został mimowolnym nadawcą żądania echa zalany zostanie pakietami potwierdzenia, co może doprowadzić do jego zablokowania**

Rozproszony DoS



Aby atak był skuteczny potrzebnych jest zwykle od kilkuset do kilku tysięcy komputerów z zainstalowanym oprogramowaniem agentów. Faza instalacji przebiega w kilku etapach:

1. **skanowanie dużej liczby komputerów pod kątem posiadania znanej luki**
2. **przejęcie kontroli nad wrażliwymi hostami**
3. **zainstalowanie agenta**
4. **użycie zdobytego komputera do dalszego skanowania**

Metody wykrywania i obrony przed atakami DoS

Extended IP access list 101
permit icmp any any echo (2 matches)
permit icmp any any echo-reply (21374 matches)
permit tcp any any established (150 matches)
permit tcp any any (15 matches)
permit ip any any (45 matches)

no access-list 101
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply log-input
access-list 101 permit tcp any any established
access-list 101 permit tcp any any
access-list 101 permit ip any any

Zapobieganie atakom DoS obejmuje między innymi:

- **Skonfigurowanie list dostępu na routerach i zaporach ogniowych**
- **Używanie i udostępnianie jedynie tych usług, które są niezbędnie potrzebne**
- **Ustalenie systemu ograniczeń na zasoby dyskowe, wykorzystanie procesora i przepustowość sieci**
- **Wprowadzenie systemu monitorowania dostępności i wykorzystania zasobów.**
- **Ustanowienie odpowiedniej polityki zarządzania hasłami, zwłaszcza kont użytkowników uprzywilejowanych**
- **Takie skonstruowanie topologii sieci by serwery nie przeszkadzały sobie nawzajem**
- **Aplikowanie łat na systemy oraz serwisy jak tylko luka zostanie odkryta**
- **Regularne czytanie list dyskusyjnych poświęconych bezpieczeństwu, zwłaszcza aplikacji zainstalowanych w firmie**
- **Używanie systemów IDS w celu możliwie wczesnego wykrycia podejrzanych działań w sieci**
- **Ustalenie systemu backupów**
- **Przygotowanie narzędzi i procedur pozwalających na szybkie ustalenie źródła ataku i opracowanie działań prowadzących do szybkiego jego odcięcia. Blokada powinna zostać założona możliwie blisko źródła, co w przypadku ataków DDoS może być niewykonalne.**

Literatura:

1) V. Ahuja. *Network & Internet Security*. Academic Press, Inc, 1996. (*tłum. MIKOM 1997*).
2) D. Atkins i inni. *Internet Security. Professional Reference.* New Riders Publishing, 1997 (*tłum. LT&P 1997*).
3) S. Garfinkel, G. Spafford. *Practical Unix and Internet Security,* O'Reilly&Associates Inc. 1996. (*tłum. RM 1997*).
4) J. Hruska, Computer viruses and antivirus warfare, Prentice Hall Int. 1992 (*tłum*. WKŁ 1995).
5) L. Klander. *Hacker Proof.* Jamsa Press, 1997. (*tłum. MIKOM 1998*).
6) S. Waldbusser. *Remote Network Monitoring Management Information Base.* RFC 1271, Nov 1991.