

**ZAGROZENIA W
SIECI, ATAKI
I ZAPOBIEGANIE**

Uwagi ogólne

- Przez incydent rozumiemy grupę ataków, która może być wyróżniona spośród innych ataków ze względu na charakterystyczne dla tych ataków grupy atakujących, rodzaje ataków, cele ataków oraz czas.
- Proces obsługi incydentu ma prowadzić do ustalenia intruzów, sposobu ataku i obiektów ataku.
- Intruzów możemy podzielić ze względu na uzyskiwane korzyści z ataku zgodnie z klasyfikacją zaproponowaną przez CERT (Computer Emergency Response Team):
 - **Hackers (hakerzy)** – atakujący, którzy dokonują naruszenia bezpieczeństwa dla samego faktu i potwierdzenia swoich umiejętności technicznych.
 - **Spies (szpiegowie)** – atakujący w celu osiągnięcia informacji, która można wykorzystać w sprawach politycznych.
 - **Terrorists (terrorysty)** – atakujący, którzy próbują wywołać zagrożenie w celu osiągnięcia korzyści politycznych.
 - **Corporate raiders (szpiegowie przemysłowi)** – atakujący, częstokroć pracownicy, prowadzący swoją nielegalną działalność w stosunku do konkurencji, w celu osiągnięcia korzyści finansowych.
 - **Professional criminals (przestępcy)** – atakujący komputery w celu uzyskania osobistych korzyści finansowych.

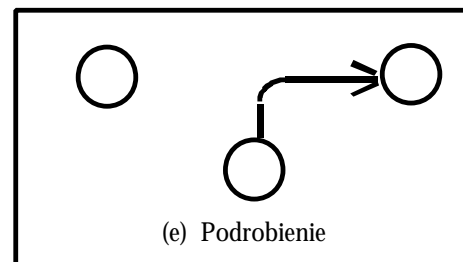
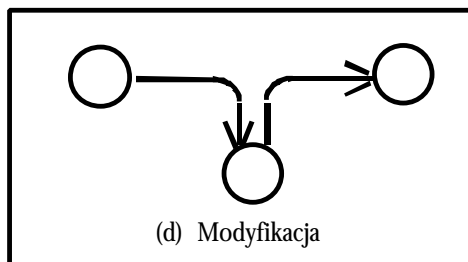
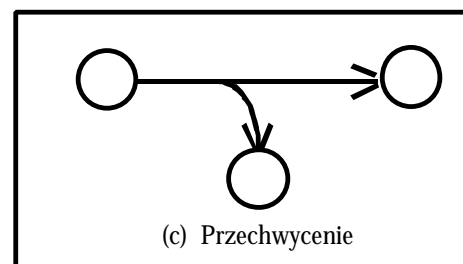
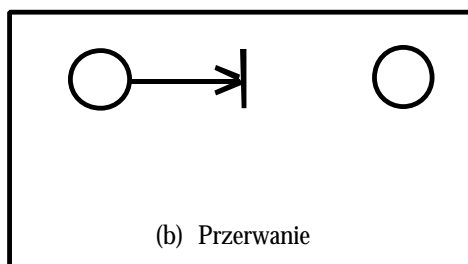
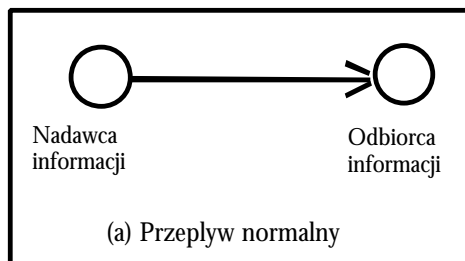
- **Vandals (wandale)** – atakujący w celu dokonania zniszczenia.
- **Voyeur (turysci)** – atakujący dla samego doznania odczucia strachu związanego z faktem uzyskiwania istotnych informacji.
- Bezpieczeństwo sieci koncentruje się na zapewnieniu:
 - **Dostępności** – elementy sieci, dane i usługi są dostępne zawsze, kiedy są potrzebne.
 - **Kontroli dostępu** – usługi i dane dostępne są tylko dla uprawnionych. Poziomą dostępność może być zróżnicowana dla różnych użytkowników.
 - **Integralności** – elementy sieci oraz informacje nie są zniszczone, uszkodzone lub skradzione poprzez zewnętrzną ingerencję lub wewnętrzną niekompetencję.
 - **Poufności** – informacje przesyłane i składowane w sieci nie mogą być czytane przez nieuprawnione podmioty.

Klasyfikacja zagrożeń bezpieczeństwa

- Ze względu na miejsce skąd przeprowadzany jest atak możemy wyróżnić:
 - **Zewnętrzne** – atak przeprowadzany jest z systemu znajdującego się poza atakowaną siecią, np. atak szpiega przemysłowego jego macierzystego systemu innego niż atakowany.
 - **Wewnętrzne** – atak przeprowadzany jest z systemu w atakowanej sieci, np. atak przeprowadza pracownik w celu sprawdzenia swoich umiejętności.

- Ze względu na zamiar ataku możemy wyróżnić:
 - **Zamierzony** – atakujący zdaje sobie sprawę z tego, co robi, np. atak szpiega w celu uzyskania konkretnych informacji.
 - **Niezamierzony** – atakujący przypadkowo dokonuje ataku, np. pracownik przypadkowo obchodzi system autoryzacji.
- Ze względu na efekt ataku możemy wyróżnić:
 - **Aktywny** – w wyniku ataku system komputerowy traci integralność np. atak wandalą, który usuwa system operacyjny serwera.
 - **Pasywny** – atak ten polega na wejściu do systemu bez dokonywania żadnych zmian, np. szpieg przemysłowy kradnie ważne dane.

Ataki na bezpieczeństwo:



Klasyfikacja ataków na systemy informatyczne

- Klasyfikacja empiryczna wg CERT Polska:

- **Skanywanie** – przegląd adresów sieciowych oraz portów sieciowych.
 - **Atak na system operacyjny** – ogół ataków na systemy operacyjne w celu przejęcia kontroli na nimi.
 - Spamming – rozsyłanie niechcianej poczty elektronicznej.
 - Atak na serwer (WWW, DNS, inne).
 - Odmowa serwisu – unieruchomienie usługi.
 - Atak na system poczty.
 - Social engineering – psychologia społeczna ukierunkowana na zdobywanie informacji o atakowanym systemie od osób pracujących z tym systemem.
 - Nielegalne oprogramowanie – dystrybucja nielegalnego oprogramowania.
 - Rozpowszechnianie nielegalnych i obrazliwych treści.
 - Inne – pozostałe, niemieszczące się w poprzednich kategoriach.
 - Wspólna klasyfikacja empiryczna wg JANET CERT Anglia i CERT-NL Holandia:
 - Abusive mail – poczta elektroniczna zawierająca obrazliwe treści.
 - Denial of Service – unieruchomienie usługi.
 - LAN Sniffing – podsłuchiwanie transmisji w sieci.
 - Other – inne niemieszczące się w kategoriach.
-

- Probe – sondowanie atakowanego systemu np. poprzez skanowanie.
- Root compromise – uzyskanie uprawnień super użytkownika.
- Spam – rozsyłanie niechcianej poczty elektronicznej.
- Trojan – konie trojańskie.
- Unauthorized use – nieuprawnione użycie.
- Virus – wirusy komputerowe.
- Warez – rozpowszechnianie nielegalnego oprogramowania.

Skanowanie (weszenie)

- Skanowanie adresów sieciowych – przeglądanie kolejnych adresów IP z założonego zakresu w celu wytypowania ofiary lub też poznania topologii atakowanej sieci. Z reguły można poznać topologie części sieci niechronionej. Wiedza zdobyta w ten sposób może również pomóc określeniu typu urządzenia systemu, z jakiego jest próbkowany.

Najprostsze skanowanie (i najłatwiej blokowane) można przeprowadzić za pomocą protokołu ICMP (Internet Message Protocol – internetowy protokół komunikatów kontrolnych) typu 8, czyli wysyłany jest popularny Ping (ICMP echo_request) a istniejący host powinien odpowiedzieć ICMP typu 0 (echo_reply), jeśli istnieje lub nie są zablokowane odpowiedzi, w hostie lub pośredniczącym urządzeniu.

- Skanowanie portów sieciowych – przeglądaniem kolejnych portów sieciowych systemu komputerowego w celu uzyskania informacji na temat, jakie porty są otwarte i jakie usługi są realizowane przez dany system komputerowy. Jest to pomocne przy określeniu:
 - usług TCP i UDP, działających na wybranym na cel komputerze,
 - typu systemu operacyjnego, z którego korzysta wybrany na cel komputer,
 - konkretnych aplikacji lub wersji świadczonych usług.

Mozna skanować porty TCP (Transmission Control Protocol) i UDP (User Datagram Protocol). Wiedza zdobyta w ten sposób może również pomóc określeniu typu urządzeń systemu zabezpieczeń. Skanowanie można zrealizować w prosty sposób poprzez badanie rezultatów funkcji connect(). Funkcja ta służy do inicjowania połączenia sieciowego między gniazdem (socket) programu klienta i odległym gniazdem serwera. Metoda ta jest najprostsza i jest łatwa do wykrycia przez skanowany system.

Tabela: Metody wykrywania i skanowania komputerów w sieci (flagi ACK- potwierdzenie, SYN- synchronizacja, FIN- zakończenie, RST- reset).

Nazwa metody	Opis	Uzyskane informacje
UDP Echo Port	Próba połączenia z portem 7 UDP Echo. Ustanowienie połączenia objawia się wysłaniem do klienta własnego	Otrzymanie odpowiedzi oznacza, że host istnieje i jest to prawdopodobnie *NIX. Brak odpowiedzi nic nie mówi,

	datagramu.	poniewaz port ten jest rzadko otwarty.
TCP Echo Port	Próba polaczenia z portem 7 TCP Echo. Ustanowienie polaczenia objawia sie wyslaniem do klienta tekstu Hello.	Otrzymanie odpowiedzi oznacza, ze host istnieje i jest to prawdopodobnie *NIX. Brak odpowiedzi nic nie mówi, poniewaz port ten jest rzadko otwarty.
UDP Closed Ports	Wykorzystywana jest odpowiedz zamknietego, nienasluchujacego portu UDP. Port taki powinien dac odpowiedz ICMP_PORT_UNREACH.	Otrzymanie odpowiedzi pozwala stwierdzic istnienie hosta. Brak odpowiedzi port jest otwarty, host nie istnieje, zagubiono datagram UDP.
TCP SYN – skanowanie pólotwarte	Wysylany jest pakiet TCP z ustawiona flaga SYN. Port docelowy odpowie, jesli jest otwarty: pakietem z ustawionymi flagami SYN i ACK, natomiast zamkniety flagami RST i ACK. Komputer skanujacy wysyla RST/ACK, dzieki czemu polaczenie nie zostanie nawiązane	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta lub filtrowanie ruchu.
TCP ACK –	Wyslanie pakietu TCP z	Otrzymanie odpowiedzi mówi o

do kontroli działania zapory ogniowej	ustawiona flaga ACK w odpowiedzi niezależnie od stanu portu otrzymujemy pakiet z flaga RST.	istnieniu hosta, brak odpowiedzi sugeruje brak hosta lub filtrowanie ruchu.
TCP SYN ACK	Wysłanie pakietu z ustawionymi flagami SYN i ACK niezależnie od stanu portu otrzymujemy pakiet z flaga RST.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta lub filtrowanie ruchu może również sugerować że host korzysta z kodu sieciowego BSD.
TCP FIN	Wysłanie pakietu z ustawioną flagą FIN na zamknięty port zwróci pakiet oflagowany przez RST i ACK, Otwarty port nie odpowie.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, otwarcie portu lub filtrowanie ruchu.
TCP NULL	Testuje odpowiedź na pakiet, z wyłączonymi wszystkimi flagami nagłówka, wysłany na zamknięty port. W odpowiedzi otrzymujemy pakiet z ustawionymi flagami RST i ACK	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, otwarcie portu lub filtrowanie ruchu.
TCP XMAS Tree	Testuje odpowiedź na pakiet, z wyłączonymi wszystkimi bitami nagłówka (SYN, ACK, FIN, RST, URG, PSH), wysłany na zamknięty port. W odpowiedzi otrzymujemy pakiet z	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, otwarcie portu, że systemem hosta jest Windows lub filtrowanie ruchu.

	ustawionymi flagami RST i ACK	
ICMP Echo Request Type 8	Ping. Wysyłamy datagram ICMP echo_request a otrzymujemy od aktywnego hosta odpowiedź ICMP Typ 0 echo_replay	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta lub filtrowanie ruchu.
ICMP Broadcast	Wysłanie datagram ICMP Typ 8 echo_request do sieci lub na adres broadcast, otrzymujemy echo_replay od aktywnych hostów.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, ze systemem, hosta jest Windows lub filtrowanie ruchu.
ICMP Router Solicitation Type 10	Wysłanie ICMP Typ 10 do hosta routera lub hosta będącego routerem powoduje jego odpowiedź.	Otrzymanie odpowiedzi mówi o istnieniu routera, brak odpowiedzi sugeruje nieistnienie routera, wyłączona jest implementacja ICMP Router Solicitation lub filtrowanie ruchu.
ICMP Timestamp Request Type 13	Wysłanie zadania Timestamp (pytanie o aktualny czas) do hosta, który zwraca nam aktualny (systemowy) czas.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, ze systemem, hosta jest Windows lub filtrowanie ruchu.
ICMP Information Request Type	Odpytywanie hosta, aby wykryć jego adres sieciowy.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, ze

15		systemem, hosta jest Windows lub filtrowanie ruchu.
ICMP Address Mask Request Type 17	Zadanie adresu maski podsieci hosta.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta, ze systemem, hosta jest *NIX lub filtrowanie ruchu.
Timeout Packet Fragmentation	Wyslanie jednego pakietu z pofragmentowanym offsetem, po przekroczeniu czasu oczekiwania host zwróci komunikat zbieranie fragmentów przekroczyło zadany czas.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie hosta..
Invalid IP Heder Length	Uzycie nieprawidłowej dlugosci nagłówka prowadzi do wygenerowania przez zdalny komputer pakietu ICMP Typ 12 z kodem bledu 0 i 2.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie, hosta.
Invalid IP Field Values	Wyslanie pakietu z bledna wartoscia w polu IP PROTO np. 0 co powoduje odeslanie bledu poprzez ICMP Typ 3 kod 3.	Otrzymanie odpowiedzi mówi o istnieniu hosta, brak odpowiedzi sugeruje nieistnienie, hosta.

Przykłady skanowania portów

- TCP SYN – skanowanie półotwarte
-

- Przygotowujemy pakiet TCO z ustawioną flagą SYN. Wysłanie pakietu na otwarty port spowoduje zwrócenie przez komputer docelowy SYN/ACK. Jeśli nie zostanie zwrócony żaden pakiet, to można przyjąć, że host jest chroniony firewallem lub port jest filtrowany.

```
jp@dev:~# hping XXX.XXX.XXX.XXX -c 1 -S -p 23
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.XXX (eth0 XXX.XXX.XXX.XXX): S set, 40
headers + 0 data bytes
50 bytes from 192.168.1.1: flags=SA seq=0 ttl=64 id=1252
win=32696 rtt=0.9 ms
```

- Otrzymany pakiet zawiera ustawiony SA (SYN/ACK). Jeśli wysłamy ten sam pakiet na port zamknięty, to w odpowiedzi otrzymamy RA (RST/ACK). Host jest jednak aktywny:

```
jp@dev:~# hping XXX.XXX.XXX.XXX -c 1 -S -p 2
eth0 default routing interface selected (according to /proc)
HPING atlanta (eth0 XXX.XXX.XXX.XXX): S set, 40
headers + 0 data bytes
50 bytes from XXX.XXX.XXX.XXX: flags=RA seq=0 ttl=255 id=1254
win=0 rtt=0.7 ms
```

□ **TCP ACK** – kontrola zapory ogniowej.

- Ustawienie bitu ACK i wysłanie na otwarty/zamknięty port spowoduje otrzymanie zwrotnego pakietu z ustawionym bitem RST. Host może oczywiście zignorować pakiet zgodnie z regulami zdefiniowanymi na routerze lub firewallu, ale wtedy wiadomo, że port jest filtrowany albo wyłączony.

- TCP ACK stosuje się zwykle w powiązaniu z innymi pakietami. Np. jeśli po próbie połączenia z hostem przy pomocy pełnego połączenia (TCP Echo port) system nie odpowie, to możliwe są dwie sytuacje: port jest filtrowany albo nieaktywny. Aby to sprawdzić należy wysłać TCP ACK. Otrzymany zwrotnie pakiet z ustawionym RST oznacza, że:
 - * TCP Echo port 7 jest filtrowany,
 - * Pakiety z ustawioną flagą SYN są blokowane na porcie 7
 - * TCP ACK pakiety docierają do hosta.
- Wniosek drugi może być jednak tylko pozornie prawdziwy. Aby potwierdzić jego słuszność można jednak przeprowadzić krótką analizę:
 - * pełne połączenie wymaga sekwencji pakietów SYN -> SYN/ACK -> ACK; jeśli TCP Echo port 7 jest filtrowany, to w odpowiedzi nie otrzymamy SYN/ACK.
 - * jeśli jednak mimo braku odpowiedzi wysyłamy TCP ACK i otrzymamy RST, to jest oczywiste, że host jest aktywny i pakiety TCP SYN są blokowane na porcie 7
- Wysłanie TCP ACK na port zamknięty (NON-LISTENING port):

```
jp@dev:~ # hping XXX.XXX.XXX.XXX -A -c 1 -p 2
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.XXX (eth0 XXX.XXX.XXX.XXX): A set, 40 headers + 0 data bytes
50 bytes from XXX.XXX.XXX.XXX: flags=R seq=0 ttl=255 id=1048 win=0 rtt=0.5 ms
```

- Wysłanie TCP ACK na port otwarty (LISTENING port):

```
jp@dev:~ # hping XXX.XXX.XXX.XXX -A -c 1 -p 23
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.XXX (eth0 XXX.XXX.XXX.XXX): A set, 40 headers + 0 data bytes
50 bytes from XXX.XXX.XXX.XXX: flags=R seq=0 ttl=255 id=1052 win=0 rtt=0.5 ms
```

□ TCP SYN/ACK (działa w systemie Linux/Windows)

- Pakiet TCP SYN/ACK pakiet wysyłany na dowolny port (otwarty/zamknięty). Jeśli zwrótnie otrzymamy TCP RST, to port żyje.

```
jp@dev:~ # hping XXX.XXX.XXX.XXX -c 1 -S -A -p 23
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.XXX (eth0 XXX.XXX.XXX.XXX): SA set, 40
headers + 0 data bytes
50 bytes from XXX.XXX.XXX.XXX: flags=R seq=0 ttl=128 id=31029
win=0 rtt=0.5 ms
```

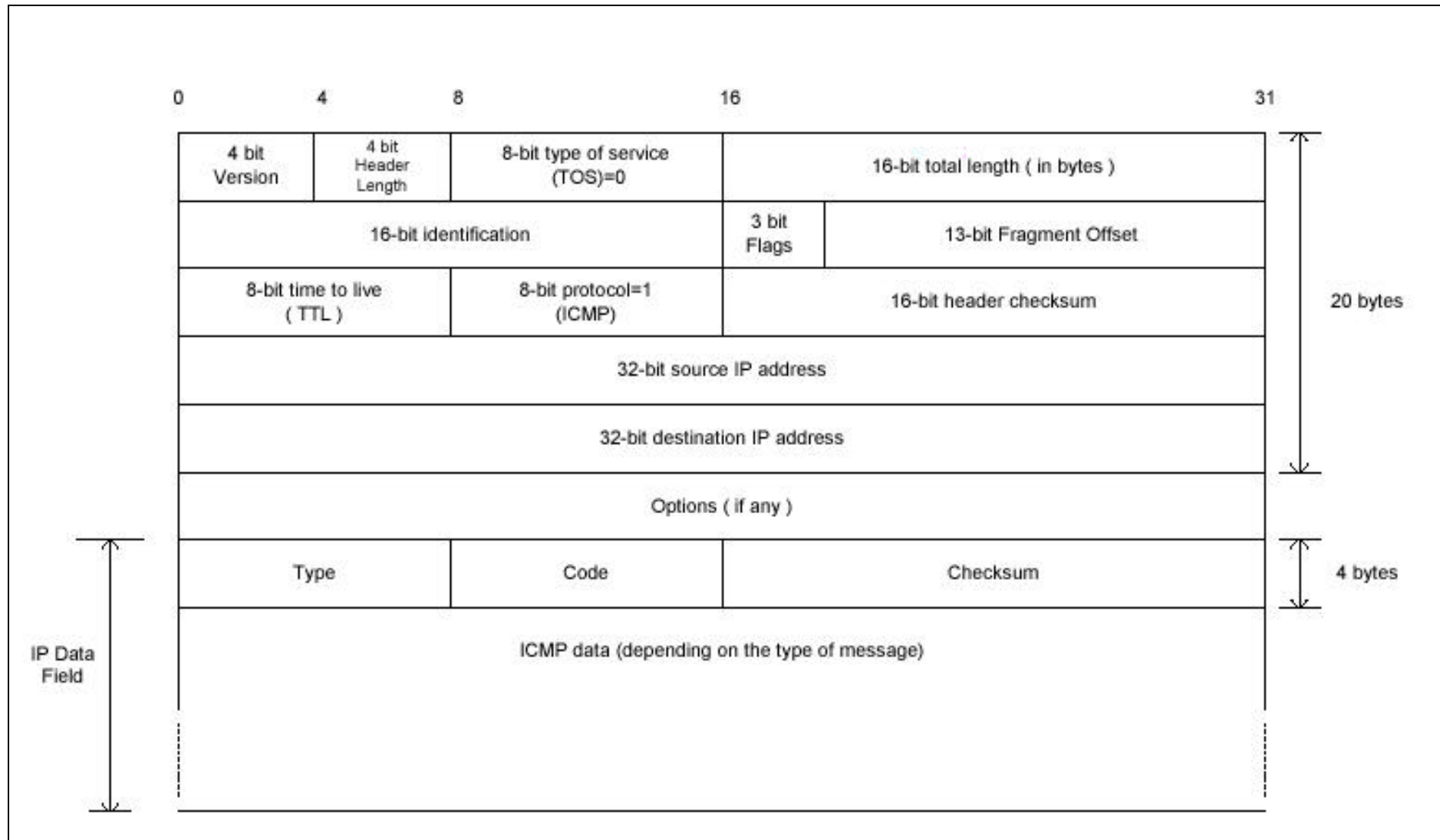
□ TCP FIN

- Jeśli pakiet TCP FIN zostanie wysłany na zamknięty port, to wtedy, w odpowiedzi otrzymamy TCP RST/ACK. Ten sam pakiet wysłany na port otwarty jest ignorowany. Umożliwia to skanowanie tylko portów zamkniętych.

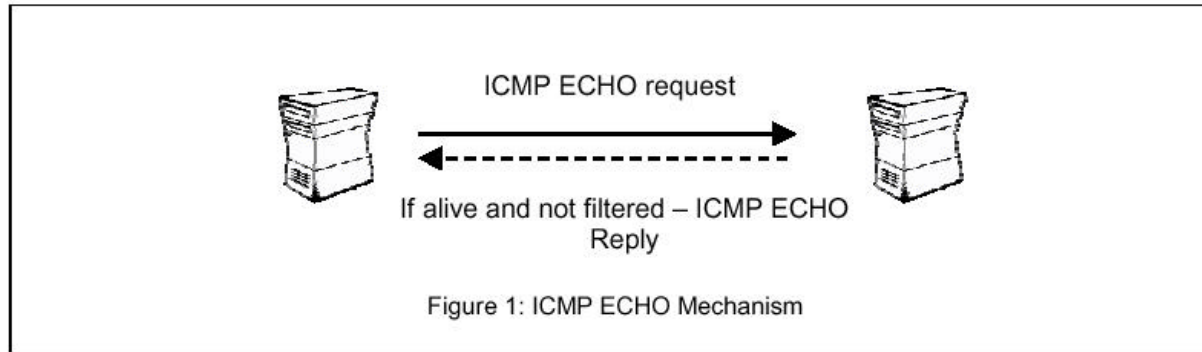
```
dethy@dev:~ # hping XXX.XXX.XXX.XXX -c 1 -F -p 2
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.XXX (eth0 XXX.XXX.XXX.XXX): F set, 40
headers + 0 data bytes
50 bytes from XXX.XXX.XXX.XXX: flags=RA seq=0 ttl=255
id=1260 win=0 rtt=0.5 ms
```

- Zignorowanie pakietu TCP FIN na porcie otwartym może oznaczać, że:
 - * pakiet FIN jest blokowany przez firewall/router/ACLs,
 - * ruch na tym porcie jest filtrowany
 - * odpytany port jest otwarty
 - * host jest wyłączony (nieaktywny).

□ **Format ICMP**



□ **ICMP Echo Request (Type 8)**



- Popularny PING, umożliwiajacy sprawdzenie istnienia polaczenia (czesto blokowany). Zawartosc tego pakietu jest nastepujaca:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9 0 1
Type		Code	Checksum
Identifier			Sequence Number
Data			

- Odpowiedz ma postac:

```
jp@dev:~ # hping -1 xxx.xxx.xxx.xxx -c 1 -C 8
eth0 default routing interface selected (according to /proc)
HPING xxx.xxx.xxx.xxx (eth0 xxx.xxx.xxx.xxx):
```

```
icmp mode set, 28 headers + 0
data bytes
50 bytes from XXX.XXX.XXX.XXX: icmp_seq=0 ttl=255 id=1273
rtt=0.4 ms
```

- Wiadomosci zarejestrowane przy pomocy programu **snort** (<http://www.clark.net/~roesch/security.html>.)::

```
01/26-13:16:25.746316 XXX.XXX.XXX.XXX -> YYY.YYY.YYY.YYY
ICMP TTL:64 TOS:0x0 ID:6059
ID:5721 Seq:1 ECHO
89 D7 8E 38 27 63 0B 00 08 09 0A 0B 0C 0D 0E 0F ...8'c.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

01/26-13:16:25.746638 XXX.XXX.XXX.XXX -> YYY.YYY.YYY.YYY
ICMP TTL:255 TOS:0x0 ID:6072
ID:5721 Seq:1 ECHO REPLY
89 D7 8E 38 27 63 0B 00 08 09 0A 0B 0C 0D 0E 0F ...8'c.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567
```

□ **ICMP Broadcast**

- Rozglaszanie jest metoda polegajaca na przeslaniu do wszystkich hostów w sieci pakietu **ICMP Echo Request (Type 8)**. Wszystkie hosty, które sa aktywne odpowiadaja swoim adresem karty sieciowej. Umozliwia to proste mapowanie IP na adres karty sieciowej.
- Jest to jednak ryzykowne z punktu widzenia bezpieczenstwa i dlatego zwykle uslugta ta jest blokowana (Windows NT/2000 nie odpowiadaja).

- Jesli wyslemy zwykly pakiet z zapytaniem na adres sieci, to otrzymamy

```
jp@dev:~ # hping -1 XXX.XXX.XXX.0 -c 2
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.0 (eth0 XXX.XXX.XXX.0): icmp mode set,
                                     28 headers + 0

data bytes
 28 bytes from XXX.XXX.XXX.3: icmp_seq=0 ttl=255 id=13013
                                     rtt=0.4 ms
 50 bytes from XXX.XXX.XXX.1: icmp_seq=0 ttl=255 id=426
                                     rtt=0.6 ms
 50 bytes from XXX.XXX.XXX.2: icmp_seq=0 ttl=255 id=15319
                                     rtt=0.8 ms

--- XXX.XXX.XXX.0 hping statistic ---
1 packets tramitted, 3 packets received, -100% packet loss
round-trip min/avg/max = 0.4/0.6/0.8 ms
```

- Jesli to samo wyslemy na adres rozgloszeniowy, to otrzymamy podobna odpowiedz:

```
jp@dev:~ # hping -1 XXX.XXX.XXX.255 -c 2
eth0 default routing interface selected (according to /proc)
HPING XXX.XXX.XXX.255 (eth0 XXX.XXX.XXX.255): icmp mode set,
                                     28 headers + 0

data bytes
 28 bytes from XXX.XXX.XXX.3: icmp_seq=0 ttl=255 id=13098
                                     rtt=0.4 ms
 50 bytes from XXX.XXX.XXX.1: icmp_seq=0 ttl=255 id=730
                                     rtt=0.7 ms
 50 bytes from XXX.XXX.XXX.2: icmp_seq=0 ttl=255 id=15327
                                     rtt=0.8 ms
```

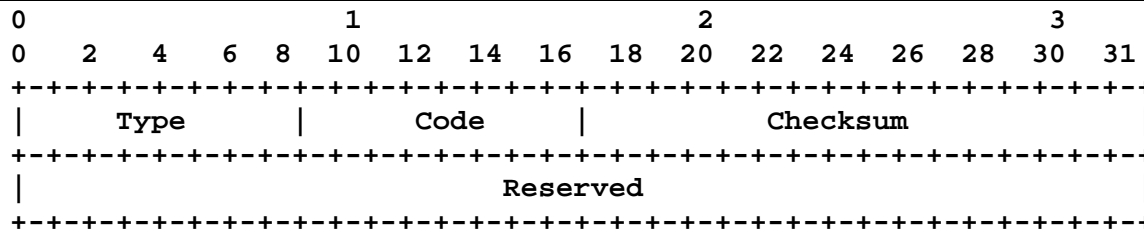
```

--- XXX.XXX.XXX.255 hping statistic ---
1 packets transmitted, 3 packets received, -100% packet loss
round-trip min/avg/max = 0.4/0.7/0.8 ms

```

□ ICMP Router Solicitation (Type 10)

- Zadanie to jest prosba o podanie adresów routerów w sieci i umieszczenia tej informacji w tabeli routingu. Tabela jest odswiezana okresowo, na podstawie otrzymanej informacji ICMP Type 9.
- Jesli wiec serwer odpowie ICMP Type 9 na zadanie ICMP Type 10, to mozemy byc pewni, ze jest to router.



- Program **hping** nie posiada zaimplementowanych ICMP Type 10,13,15,17. Pomoca sluzi inny program **icmpush** (<http://hispahack.ccc.de>)

```

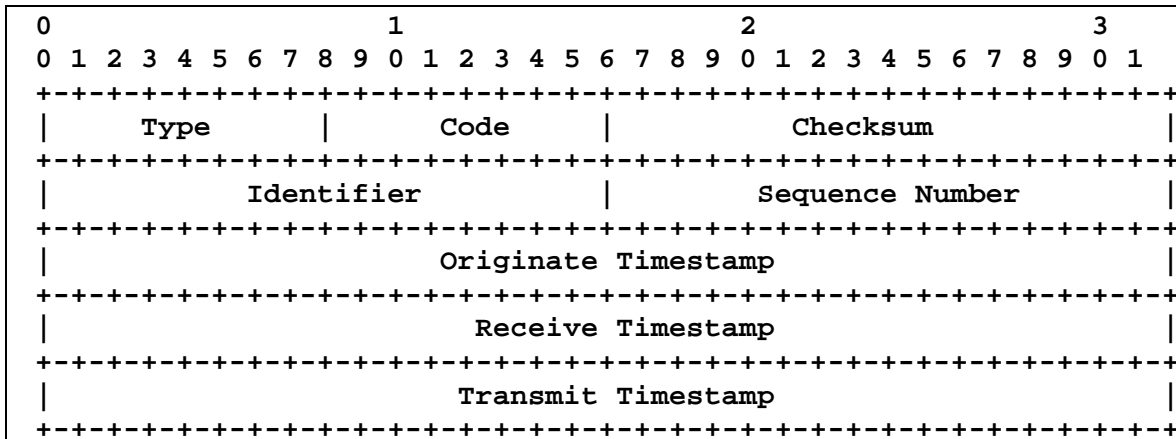
jp@dev:~ # ./icmpush -vv -rts XXX.XXX.XXX.XXX
-> Outgoing interface = XXX.XXX.XXX.XXX
-> ICMP total size = 20 bytes
-> Outgoing interface = XXX.XXX.XXX.XXX
-> MTU = 1500 bytes
-> Total packet size (ICMP + IP) = 40 bytes

```

```
ICMP Router Solicitation packet sent to XXX.XXX.XXX.XXX
                                     (XXX.XXX.XXX.XXX)
```

```
Receiving ICMP replies ...
XXX.XXX.XXX.XXX -> Router Advertisement (XXX.XXX.XXX.XXX)
./icmpush: Program finished OK
```

- Z odpowiedzi wynika, że znaleźliśmy router!
- **ICMP Timestamp Request (Type 13)**
- Na zadanie serwer odpowiada bieżącym czasem. Niektóre systemy operacyjne (np. Windows 95/98/NT) ignorują zadanie. Umożliwia to poznanie typu hosta (jeśli system nie odpowie, to może to oznaczać, że jest niektywny; problem ten można jednak rozstrzygnąć podając np. TCP ACK).



- Przykładowe zadanie ICMP timestamp wysłane na serwer Linux oraz odpowiedź ma postać (systemy Windows nie odpowiadają):

```
dethy@dev:~ # ./icmpush -vv -tstamp XXX.XXX.XXX.XXX
-> Outgoing interface = XXX.XXX.XXX.XXX
-> ICMP total size = 20 bytes
-> Outgoing interface = XXX.XXX.XXX.XXX
-> MTU = 1500 bytes
-> Total packet size (ICMP + IP) = 40 bytes
ICMP Timestamp Request packet sent to XXX.XXX.XXX.XXX
                                   (XXX.XXX.XXX.XXX)

Receiving ICMP replies ...
XXX.XXX.XXX.XXX -> Timestamp Reply transmited at 03:50:32
./icmpush: Program finished OK
```

- Wiadomości zarejestrowane przy pomocy programu **snort**:

```
01/26-13:51:29.342647 192.168.5.1 -> 192.168.5.5
ICMP TTL:254 TOS:0x0 ID:13170
TIMESTAMP REQUEST
88 16 D8 D9 02 8B 63 3D 00 00 00 00 00 00 00 00 .....c=.....

01/26-13:51:29.342885 192.168.5.5 -> 192.168.5.1
ICMP TTL:255 TOS:0x0 ID:6096
TIMESTAMP REPLY
88 16 D8 D9 02 8B 63 3D 02 88 50 18 02 88 50 18 .....c=..P...P.
2A DE 1C 00 A0 F9                               *.....
```

▣ **ATAKI NA NAGŁÓWEK IP (IP HEADER)**

- Utworzenie różnych anomalii w nagłówku IP znacznie zwiększa szanse na wykrycie filtrowanego hosta lub ukrytego za firewallem. Wiele systemów IDS odrzuca tego typu pakiety.
- Techniki ataku: **Timeout Packet Fragmentation, Invalid Header Length, Invalid Field Values**

▣ **Timeout Packet Fragmentation**

- Przygotowywany jest pakiet z częściowo wypełnionym offsetem i wysyłany do hosta. Następnie zamiast wysłania kolejnego fragmentu datagramu, klient zwleka, zmuszając serwer do wysłania sygnału **timeout** (*ICMP Type 11 Code 1 Time Exceeded Fragment Reassembly*).
- Przykład:

```
jp@dev:~ # hping -c 1 -x -y XXX.XXX.XXX.XXX
eth0 default routing interface selected (according to /proc)
HPING dev (eth0 XXX.XXX.XXX.XXX): NO FLAGS are set,
          40 headers + 0 data bytes

--- dev hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

hping sygnalizuje brak odpowiedzi (100% packet loss), ponieważ nie sprawdza ICMP datagram. Pakiet **tcpdump** daje jednak następującą informację:

```
20:41:09.309085 YYY.YYY.YYY.YYY > XXX.XXX.XXX.XXX:
          icmp: ip reassembly time
```

```
exceeded [tos 0xc0] (ttl 255, id 3375)
```

❑ Invalid Header Length

- Przygotowanie i wysłanie nagłówka IP z niewłaściwą długością wymusi odpowiedź postaci *ICMP Type 12 - Parameter Problem*. Kod błędu może mieć jedną z dwóch wartości:

```
0 - Pointer indicates the error  
2 - Bad Length
```

- Przykład – korzystając z programu **isic** (IP Stack Integrity Checker) można przygotować pakiet o nieprawidłowej długości 66 bajtów:

```
jp@dev:~ # ./isic -s YYY.YYY.YYY.YYY -d XXX.XXX.XXX.XXX  
                                           -p 1 -v 0 -F 0 -I 66 -D  
  
Compiled against Libnet 1.0.1b  
Installing Signal Handlers.  
Seeding with 5099  
No Maximum traffic limiter  
Bad IP Version = 0%   Odd IP Header Length = 100%  
                                           Frag'd Pcnt = 0%  
YYY.YYY.YYY.YYY -> XXX.XXX.XXX.XXX tos[137] id[0] ver[4] frag[0]  
  
Wrote 1 packets in 0.00s @ 5649.72 pkts/s
```

Pakiet tcpdump przechwyci następującą odpowiedź:

```
21:39:03.755839 XXX.XXX.XXX.XXX > YYY.YYY.YYY.YYY:  
                                           icmp: parameter problem -  
octet 20 [tos 0xd0] (ttl 255, id 21508)
```

Metoda ta umożliwia przejście wielu źle określonych form ACL i źle skonfigurowanych systemów filtrowania

□ **Invalid Field Values**

- Ataki tej grupy polegają na używaniu niewłaściwych wartości pól nagłówka IP. Np. IP PROTO jest polem 8-bitowym (max. wartość 256). Atak na IP PRO|TO polega na podaniu nieobsługiwanej wartości numeru protokołu (numer ten można znaleźć, ponieważ host zwróci w takim przypadku błąd *ICMP Type 3 Code 3 – Destination Unreachable Protocol Unreachable*).
- Jeśli serwer nie odpowie, to można się domyślać, że wybrany protokół jest aktywny na serwerze.
- Przykład – korzystając z programu **apsend** można przygotować odpowiedni pakiet o PROT = 0:

```
dethy@dev:~ # perl apsend -s YYY.YYY.YYY.YYY -d
                    XXX.XXX.XXX.XXX -b 8 -p 8
--protocol 0
Packet: 1 from YYY.YYY.YYY.YYY(port: 8) to XXX.XXX.XXX.XXX
                    (port: 8).
Protocol: 0  Type of Service(ToS): 16  ID: 0
```

Serwer powinien odpowiedzieć błędem. Z tcpdump mamy:

```
21:58:21.128201 YYY.YYY.YYY.YYY > XXX.XXX.XXX.XXX:
```

```
icmp: dev.synnergy.net
protocol 0 unreachable [tos 0xd0] (ttl 255, id 24133)
```

Przykłady identyfikacji typu systemu operacyjnego

Techniki identyfikowania typu systemu operacyjnego bazują na analizowaniu stosu protokołów TCP/IP. Stos jest różnie implementowany przez różnych producentów systemów (wynika to z różnic w interpretacji wskazówek zawartych w dokumentach RFC). Sprawdzając te różnice można z dużym prawdopodobieństwem określić system operacyjny.

□ Techniki aktywne

- Za Fyodorem (twórca sniffera nmap) można wyróżnić następujące rodzaje sond:
 - (a) Sonda FIN - wysyłamy pakiet FIN (lub jakikolwiek inny bez ustawionej flagi ACK lub SYN) na otwarty port i czekamy na odpowiedź. Prawidłowe zachowanie (wg. RFC 793) to brak odpowiedzi, lecz mnóstwo implementacji takich jak MS Windows, BSDI, CISCO, HP/UX, MVS i IRIX odsyłają odpowiedź RST/ACK.
 - (b) Sonda fałszywej flagi – idea polega na wysłaniu niezdefiniowanej flagi TCP (64 albo 128) w nagłówku TCP pakietu SYN. Maszyny z Linuxem wcześniejszym niż 2.0.35 zachowują tę flagę przy przesyłaniu odpowiedzi.
 - (c) Próbkowanie TCP ISN - pomysł opiera się na znalezieniu wzoru w początkowej sekwencji numerów wybranych przez implementacje TCP przy odpowiedzi na połączenie. Można podzielić je na kilka grup, takich jak tradycyjne 64K (dużo starszych UNIXów), losowa inkrementacja (nowsze wersje Solarisa, IRIX, FreeBSD, Digital UNIX, Cray i wiele

innych), prawdziwe losowe (Linux 2.0.*, OpenVMS, nowsze AIX etc.). Komputery z Windows (i kilka innych) używają inkrementacji opartej na upływie czasu, gdzie ISN jest inkrementowany o stałą, niewielką część podczas każdego 'tyknięcia zegara'. Jest to niemal tak proste do odgadnięcia, jak stare implementacje (64K). Oczywiście ulubiona technika jest 'constant' - te urządzenia zawsze używają tego samego ISN.

- (d) Monitorowanie bitu "Don't Fragment" - wiele systemów operacyjnych ustawia bit "Don't Fragment" w nagłówku IP pakietów, które wysyłają w celu polepszenia wydajności. Nie wszystkie OS'y ustawiają ten bit i robią to na różne sposoby, więc zwracając uwagę na ten bit możemy uzyskać sporo informacji na temat danego komputera.
- (e) Początkowy rozmiar okna TCP (TCP Initial Window) - dotyczy to sprawdzania rozmiaru okna powracających pakietów. Stare skanery stwierdzają "BSD 4.4" przy wykryciu niezerowego okna pakietu RST. Nowsze, np. **nmap** zwracają uwagę na dokładny rozmiar okna, który jest stały dla danego OS'a. Ten test daje nam sporo informacji, ponieważ część OS'ów może być dokładnie zidentyfikowana tylko na podstawie rozmiaru okna (np. AIX używa rozmiaru 0x3F25, NT5 Microsoft - 0x402E). Co ciekawe, jest to dokładnie ten sam
- (f) Wartość ACK - choć może się wydawać, że powinien to być standard, różne implementacje używają różnych wartości ACK w różnych przypadkach. Na przykład, powiedzmy, że wysłano FIN|PSH|URG, aby zamknąć port TCP. Większość implementacji ustawi jako ACK otrzymana od nadawcy wartość początkowa (initial sequence number - SEQ), chociaż np. Windows odesła SEQ+1. Jeśli zostanie wysłany ciąg

SYN|FIN|URG|PSH na otwarty port, Windows zachowuje sie bardzo niekonsekwentnie. Czasami odesle twoje SEQ, czasami SEQ++, a jeszcze kiedy indziej pseudolosowa wartosc.

- (g) Badanie komunikatów błędów ICMP (ICMP Error Message Quenching) - co mądrzejsze OS'y, zgodnie z RFC 1812 limitują liczbę zwracanych komunikatów o błędach. Na przykład, kernel Linuxa (net/ipv4/icmp.h) ogranicza liczbę wygenerowanych wiadomości 'destination unreachable' do 80 na 4 sekundy. Jeśli ta liczba zostanie przekroczona, wprowadza przerwy 1/4 sekundy. Jedynym sposobem na przeprowadzenie tego testu jest wysłanie jakiejś ilości pakietów na losowy, wysoki port UDP i zliczenie ilości powracających pakietów 'destination unreachable'.
- (h) Cytowanie komunikatów ICMP (ICMP Message Quoting) - RFC mówią, że pakiety ICMP z informacjami o błędach zawierają w sobie części komunikatów, które ten błąd wywołują. Dla 'port unreachable' prawie wszystkie implementacje odsyłają tylko zadany nagłówek IP i 8 bajtów. Jednak Solaris odsyła trochę więcej, a Linux jeszcze więcej niż Solaris. Największą zaletą jest fakt, że ta metoda pozwala rozpoznać maszyny z Solarisem i Linuxem nawet wtedy, kiedy nie mają otwartych żadnych portów.
- (i) Zgodność odsyłanych nagłówków ICMP (ICMP Error message echoing integrity) - komputery odsyłają część oryginalnego komunikatu razem z błędem 'port unreachable'. Subtelne różnice w zwracanych nagłówkach IP występujące w różnych implementacjach pozwalają na identyfikację typu systemu operacyjnego. Na przykład, AIX i BSD odsyłają pole 'IP total length' powiększone o 20 bajtów.

- (j) Typ usługi TOS (Type of Service) – sprawdzany jest TOS pakietów zwracanych z bledem 'port unreachable'. Prawie wszystkie implementacje uzywaja 0, ale Linux uzywa 0xC0. Nie jest to zadna ze standardowych wartosci TOS, lecz czesc nieuzywanego pola pierwszenstwa (precedence field).
- (k) Obsluga fragmentacji - technika ta polega ona na tym, ze różne implementacje skladaja w różny sposób podzielone fragmenty IP. Czesc nadpisuje stara czesc nowymi danymi, czesc 'puszcza przodem' starsze dane.
- (l) Opcje TCP - to naprawde zyla zlota, jesli chodzi o wyciaganie informacji. Ich piekno polega na tym, ze:
- sa to opcje ogólne, wiec nie wszystkie maszyny maja je zaimplementowane
 - mozna sie dowiedziec, czy dany host ma je zaimplementowane wysylajac pakiet z ustawiona odpowiednia opcja. Jesli dany host akceptuje taka opcje, odsyla ja ustawiona.
 - mozna sprawdzic naraz kilka opcji, wysylajac tylko jeden pakiet;

```
nmap -O 127.0.0.1
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1544 ports scanned but not shown below are in state: closed)
```

Port	State	Service
22/tcp	open	ssh
25/tcp	open	smtp
111/tcp	open	sunrpc
515/tcp	open	printer
6000/tcp	open	X11

No exact OS matches for host (If you know what OS is running on it, see <http://www.insecure.org/cgi-bin/nmap-submit.cgi>).

TCP/IP fingerprint:

```
SInfo(V=2.54BETA30%P=i686-pc-linux-gnu%D=3/16%Time=3C933763%O=22%C=1)
TSeq(Class=RI%gcd=1%SI=1BBD33%IPID=Z%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=1BC3AD%IPID=Z%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=1BC39C%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLen=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Uptime 0.006 days (since Sat Mar 16 13:07:34 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 7 seconds

Komentarz:

```
T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
```

- Test T1 - w tym testie wysyłamy pakiet SYN z ustawionymi kilkoma opcjami na otwarty port. DF=Y oznacza, że bit "Don't fragment" może zostać ustawiony w odpowiedzi. W=7FFF oznacza, że okno oferowane (advertised window) musi być równe 0x7FFF. ACK=S++ mówi, że potwierdzenie jakie otrzymamy musi być naszą sekwencją inicjującą (initial sequence) zwiększoną o 1. Flags=AS oznacza, że flagi ACK i SYN zostały wysłane w odpowiedzi. Ops=MNWNNT mówi, że opcje w odpowiedzi muszą być w następującym w takim porządku porządku

```
T2(Resp=N)
```

- Test 2 dotyczy TCP NULL z takimi samymi opcjami na otwarty port. Resp=Y oznacza, że musimy uzyskać odpowiedź.
- Itp.
- **Techniki pasywne (klasyczne)**
- Techniki pasywne korzystają ze standardowych narzędzi, np. telnet.

```
playground~> telnet hpux.u-aizu.ac.jp
Trying 163.143.103.12 ...
Connected to hpux.u-aizu.ac.jp.
Escape character is '^]'.

HP-UX hpux B.10.01 A 9000/715 (ttyp2)

login:
```

- Wiadomości zarejestrowane przy pomocy programu **snort**:

```
01/26-13:16:25.746316 XXX.XXX.XXX.XXX -> YYY.YYY.YYY.YYY
TCP TTL:255 TOS:0x0 ID:6059
**S***A* Seq☺xD3B709A3 AC: 0x23BE5372 Win: 0x2798
TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS 1460
```

Ataki typu DoS (odmowa obsługi)

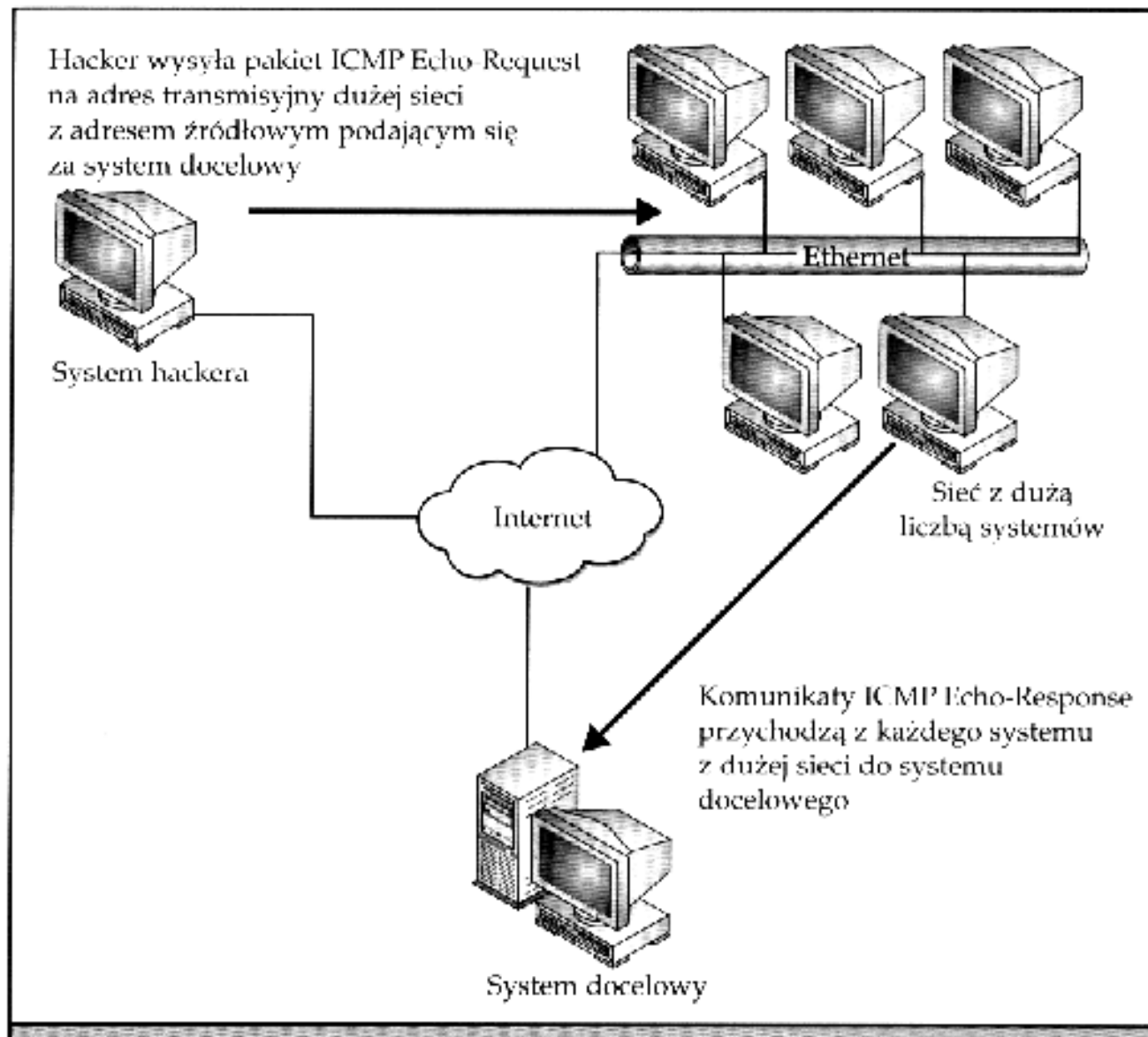
- Jest to grupa ataków mająca na celu spowodowanie awarii lub przeciążenia hosta lub usługi w sieci. Jest to bardzo popularny atak mogący wyrządzić duże szkody. Zwłaszcza, że spowodowanie przeciążenia powodujące wyłączenie jakiegoś systemu może doprowadzić nie

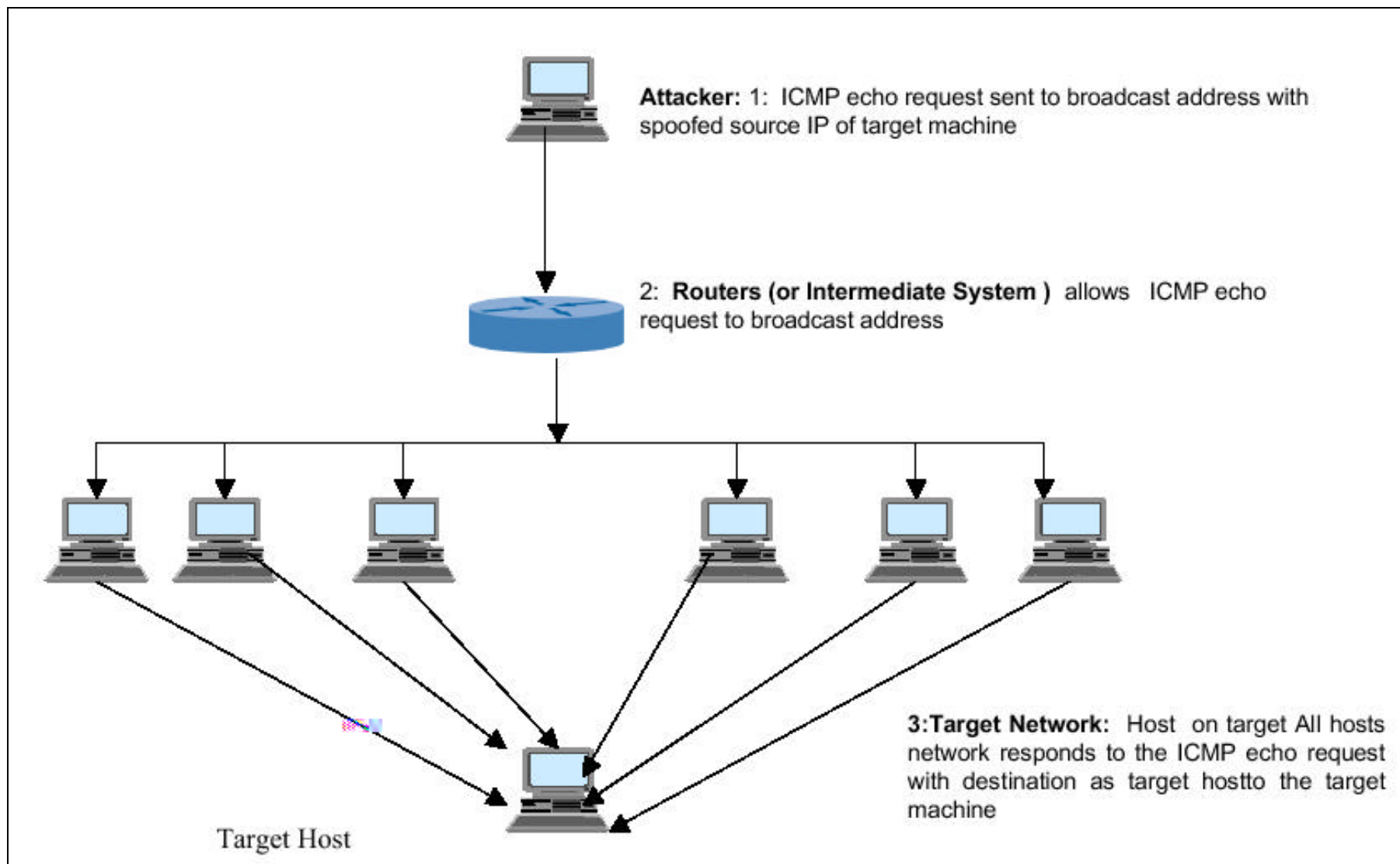
tylko do utraty dostępności usługi, uszkodzenie danych lub hosta, ale może również pomóc w przejściu przez zabezpieczenia. Ataków tego typu jest bardzo wiele i wciąż powstają nowe.

- Dla przykładu i pokazaniu możliwych implikacji można przyjrzeć się atakowi **Smurf Attack**.

Atak ten wykonywany jest następująco:

- Atakujący tworzy *ping* (ICMP echo request) i wysyła go na adres rozgłoszeniowy sieci, podając jednocześnie jako adres źródła IP ofiary.
- Każdy host w sieci przygotowuje odpowiedź ICMP echo reply i odsyła go na adres IP ofiary.
- Wzmoczony ruch pakietów ICMP echo reply może w prosty sposób przeciążyć system ofiary.
- Przeciążenie atakowanego hosta (łącznie z jego wyłączeniem) pozwala na wyłączenie go z sieci, pozwala przeciążyć przechwytywanie pakietów w systemach IDS lub spowodować reset komputera, co czasami jest konieczne dla uruchomienia nowej usługi (z koniem trojańskim).





Konie Trojańskie

- Konie trojańskie są to niewinnie wyglądające programy, które pozwalają na zdalną kontrolę (w mniejszym lub większym stopniu) komputera celu. Programy tego typu są bardzo niebezpieczne ze względu na to, iż możemy wykonywać operacje na komputerze, który znajduje się w chronionej sieci. Dla przykładu możemy tutaj wspomnieć o jednym z popularniejszych programie **BackOrifice2000**, który pozwala na opanowanym komputerze dokonywać dowolnych czynności takich jak kopiowanie, uruchamianie programów, przejmowanie pulpitu.

Wirusy

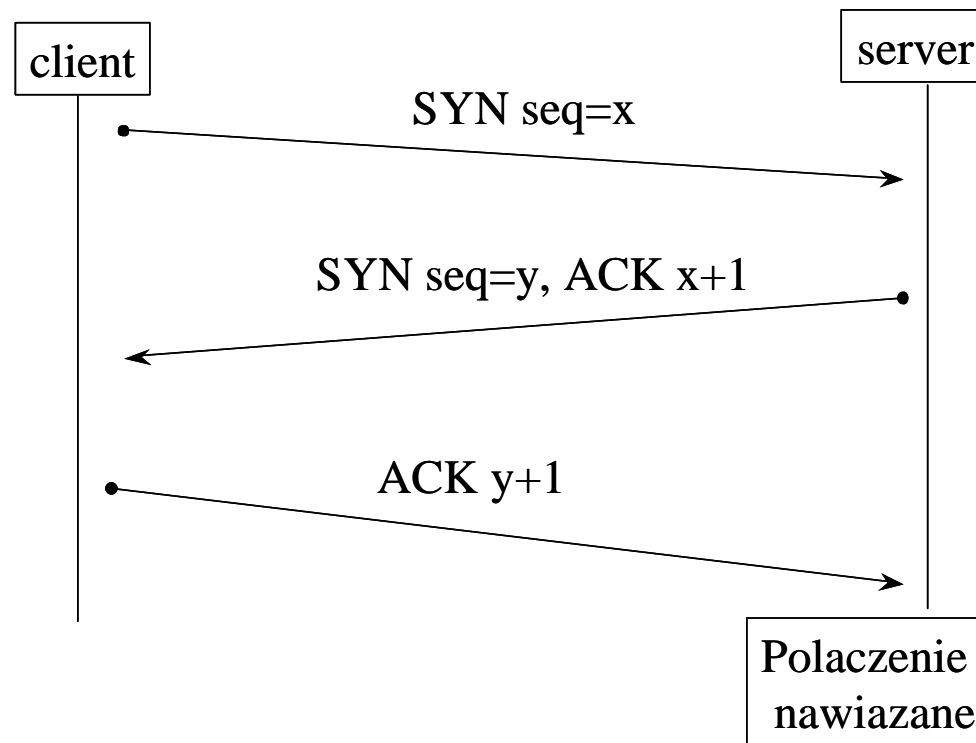
- Programy, które samo replikując się w danym środowisku mogą doprowadzić do uszkodzenia systemu, sprzętu, danych lub też otworzyć drzwi dla intruza w niezabezpieczonym serwerze. Wszystkie wirusy są niebezpieczne i naruszają bezpieczeństwo systemu, ale w naszym przypadku zwróćmy uwagę na popularnego wirusa roku 2001 **CodeRed II**. Wirus ten, rozprzestrzeniał się poprzez pocztę elektroniczną, zaraził serwery NT instalując w nich pakiety rootkit otwierające drzwi do serwerów dla przypadkowych intruzów. Pierwsza wersja tego wirusa dokonywała automatycznych ataków na serwery NT wykorzystując znane dziury w systemach zabezpieczeń.

Podsluchiwanie sieci (sniffing)

- Podsluchiwanie sieci LAN jest to ogół technik służących do podsluchiwania ruchu w sieci. Pozwala on na poznanie topologii sieci, zdobycia nie zakodowanych hasel do różnych usług i zasobów niekodujących przesyłanych hasel (telnet, POP3). Jak i zdobycie zakodowanych hasel, które następnie można próbować złamać (LanMan NT), pozwala również na podsłuch innych danych przesyłanych siecią. Dużą zaletą dla intruza jest dość trudne wykrycie przeprowadzanego podsłuchu.

Podszywanie się (spoofing)

- Nawiązanie sesji TCP:
- Atak ten polega zwykle na:
 - odgadnięciu liczby sekwencyjnej protokołu TCP
 - przeprowadzeniu ataku Syn Flood na host docelowy
- Przypomnienie:
 - Host na TCP SYN odpowiada SYN|ACK, jeśli port docelowy jest otwarty, w przeciwnym przypadku - RST|ACK.
 - Na TCP SYN|ACK host odpowiada RST, na RST nie odpowiada niczym.



- Strony scenariusza ataku:
 - host X – komputer hackera
 - host B – „milczący” host (nie nadaje żadnych pakietów)
 - host A – cel ataku, ofiara podatna na skanowanie TCP SYN
- Scenariusz ataku podszywania się pod IP
 - Host X monitoruje nagłówki IP, wychodzące z hosta B

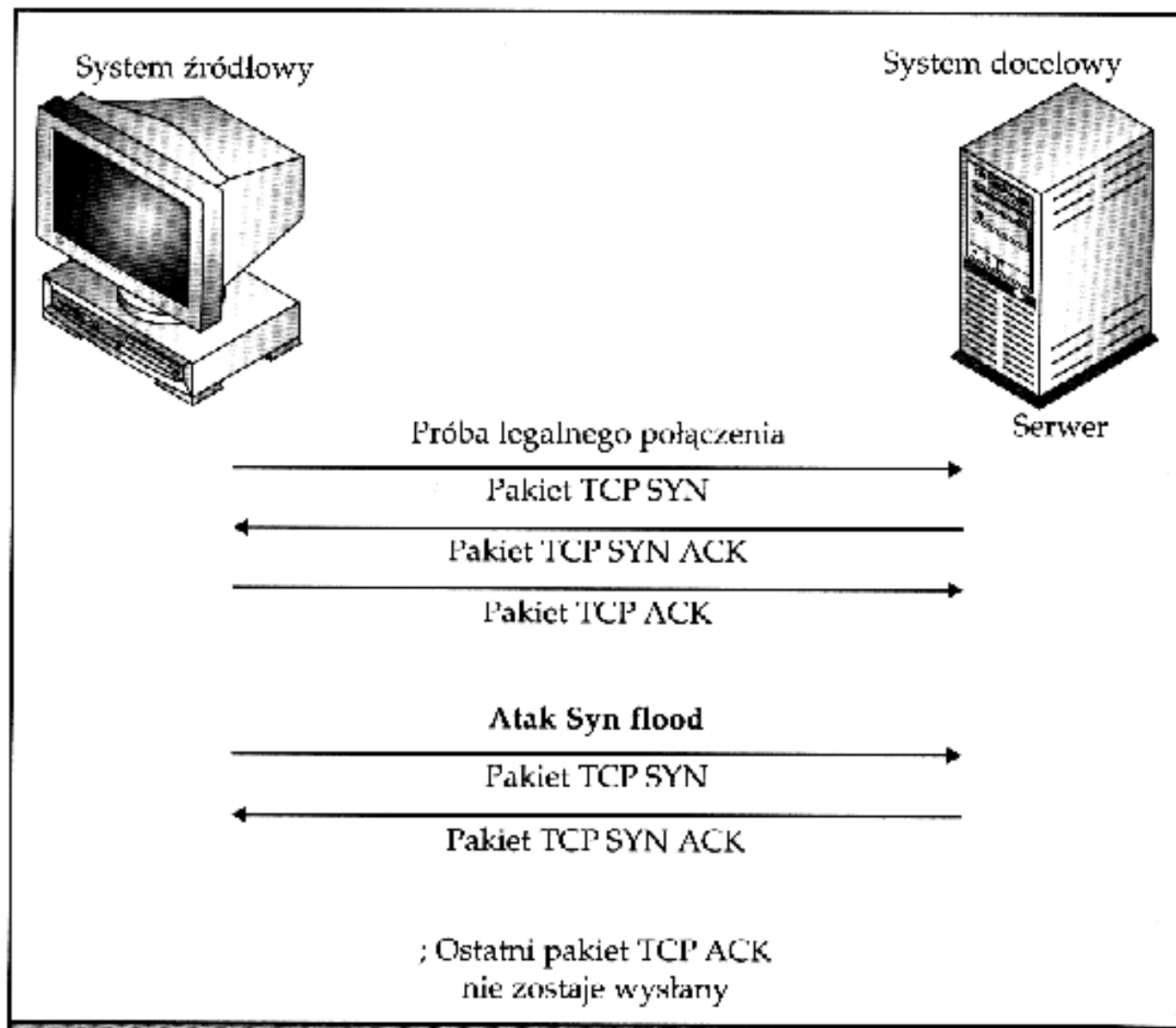
```
#hping B -r
```

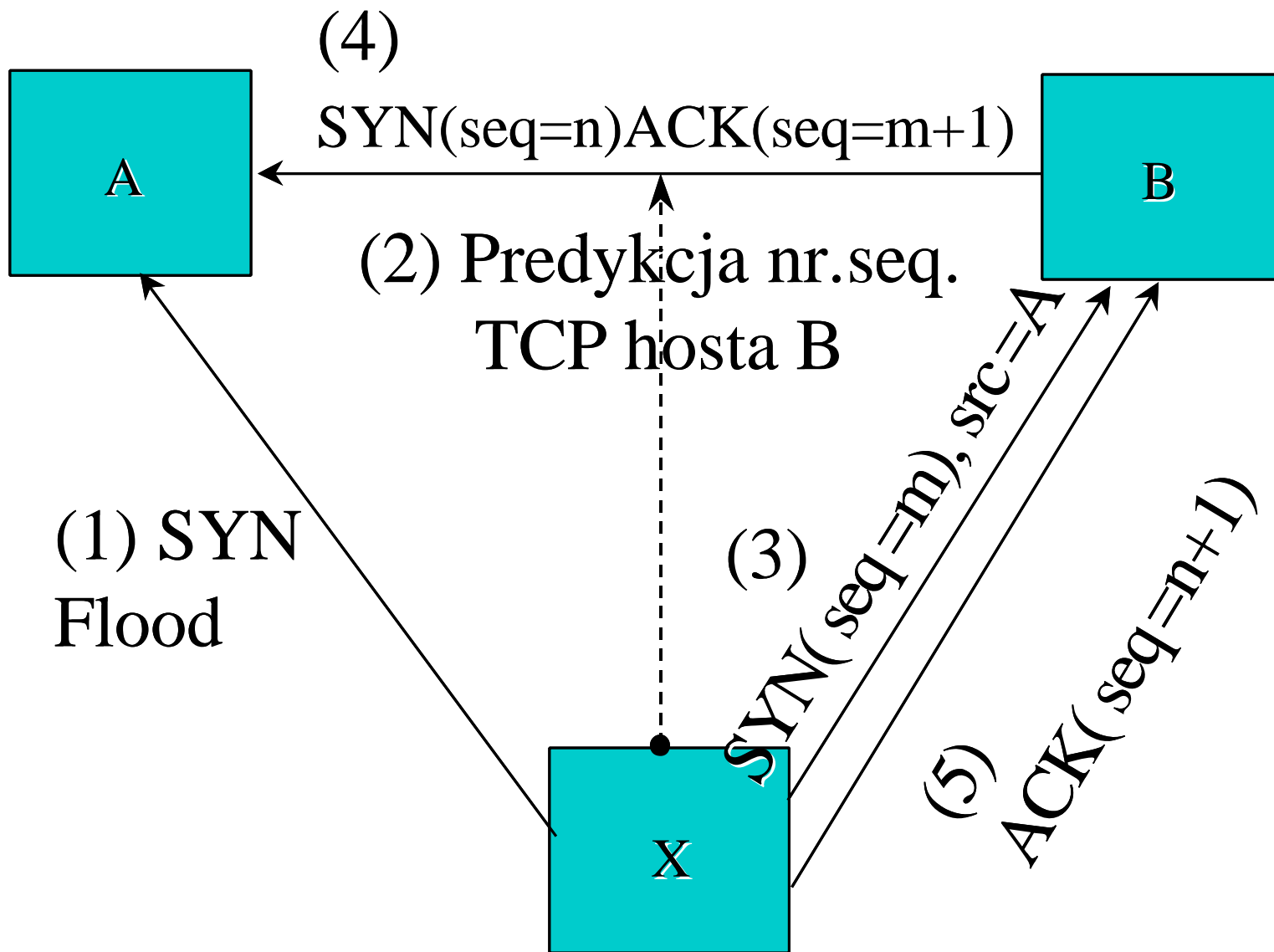
```
HPING B (eth0 xxx.yyy.zzz.jjj): no flags are set, 40 data bytes
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=0 ttl=64
                                id=41660 win=0 time=1.2 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=1 ttl=64
                                id=+1 win=0 time=75 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=2 ttl=64
                                id=+1 win=0 time=91 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=3 ttl=64
                                id=+1 win=0 time=90 ms
-cut-
```

- Teraz X wysyla SYN na port Y hosta A, podajac sie za B. Jesli port Y hosta A jest otwarty, to A wysle odpowiedz SYN|ACK do B (host A nie wie, ze w rzeczywistosci nadawca jest host X). Host B odpowie na SYN|ACK wyslaniem RST do hosta A.
 - Jesli X wysle na port Y hosta A kolejno kilka TCP SYN, to A odpowie B kilkoma SYN|ACK, na co z kolei B wysle kilka RST, itd.
 - Host B wysyla wiec pakiety.
- Atak ten latwo jets zrealizowac uruchamiajac dwie oddzielne konsole pod Linuxem.
- Pierwsza sesja podsłuchuje hosta B:

```
hping B -r
```
 - Drugua sesja sluzi do podszywania sie pod B (wysyla TCP SYN):

```
hping A -a B -S
```
- Przyklad 2 podszywania sie pod adres IP:
- Scenariusz ataku SYN FLOOD





Psychologia społeczna

- Psychologia społeczna jest to ogół zagadnień składających się na zdobywaniu informacji dzięki zdobyciu zaufania użytkownika. Czyli przekonanie np. użytkownika do podania hasła dostępu lub też wywiad z użytkownikiem pozwalający na zdobycie informacji dotyczących stosowanych zabezpieczeń, topologii sieci, polityki bezpieczeństwa, systemów autoryzacji.

Metody prowadzenia ataków

Przykład 1

- Załóżmy, że naszym celem jest host o adresie 196.3x.2x.7x i znanych aktywnych usługach (portach). Załóżmy, że znamy także rodzaj systemu, który jest uruchomiony na naszym hoscie – celu.
- Znajdźmy jakiegos użytkownika tego systemu, najlepiej root'a.
- Sprawdźmy, czy dostana jest usługa **finger**:

```
jp@dev# finger @196.3x.2x.7x
[196.3x.2x.7x]
finger: read: Connection refused
```

- Usług **finger** nie jest aktywna. Spróbujmy więc na porcie 35 (SMTP). Jeśli **sendmail** nie jest skonfigurowany, to aktywne powinny być polecenia VRFY i EXPN:

```
# telnet 196.3x.2x.7x 25
Trying 196.3x.2x.7x...
Connected to xxx.xx.co.za.
Escape character is '^]'.
220 xxx.xx.co.za ESMTP Sendmail 8.7.1/8.7.1; Mon, 14 Aug 2000
      00:34:01 +0100 (BST)
vrfy test
250 user test@xxx.xx.co.za
vrfy user
550 user... User unknown
```

```
vrfy u46b00
550 u46b00... User unknown
vrfy root
250 <root@xxx.xx.co.za>
expn root
250 <root@xxx.xx.co.za>
vrfy guest
550 guest... User unknown
vrfy mail
550 mail... User unknown
expn webmaster
550 webmaster... User unknown
expn postmaster
250 <root@xxx.xx.co.za>
```

- Użytkownik "test" istnieje, zaś "user" i "u46b00" - nie. Istnieje także użytkownik "root". "root" nie ma żadnych aliasów, ale "postmaster" jest uprawniony przez "root".
- Użytkownik „test” jest identyfikatorem, który jest bardzo często używany. Spróbujmy zgadnąć jego hasło:

```
# telnet 196.3x.2x.7x
Trying 196.3x.2x.7x...
Connected to xxx.xx.co.za.
Escape character is '^]'.
HP-UX u46b00 B.10.20 A 9000/831 (ttypl)
login: test
Password:
Login incorrect
login: test
Password:
Login incorrect
login: test
```

Password:

Login incorrect Connection closed by foreign host.

- Interesujące, użytkownik "test" nie posługuje się hasłem "test", "test1" lub "test01". Możemy próbować dalej, ale jest to droga, która nie może prowadzić do sukcesu. Spróbujmy raczej sprawdzić innych użytkowników.

```
# ftp 196.3x.2x.7x
Connected to 196.3x.2x.7x.
220 u46b00 FTP server (Version 1.7.212.2
          Tue Apr 21 12:14:46 GMT 1998) ready.
Name (196.3x.2x.7x:roelof): anonymous
331 Guest login ok, send indent as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd/etc
250 CWD command successful.
ftp> get passed
local: passwd remote: passwd
227 Entering Passive Mode (196,3x,2x,7x,8,186)
150 Opening BINARY mode data connection for passed (7695
          bytes).
100% |*****| 7695 00:00 ETA
226 Transfer complete.
7695 bytes received in 2.06 seconds (3.64 KB/s)
ftp> exit 221 Goodbye.
~/perl/telnet/brute more passwd root:*:0:3::/var/sam:/usr/bin/false
root:*:0:3::/var/sam:/usr/bin/false daemon:*:1:5::/var/sam:/usr/bin/false
bin:*:2:2::/var/sam:/usr/bin/false sys:*:3:3::/var/sam:/usr/bin/false
adm:*:4:4::/var/sam:/usr/bin/false uucp:*:5:3::/var/sam:/usr/bin/false
lp:*:9:7::/var/sam:/usr/bin/false nuucp:*:11:11::/var/sam:/usr/bin/false
```

```
hpdb:*:27:1::/var/sam:/usr/bin/false
-----cut-----
```

- Nowe systemy UNIX przechowują hasła w pliku „shadow”. Przyjrzyjmy się jednak dokładniej plikowi passwd:

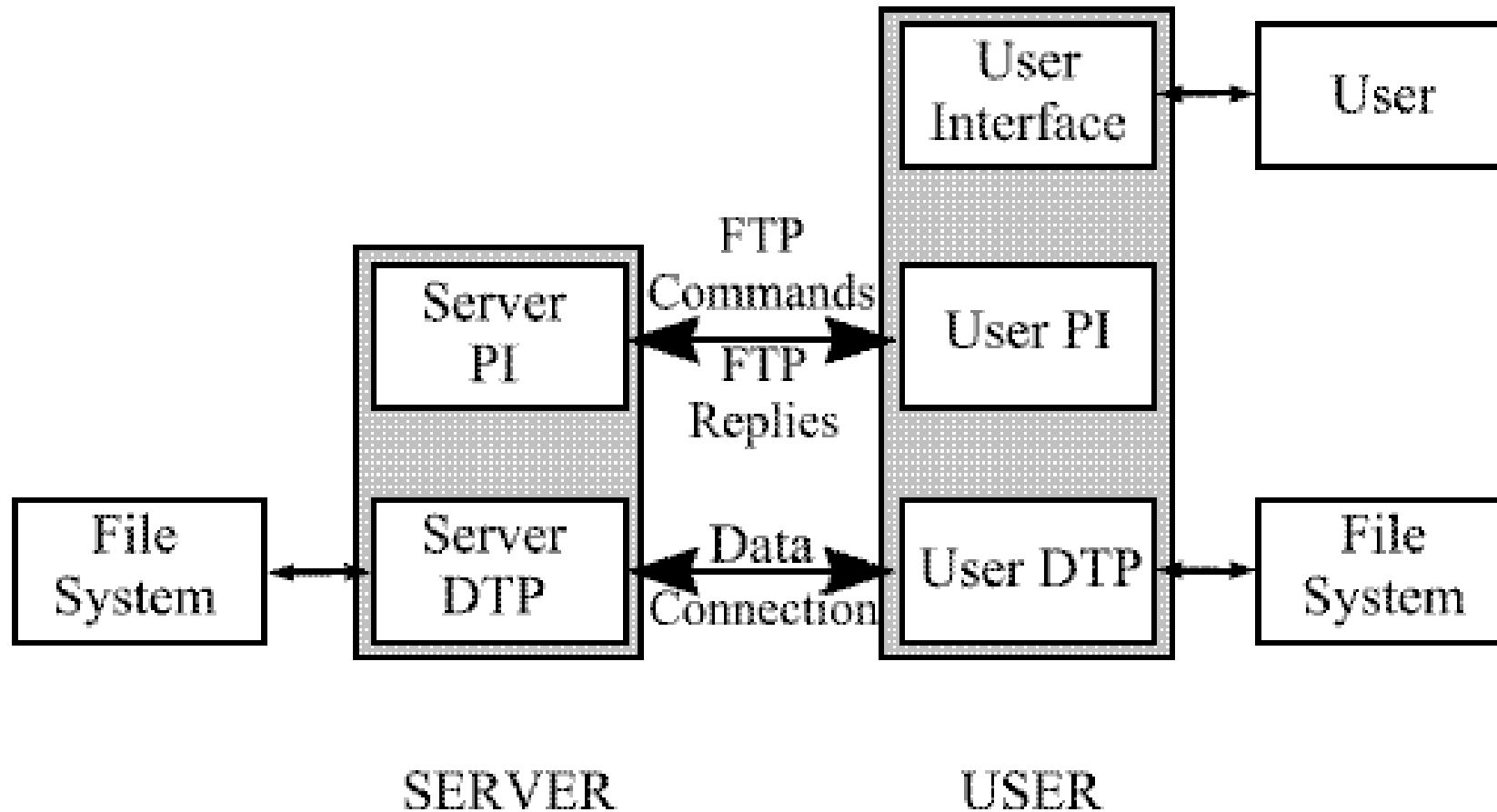
```
pro:*:100:100::/var/sam:/usr/bin/false mis2000:*:208:1000::/var/sam:/usr/bin/false
lab:*:369:2000::/var/sam:/usr/bin/false oracle:*:101:100::/var/sam:/usr/bin/false
doggy:*:541:2000::/var/sam:/usr/bin/false f399:*:611:2000::/var/sam:/usr/bin/false
```

- To są nazwy użytkowników, którzy nie pasują do opisu tzw. „zwykłych” użytkowników – ich hasła są „słabe” i używane przez więcej niż jedną osobę. Zaczniemy od "mis2000":

```
# telnet 196.3x.2x.7x
Trying 196.3x.2x.7x...
Connected to xxx.xx.co.za.
Escape character is '^]'.
HP-UX u46b00 B.10.20 A 9000/831 (ttypl)
login: mis2000
Please wait...checking for disk quotas
What is your terminal type?
```

- Wspaniale! System nie pyta nas o hasło! Jesteśmy w systemie!

Przykład 2: FTP Protocol - Cele i opis działania protokołu



Objasnienia:

PI - interpretator protokołu (*Protocol Interpreter*).

DTP - proces przekazywania danych (*Data Transfer Process*).

User-PI – interpretator protokołu ze strony użytkownika (*User-Protocol Interpreter*).

Server-PI – interpretator protokołu ze strony serwera (*Server-Protocol Interpreter*).

User-DTP- proces transmisji danych ze strony użytkownika (*User-Data Transfer Process*).

Server-DTP - proces transmisji danych ze strony serwera (*Server-Data Transfer Process*).

Data Connection – łącze danych, przez które odbywa się transmisja plików.

FTP Commands – komendy FTP służące do komunikacji pomiędzy User-PI a Server-PI.

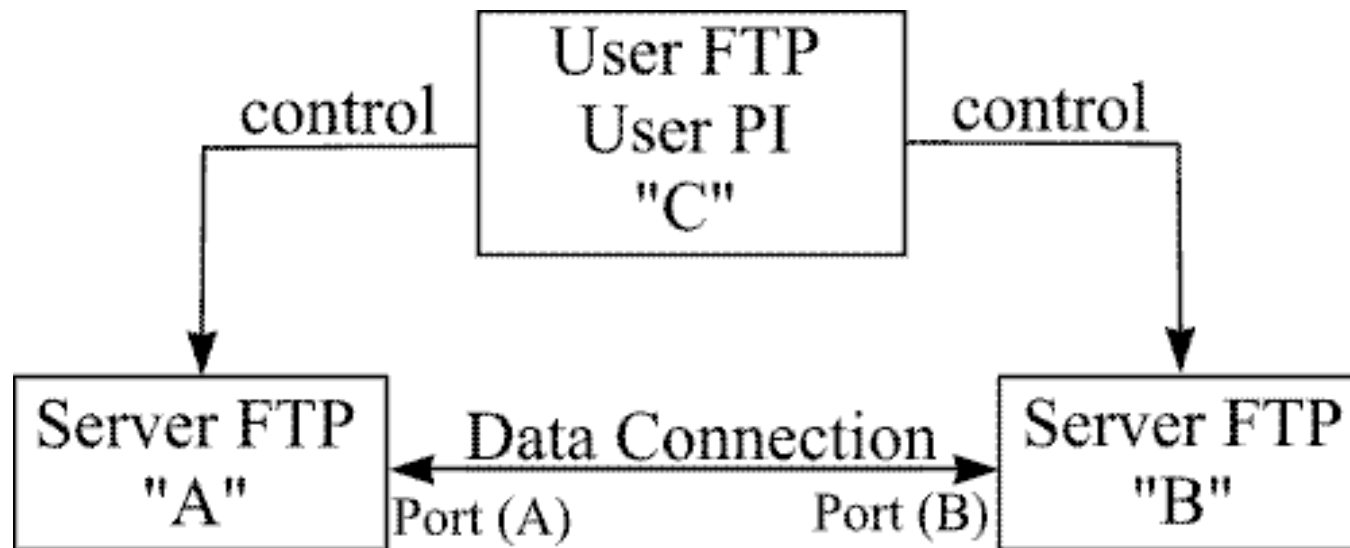
- Definiują one parametry dla połączenia danych (data connection): port danych, tryb przekazu, typy przedstawienia danych i struktur oraz typ operacji (zachowanie, zmiana, dodawanie, kasowanie etc.).
- Najczęściej używane komendy pokazuje poniższa tabelka

Komenda	Opis
ABOR	Przerywa polecenie FTP i każdą transmisję danych
LIST	Wyswietla listę plików i katalogów
PASS	Hasło umożliwiające zalogowanie się
QUIT	Wylogowanie się z serwera
RETR	Pobranie pliku
STOR	Umieszczenie pliku
SYST	Serwer zwraca informacje o rodzaju systemu
TYPE	Określa typ pliku
USER	Nazwa użytkownika

- **Sposób działania:**

- Protokół ze strony użytkownika (User-PI) i serwera (Server-PI) jest zrealizowany w różny sposób.
- Server-PI - oczekuje na określonym porcie na połączenie zainicjowane przez User-PI. Przyjmuje standardowe polecenia od User-PI, wysyła odpowiedzi i kieruje zadania do Server-DTP.
- User-PI - jest inicjatorem tworzenia połączenia ze swego portu z procesem Server-PI, wysyła polecenia FTP oraz zarządza User-DTP, o ile taki proces się pojawi.
- Server-DTP oraz User-DTP mogą działać według dwóch trybów: aktywnego i pasywnego.
- **Tryb aktywny:**
 - Server-DTP - inicjuje połączenia z procesem User-DTP. Zgodnie z poleceniami od Server-PI ustawia parametry do wysyłania i odbioru (adres oraz port serwera i klienta) i transmituje dane.
 - User-DTP – oczekuje na określonym porcie na połączenie zainicjowane przez Server-DTP.
- **Tryb pasywny:**
 - Server-DTP - oczekuje na określonym porcie na połączenie zainicjowane przez User-DTP.

- User-DTP - inicjuje połączenia z procesem Server-DTP. Zgodnie z poleceniami od User-PI ustawia parametry do wysyłania i odbioru (adres oraz port serwera i klienta) i transmituje dane.
 - User-PI inicjalizuje połączenie kierujące (control connection), podczas którego przekazuje standardowe polecenia FTP do Server-PI. Jest ono realizowane za pomocą protokołu Telnet, w związku z czym użytkownik może w prosty sposób sam połączyć się z Server-PI (np. z poziomu terminala) i pominąć tym samym User-PI. Standardowe odpowiedzi przesyłane są z Server-PI tą samą drogą.
- W trybie aktywnym User-DTP powinien „słuchać” na określonym porcie danych (data port), a serwer inicjować połączenie (data connection) i transmitowanie danych (data transfer) zgodnie ze zdefiniowanymi parametrami. Ważne jest to, że port danych (data port) nie koniecznie musi znajdować się na tym samym hostcie, który przekazuje polecenia poprzez połączenie kierujące (control connection).
 - Możliwa jest więc sytuacja, w której użytkownik przekazuje pliki pomiędzy dwoma hostami, z których żaden nie jest lokalny. W tym przypadku użytkownik tworzy połączenie kierujące (control connection) z dwoma serwerami, a następnie organizuje pomiędzy nimi połączenie danych (data connection). W ten sposób informacja kierująca przechodzi przez User-PI a dane przekazywane są pomiędzy procesami Server-DTP.
 - Diagram sytuacji, w której użytkownik przesyła pliki pomiędzy dwoma nielokalnymi hostami:



- Protokół wymaga, żeby połączenie kierujące było otwarte w czasie transmisji danych. Użytkownik sam musi podjąć decyzję o jego zamknięciu, gdy zakończy korzystanie z serwisów mimo, że to serwer wykona zamknięcie. Serwer może też zakończyć przekaz danych, jeśli połączenia kierujące zostaną zamknięte bez żadnego polecenia.
- Istotną cechą protokołu FTP jest fakt, iż połączenie danych (data connection) może być użyte do jednoczesnego wysyłania i odbierania. Połączenie danych używane jest do następujących zadań:
 - przesyłanie plików od klienta do serwera.
 - przesyłanie plików od serwera do klienta.

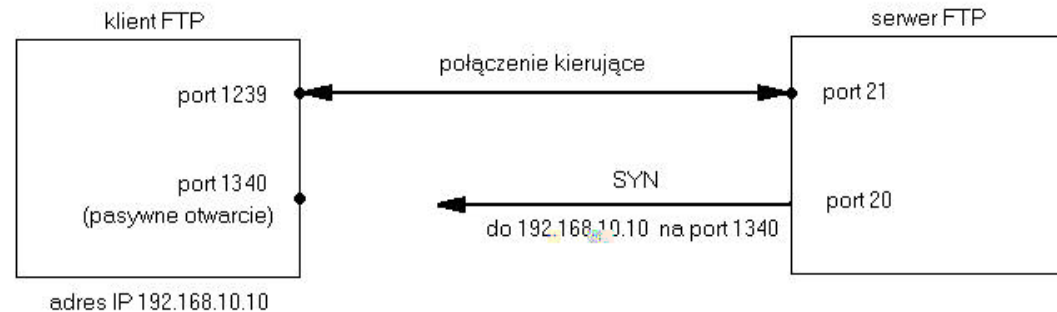
□ **Tryb aktywny przesyłania danych:**

- (1) Klient wybiera jeden z efemerycznych portów hosta, na którym jest uruchomiony. Następnie wykonuje pasywne otwarcie na tym porcie (oczekuje na połączenie od serwera).
- (2) Klient wysyła do serwera informacje na temat wybranego przez siebie portu, używając do tego celu polecenia PORT.



Na rysunku widac stan polaczenia po wykonaniu kroku 2. Zakladamy, ze portem uzywanym przez klienta dla polaczenia kierujacego jest port 1239, natomiast dla polaczenia danych – port 1340. Polecenie PORT wyslane przez klienta sklada sie z szesciu oddzielonych przecinkami liczb dziesietnych w kodzie ASCII. Pierwsze cztery okreslaja adres IP klienta. Pozostale dwie informuja o numerze portu, który w naszym przykladzie obliczany jest nastepujaco: $5 * 256 + 60 = 1340$.

- (3) Serwer odbiera informacje o numerze portu wybranego przez klienta, a następnie wykonuje aktywne otwarcie do tego portu. Po stronie serwera połączenie danych używa portu o numerze 20.



Rysunek przedstawia stan połączenia po wysłaniu aktywnego otwarcia (SYN) przez serwer.

□ **Tryb pasywny przesyłania danych:**

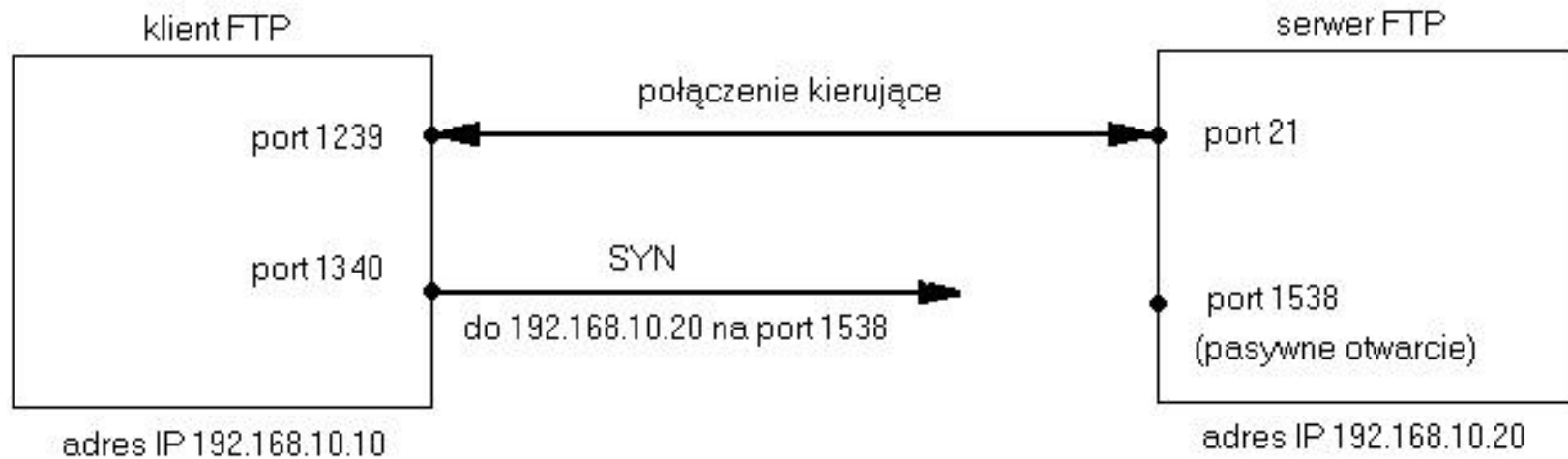
- (1) Klient wysyła do serwera zadanie zastosowania trybu pasywnego.
- (2) Serwer wybiera jeden z efemerycznych portów hosta, na którym jest uruchomiony. Następnie wykonuje pasywne otwarcie na tym porcie (oczekuje na połączenie od klienta).
- (3) Serwer wysyła do klienta informacje na temat wybranego przez siebie portu, używając do tego celu polecenia PORT.

Na rysunku widac stan polaczenia po wykonaniu kroku 3. Analogicznie jak w przypadku trybu aktywnego zakladamy, ze portem uzywanym przez klienta dla polaczenia kierujacego jest port 1239, natomiast dla polaczenia danych – port 1340. W odrzniczeniu jednak od trybu aktywnego, portem dla polaczenia danych na serwerze jest port efemeryczny 1538. Polecenie PORT wyslane przez serwer dziala tak samo jak w trybie aktywnym.



- (4) Klient odbiera informacje o numerze portu wybranego przez serwer, a następnie wykonuje aktywne otwarcie do tego portu. Po stronie klienta polaczenie danych uzywa takze portu efemerycznego.

Stan polaczenia po wyslaniu aktywnego otwarcia (SYN) przez klienta:



□ **Przykład ataku – faza analizy**

- Do testów użyto następujące komputery:

Rubycon: adres IP - 212.14.1.69, oprogramowanie - standardowy Linuxowy klient FTP

Phobos: adres IP - 212.14.1.71, oprogramowanie - serwer wu-ftpd w wersji 2.6.1

Wymiana pakietów, która odbyła się między serwerem a klientem podczas powyższego testu zarejestrowana została za pomocą narzędzia **ngrep** uruchomionego na komputerze **phobos**.

- Przykład polega na ściągnięciu pliku „test.txt” z serwera phobos na serwer rubycon. Klienta FTP uruchamiamy z opcja -d, dzięki czemu będzie on wyświetlał na monitorze polecenia i odpowiedzi wymieniane przez połączenie kierujące.
- Akcja użytkownika (1):

```
rubycon:~$ ftp -d phobos
[1] łączymy się z serwerem phobos
Connected to phobos.man.szczecin.pl
[2] klient wykonuje aktywne otwarcie połączenia kierującego

220 phobos.man.szczecin.pl FTP server ready.
[3] serwer przedstawia się i informuje, że jest gotowy
```

- Wymiana pakietów (1):

```
[2] Three-Way Handshaking - standardowa procedura nawiązywania
    połączenia TCP
T 212.14.1.69:4560 -> 212.14.1.71:21 [S]
T 212.14.1.71:21 -> 212.14.1.69:4560 [AS]
#
T 212.14.1.69:4560 -> 212.14.1.71:21 [A]
#

[3] serwer przedstawia się i informuje, że jest gotowy
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
220 phobos.man.szczecin.pl FTP server ready
#
T 212.14.1.69:4560 -> 212.14.1.71:21 [A]
```

- Akcja użytkownika (2):

```
Name (phobos:cadence): cadence
  [4] klient prosi o podanie nazwy uzytkownika
---> USER cadence
  [5] logujemy sie jako uzytkownik 'cadence'
```

- Wymiana pakietów (2):

```
[5] logujemy sie jako uzytkownik 'cadence'
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
  USER cadence.
#
T 212.14.1.71:21 -> 212.14.1.69:4560 [A]
```

- Akcja uzytkownika (3):

```
Password:
  [6] podajemy haslo, które nie jest wyswietlane na ekranie
---> PASS XXXX
  [7] klient wysyla haslo do serwera
230 User cadence logged in.
---> SYST
  [8] klient sprawdza wersje systemu operacyjnego serwera
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
```

- Wymiana pakietów (3):

```
[7] klient wysyla haslo do serwera
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
  PASS haslo5..
#
T 212.14.1.71:21 -> 212.14.1.69:4560 [A]
```

```
#
Serwer informuje o poprawnym zalogowaniu sie uzytkownika
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
  230 User cadence logged in...

#
[8] klient sprawdza wersje systemu operacyjnego serwera
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
  SYST..

#
Serwer informuje o wersji swojego systemu operacyjnego
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
  215 UNIX Type: L8..

#
T 212.14.1.69:4560 -> 212.14.1.71:21 [A]
```

- Akcja uzytkownika (4)

```
ftp> get test.txt
  [9] pobieramy z serwera plik 'test.txt'

local: test.txt remote: test.txt

---> TYPE I
  [10] klient zada typu binarnego do transferu pliku

200 Type set to I.
```

- Wymiana pakietów(4)

```
#
[10] klient zada typu binarnego do transferu pliku
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
```

```
TYPE I..
#
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
  200 Type set to I...
```

- Akcja użytkownika (5)

```
---> PORT 212,14,1,69,17,218
  [11] polecenie PORT

200 PORT command successful.
```

- Wymiana pakietów (5)

```
#
[11] polecenie PORT
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
  PORT 212,14,1,69,17,218..
#
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
  200 PORT command successful...
```

- Analiza bezpieczeństwa:

Klient wysyła do serwera polecenie PORT zawierające adres i numer portu, na który serwer powinien wykonać aktywne otwarcie połączenia danych. Serwer FTP przyjmuje te dane niezależnie od zawartych w nich informacji.

Wniosek: Intruz działający na kliencie FTP może zmusić serwer do wykonania połączenia na dowolny port dowolnego komputera w Internecie. Jest to błąd bezpieczeństwa w specyfikacji protokołu FTP.

- Akcja użytkownika (6):

```
---> RETR test.txt
      [12] klient zada od serwera przesłania pliku 'test.txt'

150 Opening BINARY mode data connection for test.txt (44 bytes).
      [13] serwer informuje o otwarciu połączenia danych
```

- Wymiana pakietów (6):

```
#
[12] klient zada od serwera przesłania pliku 'test.txt'
T 212.14.1.69:4560 -> 212.14.1.71:21 [AP]
  RETR test.txt..

#
serwer nawiązuje połączenie danych na porcie podanym przez klienta w poleceniu PORT
T 212.14.1.71:20 -> 212.14.1.69:4570 [S]
T 212.14.1.69:4570 -> 212.14.1.71:20 [AS]
#
T 212.14.1.71:20 -> 212.14.1.69:4570 [A]

#
[13] serwer informuje o otwarciu `połączenia danych
T 212.14.1.71:21 -> 212.14.1.69:4560 [AP]
  150 Opening BINARY mode data connection for test.txt (44 bytes)...
```

- Analiza bezpieczeństwa:

Aby wysłać zadany przez klienta plik, serwer wykonuje aktywne otwarcie połączenia danych na port podany przez klienta w poleceniu PORT. Klient nie sprawdza jednak, czy

komputer, który to połączenie wykonuje jest faktycznie tym komputerem, z którego klient chciał pobrać plik.

Wniosek: Intruz jest w stanie połączyć się z otwartym portem klienta, zanim zrobi to serwer. Jest to błąd bezpieczeństwa w specyfikacji protokołu FTP.

□ **Przykład ataku – atak Denial of Service na FTP**

- Zakładamy, że intruz wie już kiedy użyjemy klienta FTP i w jakim działa on trybie. Aby zasymulować atak **Denial of Service** na serwer FTP, w odpowiednim momencie odłączymy ten komputer fizycznie od sieci. Po kilku sekundach przyłączymy go z powrotem, aby zaobserwować jak zachowa się oprogramowanie.
- Do testów wykorzystamy następujące komputery:
- Do testów użyto następujące komputery:

Rubycon: adres IP - 212.14.1.69, oprogramowanie - standardowy Linuxowy klient FTP

Phobos: adres IP - 212.14.1.71, oprogramowanie - serwer wu-ftp w wersji 2.6.1

Intruz: adres IP - xxx.xxx.xxx.xxx, oprogramowanie - Linux

- Test polega na wylistowaniu zawartości katalogu domowego użytkownika „cadence”.

EKRAN KLIENTA

```
rubycon:~$ ftp -d phobos
Connected to phobos.man.szczecin.pl.
220 phobos.man.szczecin.pl FTP server ready
Name (phobos:cadence): cadence
---> USER cadence
331 Password required for cadence.
Password:
---> PASS XXXX
230 User cadence logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
```

- (1) W tym momencie jesteśmy zalogowani na serwerze **phobos**. Kolejnym krokiem będzie wylistowanie zawartości katalogu domowego. Intruz powinien teraz rozpocząć atak **Denial of Service** na serwer **phobos**. Aby to zasymulować odłączymy fizycznie serwer od sieci.

EKRAN KLIENTA

```
ftp> dir
---> PORT 212,14,1,69,16,219
```

- (2) Ze wcześniejszych doświadczeń wynika, że w tym momencie klient utworzył dla serwera port dla połączenia danych. Obliczamy go na podstawie polecenia PORT:

$16*256+219=4316$. Wiemy wiec, ze intruz musi polaczyc sie z portem 4316 komputera rubycon.

EKRAN INTRUZA

```
telnet rubycon 4316
Trying 212.14.1.69...
Connected to rubycon.man.szczecin.pl.
Escape character is '^]'.
```

- (3) Jak widac rubycon.man.szczecin.pl przyjal polaczenie od nieznanego komputera. Mozemy juz podlaczyc serwer **phobos** do sieci.

EKRAN KLIENTA

```
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for file
list.
```

- (4) Serwer akceptuje odebranie polecenia PORT i stara sie polaczyc z portem nr 4316 klienta. Z tym portem jest juz jednak zestawione polaczenie wykonane z komputera intruza. Oprogramowanie nie zgłasza jednak zadnego bledu.
- (5) Sprawdzmy teraz czy intruz jest w stanie wyslac klientowi jakiegokolwiek dane.

EKRAN INTRUZA

```
Ten tekst wpisuje Intruz na swoim komputerze.  
Nastepnie przerywa polaczenie.
```

EKRAN KLIENTA

```
150 Opening ASCII mode data connection for file  
list.  
Ten tekst wpisuje Intruz na swoim komputerze.  
Nastepnie przerywa polaczenie.  
226 Transfer complete.  
ftp>
```

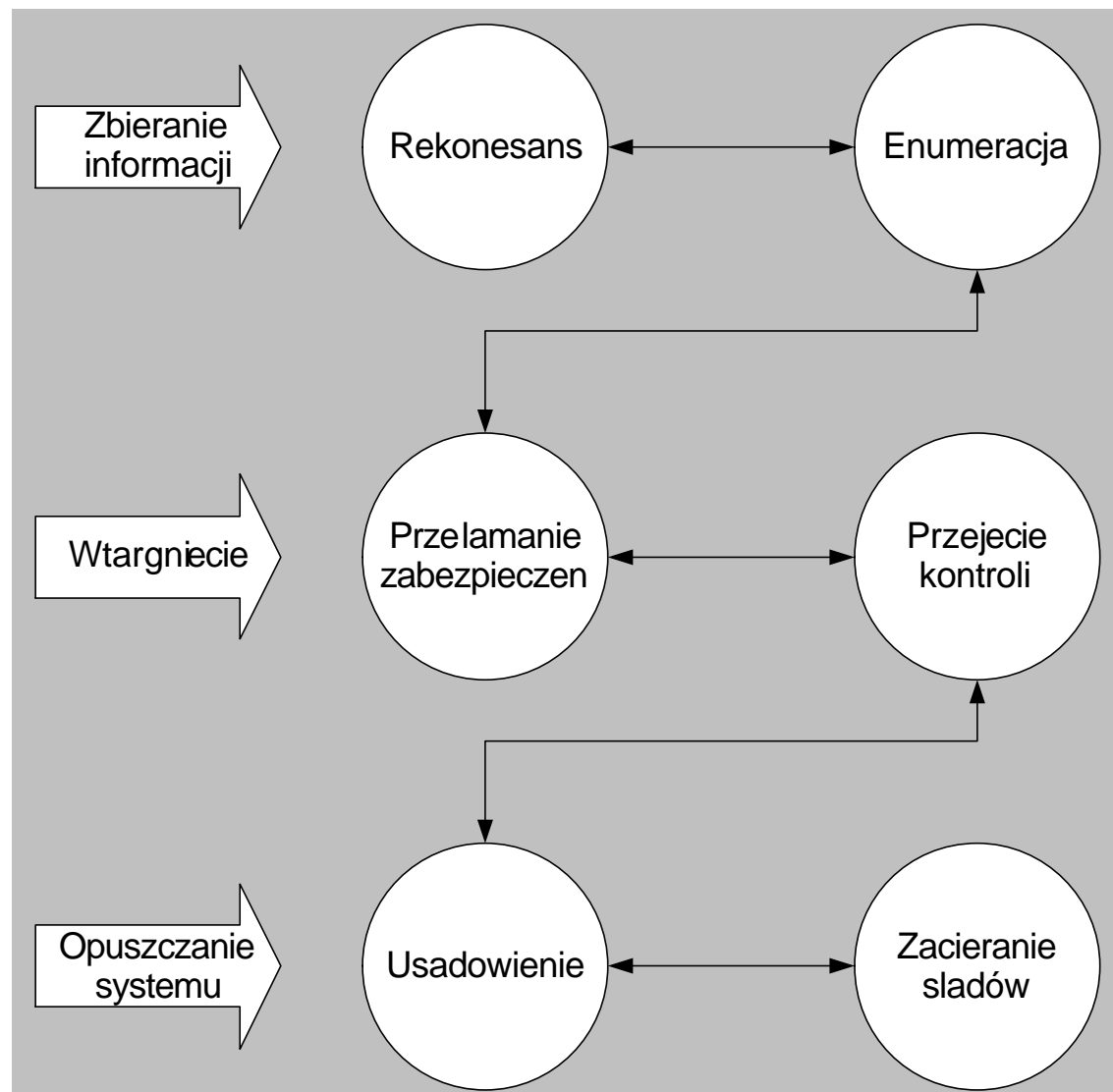
- (6) Przeprowadziliśmy wiec udany atak na protokól FTP. Intruzowi udalo sie podstawic dowolny tekst zamiast wyniku dzialania polecenia dir. W sytuacji, gdyby polaczenie to bylo wykonywane w trybie pasywnym, intruz musialby polaczyc sie z odpowiednim portem na serwerze **phobos**. W takim przypadku w wyniki ataku otrzymalby informacje o zawartosci katalogu domowego uzytkownika.

Anatomia ataku

- Jest to droga, jaka powinien przejść intruz, aby skutecznie zaatakować chroniony system. Poprzez akcje będziemy rozumieli czynność o pewnej złożoności, realizującą jakiś cel. Możemy wyróżnić następujące akcje
 - **Próbkowanie** – próba dostępu do obiektu poprzez zbadanie jego charakterystyki.
 - **Skanowanie** – próba dostępu do wielu obiektów na raz poprzez ustalenie obiektu z oczekiwaną charakterystyką.
 - **Przepelnienie** – dostęp do obiektu poprzez nagłe przepelnienie jego możliwości przetwarzania.
 - **Uwierzytelnienie** – przedstawienie się jako osoba uprawniona oraz w razie konieczności przekazanie informacji potrzebnej do poprawnego uwierzytelnienia.
 - **Ominiecie** – ominiecie procesu zabezpieczającego poprzez zastosowanie alternatywnej drogi osiągnięcia obiektu.
 - **Podszywanie** – przedstawianie się, w trakcie połączenia sieciowego, jako użytkownik posiadający prawo dostępu do zasobów
 - **Czytanie** – dostęp z prawami czytania do informacji przez osobę nieuprawnioną.
 - **Kopiowanie** – dostęp z możliwością kopiowania do informacji przez osobę nieuprawnioną.

- **Kradzież** – przejęcie zasobów przez osobę nieuprawnioną bez pozostawienia kopii w uprawnionej lokalizacji.
 - **Modyfikacja** – zmian zawartości lub charakterystyki obiektu ataku.
 - **Usunięcie** – usunięcie (zniszczenie) obiektu ataku.
- Dodatkowo do przedstawionych akcji należy dodać dla opisu ataków dwa pojęcia:
- **Rekonesans** – nieinwazyjne zbieranie wszelkich informacji o danej organizacji. Sprowadzenie nazwy organizacji do konkretnego zakresu nazw domen, adresów IP komputerów bezpośrednio podłączonych do Internetu, bloków sieci.
 - **Enumeracja** – inwazyjne zbieranie informacji na temat zasobów sieciowych i ich udostępnieniu, użytkowników grup oraz aplikacji i etykiet; enumeracja wiąże się z aktywnymi połączeniami z hostami i ukierunkowanymi zapytaniami; z tego powodu intruz powinien zostać zauważony, a jego działania odnotowane.

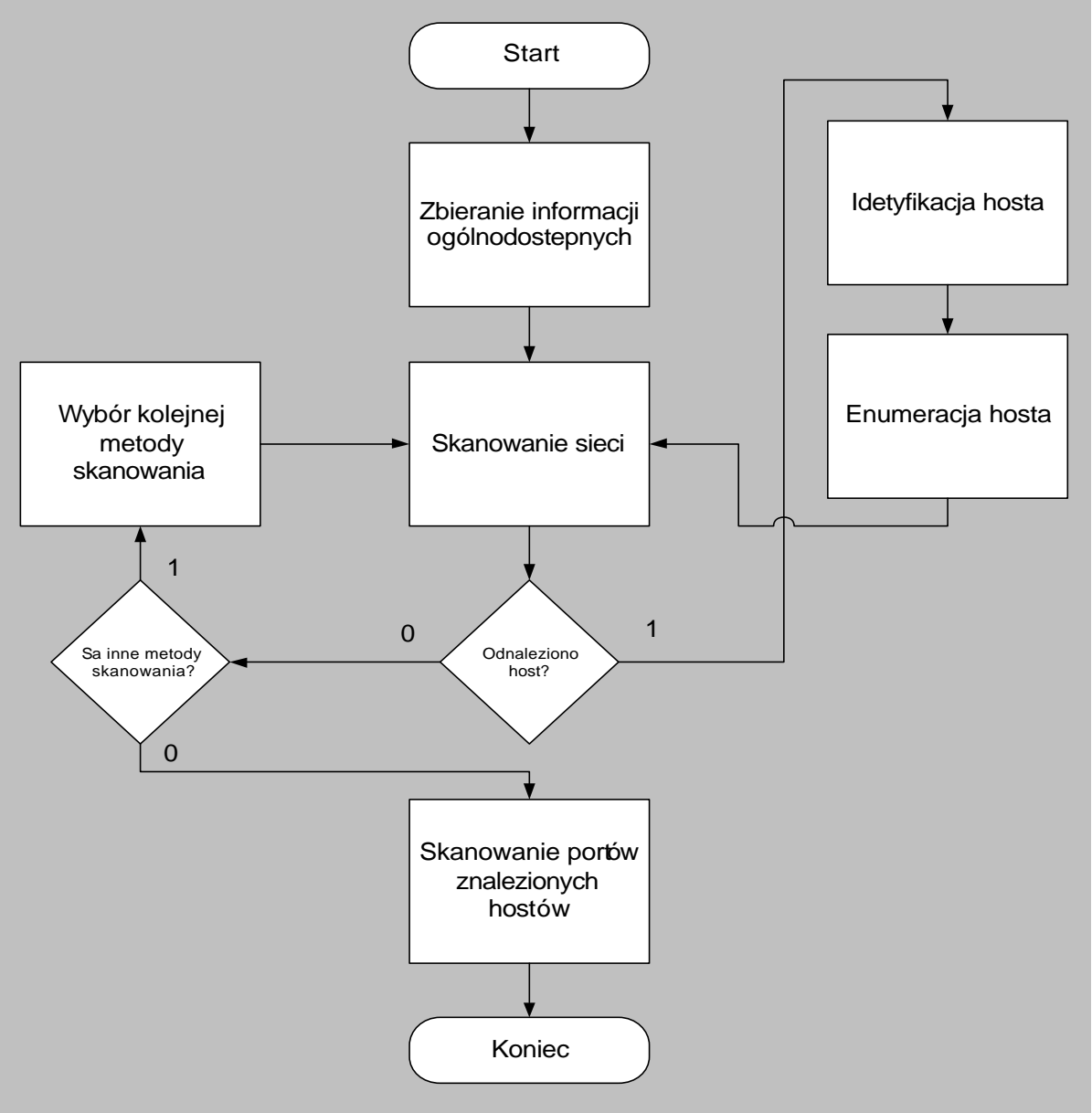
Etapy ataku na siec



- **Zbieranie informacji** – zdobycie maksymalnej ilości informacji o systemie jak i o organizacji, której jest właścicielem. Pozwoli to na zaplanowanie ataku i wyznaczenie słabych punktów systemu.
- **Wtargnięcie** – na tym etapie dokonuje się prób wejścia do systemu i rozszerzenia kontroli, czyli zdobycia maksymalnych uprawnień w zdobytym systemie.
- **Opuszczanie systemu** – na tym etapie po udanym ataku intruz przeprowadza modyfikacje zdobytego systemu tak, aby móc się tam dostać kolejny raz bez konieczności przelamywania zabezpieczeń. Kolejnym elementem tego etapu jest usunięcie widocznych śladów bytności intruza w systemie, czyli na przykład usunięcie pozycji logów świadczących włamaniu.

Zbieranie informacji

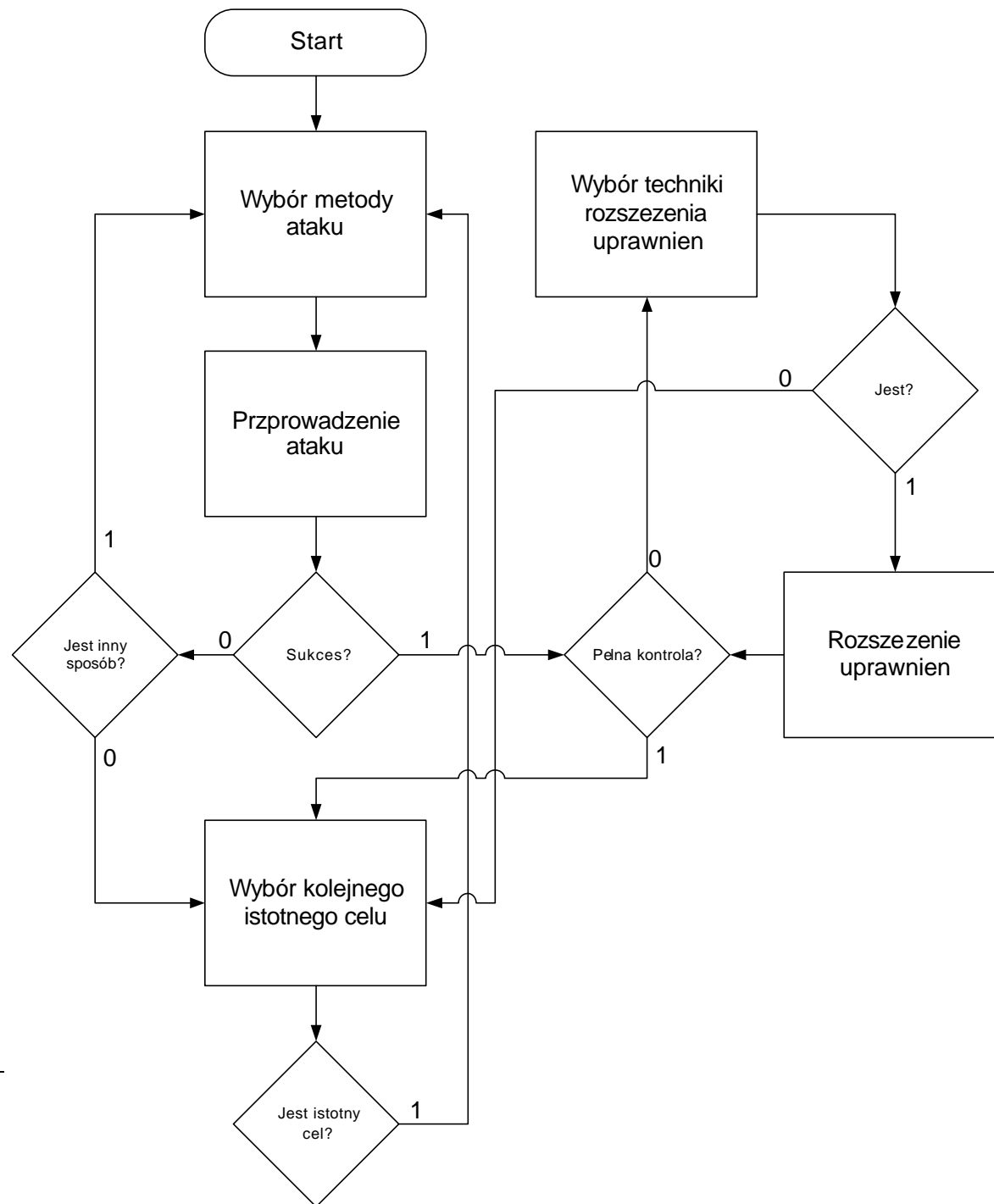
- Zbieranie informacji jest ważnym elementem każdego ataku. Może być realizowane poprzez zasięgnięcie informacji z ogólnie dostępnych źródeł takich jak strona WWW organizacji, wszelkie wyszukiwarki, itp. Zbieranie informacji od rejestratorów domen pozwoli na poznanie nazw domen kontaktów administracyjnych, daty utworzenia, serwery DNS, zakres przyznanych adresów IP.



- W dalszym etapie można przystąpić do zbierania informacji fizycznych o sieci poprzez jej skanowanie. Skanowanie sieci pozwoli określić jej topologię aktywne i widoczne hosty, urządzenia sieciowe, czyli potencjalne cele ataku.
- Dalej przeprowadzona zostaje głębsza analiza obiektu. Polega ona na skanowaniu wykrytych hostów, dokonania enumeracji, próbkowania. Intruz zaopatrzony w takie informacje może przystąpić do rozpoczęcia ataku.

Atak, wtargnięcie

- Intruz uzbrojony w informacje może przystąpić do wyboru technik i akcji, jakich użyje do przeprowadzenia ataku oraz osiągnięcia głównego celu wtargnięcia.
- W momencie użycia dostępnej techniki może się okazać, że system nie jest na nią podatny, należy dobrać więc inny sposób ataku.
- Po uzyskaniu dostępu do obiektu ataku może zaistnieć sytuacja, że poziom uzyskanego dostępu jest niewystarczający, więc należy spróbować rozszerzyć przywileje dostępu.
- Możliwe, że nie jest konieczna dokonanie złamania systemu hosta a wystarczające jest jego ominięcie.
- Procedura ta jest powtarzana aż napastnik przejmie kontrolę nad najbardziej istotnymi obiektami danej sieci. Do tej kategorii należy zaliczyć systemy odpowiedzialne za ochronę sieci routery, hosty bastionowe, elementy systemu firewall itp. oraz systemy interesujące z uwagi na cel ataku np. chronione wewnętrzne serwery baz danych atakowanej organizacji.



- Istotnym elementem tego etapu jest wybór odpowiednich celów, które będą umożliwiały uzyskanie dostępu do chronionej sieci. Ponieważ atak obarczony dużym ryzykiem wykrycia musi być przeprowadzony relatywnie szybko, delikatnie i bez zbędnych akcji typu włamania, na hosta, który można ominąć lub łamania zabezpieczenia, które także można ominąć.

Opuszczanie systemu

- Odchodząc intruzy stara się zabezpieczyć ewentualny powrót do zdobytego systemu z pominięciem walki z zabezpieczeniami.
- W tym celu, o ile to możliwe, zakładane są wszelkiego rodzaju tylne wejścia (backdoor), zmieniana konfiguracja zdobytych hostów itp.
- Ostatnim etapem jest usunięcie śladów bytności intruzy głównie w systemach monitorujących, czyli usuwanie wpisów z logów systemowych o zaistniałym włamaniu.
- W tym celu należy odnaleźć wszelkie miejsca gdzie logi są składowane czy w systemach zdobytych, czy też w innych dedykowanych systemach gdzie te informacje są składowane.

Rodzaje ataków.

- W poprzednich rozdziałach przy przeprowadzaniu klasyfikacji dokonano opisu pewnych typów ataków tutaj zostaną wymienione najważniejsze rodzaje ataków:

- (a) Ataki przeprowadzane kanałem poleceń (command-channel attacks) – są to ataki skierowane bezpośrednio na serwer danej usługi, wysyłając mu polecenia przez jego zwykły kanał poleceń. Istnieją dwa podstawowe typy tego ataku:
- (b) Wysyłanie legalnych danych, aby wywołać niepożądany skutek,
- (c) Wysyłanie błędnych danych i wykorzystanie usterek w obsłudze błędnych danych wejściowych.
- (d) Ataki wykorzystujące dane (data-driven attack) – jest to grupa ataków mających związek z danymi przesyłanymi przez protokół nie zaś z serwerem implementującym ten protokół. Można wyróżnić następujące typy tych ataków:
 - (e) Ataki mające związek z wrogimi danymi,
 - (f) Ataki naruszające legalne dane.
 - (g) Ataki na usługi trzecie (third-party attacks) – nie wykorzystują one usług, które mają być udostępnione, ale wykorzystują luki, które zostały otwarte w celu udostępnienia jednej usługi, aby zaatakować zupełnie inną.
 - (h) Sfałszowanie uwierzytelnienia klientów (false authentication) – intruz rozbija mechanizm uwierzytelnienia i jest postrzegany jako jeden z legalnych użytkowników.
 - (i) Przejmowanie sesji (hijacking) – w tym ataku intruz przejmuje otwartą sesję terminala lub logowania od użytkownika uwierzytelnionego i autoryzowanego przez system. Ataki tego

typu są bardzo trudne do przeprowadzenia jednak powiedzenie się takiego ataku przynosi intruzowi wiele zysku.

- (j) Przechwytywanie pakietów (sniffing) – polega to na przechwytywaniu przepływających siecią pakietów, co umożliwi zdobycie niezaszyfrowanych lub łatwych do rozszyfrowania danych.
 - (k) Wprowadzanie i modyfikacja danych – intruz, który zdoła przejąć połączenie może dokonać zmian w przepływających przez nie danych. Intruz kontrolujący router między klientem i serwerem, może nie tylko przechwycić pakiet i odczytać go, ale również dokonać jego modyfikacji. W rzadszych przypadkach zdoła to zrobić bez kontroli nad routerem, wysyłając zmodyfikowany pakiet tak, aby przybył przed oryginalnym pakietem.
 - (l) Odtworzenie (replays) – intruz niemożący zmienić danych ani przejąć połączenia może rejestrować przepływające dane i wysłać je ponownie.
 - (m) Blokada usług (denial of service) – jest to jeden z typów ataków, którym nie udało się całkowicie zapobiec. Polega on próbie przeciążenia usługi a dalej serwera, zbyt dużą ilością zadań (np. ilość połączeń).
- Przedstawiona klasyfikacja rodzajów ataków pozwala na przyporządkowanie konkretnych istniejących ataków do którejś z grup.

