

Metody i techniki rekonesansu

Jak już powiedziano, pierwszym krokiem realizowanym przez ewentualnego napastnika jest zbieranie informacji o celu przyszłego ataku. Fazę tę można nazwać rekonesansem. Pozwala ona agresorom na utworzenie pełnego lub częściowego profilu jej zabezpieczeń. Jest to chyba najbardziej pracowity element badania zabezpieczeń. Co może zidentyfikować agresor?

- nazwę domeny,
- bloki sieci,
- adresy IP komputerów osiągalnych poprzez usługi działające na zidentyfikowanych komputerach,
- architekturę i zainstalowany system operacyjny,
- mechanizmy kontroli dostępu,
- systemy wykrywania intruzów i zapory sieciowe,
- używane protokoły,
- numery linii telefonicznych,
- mechanizmy autoryzacji dla zdalnego dostępu.

Krok 1, to przeszukiwanie ogólnie dostępnych źródeł, takich jak:

- strony www,
- artykuły i informacje prasowe,
- listy dyskusyjne,
- serwisy wyszukiwawcze.

Wiele informacji można czasami znaleźć w komentarzach w kodzie źródłowym strony www. Często można tam znaleźć:

- informacje o lokalizacji,
- powiązane firmy i jednostki organizacyjne,
- informacje o przejęciach i fuzjach,
- numery telefonów,
- adresy kontaktowe i adresy e-mail,
- informacje o polityce prywatności i zabezpieczeń,
- łącza do innych serwerów powiązanych z organizacją.

Do identyfikacji nazw domen i sieci związanych z daną organizacją można wykorzystywać bazy danych **whois**. Większość informacji potrzebnych agresorom można uzyskać poprzez zapytania:

- o rejestratora,
- o organizację,
- o domenę,
- o sieć,
- o kontakt.

Niektóre bardziej znane serwisy *whois*, to:

- www.allwhois.com
- www.arin.net - American Registry for Internet Numbers
- www.samspace.org - SamSpace
- www.apnic.net - Asia-Pacific Network Information Center
- www.ripe.net - Reseaux IP Europeens
- www.dns.pl - Naukowa i Akademicka Sieć Komputerowa

Przeciwdziałanie polega przede wszystkim na usunięciu wszystkich informacji, które mogłyby pomóc w zdobyciu dostępu do naszej sieci. Warto zajrzeć do RFC 2196 - *Site Security Handbook*.

W kolejnym kroku powinna mieć miejsce kontrola serwerów DNS. Jednym z najpoważniejszych błędów jakie może popełnić administrator systemu, jest umożliwienie nieautoryzowanym użytkownikom na dokonanie przesłania strefy serwera DNS. Takie przesłanie umożliwia serwerowi zapasowemu uaktualnienie swojej bazy i jest ono dla nich niezbędne. Niektóre serwery udostępniają kopię strefy każdemu, kto o nią poprosi. Poważny problem występuje wtedy, gdy organizacja nie używa DNS do segregowania informacji na wewnętrzne i zewnętrzne. Udostępnienie informacji o wewnętrznych adresach IP można porównać do udostępnienia pełnego planu sieci wewnętrznej. Można do tego wykorzystać program **nslookup**. W sieci można znaleźć również inne narzędzia umożliwiające przeprowadzenie takiego badania.

W rekordach HINFO możemy znaleźć opis platformy programowo sprzętowej. Niekiedy będzie tam również informacja, że są to systemy testowe (zwykle słabo zabezpieczone). Rekordy MX określają serwery pocztowe.

Przeciwdziałanie polega na umożliwieniu przesyłania informacji o strefie jedynie autoryzowanym serwerom. Informacje o tym można znaleźć w dokumentacji określonych serwerów. Dodatkowo należy rozdzielić serwery DNS na wewnętrzne i zewnętrzne.

Kolejny krok to tzw. badanie sieci. W tym kroku następuje próba określenia topologii sieci oraz potencjalnych ścieżek dostępu do nich. Można użyć programu **traceroute** (UNIX) lub **tracert** (Windows). Są również podobne programy udostępniające interfejs graficzny, np. **Visual Route** lub **Neo Trace**. Narzędzia te umożliwiają poznanie ścieżki pokonywanej przez pakiet w drodze do komputera docelowego.

Przeciwdziałanie polega na zastosowaniu odpowiedniego systemu wykrywania włamań, blokującego omówione żądania. Można skonfigurować graniczne routery tak aby ograniczały ruch pakietów ICMP i UDP do konkretnych komputerów.

3. Techniki skanowania

W tej chwili technika skanowania budzi w wielu środowiskach zastrzeżenia co do jej legalności. Poniżej przytaczam fragment wiadomości umieszczonej na witrynie CERT Polska.

Pod koniec roku 2000, jeden z lokalnych sądów w Stanach Zjednoczonych uznał, że skanowanie portów komputerowych nie jest niezgodne z prawem, pod warunkiem oczywiście że nie wyrządza szkody. Sąd przychylił się do głosu obrony i uznał, że czas spędzony na rozpatrywaniu przypadku skanowania sieci, czy komputera, nie może być wzięty pod uwagę przy określeniu poniesionych strat finansowych. Strata może być uznana tylko wtedy jeśli następuje naruszenie integralności i dostępności sieci. "Jest to dobra decyzja dla naukowców związanych z bezpieczeństwem teleinformatycznym" - stwierdził obrońca oskarżonego.

Decyzja wydaje się być kontrowersyjna. Wszelkie klasyfikacje przypadków naruszenia bezpieczeństwa teleinformatycznego zawierają przypadek skanowania sieci, czy pojedynczego komputera. Oczywiście, rzadko kiedy dochodzi w wyniku samego skanowania do naruszenia bezpieczeństwa, chociaż nie jest to wykluczone. Jest to jednak niezaprzeczalnie sposób na zebranie informacji, która w rezultacie może posłużyć do dokonania zasadniczego włamania. Co więcej - to właśnie włamanie może nie zostać wykryte właśnie dzięki wcześniejszemu skanowaniu. Wątpliwe jest również określenie tej decyzji jako korzystnej dla naukowców. Poważni naukowcy tego typu eksperymenty dokonują w laboratoriach.

3.1. Cele skanowania

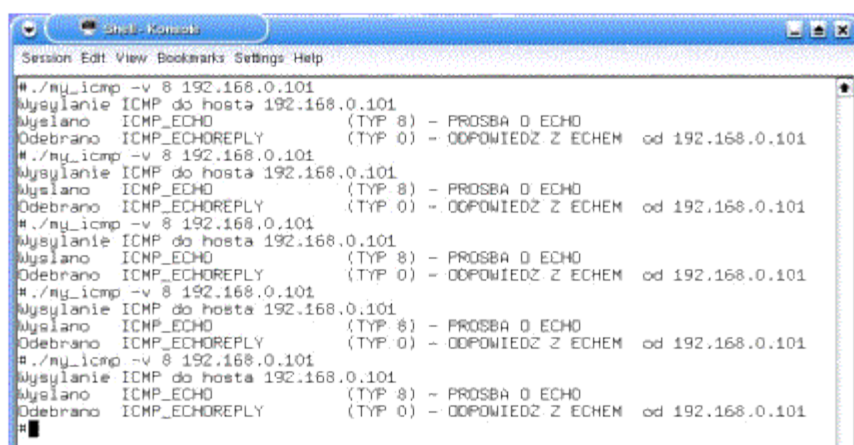
Skanowanie jest powszechnie stosowaną metodą zdalnego wykrywania komputerów i usług udostępnianych przez te komputery. Metoda ta polega na próbkowaniu aktywności badanego komputera, poprzez wysyłanie do niego specjalnie spreparowanych pakietów i oczekiwaniu na odpowiedź. Po odebraniu odpowiedzi przystępujemy do jej interpretacji. Niekiedy również brak odpowiedzi niesie dla skanującego informację odnośnie aktywności badanego komputera lub usługi.

Skanowanie pełni rolę wywiadu, który dostarcza informacji o zdarzeniach i urządzeniach w sieci. Pozwala stwierdzić, które urządzenia i serwisy sieciowe działają, a które nie - co niejednokrotnie jest informacją równie istotną. Można zdalnie określić czy dany komputer jest aktywny, rozpoznać uruchomione na nim serwisy oraz system operacyjny.

Skanowanie może pomóc również w rozpoznaniu topologii sieci i konfiguracji urządzeń dostępowych (np.: list kontroli dostępu, tablic rutowania). Jest ono wykorzystywane przez administratorów do rozwiązywania problemów z siecią, jak również przez intruzów w celach rozpoznawczych.

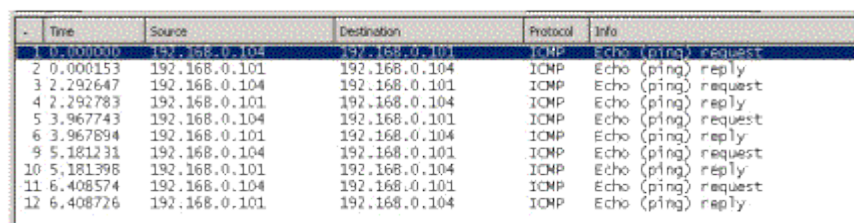
3.2. Skanowanie ICMP

Najprostszą, najczęściej stosowaną ale i coraz mniej skuteczną metodą skanowania jest wysłanie pakietu *ICMP echo request*, czyli popularnego *pinga*. Na tej podstawie można stwierdzić czy docelowe urządzenie jest osiągalne. Brak odpowiedzi nie świadczy jednak o tym, że komputer nie jest nieosiągalny. Powodów braku odpowiedzi może być wiele: zaporą ogniową filtrująca pakiety ICMP, wyłączony serwis na docelowym komputerze i wiele innych. Na rys. 1 przedstawiono przykładowy raport uzyskiwany z programu skanującego, a na rys. 2, obraz ruchu sieciowego związanego z tym rodzajem skanowania.



```
Session Edit View Bookmarks Settings Help
#./nmap_icmp -v 8 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.101
#./nmap_icmp -v 8 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.101
#./nmap_icmp -v 8 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.101
#./nmap_icmp -v 8 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.101
#./nmap_icmp -v 8 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.101
#
```

Rys. 1. Raport skanera wysyłającego pakiety *ICMP echo request*



Time	Source	Destination	Protocol	Info
0.000000	192.168.0.104	192.168.0.101	ICMP	Echo (ping) request
0.000153	192.168.0.101	192.168.0.104	ICMP	Echo (ping) reply
2.292647	192.168.0.104	192.168.0.101	ICMP	Echo (ping) request
2.292783	192.168.0.101	192.168.0.104	ICMP	Echo (ping) reply
3.967743	192.168.0.104	192.168.0.101	ICMP	Echo (ping) request
3.967894	192.168.0.101	192.168.0.104	ICMP	Echo (ping) reply
5.181231	192.168.0.104	192.168.0.101	ICMP	Echo (ping) request
5.181398	192.168.0.101	192.168.0.104	ICMP	Echo (ping) reply
6.408574	192.168.0.104	192.168.0.101	ICMP	Echo (ping) request
6.408726	192.168.0.101	192.168.0.104	ICMP	Echo (ping) reply

Rys. 2. Ruch sieciowy podczas skanowania pakietami *ICMP echo request*

Zdarza się, że na maszynie filtrującej ruch blokowane są wyłącznie pakiety *ICMP Echo Request/Reply*. W takim przypadku można próbować wysłać pakiety *Timestamp Request* (ICMP - typ 13) albo *Address Mask Request* (ICMP - typ 17). Są to zapytania kontrolne ICMP, na które docelowy komputer może odpowiedzieć. W pierwszym przypadku będzie to aktualny czas obowiązujący na zdalnej maszynie. Drugi przypadek to sytuacja, w której bezdyskowa stacja robocza pobiera maskę podsieci w czasie startu. Komunikaty te można wysłać wykorzystując narzędzia takie jak *icmpush* oraz *icmquery*. W trakcie badań wykorzystano własny program autora. Na rys. 3 i 4 przedstawiono raporty programu skanującego. Na rys. 5 i 6 można zobaczyć obraz ruchu sieciowego związanego z obydwojema metodami skanowania. Podczas skanowania przy pomocy pakietów *ICMP Address Mask Request* nie uzyskano odpowiedzi ani od systemów Windows ani od systemu Linux.

```

Shell - Konsola
Session Edit View Bookmarks Settings Help
#./my_icmp -v 13 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_TIMESTAMP (TYP 13) - PROSBA O CZAS
Odebrano ICMP_TIMESTAMPREPLY (TYP 14) - ODPOWIEDZ Z CZASEM od 192.168.0.101
#./my_icmp -v 13 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_TIMESTAMP (TYP 13) - PROSBA O CZAS
Odebrano ICMP_TIMESTAMPREPLY (TYP 14) - ODPOWIEDZ Z CZASEM od 192.168.0.101
#./my_icmp -v 13 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_TIMESTAMP (TYP 13) - PROSBA O CZAS
Odebrano ICMP_TIMESTAMPREPLY (TYP 14) - ODPOWIEDZ Z CZASEM od 192.168.0.101
#./my_icmp -v 13 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_TIMESTAMP (TYP 13) - PROSBA O CZAS
Odebrano ICMP_TIMESTAMPREPLY (TYP 14) - ODPOWIEDZ Z CZASEM od 192.168.0.101
#./my_icmp -v 13 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_TIMESTAMP (TYP 13) - PROSBA O CZAS
Odebrano ICMP_TIMESTAMPREPLY (TYP 14) - ODPOWIEDZ Z CZASEM od 192.168.0.101
#

```

Rys. 3. Raport skanera wysyłającego pakiety *ICMP Timestamp Request*

-	Time	Source	Destination	Protocol	Info
3	0.001677	192.168.0.104	192.168.0.101	ICMP	Timestamp request
4	0.001831	192.168.0.101	192.168.0.104	ICMP	Timestamp reply
5	1.475592	192.168.0.104	192.168.0.101	ICMP	Timestamp request
6	1.475744	192.168.0.101	192.168.0.104	ICMP	Timestamp reply
7	2.464171	192.168.0.104	192.168.0.101	ICMP	Timestamp request
8	2.464326	192.168.0.101	192.168.0.104	ICMP	Timestamp reply
9	3.488953	192.168.0.104	192.168.0.101	ICMP	Timestamp request
10	3.489102	192.168.0.101	192.168.0.104	ICMP	Timestamp reply
11	4.587850	192.168.0.104	192.168.0.101	ICMP	Timestamp request
12	4.588009	192.168.0.101	192.168.0.104	ICMP	Timestamp reply

Rys. 4. Ruch sieciowy podczas skanowania pakietami *ICMP Timestamp Request*

```

Shell - Konsola
Session Edit View Bookmarks Settings Help
#./my_icmp -v 17 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_ADDRESS (TYP 17) - PROSBA O MASKE PODSIECI
BRAK KOMPLETU ODPOWIEDZI (timeout)
#./my_icmp -v 17 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_ADDRESS (TYP 17) - PROSBA O MASKE PODSIECI
BRAK KOMPLETU ODPOWIEDZI (timeout)
#./my_icmp -v 17 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_ADDRESS (TYP 17) - PROSBA O MASKE PODSIECI
BRAK KOMPLETU ODPOWIEDZI (timeout)
#./my_icmp -v 17 192.168.0.101
Wysyłanie ICMP do hosta 192.168.0.101
Wysłano ICMP_ADDRESS (TYP 17) - PROSBA O MASKE PODSIECI
BRAK KOMPLETU ODPOWIEDZI (timeout)
#

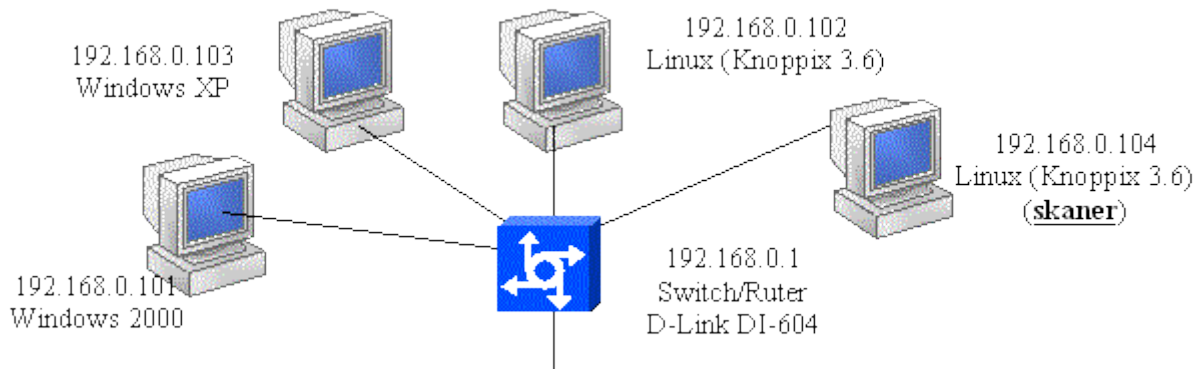
```

Rys. 5. Raport skanera wysyłającego pakiety *ICMP Address Mask Request*

-	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.104	192.168.0.101	ICMP	Address mask request
4	8.785415	192.168.0.104	192.168.0.101	ICMP	Address mask request
5	16.109807	192.168.0.104	192.168.0.101	ICMP	Address mask request
6	23.403163	192.168.0.104	192.168.0.101	ICMP	Address mask request

Rys. 6. Ruch sieciowy podczas skanowania pakietami *ICMP Address Mask Request*

Można również wysłać pakiety ICMP na adres rozgłoszeniowy sieci. Pakiety takie będą jednak prawdopodobnie ignorowane przez odbiorców pracujących z systemami Windows. Powodem jest możliwość nadużycia, którą wykorzystuje jeden ze znanych ataków DoS- SMURF). Wyniki takiego skanowania przedstawiono na rys. 8 i 9. Skanowanie przeprowadzono w sieci, której topologię przedstawiono na rys. 7.



Rys. 7. Topologia sieci wykorzystanej w skanowaniu pakietami rozgłoszeniowymi

```

Shell - Konsola
Session Edit View Bookmarks Settings Help
# ./my_icmp -v8 255.255.255.255
Wysylanie ICMP do hosta 255.255.255.255
Wyslano ICMP_ECHO (TYP 8) - PROSBA O ECHO
Odebrano ICMP_ECHO (TYP 8) - PROSBA O ECHO od 192.168.0.104
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.104
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.102
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.1

# ./my_icmp -v8 255.255.255.255
Wysylanie ICMP do hosta 255.255.255.255
Wyslano ICMP_ECHO (TYP 8) - PROSBA O ECHO
Odebrano ICMP_ECHO (TYP 8) - PROSBA O ECHO od 192.168.0.104
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.104
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.102
Odebrano ICMP_ECHOREPLY (TYP 0) - ODPOWIEDZ Z ECHEM od 192.168.0.1
  
```

Rys. 8. Raport skanera wysyłającego pakiety ICMP echo request na adres rozgłoszeniowy

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.104	255.255.255.255	ICMP	Echo (ping) request
2	0.000274	192.168.0.102	192.168.0.104	ICMP	Echo (ping) reply
3	0.000401	192.168.0.1	192.168.0.104	ICMP	Echo (ping) reply
6	8.427390	192.168.0.104	255.255.255.255	ICMP	Echo (ping) request
7	8.427654	192.168.0.102	192.168.0.104	ICMP	Echo (ping) reply
8	8.427795	192.168.0.1	192.168.0.104	ICMP	Echo (ping) reply

Rys. 9. Ruch sieciowy podczas skanowania pakietami ICMP echo request wysłanymi na adres rozgłoszeniowy

Podczas eksperymentu, zgodnie z oczekiwaniami, systemy Windows nie odpowiedziały na wysłane pakiety. Systemy Linuks, oraz ruter zgłosiły swoją obecność wysyłając odpowiedzi do skanera. Na rys.9 nie widać pakietów odpowiedzi wysyłanych przez system 192.168.0.104 do samego siebie. Wynika to ze sposobu funkcjonowania snifera.

3.3. Skanowanie TCP

Pewne cechy protokołu TCP sprawiają, że jest on bardziej przydatny do skanowania niż np. protokół UDP. Niektóre techniki skanowania, w tym skanowanie z ukryciem tożsamości skanującego, wykorzystują cechę zorientowania na połączenia (*connection-oriented*) protokołu TCP. W skanowaniu ważne może być również śledzenie numerów sekwencyjnych oraz odpowiedzi systemu po otrzymaniu pakietu TCP z włączonymi określonymi flagami. Z reguły stosowane są pakiety nie zawierające danych, gdyż ważny jest fakt, czy zdalny system odpowiedział, a nie zawartość pola danych pakietu. Pewne techniki wykorzystują fragmentację pakietów w warstwie sieciowej, które pozwalają ukryć nagłówek TCP w kilku pakietach IP utrudniając detekcję skanowania.

Skanowanie połączeniowe

Najprostszą techniką skanowania portów z wykorzystaniem TCP jest metoda połączeniowa (*TCP connect*). Nazwa jej pochodzi od systemowej funkcji *connect()*, która służy do nawiązania pełnego połączenia ze zdalnym portem. Jeśli w fazie nawiązywania połączenia serwer odpowie

pakiem z flagami SYN/ACK znaczy to, że port jest otwarty w trybie nasłuchu. Pakeć z flagami RST/ACK indykuje zamknięty port. Skanowanie kończy wysłanie pakieta z flagą RST. Wadą tej metody jest łatwość jej wykrycia i zablokowania. Zalety to szybkość oraz możliwość wykonania w przez każdego użytkownika.

```

C:\> Wiersz polecenia

E:\BSI>nmap -sT -p445 192.168.0.105

Starting nmap V. 3.00 < www.insecure.org/nmap >
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
445/tcp   open       microsoft-ds

Nmap run completed -- 1 IP address (1 host up) scanned in 9 seconds

E:\BSI>nmap -sT -p135 192.168.0.105

Starting nmap V. 3.00 < www.insecure.org/nmap >
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
135/tcp   open       loc-srv

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds

E:\BSI>_

```

Rys. 10. Raport skanera *nmap* wykrywającego otwarte porty 445 i 135 (metoda połączeniowa)

Source	Destination	Protocol	Info
192.168.0.101	192.168.0.105	TCP	445 > 3781 [SYN, ACK] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	3781 > 445 [ACK] Seq=1 Ack=1 Win=17520 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	3781 > 445 [RST] Seq=1 Ack=1 Win=0 Len=0
192.168.0.101	192.168.0.105	TCP	3782 > 135 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	135 > 3782 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	3782 > 135 [ACK] Seq=1 Ack=1 Win=65535 Len=0
192.168.0.101	192.168.0.105	TCP	3782 > 135 [RST] Seq=1 Ack=1 Win=0 Len=0

Rys.11. Ruch sieciowy podczas wykrywania przez skaner *nmap* otwartych portów 445 i 135 (metoda połączeniowa)

```

C:\> Wiersz polecenia

E:\BSI>nmap -sT -p25 192.168.0.105

Starting nmap V. 3.00 < www.insecure.org/nmap >
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
25/tcp    filtered   smtp

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

E:\BSI>nmap -sT -p21 192.168.0.105

Starting nmap V. 3.00 < www.insecure.org/nmap >
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
21/tcp    filtered   ftp

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

E:\BSI>

```

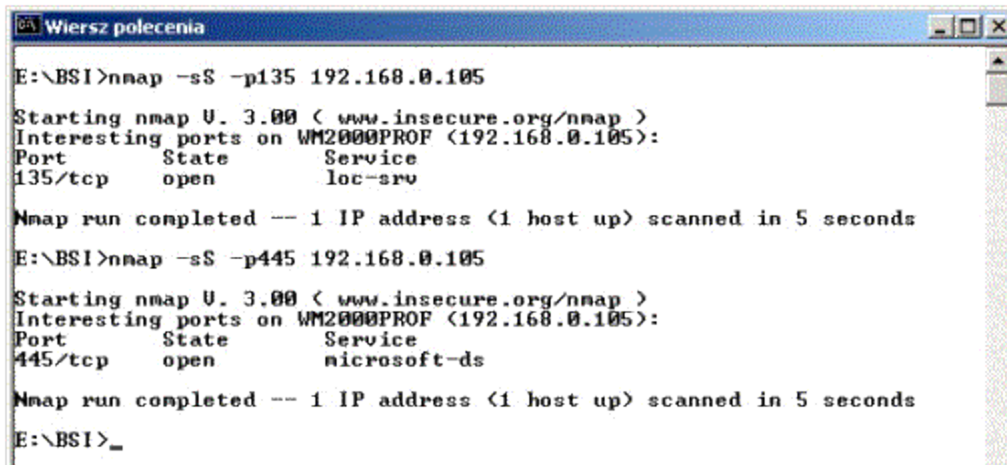
Rys. 12. Raport skanera *nmap* wykrywającego zamknięte porty 21 i 25 (metoda połączeniowa)

Source	Destination	Protocol	Info
192.168.0.101	192.168.0.105	TCP	3822 > 25 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	25 > 3822 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.101	192.168.0.105	TCP	3821 > 25 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	25 > 3821 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
192.168.0.101	192.168.0.105	TCP	3823 > 21 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	21 > 3823 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.101	192.168.0.105	TCP	3824 > 21 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	21 > 3824 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

Rys.13. Ruch sieciowy podczas wykrywania przez skaner *nmap* zamkniętych portów 21 i 25 (metoda połączeniowa)

Skanowanie półotwarte

Nietrudno zaobserwować, że system docelowy dostarcza informacji o statusie portu już w trakcie trwania procesu nawiązywania połączenia po nadesłaniu odpowiedzi na pakiet SYN. Spostrzeżenie to wykorzystuje technika półotwarcia. Polega ona na wysłaniu pakietu RST zaraz po otrzymaniu w drugiej fazie połączenia pakietu SYN/ACK lub RST/ACK. Swego czasu zaletą tej metody była jej utrudniona wykrywalność co znalazło swój wyraz w pierwotnej nazwie (*TCP SYN stealth*). Teraz nie jest to już prawdą. Technika ta zbliżona jest do ataku *DoS SYN Flood*. Dlatego też jest często wykrywana przez systemy IDS lub odfiltrowywana na bramkach dostępowych. Wadą tej metody jest konieczność posiadania uprawnień superużytkownika w systemie Linux. Potrzebne są bowiem uprawnienia do tworzenia tzw. gniazd surowych (*raw socket*) - zastrzeżone dla administratora. Detekcja portów zamkniętych przebiega tak samo jak w metodzie połączeniowej.

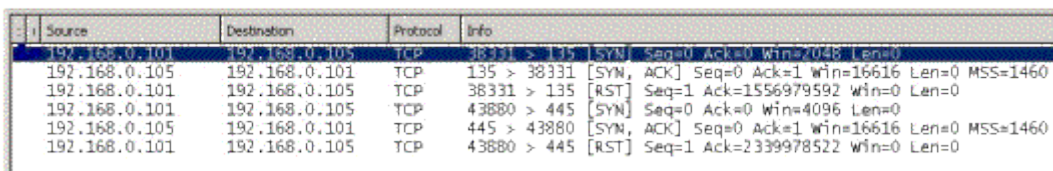


```
E:\BSI>nmap -sS -p135 192.168.0.105
Starting nmap U. 3.00 ( www.insecure.org/nmap )
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
135/tcp   open       loc-srv

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
E:\BSI>nmap -sS -p445 192.168.0.105
Starting nmap U. 3.00 ( www.insecure.org/nmap )
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
445/tcp   open       microsoft-ds

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
E:\BSI>_
```

Rys. 14. Raport skanera *nmap* wykrywającego otwarte porty 445 i 135 (metoda półotwarta)



Source	Destination	Protocol	Info
192.168.0.101	192.168.0.105	TCP	38331 > 135 [SYN] Seq=0 Ack=0 Win=0 Len=0
192.168.0.105	192.168.0.101	TCP	135 > 38331 [SYN, ACK] Seq=0 Ack=1 Win=16616 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	38331 > 135 [RST] Seq=1 Ack=1556979592 Win=0 Len=0
192.168.0.101	192.168.0.105	TCP	43880 > 445 [SYN] Seq=0 Ack=0 Win=0 Len=0
192.168.0.105	192.168.0.101	TCP	445 > 43880 [SYN, ACK] Seq=0 Ack=1 Win=16616 Len=0 MSS=1460
192.168.0.101	192.168.0.105	TCP	43880 > 445 [RST] Seq=1 Ack=2339978522 Win=0 Len=0

Rys.15. Ruch sieciowy podczas wykrywania przez skaner *nmap* otwartych portów 445 i 135 (metoda półotwarta)

Techniki specjalne TCP

Podobnie jak w przypadku skanowania półotwartego, stosowanie technik określanych mianem specjalnych miało na celu utrudnienie wykrycia faktu skanowania. Obecnie większość z nich należy do podstawowego zbioru zdarzeń wykrywanych przez systemy detekcji intruzów. Wszystkie techniki przedstawione w niniejszym punkcie wykorzystują podstawową zasadę zapisaną w RFC 793 określającą, że system powinien odpowiedzieć pakietem RST na każdy pakiet niezgodny z kolejnością nawiązywania połączenia TCP, jeżeli jest on kierowany do portu zamkniętego. Wobec tego skanowanie będzie polegało na wysłaniu pakietów z ustawioną flagą FIN, z flagami SYN/ACK (drugi etap nawiązywania połączenia), z wszystkimi ustawionymi flagami (pakiet XMAS), bez ustawionych flag (pakiet NULL).

Detekcję portu zamkniętego systemu Linux przy pomocy technik specjalnych pokazano na rys. 16 i 17. Obrazy uzyskane podczas detekcji portu otwartego można zobaczyć na rys. 18 i 19. W tym przypadku, jak można zauważyć, nie jest wysyłana żadna odpowiedź przez system skanowany.

Niektóre systemy (np.: Windows) są na tą technikę odporne, gdyż zwracają pakiet RST również w przypadku skanowania portu otwartego. Taki przypadek dla portu otwartego pokazano na rys. 20 i 21. Detekcji poddano port 445 systemu Windows 2000. Jest on otwarty, co można stwierdzić np. na podstawie obrazów zamieszczonych na rys. 14 i 15. Program *nmap* błędnie rozpoznaje stan portów. Wynika to z odpowiedzi uzyskanych od systemu badanego. Przebieg detekcji portu zamkniętego przebiega według reguły przedstawionej powyżej.

```

Wiersz polecenia
E:\BS1>nmap -sF -P0 -n -p25 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.104) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BS1>nmap -sX -P0 -n -p25 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.104) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BS1>nmap -sN -P0 -n -p25 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.104) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BS1>_

```

Rys. 16. Raport skanera *nmap* dokonującego detekcji zamkniętego portu systemu Linux poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)

Source	Destination	Protocol	Info
192.168.0.101	192.168.0.104	TCP	41063 > 25 [FIN] Seq=0 Ack=0 Win=4096 Len=0
192.168.0.104	192.168.0.101	TCP	25 > 41063 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.101	192.168.0.104	TCP	38418 > 25 [FIN, PSH, URG] Seq=0 Ack=0 Win=2048 Urg=0 Len=0
192.168.0.104	192.168.0.101	TCP	25 > 38418 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.101	192.168.0.104	TCP	38830 > 25 [] Seq=0 Ack=0 Win=4096 Len=0
192.168.0.104	192.168.0.101	TCP	25 > 38830 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

Rys.17. Ruch sieciowy podczas detekcji przez skaner *nmap* zamkniętego portu systemu Linux poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)

```

Wiersz polecenia
E:\BS1>nmap -sF -P0 -n -p68 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on (192.168.0.104):
Port      State      Service
68/tcp    open      dhcpclient
Nmap run completed -- 1 IP address (1 host up) scanned in 13 seconds
E:\BS1>nmap -sX -P0 -n -p68 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on (192.168.0.104):
Port      State      Service
68/tcp    open      dhcpclient
Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
E:\BS1>nmap -sN -P0 -n -p68 192.168.0.104
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on (192.168.0.104):
Port      State      Service
68/tcp    open      dhcpclient
Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
E:\BS1>_

```

Rys. 18. Raport skanera *nmap* dokonującego detekcji otwartego portu systemu Linux poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)

Source	Destination	Protocol	Info
192.168.0.101	192.168.0.104	TCP	40033 > 68 [FIN] Seq=0 Ack=0 Win=1024 Len=0
192.168.0.101	192.168.0.104	TCP	40034 > 68 [FIN] Seq=0 Ack=0 Win=1024 Len=0
192.168.0.101	192.168.0.104	TCP	55559 > 68 [FIN, PSH, URG] Seq=0 Ack=0 Win=1024 Urg=0 Len=0
192.168.0.101	192.168.0.104	TCP	55560 > 68 [FIN, PSH, URG] Seq=0 Ack=0 Win=1024 Urg=0 Len=0
192.168.0.101	192.168.0.104	TCP	57522 > 68 [] Seq=0 Ack=0 Win=2048 Len=0
192.168.0.101	192.168.0.104	TCP	57523 > 68 [] Seq=0 Ack=0 Win=2048 Len=0

Rys.19. Ruch sieciowy podczas detekcji przez skaner *nmap* otwartego portu systemu Linux poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)


```

Wiersz polecenia
E:\BSI> nmap -sF -P0 -n -p445 192.168.0.105
Starting nmap U. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.105) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BSI> nmap -sX -P0 -n -p445 192.168.0.105
Starting nmap U. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.105) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BSI> nmap -sN -P0 -n -p445 192.168.0.105
Starting nmap U. 3.00 ( www.insecure.org/nmap )
The 1 scanned port on (192.168.0.105) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
E:\BSI>_

```

Rys. 20. Błędny raport skanera *nmap* dokonującego badania faktycznie otwartego portu systemu Windows poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)

#	Source	Destination	Protocol	Info
1	192.168.0.101	192.168.0.105	TCP	55270 > 445 [FIN, Seq=0 Ack=0 Win=2048 Len=0
2	192.168.0.105	192.168.0.101	TCP	445 > 55270 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
3	192.168.0.101	192.168.0.105	TCP	62067 > 445 [FIN, PSH, URG] Seq=0 Ack=0 Win=3072 Urg=0 Len=0
4	192.168.0.105	192.168.0.101	TCP	445 > 62067 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
5	192.168.0.101	192.168.0.105	TCP	62405 > 445 [] Seq=0 Ack=0 Win=2048 Len=0
6	192.168.0.105	192.168.0.101	TCP	445 > 62405 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

Rys.21. Ruch sieciowy podczas badania przez skaner *nmap* otwartego portu systemu Windows poprzez zastosowanie technik specjalnych (FIN, XMAS, NULL)

W sytuacjach przedstawionych wyżej należałoby bardziej zagłębić się w szczegóły implementacyjne stosów TCP/IP poszczególnych systemów operacyjnych. Niekiedy da się wówczas zaobserwować pewne prawidłowości, które można wykorzystać podczas interpretowania wyników skanowania portów. Polegają one na analizie pola określającego wielkość okna oraz pola TTL (*time to live*) otrzymanego pakietu RST. Niektóre systemy operacyjne, w przypadku portów zamkniętych zwracają pakiet RST z ustawionym polem TTL na wartość wyższą niż dla portów otwartych.

Sprawdzanie wielkości okna jest podobną techniką. W przypadku pokazanym na rys. 22 badanie polegało na wysyłaniu pakietów SYN/ACK do systemu OpenBSD przy pomocy programu *hwing*. W zwracanych (zaznaczonych) pakietach RST wielkość okna różna od zera oznacza otwarty port. Jest to prawdą np. dla systemów z rodziny BSD (FreeBSD, OpenBSD) oraz niektórych systemów Unix (AIX, HP-UX), choć pojawiły się fazy w ich ostatnich wersjach. W przykładzie pokazanym na rys. 22 pole TTL nie zdradza stanu portu. Natomiast dobre rezultaty daje analiza wielkości okna.

#	Source	Destination	Protocol	Info
00:00:39:dd:41:b0	ff:ff:ff:ff:ff:ff	ARP	who has 192.168.0.102? Tell 192.168.0.101	
00:02:44:4b:92:ac	00:00:39:dd:41:b0	ARP	192.168.0.102 is at 00:02:44:4b:92:ac	
192.168.0.101	192.168.0.102	TCP	24876 > 20 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	
192.168.0.102	192.168.0.101	TCP	20 > 24876 [RST] Seq=1 Ack=4294942420 Win=0 Len=0	
192.168.0.101	192.168.0.102	TCP	24876 > 21 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	
192.168.0.102	192.168.0.101	TCP	21 > 24876 [RST] Seq=1 Ack=4294942420 Win=36384 Len=0	
192.168.0.101	192.168.0.102	TCP	24876 > 22 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	
192.168.0.102	192.168.0.101	TCP	22 > 24876 [RST] Seq=1 Ack=4294942420 Win=0 Len=0	
192.168.0.101	192.168.0.102	TCP	24876 > 23 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	
192.168.0.102	192.168.0.101	TCP	23 > 24876 [RST] Seq=1 Ack=4294942420 Win=36384 Len=0	
192.168.0.101	192.168.0.102	TCP	24876 > 24 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	
192.168.0.102	192.168.0.101	TCP	24 > 24876 [RST] Seq=1 Ack=4294942420 Win=0 Len=0	
192.168.0.101	192.168.0.102	TCP	24876 > 25 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0	

Rys.22. Detekcja otwartych portów 21 i 23 systemu OpenBSD poprzez analizę wartości pola WINDOW w odbieranych pakietach RST

Jak widać z załączonych przykładów poprawna interpretacja wyników uzyskanych przy pomocy opisanej metody wymaga wiedzy odnośnie systemu operacyjnego zainstalowanego na skanowanym komputerze. Dlatego też z reguły trzeba najpierw ustalić jego rodzaj.

3.4. Skanowanie UDP

W przypadku bezpołączeniowego protokołu UDP reakcja zdalnego systemu może być dwójaka. Aktywny system w momencie otrzymania datagramu UDP na zamknięty port powinien wysłać komunikat *ICMP Destination Unreachable* (typ 3) a dokładniej mówiąc *ICMP Port Unreachable* (typ 3, kod 3). Przypadek taki został przedstawiony na rys. 23 i 24. W przeciwnym przypadku, gdy port jest otwarty, nie należy się spodziewać odpowiedzi, gdyż w przypadku UDP nie występuje potwierdzenie odebrania pakietu. Taka sytuację można zobaczyć na rys. 25 i 2.

Czasami można uzyskać odpowiedź z portu otwartego, gdy serwer usługi ulokowanej w tym porcie próbuje odpowiedzieć na domniemane żądanie. Zależać to będzie przede wszystkim od sposobu budowania pakietu skanującego.

```

C:\> Wiersz polecenia

E:\BSI>nmap -sU -p 25 -P0 192.168.0.104

Starting nmap V. 3.00 < www.insecure.org/nmap >
The 1 scanned port on <192.168.0.104> is: closed

Nmap run completed -- 1 IP address <1 host up> scanned in 9 seconds

E:\BSI>

```

Rys. 23. Raport skanera *nmap* dokonującego detekcji zamkniętego portu 25 UDP

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.101	192.168.0.104	UDP	Source port: 52091 Destination port: 25
2	0.001438	192.168.0.104	192.168.0.101	ICMP	Destination unreachable
3	12.656296	192.168.0.101	192.168.0.104	UDP	Source port: 38279 Destination port: 25
4	12.667025	192.168.0.104	192.168.0.101	ICMP	Destination unreachable
5	23.299364	192.168.0.101	192.168.0.104	UDP	Source port: 55998 Destination port: 25
6	23.310802	192.168.0.104	192.168.0.101	ICMP	Destination unreachable


```

Frame 2 (70 bytes on wire, 70 bytes captured)
Ethernet II, Src: 00:0c:29:34:20:1b, Dst: 00:00:39:dd:41:b0
Internet Protocol, Src Addr: 192.168.0.104 (192.168.0.104), Dst Addr: 192.168.0.101 (192.168.0.101)
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
Checksum: 0x7f54 (correct)
Internet Protocol, Src Addr: 192.168.0.101 (192.168.0.101), Dst Addr: 192.168.0.104 (192.168.0.104)
User Datagram Protocol, Src Port: 52091 (52091), Dst Port: 25 (25)

```

Rys.24. Ruch sieciowy podczas detekcji przez skaner *nmap* zamkniętego portu 25 UDP

```

C:\> Wiersz polecenia

E:\BSI>nmap -sU -p 445 -P0 192.168.0.105

Starting nmap V. 3.00 < www.insecure.org/nmap >
Interesting ports on WM2000PROF (192.168.0.105):
Port      State      Service
445/udp   open       microsoft-ds

Nmap run completed -- 1 IP address <1 host up> scanned in 18 seconds

E:\BSI>

```

Rys. 25. Raport skanera *nmap* dokonującego detekcji otwartego portu 445 UDP

No.	Source	Destination	Protocol	Info
1	C 192.168.0.101	192.168.0.105	NBNS	Name query NBSTAT *<00><00><00><00><00><00><00><00>
2	C 00:0c:29:32:17:47	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.101? Tell 192.168.0.105
3	C 00:00:39:dd:41:b0	00:0c:29:32:17:47	ARP	192.168.0.101 is at 00:00:39:dd:41:b0
4	C 192.168.0.105	192.168.0.101	NBNS	Name query response NBSTAT
5	C 192.168.0.101	192.168.0.105	UDP	Source port: 41345 Destination port: 445
5	C 192.168.0.101	192.168.0.105	UDP	Source port: 41346 Destination port: 445

Rys.26. Ruch sieciowy podczas detekcji przez skaner *nmap* otwartego portu 445 UDP

Jeżeli skanującemu zależy na zbadaniu osiągalności komputera a nie portu, to technika skanowania UDP również może znaleźć zastosowanie. Odpowiedź *ICMP Port Unreachable* (typ 3, kod 3) świadczy o osiągalności badanego węzła. Podobnie, z zastrzeżeniem uwag sformułowanych niżej, można by było interpretować brak odpowiedzi. Nieosiągalność węzła sygnalizowana może być przez ostatni przed badanym węzłem ruter, zwróceniem komunikatu *ICMP Destination Unreachable* (typ 3, kod 1).

Technika skanowania przy pomocy protokołu UDP nie należy do najskuteczniejszych ze względu na fakt, że wiele bramek (ściany ogniowe, routery brzegowe) odfiltrowuje datagramy UDP skierowane na inne porty niż 53 (DNS). Datagramy UDP są łatwo wykrywalne, ze względu na ich

małą popularność. Duża liczba systemów nie odpowiada prawidłowo na datagramy UDP, inne mają wprowadzone ograniczenia, np. co do ilości i częstości generowanych pakietów ICMP. Często jest także filtrowanie pakietów ICMP przez zapory ogniowe oraz routery. Dlatego brak odpowiedzi na datagram UDP o niczym nie świadczy. Niestety duża część dostępnych aktualnie skanerów brak odpowiedzi jednoznacznie interpretuje jako wykrycie portu otwartego, co oczywiście nie jest prawidłowym działaniem.

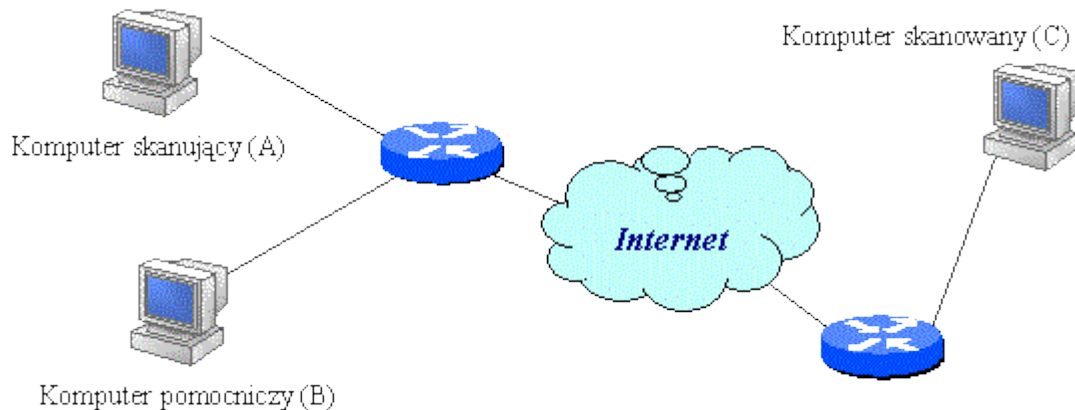
3.5. Inne techniki skanowania

Mapowanie odwrotne

Metodą wykrywania komputerów funkcjonujących w sieci może być wysyłanie pakietów z ustawioną flagą RST, tzw. *inverse mapping*. Metoda ta wykorzystywana jest z reguły do poznania topologii sieci - stwierdzenia, czy dany komputer istnieje czy nie. W typowym przypadku router po otrzymaniu pakietu skierowanego do hosta, który nie istnieje, wygeneruje komunikat *ICMP host unreachable* lub *ICMP time exceeded*. Dzieje się tak ponieważ router najpierw wyśle do podsieci zapytanie ARP o adres MAC komputera o zadanym adresie, a kiedy nie otrzyma odpowiedzi, zwróci komunikat o błędzie. Do routera można wysłać dowolny pakiet, ale jeśli będzie on typowy (np.: ping - *ICMP echo request* lub SYN/ACK) to prawdopodobnie zostanie zapisany w logach. Jeśli natomiast będzie to pakiet z ustawioną flagą RST i losowym numerem ACK to istnieje duże prawdopodobieństwo, że zostanie zignorowany przez systemy ochrony, a wygeneruje komunikat interesujący osobę, która przeprowadza rozpoznawanie. Sytuacja taka odpowiada zdarzeniu, kiedy zdalny system zamyka połączenie z hostem z chronionej sieci - nie ma więc powodów by takiego pakietu nie przyjąć. Dopóki zapory ogniowe lub inne programy ochronne nie będą śledzić wszystkich otwartych połączeń, skanowanie takie będzie skuteczne. Metoda ta pozwala jedynie stwierdzić, czy dany komputer nie jest aktywny. Brak odpowiedzi może oznaczać aktywność hosta, choć równie prawdopodobne jest to, że router nie wygenerował komunikatu ICMP, komunikat się zgubił, lub wysłany przez skanowanego pakiet został odfiltrowany w drodze powrotnej.

Mapowanie odwrotne z podszywaniem się

Odmianą powyższej metody jest tzw. *spoofed inverse mapping*, czyli skanowanie z ukryciem tożsamości (adresu) skanującego. Polega to na wykorzystaniu do skanowania jeszcze jednego komputera. Załóżmy, że komputerem skanującym jest A, komputerem pomocniczym B, a skanowaniu podlega komputer C. Ważne jest by wszystkie pakiety wysyłane z komputera B przechodziły przez A. W praktyce oznacza to, że A i B muszą znajdować się w jednym segmencie sieci.



Rys.27. Topologia sieci wymagana do przeprowadzenia skanowania metodą mapowania odwrotnego z podszywaniem się

Teraz postąpić można dwojako:

Wysłać do komputera B pakiety z włączoną flagą ACK i sfałszowanym adresem źródłowym wskazującym na C. Komputer B odpowie na takie pakiety segmentami RST skierowanymi do hosta C. Ponieważ skanujący komputer A znajduje się po drodze do hosta B, będzie on w stanie wychwycić odpowiedź C na pakiety RST.

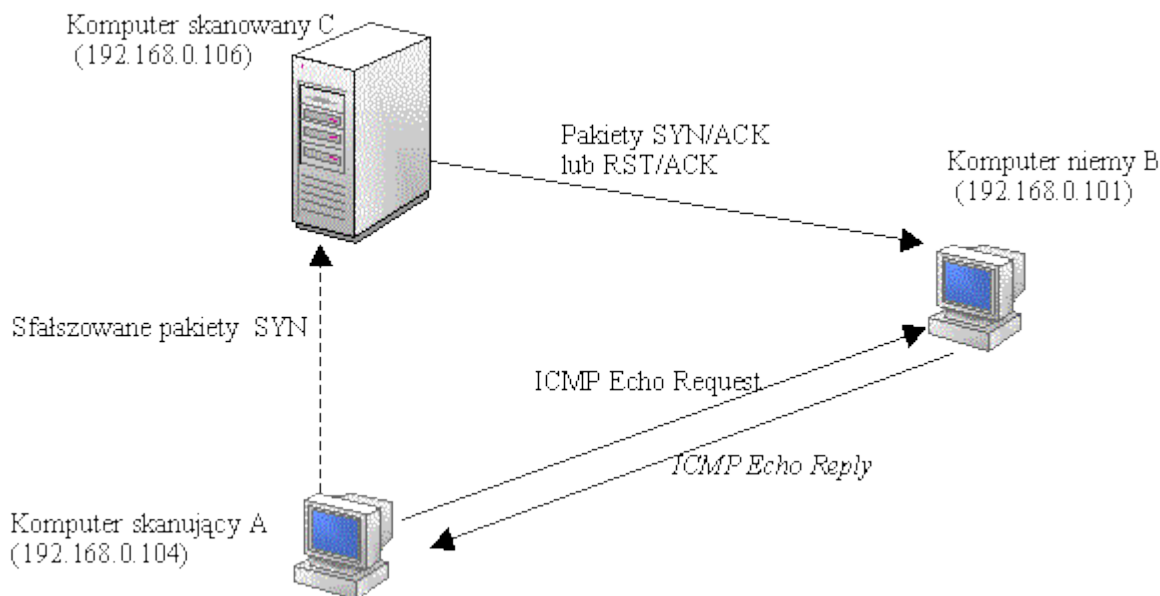
Aby nie zostawić śladu w logach na komputerze B można od razu wysłać pakiety RST ze sfałszowanym adresem źródła (wskazującym na B) do komputera C. Jeśli bramka wyśle komunikat ICMP wskazujący na brak hosta C, to skanujący komputer A zobaczy go. Ujemną stroną takiego postępowania jest brak pewności pełnej anonimowości, bowiem podrobione pakiety zawierają pewne cechy identyfikujące konkretny system operacyjny.

Metoda *idle scan*

Metoda *idle scan* zależna jest od implementacji stosu TCP/IP konkretnego systemu operacyjnego. Wykorzystuje ona wcześniej opisaną technikę skanowania SYN - czyli nawiązywania połączenia TCP. Różnica polega na wykorzystaniu trzeciego komputera jako źródła pakietów, co pozwala na ukrycie przed skanowanym własnego adresu.

Aby skorzystać z tej techniki trzeba zlokalizować w sieci komputer, który nie wysyła i nie odbiera żadnych pakietów, tzw: host niemy (*dumb*). Scenariusz procesu skanowania zakłada udział trzech komputerów:

- A - host skanujący,
- B - host niemy,
- C - host skanowany, czyli cel.



Rys.28. Topologia sieci wykorzystanej do zaprezentowania skanowania metodą *idle scan*

Technika ta wykorzystuje fakt, że wiele systemów operacyjnych umieszcza jako zawartość pola IP ID w nagłówku pakietu IP, liczby generowane w kolejności rosnącej, różniące się o stałą wartość. System Microsoft NT zwiększa to pole stopniowo o wartość 256, Linux o 1. Niektóre systemy (np. *OpenBSD*) losują te wartości, przez co nie można ich wykorzystać jako niemych hostów w tej metodzie.

Skanowanie zaczyna komputer A od wysyłania do komputera B pakietów *ICMP Echo Request*. W odbieranych pakietach *ICMP Echo Reply* analizowane są wartości pola IP ID. Wartości te powinny rosnąć w sposób regularny. Oznacza to, że komputer B nie wysyłał żadnych pakietów poza tymi, które stanowiły odpowiedź na pakiety przychodzące z komputera A. Przykład sekwencji takich odpowiedzi przedstawiono na rys. 29.

```

>hping2 192.168.0.101
HPING 192.168.0.101 (eth0 192.168.0.101): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.101 ttl=128 id=1857 sport=0 flags=RA seq=16 win=0 rtt=4.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1858 sport=0 flags=RA seq=17 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1859 sport=0 flags=RA seq=18 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1860 sport=0 flags=RA seq=19 win=0 rtt=2.7 ms
len=46 ip=192.168.0.101 ttl=128 id=1861 sport=0 flags=RA seq=20 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1862 sport=0 flags=RA seq=21 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1863 sport=0 flags=RA seq=22 win=0 rtt=2.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1864 sport=0 flags=RA seq=23 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1865 sport=0 flags=RA seq=24 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1866 sport=0 flags=RA seq=25 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1867 sport=0 flags=RA seq=26 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1868 sport=0 flags=RA seq=27 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1869 sport=0 flags=RA seq=28 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1870 sport=0 flags=RA seq=29 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1871 sport=0 flags=RA seq=30 win=0 rtt=1.9 ms
len=46 ip=192.168.0.101 ttl=128 id=1872 sport=0 flags=RA seq=31 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1873 sport=0 flags=RA seq=32 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1874 sport=0 flags=RA seq=33 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1875 sport=0 flags=RA seq=34 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1876 sport=0 flags=RA seq=35 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1877 sport=0 flags=RA seq=36 win=0 rtt=1.4 ms
len=46 ip=192.168.0.101 ttl=128 id=1878 sport=0 flags=RA seq=37 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1879 sport=0 flags=RA seq=38 win=0 rtt=1.3 ms
len=46 ip=192.168.0.101 ttl=128 id=1880 sport=0 flags=RA seq=39 win=0 rtt=1.1 ms
len=46 ip=192.168.0.101 ttl=128 id=1881 sport=0 flags=RA seq=40 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1882 sport=0 flags=RA seq=41 win=0 rtt=1.2 ms
len=46 ip=192.168.0.101 ttl=128 id=1883 sport=0 flags=RA seq=42 win=0 rtt=1.3 ms

```

Rys.29. Odpowiedzi generowane przez host niemy

Równoległe z wysyłaniem i odbieraniem pakietów ICMP, komputer A wysyła do komputera C na badany port pakiet SYN (pierwsza faza nawiązywania połączenia TCP), ze sfałszowanym adresem nadawcy wskazującym na komputer B. Komputer C odpowie na taki pakiet w sposób zdefiniowany w RFC:

- Pakietem SYN/ACK jeśli port jest otwarty w trybie nasłuchu. Na taki pakiet host B, który nic nie wie o połączeniu odpowie pakietem RST. Oznacza to, że w pakietach wysyłanych przez komputer B do komputera A, regularny do tej pory przyrost wartości pola ID IP zostanie zakłócony.
- Pakietem RST/ACK jeśli docelowy port na komputerze C jest zamknięty. Komputer B zignoruje taki pakiet i zakłócenia regularności przyrostu pola ID IP nie będzie.

Wysyłanie przez skanującego pakietów SYN, kolejno do portów 79, 80, 81 komputera skanowanego zaprezentowano na rys. 30.

```

>hping2 -S -p 79 -c 1 -a 192.168.0.101 192.168.0.106
HPING 192.168.0.106 (eth0 192.168.0.106): S set, 40 headers + 0 data bytes

--- 192.168.0.106 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ns
>
>hping2 -S -p 80 -c 1 -a 192.168.0.101 192.168.0.106
HPING 192.168.0.106 (eth0 192.168.0.106): S set, 40 headers + 0 data bytes

--- 192.168.0.106 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ns
>
>hping2 -S -p 81 -c 1 -a 192.168.0.101 192.168.0.106
HPING 192.168.0.106 (eth0 192.168.0.106): S set, 40 headers + 0 data bytes

--- 192.168.0.106 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ns
>

```

Rys. 30. Raport skanera hping2 dokonującego badania portów 79,

Na komputerze A przez cały czas analizowane powinny być zmiany w polu IP ID pakietów przychodzących z hosta B. Jeśli nastąpiło zakłócenie regularności oznacza to, że komputer B odpowiedział pakietem RST na połączenie z komputera C zdradzając przez to, że badany port jest otwarty. Taki przypadek zaznaczono na rys. 31. Ruch pomiędzy komputerem niemy a skanowanym przedstawiono na rys. 32.

len=46	ip=192.168.0.101	ttl=128	id=51535	sport=0	flags=RA	seq=25	win=0	rtt=2.0 ms
len=46	ip=192.168.0.101	ttl=128	id=51536	sport=0	flags=RA	seq=26	win=0	rtt=1.9 ms
len=46	ip=192.168.0.101	ttl=128	id=51537	sport=0	flags=RA	seq=27	win=0	rtt=1.8 ms
len=46	ip=192.168.0.101	ttl=128	id=51538	sport=0	flags=RA	seq=28	win=0	rtt=1.8 ms
len=46	ip=192.168.0.101	ttl=128	id=51539	sport=0	flags=RA	seq=29	win=0	rtt=2.2 ms
len=46	ip=192.168.0.101	ttl=128	id=51540	sport=0	flags=RA	seq=30	win=0	rtt=1.5 ms
len=46	ip=192.168.0.101	ttl=128	id=51541	sport=0	flags=RA	seq=31	win=0	rtt=1.8 ms
len=46	ip=192.168.0.101	ttl=128	id=51542	sport=0	flags=RA	seq=32	win=0	rtt=1.7 ms
len=46	ip=192.168.0.101	ttl=128	id=51543	sport=0	flags=RA	seq=33	win=0	rtt=1.6 ms
len=46	ip=192.168.0.101	ttl=128	id=51544	sport=0	flags=RA	seq=34	win=0	rtt=1.6 ms
len=46	ip=192.168.0.101	ttl=128	id=51546	sport=0	flags=RA	seq=35	win=0	rtt=1.5 ms
len=46	ip=192.168.0.101	ttl=128	id=51547	sport=0	flags=RA	seq=36	win=0	rtt=1.5 ms
len=46	ip=192.168.0.101	ttl=128	id=51548	sport=0	flags=RA	seq=37	win=0	rtt=1.6 ms
len=46	ip=192.168.0.101	ttl=128	id=51549	sport=0	flags=RA	seq=38	win=0	rtt=1.5 ms
len=46	ip=192.168.0.101	ttl=128	id=51550	sport=0	flags=RA	seq=39	win=0	rtt=2.0 ms
len=46	ip=192.168.0.101	ttl=128	id=51551	sport=0	flags=RA	seq=40	win=0	rtt=1.5 ms

Rys. 31. Zakłócenie regularności w wartościach pola IP ID sygnalizujące port otwarty

Source	Destination	Protocol	Info
00:0c:29:49:3c:93	ff:ff:ff:ff:ff:ff	ARP	who has 192.168.0.101? Tell 192.168.0.106
00:00:39:dd:41:b0	00:0c:29:49:3c:93	ARP	192.168.0.101 is at 00:00:39:dd:41:b0
192.168.0.106	192.168.0.101	TCP	79 > 1718 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.106	192.168.0.101	TCP	80 > 1039 [SYN, ACK] Seq=0 Ack=1 Win=8576 Len=0 MSS=1460
192.168.0.101	192.168.0.106	TCP	1039 > 80 [RST] Seq=1 Ack=1976890915 Win=0 Len=0
192.168.0.106	192.168.0.101	TCP	81 > 1579 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

Rys. 32. Ruch pomiędzy komputerem niemy (192.168.0.101) a skanowanym (192.168.0.106)

Technika ta w oczywisty sposób wymaga niemego hosta, by zminimalizować prawdopodobieństwo fałszywego rozpoznania. Fałszywe rozpoznanie może wystąpić wówczas, gdy komputer uważany za niemy nagle rozpocznie komunikację z jakimś innym hostem, którego skanujący nie brał pod uwagę. Generowanie większej liczby sfałszowanych pakietów do komputera C może być metodą również zmniejszenia prawdopodobieństwa popełnienia pomyłki. Skanowanie to może wykorzystywać inne metody nawiązywania połączenia niż SYN, np.: *inverse mapping* omijając ewentualne systemy IDS lub inne programy wyspecjalizowane w wykrywaniu skanowania.

Metoda FTP bounce

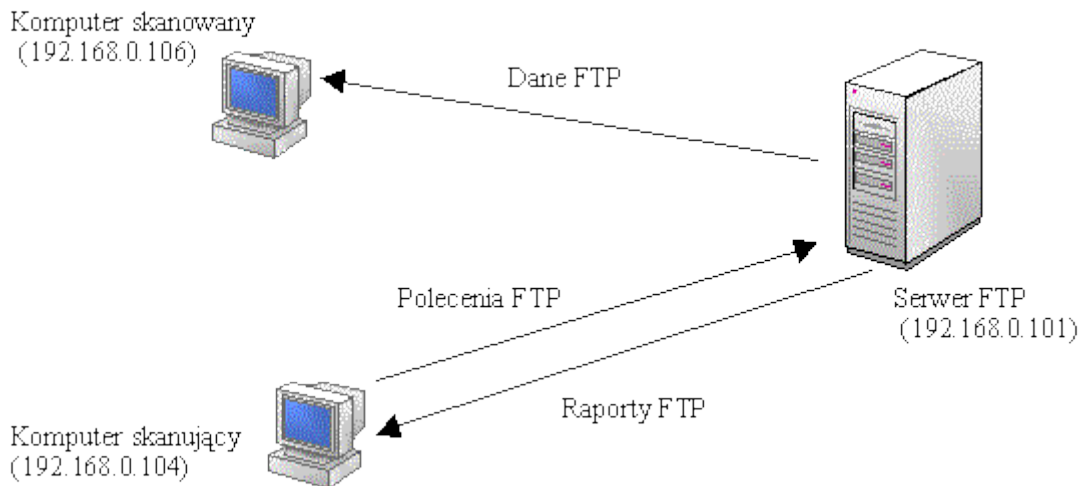
Metodą na ukrywanie tożsamości skanującego jest wykorzystanie techniki *FTP Bounce Scanning*. Wykorzystuje ona serwer FTP jako punkt pośredniczący - *proxy*. Metoda ta posługuje się właściwością protokołu FTP określoną przez RFC959, polegającą na tym, że serwer FTP może wysłać dane do innego komputera niż źródłowy czyli ten, z którego nawiązano połączenie. Właściwość ta określa się mianem FXP. Jest ona blokowana w wielu serwerach domyślnie, inne w ogóle jej nie posiadają. Czasami można zdefiniować, którzy użytkownicy serwera mogą korzystać z tej funkcji. Można na przykład pozwalać na takie transfery z określonego źródła, lub zabronić korzystania z tej funkcji użytkownikowi anonimowemu (*anonymous*).

Do skanowania wykorzystana zostaje komenda PORT określająca port docelowy oraz adres IP pod który należy wysłać dane. Od tego momentu wyniki wszystkich poleceń wydanych serwerowi FTP przesłane zostaną do komputera skanowanego. Natomiast raporty dotyczące realizacji przesyłania danych kierowane są do skanującego.

Jeśli wyspecyfikowany port na komputerze skanowanym jest otwarty, to serwer FTP zwróci przeprowadzającemu skanowanie komunikat 150 i 226. W przeciwnym wypadku pojawi się komunikat 425 *Can't build data connection: Connection refused*.

Największą zaletą tej metody jest anonimowość (nie licząc logów serwera FTP). Osoba skanująca nie musi też posiadać uprawnień superużytkownika. Główną wadą tej metody jest jej powolność.

Na rys.33 przedstawiono topologię sieci wykorzystanej do przeprowadzenia opisywanego skanowania. Serwer FTP zainstalowany został na komputerze o adresie 192.168.0.101. Skanowanie przeprowadzano z komputera o adresie 192.168.0.104. Celem skanowania był komputer o adresie 192.18.0.106.



Rys.33. Topologia sieci zastosowana do przeprowadzenia skanowania *FTP bounce*

Na rys. 34 zobaczyć można raport uzyskiwany po stronie skanującego. Do skanowania wykorzystano standardowy program *telnet*. Skanowaniu podlegały porty 79, 82.

Na rys. 35 przedstawiono obraz ruchu sieciowego podczas testowania otwartego portu 80, na rys. 36 - obraz ruchu podczas testowania zamkniętego portu 79, a na rys. 37 - obraz ruchu podczas inicjowania sesji FTP pomiędzy skanującym a serwerem FTP. Dodatkowo na rys. 38 zamieszczono fragment logu serwera FTP, w którym można zauważyć zapisy dotyczące przeprowadzonego skanowania.

```
>telnet 192.168.0.101 21
Trying 192.168.0.101...
Connected to 192.168.0.101.
Escape character is '^]'.
220-GuildFTPd FTP Server (c) 1997-2002
220-Version 0.999.13
220 Please enter your name:
USER anonymous
331 User name okay, Need password.
PASS abc@abc
230 User logged in.
PORT 192,168,0,106,0,79
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/lis (0 bytes).
425 Can't build data connection.
PORT 192,168,0,106,0,80
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/lis (0 bytes).
226 Transfer successful.
PORT 192,168,0,106,0,81
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/lis (0 bytes).
425 Can't build data connection.
PORT 192,168,0,106,0,82
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for /bin/lis (0 bytes).
425 Can't build data connection.
QUIT
221 Goodbye. Control connection closed.
Connection closed by foreign host.
>
```

Rys.34. Raport po stronie skanującego

no	Source	Destination	Protocol	Info
1	192.168.0.104	192.168.0.101	FTP	Request: PORT 192.168.0.106,0.80
2	192.168.0.101	192.168.0.104	FTP	Response: 200 PORT command successful.
3	192.168.0.104	192.168.0.101	TCP	1035 > 21 [ACK] Seq=25 Ack=30 Win=5840 Len=0 TSV=195702 TSER=176668
4	192.168.0.104	192.168.0.101	FTP	Request: LIST
5	192.168.0.101	192.168.0.104	TCP	21 > 1035 [ACK] Seq=30 Ack=31 Win=65443 Len=0 TSV=176739 TSER=195973
6	192.168.0.101	192.168.0.104	FTP	Response: 150 Opening ASCII mode data connection for /bin/lis (0 bytes).
7	192.168.0.101	192.168.0.106	TCP	4348 > 80 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
8	192.168.0.106	192.168.0.101	TCP	80 > 4348 [SYN, ACK] Seq=0 Ack=1 Win=8760 Len=0 MSS=1460
9	192.168.0.101	192.168.0.106	TCP	4348 > 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
10	192.168.0.104	192.168.0.101	TCP	1035 > 21 [ACK] Seq=31 Ack=93 Win=5840 Len=0 TSV=195974 TSER=176741
11	192.168.0.101	192.168.0.104	FTP	Response: 226 Transfer successful.
12	192.168.0.101	192.168.0.106	TCP	4348 > 80 [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
13	192.168.0.106	192.168.0.101	TCP	80 > 4348 [ACK] Seq=1 Ack=2 Win=8760 Len=0
14	192.168.0.106	192.168.0.101	TCP	80 > 4348 [FIN, ACK] Seq=1 Ack=2 Win=8760 Len=0
15	192.168.0.101	192.168.0.106	TCP	4348 > 80 [ACK] Seq=2 Ack=2 Win=65535 Len=0
16	192.168.0.104	192.168.0.101	TCP	1035 > 21 [ACK] Seq=31 Ack=119 Win=5840 Len=0 TSV=195975 TSER=176742

Rys. 35. Ruch sieciowy podczas detekcji otwartego portu 80 TCP

nr	Source	Destination	Protocol	Info
192.168.0.104	192.168.0.101	192.168.0.101	FTP	Request: PORT 192.168.0.106,0.79
192.168.0.101	192.168.0.104	192.168.0.104	TCP	21 > 1035 [ACK] Seq=0 Ack=25 Win=65480 Len=0 TSV=176181 TSER=193759
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 200 PORT command successful.
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=25 Ack=30 Win=5840 Len=0 TSV=193760 TSER=176182
192.168.0.104	192.168.0.101	192.168.0.101	FTP	Request: LIST
192.168.0.101	192.168.0.104	192.168.0.104	TCP	21 > 1035 [ACK] Seq=30 Ack=31 Win=65474 Len=0 TSV=176285 TSER=194184
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 150 Opening ASCII mode data connection for /bin/lis (0 bytes).
192.168.0.101	192.168.0.106	192.168.0.106	TCP	4347 > 79 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.106	192.168.0.101	192.168.0.101	TCP	79 > 4347 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=31 Ack=93 Win=5840 Len=0 TSV=194187 TSER=176286
192.168.0.101	192.168.0.106	192.168.0.106	TCP	4347 > 79 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.106	192.168.0.101	192.168.0.101	TCP	79 > 4347 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
192.168.0.101	192.168.0.106	192.168.0.106	TCP	4347 > 79 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.106	192.168.0.101	192.168.0.101	TCP	79 > 4347 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 425 Can't build data connection.
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=31 Ack=127 Win=5840 Len=0 TSV=194213 TSER=176298

Rys.36. Ruch sieciowy podczas detekcji zamkniętego portu 79 TCP

nr	Source	Destination	Protocol	Info
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=189758 TSER=0 Win=0
192.168.0.101	192.168.0.104	192.168.0.104	TCP	21 > 1035 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=0 TSV=0 TSER=0
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=189761 TSER=0
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 220-GuillodFTPd FTP Server (c) 1997-2002
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=1 Ack=63 Win=5840 Len=0 TSV=189828 TSER=175196
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 220 Please enter your name:
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=1 Ack=92 Win=5840 Len=0 TSV=189828 TSER=175197
192.168.0.104	192.168.0.101	192.168.0.101	FTP	Request: USER anonymous
192.168.0.101	192.168.0.104	192.168.0.104	TCP	21 > 1035 [ACK] Seq=92 Ack=17 Win=65519 Len=0 TSV=175593 TSER=191409
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 331 User name okay, Need password.
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=17 Ack=128 Win=5840 Len=0 TSV=191411 TSER=175593
192.168.0.104	192.168.0.101	192.168.0.101	FTP	Request: PASS abc@abc
192.168.0.101	192.168.0.104	192.168.0.104	TCP	21 > 1035 [ACK] Seq=128 Ack=31 Win=65505 Len=0 TSV=175744 TSER=192054
192.168.0.101	192.168.0.104	192.168.0.104	FTP	Response: 230 User logged in.
192.168.0.104	192.168.0.101	192.168.0.101	TCP	1035 > 21 [ACK] Seq=31 Ack=149 Win=5840 Len=0 TSV=192060 TSER=175758

Rys.37. Ruch sieciowy podczas inicjowania sesji FTP

```

(000001) 14-07-2005 07:31:44 - (not logged in) (192.168.0.104) > connected to ip : 192.168.0.101
(000001) 14-07-2005 07:31:44 - (not logged in) (192.168.0.104) > sending welcome message.
(000001) 14-07-2005 07:31:44 - (not logged in) (192.168.0.104) > 220 Bullet Proof FTP Server
(000001) 14-07-2005 07:31:55 - (not logged in) (192.168.0.104) > USER anonymous
(000001) 14-07-2005 07:31:55 - (not logged in) (192.168.0.104) > 331 Password required for anonymous.
(000001) 14-07-2005 07:32:05 - (not logged in) (192.168.0.104) > PASS *****
(000001) 14-07-2005 07:32:06 - anonymous (192.168.0.104) > logged in.
(000001) 14-07-2005 07:32:06 - anonymous (192.168.0.104) > 230 User anonymous logged in.
(000001) 14-07-2005 07:32:37 - anonymous (192.168.0.104) > PORT 192,168,0,106,0,79
(000001) 14-07-2005 07:32:37 - anonymous (192.168.0.104) > 200 Port command successful.
(000001) 14-07-2005 07:32:41 - anonymous (192.168.0.104) > LIST
(000001) 14-07-2005 07:32:41 - anonymous (192.168.0.104) > 150 Opening data connection for directory list.
(000001) 14-07-2005 07:32:42 - anonymous (192.168.0.104) > 425 Cannot open data connection (10061).
(000001) 14-07-2005 07:33:11 - anonymous (192.168.0.104) > PORT 192,168,0,106,0,80
(000001) 14-07-2005 07:33:11 - anonymous (192.168.0.104) > 200 Port command successful.
(000001) 14-07-2005 07:33:17 - anonymous (192.168.0.104) > LIST
(000001) 14-07-2005 07:33:17 - anonymous (192.168.0.104) > 150 Opening data connection for directory list.
(000001) 14-07-2005 07:33:17 - anonymous (192.168.0.104) > 226 Transfer ok
(000001) 14-07-2005 07:33:52 - anonymous (192.168.0.104) > PORT 192,168,0,106,0,81
(000001) 14-07-2005 07:33:52 - anonymous (192.168.0.104) > 200 Port command successful.
(000001) 14-07-2005 07:33:55 - anonymous (192.168.0.104) > LIST
(000001) 14-07-2005 07:33:55 - anonymous (192.168.0.104) > 150 Opening data connection for directory list.
(000001) 14-07-2005 07:33:56 - anonymous (192.168.0.104) > 425 Cannot open data connection (10061).
(000001) 14-07-2005 07:34:36 - anonymous (192.168.0.104) > PORT 192,168,0,106,0,82
(000001) 14-07-2005 07:34:36 - anonymous (192.168.0.104) > 200 Port command successful.
(000001) 14-07-2005 07:34:40 - anonymous (192.168.0.104) > LIST
(000001) 14-07-2005 07:34:40 - anonymous (192.168.0.104) > 150 Opening data connection for directory list.
(000001) 14-07-2005 07:34:41 - anonymous (192.168.0.104) > 425 Cannot open data connection (10061).
(000001) 14-07-2005 07:34:45 - anonymous (192.168.0.104) > QUIT
(000001) 14-07-2005 07:34:45 - anonymous (192.168.0.104) > 221 Bye bye ...
(000001) 14-07-2005 07:34:45 - anonymous (192.168.0.104) > disconnected. (00:03:01)

```

Rys.38. Fragment zawartości logu serwera FTP z raportem dotyczącym przeprowadzonego skanowania

3.6. Ukrywanie skanowania

Istnieje szereg metod ukrywania faktu skanowania portów wybranych komputerów. Po pierwsze można użyć tych technik skanowania, które nie zostawiają śladu w standardowych logach systemu. Drugą przeszkodą są specjalistyczne systemy w tym IDS, które są wyczułone na tego typu działania. Wykorzystują one algorytmy, które stwierdzają, czy wykryta działalność jest skanowaniem czy zwykłym ruchem sieciowym. Znajomość tych algorytmów pozwala je ominąć.

- *Skanowanie portów w losowej kolejności* - niektóre systemy IDS wykrywają sekwencyjne połączenia z jednego adresu źródłowego z kolejnymi portami. Wystarczy wprowadzić losowość przy wyborze portów do skanowania by ominąć to zabezpieczenie. Na rys. 39 przedstawiono przykład ruchu sieciowego w czasie skanowania połączeniowego z losowaniem kolejności skanowanych portów.

Source	Destination	Protocol	Info
192.168.0.101	192.168.0.104	TCP	3318 > 201 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3319 > 985 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3317 > 49 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3316 > 952 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3321 > 809 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3322 > 932 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3323 > 509 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3324 > 223 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3320 > 893 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3321 > 809 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3322 > 932 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3323 > 509 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3324 > 223 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3321 > 809 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3322 > 932 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3323 > 509 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3326 > 546 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3326 > 151 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3327 > 295 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3323 > 509 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3324 > 223 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3328 > 196 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3329 > 458 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3325 > 546 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3326 > 151 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3327 > 295 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
192.168.0.101	192.168.0.104	TCP	3328 > 196 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460

Rys.39. Ruch sieciowy podczas skanowania połączeniowego z losową kolejnością portów

- *Powolne skanowanie* - system IDS lub dowolny inny stwierdzi próbę skanowania systemu jeśli wykryje kolejne połączenia z jednego adresu na różnych portach w określonym czasie, np.: za skanowanie uważana może być próba nawiązania 5 połączeń na różne porty z jednego adresu w czasie 3 sekund. Wiedza o tym pozwala ustalić skanowanie na 5 połączeń w czasie 4 sekund by uniknąć wykrycia. W przypadku nieznanymi ustawień można skanowanie spowolnić do kilku pakietów na dzień.
- *Fragmentacja pakietów* - część istniejących systemów IDS nie składa fragmentowanych pakietów bądź to w obawie przed atakiem DoS bądź nie posiadają takiej funkcji. Dokument RFC791 określa minimalny rozmiar fragmentowanego pakietu na 8 oktetów, czyli znacznie mniej niż nagłówek TCP + IP, przez co flagi TCP mogą znajdować się w innym fragmencie niż nagłówek. Nie widząc całego pakietu system nie jest w stanie poprawnie go rozpoznać. Rozwiązaniem problemu jest skonfigurowanie bramki (zapory ogniowej lub rutera) tak, by składał w całość wszystkie fragmentowane pakiety.
- *Odwrócenie uwagi* - technika ta polega na stworzeniu licznych strumieni skanujących pakietów ze sfalszowanymi adresami nadawcy. Wśród zalewu pakietów co jakiś czas znajdować będzie się adres prawdziwego hosta inicjującego skanowanie. Jego wychwycenie w morzu innych pakietów jest jednak bardzo trudne i pracochłonne. Skanowanie takie można zidentyfikować badając wartość pola TTL. Jeśli wszystkie pakiety posiadają jednakową wartość znaczy to, że z dużym prawdopodobieństwem wysłane zostały z jednego miejsca. Na rys. 40 przedstawiono przykład zalewu pakietów, mającego na celu ukrycie faktu skanowania.

Source	Destination	Protocol	Info
32.12.149.62	192.168.0.104	TCP	632 > 61265 [SYN] Seq=0 Ack=0 Win=41087, bogus TC
144.80.250.84	192.168.0.104	TCP	56710 > 33605 [SYN] Seq=0 Ack=0 Win=62493, bogus
84.105.218.207	192.168.0.104	TCP	42229 > 16037 [SYN] Seq=0 Ack=0 Win=32640, bogus
131.111.191.230	192.168.0.104	TCP	61634 > 15758 [SYN] Seq=0 Ack=0 Win=38252, bogus
61.64.203.136	192.168.0.104	TCP	48697 > 61727 [SYN] Seq=0 Ack=0 Win=40960, bogus
254.203.118.111	192.168.0.104	TCP	14713 > 55463 [SYN] Seq=0 Ack=0 Win=27611, bogus
44.201.206.75	192.168.0.104	TCP	23841 > 3526 [SYN] Seq=0 Ack=0 Win=65175, bogus T
138.153.248.236	192.168.0.104	TCP	54892 > 40645 [SYN] Seq=0 Ack=0 Win=57477, bogus
84.178.237.175	192.168.0.104	TCP	49412 > 1554 [SYN] Seq=0 Ack=0 Win=49293, bogus T
142.67.218.215	192.168.0.104	TCP	47899 > 27054 [SYN] Seq=0 Ack=0 Win=7417, bogus T
81.228.153.24	192.168.0.104	TCP	59234 > 10625 [SYN] Seq=0 Ack=0 Win=16429, bogus
139.28.48.96	192.168.0.104	TCP	8493 > 3755 [SYN] Seq=0 Ack=0 Win=4937, bogus TCP
146.111.49.105	192.168.0.104	TCP	40565 > 26573 [SYN] Seq=0 Ack=0 Win=55341, bogus
41.54.33.57	192.168.0.104	TCP	46798 > 232 [SYN] Seq=0 Ack=0 Win=54163, bogus TC
242.245.53.174	192.168.0.104	TCP	24421 > 26987 [SYN] Seq=0 Ack=0 Win=52474, bogus
70.177.33.27	192.168.0.104	TCP	14821 > 37404 [SYN] Seq=0 Ack=0 Win=43958, bogus
164.203.180.202	192.168.0.104	TCP	40011 > 18112 [SYN] Seq=0 Ack=0 Win=60789, bogus
2.7.224.39	192.168.0.104	TCP	12119 > 167 [SYN] Seq=0 Ack=0 Win=24677, bogus TC
135.28.82.103	192.168.0.104	TCP	34426 > 40551 [SYN] Seq=0 Ack=0 Win=18940, bogus
185.139.28.216	192.168.0.104	TCP	14548 > 11996 [SYN] Seq=0 Ack=0 Win=1872, bogus T
19.225.172.146	192.168.0.104	TCP	745 > 3052 [SYN] Seq=0 Ack=0 Win=42535, bogus TCP
58.227.139.78	192.168.0.104	TCP	423 > 3862 [SYN] Seq=0 Ack=0 Win=42528, bogus TCP
97.50.200.108	192.168.0.104	TCP	26172 > 32209 [SYN] Seq=0 Ack=0 Win=65078, bogus
68.63.26.140	192.168.0.104	TCP	62670 > 145 [SYN] Seq=0 Ack=0 Win=19195, bogus TC
73.98.243.248	192.168.0.104	TCP	18378 > 55814 [SYN] Seq=0 Ack=0 Win=41610, bogus
229.255.209.130	192.168.0.104	TCP	64824 > 58913 [SYN] Seq=0 Ack=0 Win=36522, bogus

Rys.40. Ruch sieciowy podczas zalewu pakietów ukrywającego skanowanie

- *Falszowanie adresu nadawcy* - metoda ta może stanowić rozwinięcie pomysłu przedstawionego powyżej. W tym przypadku komputer skanujący fałszuje wszystkie adresy nadawcy dbając jedynie o to, by przynajmniej jeden z fałszowanych adresów znajdował się w jego podsieci. Dzięki temu host skanujący jest w stanie sniffować pakiety zwrotne nie zdradzając swojego adresu. Ruch sieciowy jest bardzo podobny do przedstawionego na rys. 40.
- *Skąnowanie rozproszone* - rozproszone skoordynowane skąnowanie może być wykorzystane w połączeniu z powolnym skąnowaniem w celu wykonania praktycznie niewykrywalnego skąnowania w rozsądnym czasie. Technika ta sporo zalet, jednak wymaga wcześniejszych przygotowań i znacznych zasobów.

2. Zjawisko sniffingu

Sniffing czyli węszenie jest zagrożeniem biernym. Polega na odczytywaniu danych przez węzeł, dla którego nie były one przeznaczone. Możliwość taka jest dostępna w wielu urządzeniach (np. analizator sieci). Urządzenia wykorzystujące sniffing są pożyteczne i konieczne. Mogą być jednak wykorzystywane w złych zamiarach. Np. do przechwytywania hasła, odczytywania poczty, odczytywania przesyłanych rekordów baz danych. Często węszenie stanowi etap wstępny przed przystąpieniem do ataku aktywnego.

Wszystkie interfejsy sieciowe w segmencie sieci mają dostęp do wszystkich transmitowanych w nich danych. Każdy interfejs powinien mieć inny adres. Istnieje też przynajmniej jeden adres rozgłoszeniowy (*broadcast*) odpowiadający wszystkim interfejsom. Normalnie, interfejs reaguje tylko na pakiety, które w polu adresowym mają jego adres, lub adres rozgłoszeniowy.

Sniffer przełącza interfejs w tryb podsłuchu, dzięki czemu interfejs może analizować każdy pakiet w danym segmencie sieci. Jest to bardzo przydatne narzędzie w rękach administratora, służące do ustalania przyczyn nieprawidłowego działania sieci. Można ustalić udział poszczególnych protokołów w ruchu sieciowym, udział poszczególnych hostów w generowaniu i odbieraniu pakietów.

Oprogramowanie umożliwiające *sniffing* jest w tej chwili łatwo dostępne w Internecie. Oznacza to, że mogą z niego korzystać również potencjalni intruzi. *Sniffing* danych z sieci prowadzi do utraty tajności pewnych informacji, które powinny zostać tajne. Powszechnie stosowane praktyki obejmują m.in.:

- Przechwytywanie hasła.
- Przechwytywanie numerów kont finansowych (np. kart kredytowych).
- Przechwytywanie danych prywatnych (np. zawartych w poczcie elektronicznej).
- Analizę ruchu sieciowego.
- Gromadzenie danych z usług finger,, whois, nslookup, DNS.
- Gromadzenie danych SNMP.

3. Techniki wykrywania sniferów

Wiele systemów operacyjnych udostępnia mechanizm pozwalający stwierdzić, czy interfejs sieciowy pracuje w trybie bezładnym. W systemie Linux można to stwierdzić przy pomocy polecenia *ifconfig*. Gdy interfejs pracuje w tym trybie, to w sekcji atrybutów pojawi się słowo PROMISC. Należy jednak zwrócić uwagę, że jeżeli atakujący przejmie kontrolę nad komputerem, to może podłożyć zastępcze polecenie *ifconfig*, które nie będzie sygnalizowało trybu bezładnego. W Windows można wykorzystać program *PromiscDetect*.

Część programów, monitorujących sieć przeprowadza odwrotne przeszukiwanie DNS w momencie generowania raportów wyjściowych. Ma to na celu określenie nazw komputerów o znanych adresach. Generowany jest w ten sposób dodatkowy ruch w sieci związany z DNS. Możliwe jest wobec tego monitorowanie sieci w poszukiwaniu komputerów, które przeprowadzają dużą liczbę wyszukiwań DNS. Jednak oczywiście może to być działanie przypadkowe i nie odnajdziemy podsłuchującego komputera.

Prostszym sposobem byłoby utworzenie fałszywego połączenia z adresem, który nie ma żadnego związku z lokalną siecią. Można wówczas monitorować sieć w poszukiwaniu zapytań DNS, które próbują rozwiązać sfalszowany adres, co automatycznie zdradzi podsłuchującą maszynę.

Kolejną techniką jest badanie różnic w opóźnieniu odpowiedzi na wysłane pakiety *ICMP Echo Request* (ping). Na wstępie należy sprawdzić komputer przez próbkowanie czasów odpowiedzi. Potem należy wygenerować w sieci duży ruch tak spreparowany, aby zainteresować potencjalnego snifera. W końcu czas opóźnienia jest próbkowany ponownie w celu porównania, czy zmienił się znacząco. Jednym z problemów tej metody jest fakt, że mogą wystąpić opóźnienia ponieważ medium transmisyjne będzie mocno obciążone i wzrośnie liczba kolizji.

Czasem w detekcji interfejsu pracującego w trybie bezładnym może pomóc błąd sterownika. Odkryto, że w popularnym sterowniku ethernetowym Linuxa, gdy komputer pracował w trybie bezładnym, system operacyjny nie był w stanie przeprowadzać sprawdzeń adresów MAC. Zamiast tego, sprawdzenie było przeprowadzane na poziomie protokołu IP. W normalnej sytuacji, pakiety z obcym adresem MAC

zostałyby odrzucone na poziomie sprzętowym. W trybie bezładnym tak się nie dzieje. Można więc określić, czy komputer pracuje w trybie bezładnym przez wysłanie do niej pakietu *ICMP Echo Request* z poprawnym adresem IP i niepoprawnym MAC. Jeżeli nadejdzie odpowiedź, to oznacza pracę w trybie bezładnym.

Podobnie można wysyłać zapytania ARP nie na adres rozgłoszeniowy, lecz na adres podejrzanego o sniffing komputera. Jeżeli odpowie, to oznacza pracę w trybie bezładnym.

Metoda rutowania źródłowego polega na wypełnieniu w nagłówku IP pola tej opcji. Można to wykorzystać do wykrywania sniferów pracujących w innych segmentach sieci. Wymagane jest utworzenie pakietu *ICMP Echo Request* do podejrzanego komputera ze wskazaniem trasy typu *loose-source* w celu wymuszenia jego przekierowania przez inny komputer znajdujący się w tym samym segmencie. Komputer ten powinien mieć jednak wyłączony ruting. Jeśli pojawi się odpowiedź, prawdopodobnie oznacza to, że podejrzanym przechwyił pakiet, ponieważ nie mógł go otrzymać z tego routera. Warto również sprawdzić pole TTL, aby upewnić się czy pakiet powrócił z powodu sniffingu. Jeżeli komputer wskazany jako router miałby włączony ruting, to pole TTL wskaże, czy cel odpowiedzią z przekierowania czy też bezpośrednio.

Kolejna metoda, metoda *destroy* działa w środowisku nie tylko sieci lokalnej. Metoda polega na tym, że instaluje się klienta i serwer a następnie klient loguje się do serwera za pomocą Telnetu, POP, IMAP lub innego dowolnego jawnego protokołu. Serwer jest całkowicie wirtualny, to znaczy, że nie musi mieć żadnych kont. Kiedy atakujący przechwyci dane uwierzytelniające, z pewnością spróbuje je wkrótce wykorzystać do zalogowania się. Standardowe IDS mogą zostać skonfigurowane na wykrywanie takich przypadków.

1. Co to jest enumeracja?

Enumeracją nazywamy proces wyszukiwania poprawnych kont użytkowników lub źle zabezpieczonych zasobów współdzielonych. Enumeracja jest techniką inwazyjną. Wiąże się z aktywnymi połączeniami i ukierunkowanymi zapytaniami. Większość informacji zbieranych w ten sposób wydaje się zwykle błaha. Mogą one być jednak bardzo groźne. Po zdobyciu poprawnej nazwy użytkownika lub zasobu, tylko kwestią czasu pozostaje moment, w którym intruz zdobędzie odpowiednie hasło lub znajdzie lukę związaną z protokołem udostępniania zasobu.

Do głównych rodzajów zbieranych informacji należą:

- zasoby sieciowe i ich udostępnianie,
- użytkownicy i grupy,
- aplikacje i etykiety.

Techniki enumeracji są najczęściej charakterystyczne dla konkretnego systemu operacyjnego. Stosowanie mechanizmów ochronnych pozwalających na ukrywanie informacji o rodzaju zainstalowanego systemu operacyjnego utrudnia również stosowanie odpowiednich technik enumeracji.

2. Enumeracja Windows

2.1. NetBIOS

Od początku swego istnienia Windows NT/W2K postrzegany jest jako system rozdający darmowe informacje wszystkim ciekawskim. Wynika to z wykorzystywania protokołów przesyłania danych CIFS/SMB (*Common Internet File System/ Server Message Block*) i NetBIOS.

Do zbierania informacji wykorzystywany jest często pakiet *Windows NT Resource Kit*, zwany z tego powodu *Windows NT Hacking Kit*. Zawiera on kolekcję narzędzi, które są bardzo pomocne dla administratora systemu, ale mogą być również wykorzystane przez agresorów do zdobywania cennej informacji.

Wymienione poprzednio protokoły wykorzystują porty 135, 139, oraz 445 w Windows 2000. Pierwszym krokiem podczas zdalnego korzystania z tych interfejsów jest utworzenie nieautoryzowanego połączenia z systemem. Używa się w tym celu tzw. polecenia *pustej sesji*, np:

```
net use \\192.168.1.2\IPC$ "" /user:""
```

Powoduje to połączenie z ukrytym zasobem komunikacyjnym IPC\$ jako anonimowy użytkownik z pustym hasłem. Zasób ten jest używany do komunikacji między procesowej i z racji swojego przeznaczenia umożliwia zewnętrznym procesom utworzenie anonimowego połączenia. Po stworzeniu takiej anonimowej sesji, tester nie będzie miał co prawda praw dostępu do zasobów, ale będzie mógł zidentyfikować udostępnione zasoby i użytkowników na testowanym komputerze. Większość technik korzysta z tej charakterystycznej luki w zabezpieczeniach.

Nazwy NetBIOS w standardowej postaci mają długość 16 znaków. Ostatni 16 znak określa rodzaj zasobu lub usługę związaną z nazwą. Niektóre przyrostki NetBIOS przedstawiono w tabeli na Rys. 1. Na Rys. 2, 4 pokazano przykłady enumeracji systemu Windows z wykorzystaniem różnych programów.

Nazwa NetBIOS	Przyrostek	Usługa
<nazwa komputera>	00	Workstation
<nazwa komputera>	01	Messenger
<nazwa komputera>	03	Messenger
<nazwa komputera>	06	RAS Server
<nazwa komputera>	21	RAS Client
<nazwa komputera>	30	Modem Sharing Server
<nazwa komputera>	20	Server
<nazwa użytkownika>	03	Messenger
<nazwa domeny>	00	Domain Name
<nazwa domeny>	1B	Domain Master Browser
<nazwa domeny>	1C	Domain Controller
<nazwa domeny>	1E	Browser Service Election
<_MS_BROWSE_>	01	Master Browser
<INet~Services>	1C	IIS
<IS~nazwa komputera>	00	IIS

Rys. 1. Przyrostki NetBIOS

```

MS-DOS Command Prompt
H:\>NBTSTAT -A 172.16.50.152 -c -r -S

                NetBIOS Connection Table

Local Name      State      In/Out  Remote Host      Input      Output
-----
TESTOWY         <00>      Connected  Out    172.16.50.150    220B      445B
TESTOWY         <00>      Connected  Out    172.16.50.150    166KB     2KB
TESTOWY         <03>      Listening
ADMINISTATOR    <03>      Listening
H:\>
    
```

Rys. 2. Enumeracja przy pomocy programu NBTSTAT

```

MS-DOS Command Prompt
H:\>srvcheck \\testowy
\\testowy\NETLOGON
    .\Everyone
\\testowy\Dokumenty
    DOMENA_TESTOWA\Domain Users
\\testowy\Aplikacje
    .\Everyone
\\testowy\Video
    DOMENA_TESTOWA\Domain Guests
    DOMENA_TESTOWA\Domain Users
H:\>
    
```

Rys. 3. Enumeracja przy pomocy programu SRVCHECK

```

MS Command Prompt
H:\>rmtshare \\testowy

Share name      Resource                Remark
-----
NETLOGON        H:\NT_TEST\system32\Repl\Imp... Logon server share
ADMIN$          H:\NT_TEST              Remote Admin
IPC$            C:\                     Remote IPC
C$              D:\                     Default share
D$              E:\                     Default share
E$              F:\                     Default share
Video           F:\                     Default share
F$              G:\                     Default share
G$              H:\                     Default share
H$              I:\                     Default share
Aplikacje       J:\                     Default share
N$              K:\                     Default share
Dokumenty       L:\                     Default share
The command completed successfully.

H:\>rmtshare \\testowy\Aplikacje
Share name      \\testowy\Aplikacje
Path            F:\
Remark
Maximum users   No limit
Users           0
Permissions:
  \Everyone    : READ
The command completed successfully.

H:\>rmtshare \\testowy\admin$
Share name      \\testowy\admin$
Path            H:\NT_TEST
Remark          Remote Admin
Maximum users   No limit
Users           0
No permissions specified.
The command completed successfully.

H:\>

```

Rys. 4. Enumeracja przy pomocy programu RMTSHARE

Ponieważ puste sesje wymagają dostępu do portu 139, więc najprostszą metodą ich powstrzymania jest filtrowanie ruchu związanego z NetBIOS. Można również wyłączyć usługi *Alerter* i *Messenger*. Można również wykorzystać tzw. poprawkę *RestrictAnonymous*. Polega ona na umieszczeniu w kluczu **HKLM\SYSTEM\CurrentControlSet\Control\LSA** następującej wartości:

Nazwa wartości: *RestrictAnonymous*

Typ danych: REG_DWORD

Wartość: 2

2.2. SNMP

System Windows może również udostępniać informacje podobne jak przy enumeracji NetBIOS, jeżeli zostanie na nim uruchomiony agent SNMP. Standardowo SNMP nie jest instalowany. W SNMP, serwerem jest system zarządzający klientem, agent. Jedyną operacją jaką może zainicjować agent jest pułapka, czyli uzyskanie informacji o jakimś zdarzeniu. Pozostałe operacje inicjowane są przez serwer. Jedynym zabezpieczeniem przesyłanych danych jest prymitywne uwierzytelnienie. Polega ono na przynależności do tzw. wspólnot i znajomości nazwy wspólnoty. Aby przechwycić informacje przesyłane przy pomocy SNMP wystarczy znać nazwę wspólnoty. Nazwą standardową jest PUBLIC.

Informacje o monitorowanych obiektach przechowywane są w hierarchicznej bazie MIB (*Management Information Base*). Dzięki temu serwer wie o co może spytać agentów i w jakim formacie otrzyma odpowiedź (RFC 1213). Typowe, możliwe do uzyskania informacje to:

- uruchomione usługi,
- nazwy zasobów sieciowych,

- nazwy użytkowników,
- nazwy domen,
- nazwy komputerów,
- szczegółowe informacje dotyczące konfiguracji urządzeń.

Obrona przed tego typu działaniem może polegać na:

- usunięciu agenta SNMP lub wyłączenie (niewłączenie) usługi SNMP,
- skonfigurowaniu prywatnej nazwy wspólnoty,
- określeniu dla agenta, adresów zaufanych serwerów,
- modyfikacji rejestru aby dopuszczać jedynie autoryzowany dostęp do nazwy wspólnoty,
- blokadzie portu 161 TCP i UDP w granicznych urządzeniach kontroli dostępu.

2.3. DNS

Przeźreń nazw *Active Directory* bazuje na DNS. Klienci dzięki rekordowi SRV (opisanemu w RFC 2052) mogą lokalizować w sieci serwery poszczególnych usług. Intruz może uzyskać informacje o usługach poprzez transfer strefy (np. program *nslookup*). Symbole przykładowych usług to:

`gc._tcp` katalog globalny (port 3268)

`_kerberos._tcp` kontroler domeny wykorzystujący *Kerberos* (port 88)

`_ldap._tcp` serwer LDAP (port 389)

2.4. SID

Zwykle duża część pracy związanej z uzyskaniem dostępu do konta, to zdobycie nazwy użytkownika. Ponieważ większość użytkowników stosuje łatwe hasła, więc często jest to prawie cały wysiłek z tym związany.

SID to identyfikator zabezpieczeń przypisywany systemowi podczas instalacji. Znając SID systemu można sprawdzić praktycznie każde konto i grupę użytkowników w tym systemie. Identyfikator użytkownik zawiera dodatkowy (ostatni) człon tzw. RID, czyli identyfikator względny. Dla kont i grup wbudowanych jest on ustalony: Administrator (500), Guest (501), konta lokalne lub domenowe (1000, 1001, 1002, itd.).

Przeciwdziałanie enumeracji kont realizuje się jak w przypadku przeciwdziałania zapytaniom NetBIOS. Innym sposobem jest wyłączenie usług *Alert* i *Messenger*.

UserName	ctExpiresTime	LastLogonTime	LastLogonServer	LogonHours	Sid
Administrator	ver	11/18/01 11:07 PM	testowy	All	S-1-5-21-251926106-347765527-208020174-500
Administrator	ver	11/18/01 11:07 PM	testowy	All	S-1-5-21-251926106-347765527-208020174-500
Administrator	ver	11/18/01 11:07 PM	testowy	All	S-1-5-21-251926106-347765527-208020174-500
Guest	ver	Never	testowy	All	S-1-5-21-251926106-347765527-208020174-501
IUSR_TESTOWY	ver	11/18/01 8:45 PM	testowy	All	S-1-5-21-251926106-347765527-208020174-1001
IUSR_TESTOWY	ver	11/18/01 8:45 PM	testowy	All	S-1-5-21-251926106-347765527-208020174-1001
TESTOWY\$	ver	Never	testowy	All	

Found 4 users

Rys. 5. Enumeracja SID

Informację w czasie enumeracji można również realizować poprzez łączenie się ze zdalnymi aplikacjami i obserwowanie zwracanych przez nie danych. Nazywamy to przechwytywaniem etykiet. Najprostsza metoda to nawiązanie połączenia przez **telnet** ze znanym portem wybranego na cel serwera. W razie potrzeby należy wcisnąć kilka razy **Enter**.

Obrona wymaga dużo pracy ze strony administratora. Należy sporządzić listę aplikacji, które muszą funkcjonować a następnie znaleźć odpowiednie dla nich metody wyłączenia pokazywania informacji o ich producencie, wersji, itp. Co pewien czas należy dokonywać skanowania aby upewnić się, że niepożądane informacje faktycznie nie są udostępniane.

2.5. Przechwytywanie etykiet

Informację można również uzyskać poprzez łączenie się ze zdalnymi aplikacjami i obserwowanie zwracanych przez nie danych. Nazywamy to przechwytywaniem etykiet lub pozyskiwaniem banerów. Najprostsza metoda to nawiązanie połączenia przez **telnet** ze znanym portem wybranego na cel serwera. W razie potrzeby należy wcisnąć kilka razy **Enter**.

Obrona wymaga dużo pracy ze strony administratora. Należy sporządzić listę aplikacji, które muszą funkcjonować a następnie znaleźć odpowiednie dla nich metody wyłączenia pokazywania informacji o ich producencie, wersji, itp. Co pewien czas należy dokonywać skanowania aby upewnić się, że niepożądane informacje faktycznie nie są udostępniane.

1. Co to jest spoofing?

Spoofing, czyli atak poprzez podszywanie się w tradycyjnym ujęciu oznacza działanie atakującego, polegające na oszukaniu mechanizmu autentykacji zachodzącego pomiędzy maszynami przekazującymi między sobą pakiety. Proces autoryzacji przeprowadzany jest poprzez sfalszowanie pakietów "zaufanego" hosta - należącego do atakowanej sieci. Obecnie mianem *spoofingu* określa się dowolną metodę łamania zabezpieczeń opartych na adresie lub nazwie hosta. Istnieje kilka technik podszywania. Spoofing nie jest cechą ściśle określonej warstwy OSI. Można go realizować praktycznie w każdej warstwie.

Ataki tego typu należą do grupy "technik zaawansowanych". Naruszenie bezpieczeństwa następuje w bardzo dyskretny sposób. Techniki spoofingu są bardziej skomplikowane niż inne, przez co rzadko dochodzi do ataków z wykorzystaniem tej metody.

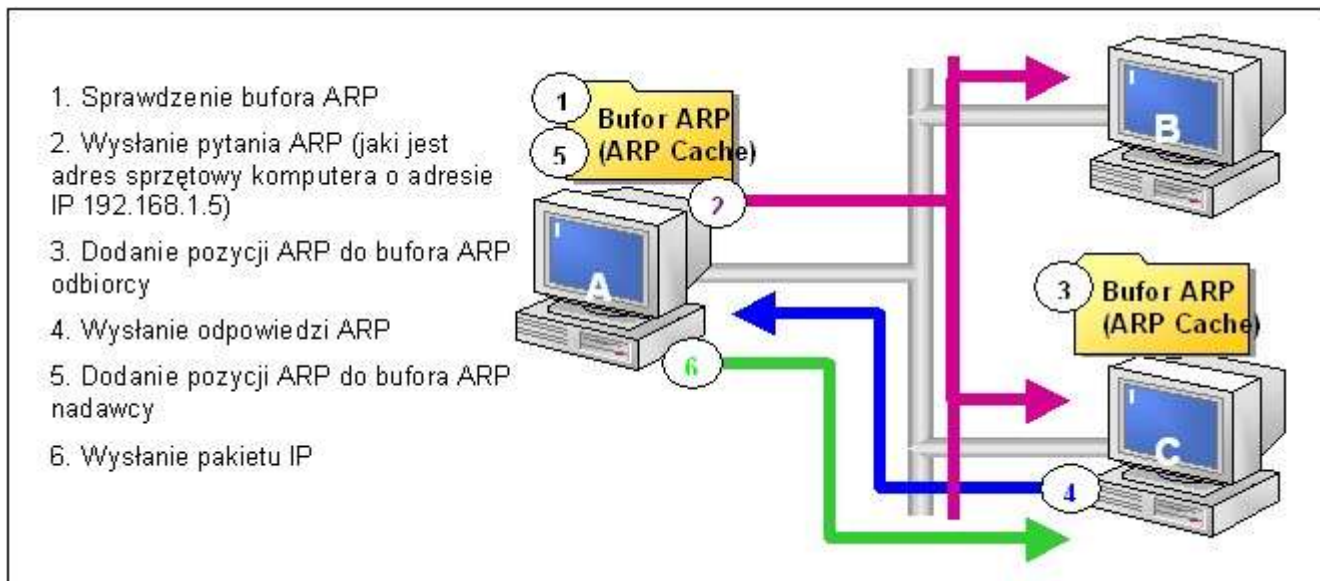
2. Spoofing ARP

ARP (*Address Resolution Protocol*) jest protokołem odpowiedzialnym za konwersję adresu IP na adres sprzętowy. Gdy datagram IP jest gotowy do wysłania, host musi dowiedzieć się, jaki jest adres sprzętowy skojarzony z docelowym adresem IP. Dla pakietów wysyłanych wewnątrz sieci lokalnej, będzie to adres interfejsu docelowego. Dla pakietów skierowanych na zewnątrz, będzie to adres jednego z *routerów*.

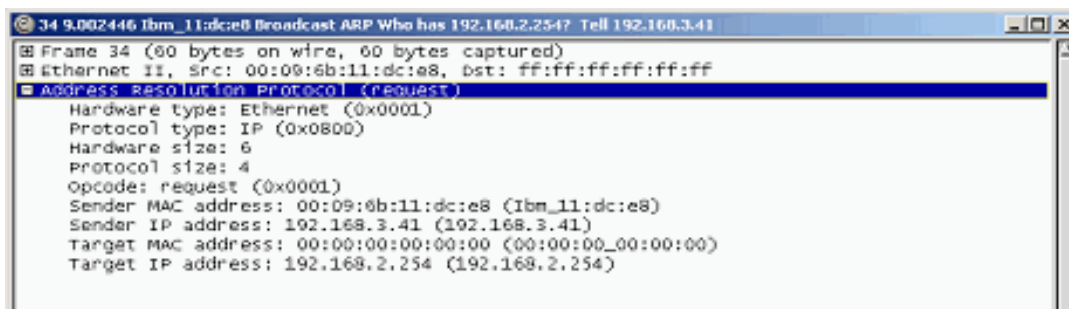
Aby zdobyć poszukiwany adres sprzętowy, host wysła zapytanie ARP używając sprzętowego adresu ogólnego (*hardware broadcast address*). Pytanie brzmi: *Jaki jest adres sprzętowy skojarzony z podanym adresem IP?* Powinien odpowiedzieć co najwyżej jeden host z sieci lokalnej. Pytanie zawiera adres IP nadawcy. Wszyscy, do których pytanie dotrze, mogą zapisać skojarzenie adresu IP i adresu sprzętowego nadawcy pytania. Zrobi to na pewno wywołany host gdyż musi odpowiedzieć na pytanie. Skojarzenia te zapisywane są w buforze ARP (*ARP cache*). Ulegają one przeterminowaniu po kilku minutach. Zawsze przed wysłaniem zapytania ARP ma miejsce sprawdzenie bufora ARP. Po przeterminowaniu wysyłane jest pytanie odświeżające zapis ARP. Jeżeli odpowiedź nie nadejdzie, to zapis jest usuwany z bufora. Jeżeli przed usunięciem zapisu z bufora stary komputer zostanie odłączony od sieci i pojawi się nowy z tym samym adresem IP i innym adresem sprzętowym, to odpowie on na pytanie odświeżające i nastąpi modyfikacja w buforach nadawcy pytania. Na rys. 1 przedstawiono przebieg opisanego procesu, a na rys. 2 można obejrzyć strukturę zapytania ARP.

Gdy dwa komputery mają ten sam adres IP, to oba odpowiedzą na zapytanie dotyczące tego adresu. Niektóre systemy mogą zignorować drugi komunikat, inne nadpiszą pierwszy zawartością drugiego. Systemy mogą również wysłać komunikat zerowania (RESET TCP/IP). Systemy nie

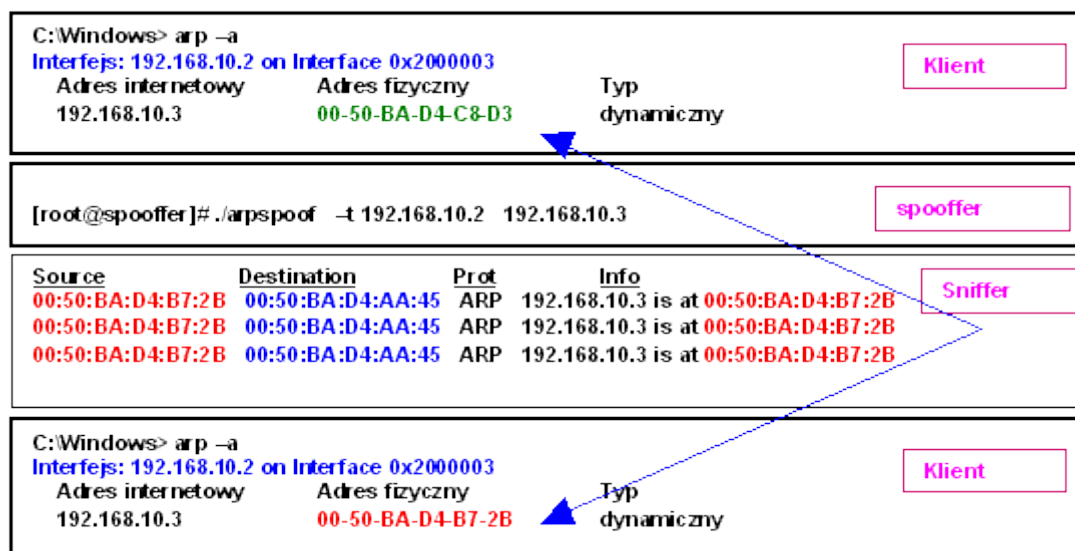
muszą sprawdzać, czy powtórzony komunikat pochodzi z tego samego źródła, czy jest próbą *spoofingu*. Włamywacz może spowodować odłączenie od sieci komputera pod który chce się podszyc i wejść w ten sposób w jego miejsce. Może jednocześnie próbować zmienić jego adres IP. Na rys. 3. przedstawiono przebieg procesu *spoofingu* ARP.



Rys. 1. Funkcjonowanie ARP



Rys. 2. Zapytanie ARP przechwycone przez sniffer



Rys. 3. Przebieg spoofingu ARP

Istotą *spoofingu* jest to, że jest on skierowany przeciwko komputerowi oszukiwanemu, a nie temu, którego adres IP został przejęty. W parze tej komputer oszukiwany jest elementem ufającym a ten, którego adres przejęto - zaufanym. Wynika z tego, że maszyny ufające nie powinny korzystać z ARP do wykrywania adresów sprzętowych komputerów zaufanych. Zamiast tego zapis taki powinien być wprowadzony do bufora ARP jako tzw. *zapis permanentny*, który nie ulega przeterminowaniu. Nie będą wysyłane pytania ARP. Ewentualne odpowiedzi, które nie były poprzedzone pytaniem nie są obsługiwane.

Wadą zapisów permanentnych jest możliwość wysyłania pakietów do niedziałających komputerów. Wadą jest również konieczność zmian wpisów w przypadku zmiany konfiguracji. Bufory ARP mają ograniczoną pojemność, co limituje liczbę zapisów permanentnych lub ogranicza czas ważności zapisów dynamicznych. W systemach Unix, Windows dostępne jest polecenie **arp** umożliwiające:

- wyświetlanie zapisów znajdujących się w buforze,
- usuwanie zapisów z bufora,
- wstawianie zapisów permanentnych,
- wstawianie grupy zapisów z pliku.

Zagrożenie *spoofingu* pomiędzy podsieciami IP jest usuwane przez zastosowanie barier sprzętowych w postaci *routerów* z permanentnymi zapisami ARP. Jeżeli komputery zaufane pracują w podsieci narażonej na *spoofing* ARP, to dzięki zapisom permanentnym *router* nie zostaną oszukane. Należy jeszcze zadbać aby komputery zaufane były chronione przed *spoofingiem* ARP udającym *router*. *Router* są jednak zwykle dobrze zabezpieczone i nie przerywają pracy.

Jeżeli nie istnieje możliwość wprowadzenia bariery sprzętowej, to zostaje szybkie wykrywanie *spoofingu* i natychmiastowa interwencja. Należy opracować procedury postępowania w takich przypadkach gdyż wykryta nieprawidłowość może być zamierzona, przypadkowa, lub być naruszeniem bezpieczeństwa.

Pasywna detekcja na poziomie hosta

Komputer odpowiadający na pakiet ARP powinien badać nie tylko adres odbiorcy ale i adres IP nadawcy. Jeżeli stwierdzi, że to jest jego adres, to może oznaczać, że inny komputer podszywa się pod niego. Takiej kontroli dokonuje większość systemów.

Aktywna detekcja na poziomie hosta

Hosty powinny wysłać pytania ARP ich własnych adresów przy starcie systemu jak i regularnie później. Jeżeli odpowiedź ARP nadejdzie, to może to oznaczać wykrycie *spoofingu*.

Detekcja na poziomie serwera

Polega na weryfikacji pytania ARP przez pytanie RARP dotyczące adresu sprzętowego zawartego w odpowiedzi. Pytanie RARP brzmi: *Jaki jest adres IP skojarzony z podanym adresem sprzętowym?*. Protokół RARP jest normalnie używany przez stacje bezdyskowe, które podczas startu muszą sprawdzić swój adres IP. Metoda pytania odwrotnego jest bardzo skuteczna również w wielu innych sytuacjach.

Detekcja na poziomie sieci przez okresowe kontrole

Okresowe kontrole powinny dotyczyć zawartości buforów ARP. Można wtedy wykryć w nich zmiany adresów. Zadaniem personelu administracyjnego powinno być bieżące utrzymywanie bazy danych z adresami sprzętowymi, adresami IP, nazwami DNS itp. Bazy takie mogą być podstawą okresowo przeprowadzanych automatycznych kontroli. Można wykorzystać protokół SNMP. W SNMP każdy komputer korzystający z IP ma agenta SNMP odpowiadającego na żądania dotyczące informacji i konfiguracji. W niektórych standardach SNMP dostępne są tabele opisujące adresy sprzętowe i IP.

Detekcja na poziomie sieci przez ciągłe monitorowanie

Interfejs sieciowy można przełączyć w *tryb ogólny*, dzięki któremu możliwe jest ciągłe analizowanie każdego pakietu w sieci. Można wtedy dodatkowo realizować analizę ruchu w sieci i opracowywać odpowiednie statystyki. Do przeprowadzenia takiego niskopoziomowego monitoringu sieci stworzono agentów SNMP z obsługą protokołu RMON.

3. Spoofing usługi routingu

Decyzja dotycząca *routingu*, to odpowiedź na pytanie: *Dokąd wysłać datagram o danym adresie IP?* Jeżeli adres docelowy zgadza się z adresem sieci podłączonej do jednego z interfejsów, wtedy datagram jest skierowywany bezpośrednio pod adres docelowy. W innym przypadku wybierany jest *router*, który przekaże datagram dalej.

Spoofing routingu polega na skłanianiu komputerów do przesyłania datagramów w miejsca inne niż te, do których powinny trafić. Może to doprowadzić do odmowy usługi. Maszyna, do której są kierowane pakiety nie odpowiada. Może zostać przechwycony wszelki ruch pomiędzy sieciami. W trakcie tego działania można prowadzić filtrowanie ruchu, wprowadzać modyfikacje, tworzyć wrażenie poprawnego funkcjonowania sieci.

Jeżeli w sieci dostępnych jest kilka *routerów* (w tym domyślny) to może się zdarzyć, że ten do którego dotrze datagram uzna, że inny będzie bardziej właściwy, to przesyła do niego datagram, a do komputera źródłowego wysyła komunikat ICMP (*Internet Control Message Protocol*)

informujący o zmianie kierunku. Komunikat ten mówi: *datagramy do sieci A.B.C.D lepiej jest przesyłać poprzez router W.X.Y.Z*. Komputer, który otrzyma ten komunikat powinien uaktualnić swoje tablice *routingu*. Datagram nie jest gubiony i nie jest potrzebne jego powtórne wysłanie, gdyż już to zrobił *router*.

Jeżeli komputer ignoruje komunikaty ICMP, to pakiety będą dostarczane mniej efektywnie. Jest to jednak metoda uniknięcia najprostszej techniki *spoofingu routingu* - wysłaniu komunikatów ICMP o zmianie kierunku. Wiele systemów nie sprawdza ważności tych komunikatów. Powinno przynajmniej nastąpić sprawdzenie, czy komunikaty takie pochodzą z jednego ze znanych *routerów*.

Można się również zabezpieczać poprzez sprawdzenie czy bufor ARP hostów mają zapisy permanentne dotyczące adresów autoryzowanych *routerów*. Zapobiega to *spoofingowi* ARP, w których jeden z komputerów mógłby udawać *router*. Mógłby on wtedy przechwytywać wszelki ruch wychodzący z sieci.

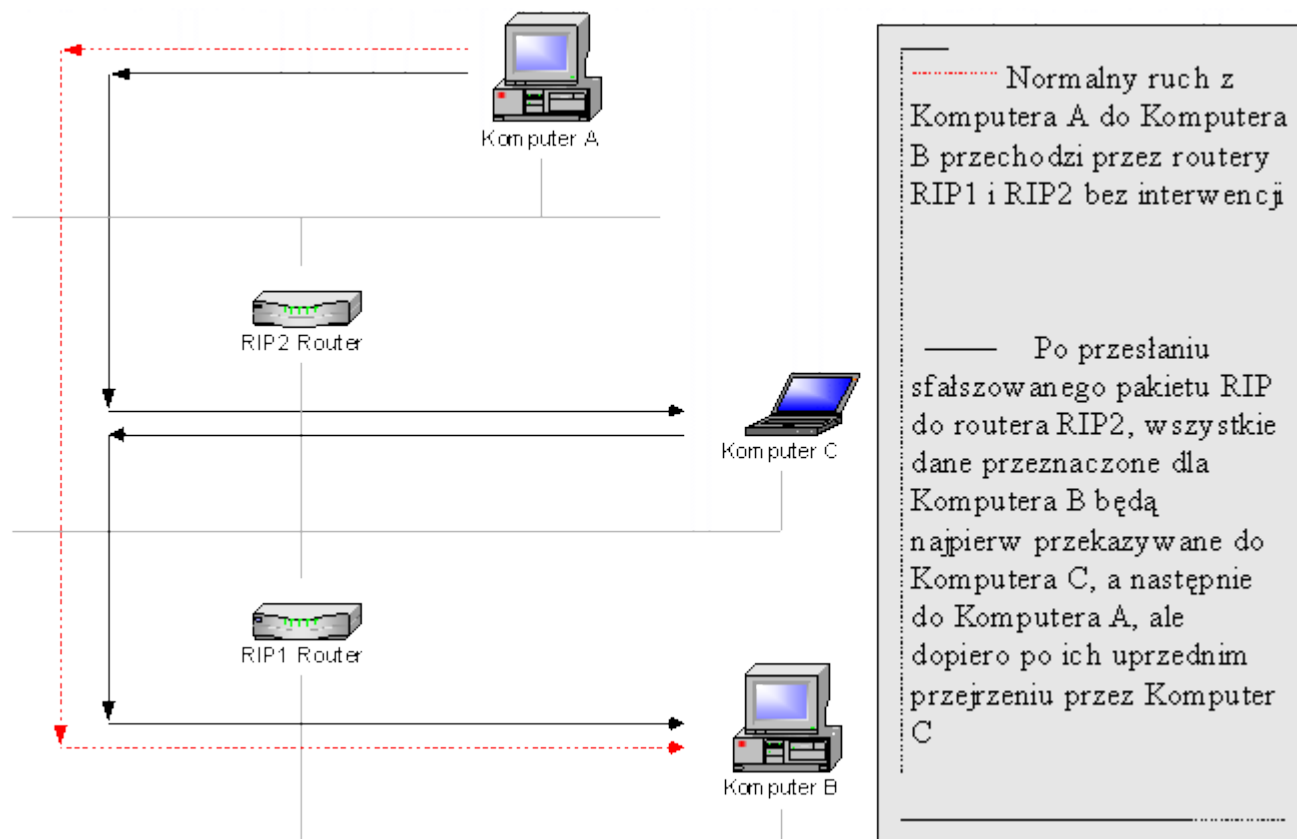
Protokół RIP (*Routing Information Protocol*) jest protokołem *routingu*, który wykorzystywany jest do informowania maszyn znajdujących się w sieci o *routerach*. Protokoły *routingu* stosuje się do przesyłania pakietów do poszczególnych sieci docelowych. Poza tym routery potrzebują protokołów *routingu* do wymiany informacji, o ile nie zostały zastosowane ręcznie konfigurowane tabele *routingu*. Protokoły *routingu* mogą być również narażone na atak, który doprowadza do powstania błędnych tablic w *routerach* jak i w zwykłych maszynach. Protokół wektorowy RIP, jest najczęściej używanym protokołem *routingu*.

Jeśli w sieci komputerowej znajdują się urządzenia routujące, bardziej zaawansowani włamywacze zaczną szukać *routerów* obsługujących protokół RIP v1 lub RIP v2. Spowodowane jest to tym, iż dane powyższego protokołu można w prosty sposób sfalszować (patrz slajd)

Włamywacz może więc w prosty sposób przesłać pakiety do *routera* RIP z nakazem przesłania ich do nieautoryzowanego systemu lub sieci. Metoda polegająca na fałszowaniu protokołu RIP może wyglądać w następujący sposób:

1. Włamywacz identyfikuje *router* RIP, który chce zaatakować przez szukanie nasłuchującego portu 520 UDP,
2. Znalezienie tablic routowania, które co pewien czas rozgłaszane są przez RIP,
3. Wybranie najlepszej drogi ataku. Może być to np. dodanie do *routera* RIP własnej ścieżki routowania,
4. Od tej pory wszystkie pakiety zostaną przekierowane do systemu agresora.

Na rys. 4 przedstawiono przykład *spoofowania* usługi *routingu*.



Rys. 4. *Spoofing* routingu

Wynika z tego, że należałoby zastosować jedną z opcji:

- zaprzestać używania pasywnego protokołu RIP na *routerach*,
- bardzo ostrożnie używać pasywnego protokołu RIP na *routerach*.

Aby zachować bezpieczeństwo, pasywny uczestnik RIP powinien brać pod uwagę tylko informacje z zaufanych źródeł. Powszechnie stosowany demon **routed** jest przesadnie ufny. Należałoby stosować taki, który np. będzie podczas startu sprawdzać plik konfiguracyjny zawierający m.in. adresy zaufanych źródeł informacji RIP. Dobrym rozwiązaniem będzie też stosowanie innych protokołów rutowania dynamicznego, mających wbudowane mechanizmy zabezpieczeń, które ograniczają możliwości fałszowania. Rezygnacja z protokołu routingu RIP na pewno znacznie utrudni atakującemu dostęp do sieci komputerowej firmy.

4. Spoofing DNS

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. Przypomnijmy sobie teraz kilka podstawowych pojęć związanych z DNS:

- **domena prosta** zawiera rekordy, które umożliwiają translację nazw na adresy IP, czyli umożliwia odpowiadanie na proste pytania DNS,
- **domena odwrotna** zawiera rekordy, które umożliwiają translację adresów IP na nazwy, czyli umożliwia odpowiadanie na odwrotne pytania DNS,
- **serwerem autorytatywnym** jest serwer odpowiedzialny za utrzymywanie dokładnej i pewnej informacji o domenie,
- **serwerem pierwotnym** jest serwer autorytatywny stanowiący pierwotne źródło informacji o domenie,
- **serwerem wtórnym** jest serwer autorytatywny, który okresowo pobiera plik strefowy z serwera głównego,
- **serwerem notatnikowym** (*Caching-Only*) jest serwer nieautorytatywny, który otrzymuje odpowiedzi od innych serwerów, zapamiętuje je i jest wobec tego w stanie udzielać odpowiedzi klientom,
- **pytania iteracyjne** jest pytaniem, na które serwer zwraca w odpowiedzi adres serwera autorytatywnego, który powinien znać odpowiedź - jeżeli sam nie potrafi na nie odpowiedzieć,
- **pytania rekursywne** jest pytaniem, na które serwer zwraca ostateczną odpowiedź, nawet jeżeli sam nie potrafi na nie odpowiedzieć - poszukuje odpowiedzi u innych serwerów.

W komunikacie DNS występuje pole identyfikatora komunikatu TID (*Transaction ID*). Miało ono:

- umożliwiać oprogramowaniu DNS poszczególnych transakcji (zapytań i odpowiedzi),
- uniemożliwiać intruzowi podszywanie się.

Pole identyfikatora ma 16 bitów, zatem można wygenerować 65535 identyfikatorów. W najprostszym przypadku intruz wyśle pakiety zawierające wszystkie możliwe identyfikatory (65535 pakietów). Ponieważ standardowa odpowiedź liczy około 200 bajtów, więc trzeba wysłać w sieć kilkanaście MB. Intruz uzyskuje jednak pewność, że jeden z tych pakietów zostanie uznany za właściwy. Trudność przeprowadzenia ataku polega jedynie na tym aby udało się wysłać dane przed udzieleniem odpowiedzi przez właściwy serwer DNS. Wykorzystuje się w tym celu ataki DoS.

Jeżeli chodzi o pola adresów IP, to oczywiście muszą one wskazywać adres komputera oszukiwanego, oraz adres serwera, pod który podszywa się intruz. Uzyskanie listy adresów serwerów DNS dowolnej domeny jest bardzo łatwe.

Kolejne dane, które musi ustalić atakujący, to port źródłowy i docelowy. Port źródłowy w preparowanym pakiecie, to zawsze 53. Z portem docelowym sytuacja jest trudniejsza, ponieważ teoretycznie komputer atakowany mógł wysłać zapytanie DNS z dowolnego wolnego portu. Praktyka wskazuje jednak, że jest to port 53 lub 1024. Testy wykazują ponadto, że odpowiedzi kierowane na port 53 są zawsze akceptowane, bez względu na to z jakiego portu wysłano zapytanie. Jest to prawdopodobnie błąd implementacji serwera.

Okazuje się ponadto, że można skrócić odpowiedź DNS do około 100 bajtów (z około 200), co znacznie ogranicza globalną ilość i czas wysyłania danych przez intruza.

Co powinien wiedzieć intruz, aby mógł przeprowadzić atak?

- Kiedy atakowany komputer zada pytanie, na które intruz chce udzielić odpowiedzi?
- Jaki powinien być identyfikator odpowiedzi?
- Jaki powinien być adres źródłowy odpowiedzi?

W pamięci podręcznej, serwer DNS przechowuje informacje dotyczące ostatnio zrealizowanych mapowań. Czas przechowywania zapisany jest w polu TTL komunikatu DNS. Jeżeli atakujący sam zada pytanie serwerowi i uczestnicząc w przygotowaniu odpowiedzi umieści w komunikacie fałszywe odwzorowanie z dużym TTL, to informacje podstawione przez intruza przez długi okres czasu będą przebywać w pamięci podręcznej serwera DNS. Każdy pytający dostanie sfałszowaną odpowiedź.

Ustawienie właściwego identyfikatora transakcji dla serwerów w Windows jest łatwe, gdyż generuje on identyfikatory przewidywalne (inkrementacja o 1). Dla serwera BIND należy wysłać paczkę pakietów odpowiedzi, gdyż numer transakcji jest losowany.

Scenariusz ataku

Atak na serwer dns1.firma.com ma na celu umieszczenie w jego pamięci podręcznej sfałszowanego zapisu dotyczącego serwera wazny.abc.com. Celem intruza jest aby użytkownik korzystający z serwera DNS dns1.firma.com, chcący połączyć się z serwerem wazny.abc.com uzyskiwał połączenie z serwerem intruz.

Krok 1: Uzyskanie listy serwerów DNS domeny abc.com. Np. poprzez usługę *whois*.

Krok 2: Zapytanie o dowolny komputer z domeny abc.com. Np. www.abc.com.

Np. *nslookup* www.abc.com dns1.firma.com

Powoduje to, że serwer dns1.firma.com zapamięta w swojej pamięci podręcznej dane dotyczące serwerów obsługujących domenę abc.com oraz dane komputera www.abc.com.

Krok 3: Atak DoS na serwery DNS domeny abc.com. Celem tego ataku jest uniemożliwienie udzielania odpowiedzi DNS przez te serwery.

Krok 4: Intruz wysyła zapytanie o adres serwera wazny.abc.com:

nslookup wazny.abc.com dns1.firma.com

Krok 5: Intruz zaczyna wysyłać do serwera dns1.firma.com sfałszowane odpowiedzi na pytania dotyczące komputera wazny.abc.com. Adres źródłowy wskazuje na jeden z serwerów DNS domeny abc.com, wartość TTL jest maksymalna z możliwych, jako adres komputera wazny.abc.com ustawiony jest adres komputera intruz. Wysyłanych jest 65535 odpowiedzi, każda z innym identyfikatorem transakcji.

Krok 6: Sprawdzenie, czy sfałszowane odwzorowanie zostało umieszczone w pamięci podręcznej serwera dns1.firma.com

nslookup wazny.abc.com dns1.firma.com

Jako adres serwera wazny.abc.com powinien zostać wyświetlony adres komputera intruz.

Krok 7: Dowolny klient z domeny firma.com wydaje polecenie połączenia z serwerem wazny.abc.com. W rzeczywistości uzyskuje połączenia z komputerem intruz.

Obrona przed atakiem

- Przeprowadzenie ataku będzie utrudnione jeżeli wyłączymy stosowanie pamięci podręcznej przez serwery DNS. Obniży to wydajność usługi DNS ale zmusi intruza do dokładnego przewidzenia kiedy jego ofiara zleci serwerowi DNS dokonanie mapowania. A to jest trudne.
- Próba obrony powinna polegać również na sprawdzeniu wszystkich odpowiedzi na pytania odwrotne za pomocą pytań prostych. Testy takie mogą być skuteczne jeżeli włamywacz zmienił pliki związane z pytaniami odwrotnymi a nie zmienił plików związanych z pytaniami prostymi. Ponadto pliki takie mogą być przechowywane na różnych serwerach i może się zdarzyć, że tylko jeden z nich zostanie przechwycony.
- Można wykorzystywać pytania iteracyjne zamiast rekursywnych. Gdy serwer nazw udziela odpowiedzi nieautorytatywnej na pytanie iteracyjne, to odpowiada nazwą serwera, który prawdopodobnie zna odpowiedź autorytatywną. Przechwycony serwer nazw może skierować pytanie do innego serwera pod kontrolą włamywacza lub stwierdzić, że sam jest autorytatywny. Test na autorytatywność powinien wykryć atak.

- Test na autorytatywność wymaga skierowania pytania do serwera nazw poziomu głównego i zacząć poszukiwania od góry, schodząc stopniowo coraz niżej. Jest to procedura pracochłonna i nie pomaga gdy przechwycony został serwer autorytatywny. Ponieważ standardy DNS wymagają aby dane dotyczące każdej z domen były powielone na komputerach nie znajdujących się w tej samej sieci ani nie pobierających energii z tego samego źródła (tzw. *wspólny punkt awarii*), więc nie wydaje się prawdopodobne przechwycenie wszystkich autorytatywnych serwerów DNS danej domeny. Ponieważ jednak jeden z serwerów pełni rolę podstawowego (*primary*), więc jego przechwycenie spowoduje, że po pewnym czasie błędne dane będą znajdowały się na wszystkich serwerach autorytatywnych danej domeny.
- Rozwiązaniem, które należy również rozważyć, przynajmniej w niektórych sieciach, jest zrezygnowanie z DNS i korzystanie z tablic statycznych.
- Dokument RFC 1788 proponuje rozwiązanie polegające na wykorzystaniu ICMP. Komputery powinny odpowiadać na komunikat ICMP proszący o zbiór nazw odpowiadających podanemu adresowi IP. Potem można przeprowadzić weryfikacje poprzez proste pytania DNS.
- Kolejną z metod ochrony serwerów DNS przed spoofingiem jest zastosowanie *DNS Security* (DNSSEC). W 1997 roku zaczęto myśleć nad systemem zabezpieczeń i uwierzytelniania w DNS. Nazwano go właśnie DNSSEC (*Domain Name Server Security*). Jest to protokół służący do bezpiecznej dystrybucji nazw domen, umożliwiający serwerowi DNS podpisywanie swoich odpowiedzi.

Porywanie sesji (hijacking)

Porwanie połączenia TCP jest aktywnym atakiem polegającym na przechwyceniu części sesji użytkownika oraz wysłaniu danych w jego imieniu. *Hijacking* możliwy jest w przypadku protokołów, które nie zabezpieczają transmisji poprzez szyfrowanie. Przesyłane czystym tekstem dane mogą zostać podsłuchane, przechwycone, a atakujący ma możliwość podania się za którąkolwiek ze stron i wysłania dowolnych informacji. *Hijacking* wykorzystuje istniejące połączenie zainicjowane przez użytkownika. Porywanie sesji ma więc sens jedynie po fazie autentykacji, kiedy połączenie wykorzystuje hasło. Dzięki temu atakujący zyskuje możliwość wykonania poleceń na serwerze z uprawnieniami zalogowanego klienta. Jest to kradzież zaufania serwera do klienta.

Aby móc porwać sesję atakujący musi znajdować się na drodze pomiędzy klientem i serwerem. Najprostszy przypadek to komputery znajdujące się w tym samym segmencie sieci LAN w jednej domenie kolizyjnej. W takim przypadku atakujący widzi wszystkie pakiety bez stosowania dodatkowych technik.

Umieszczenie maszyny atakującego w środku połączenia tak by mógł *routować* pakiety pomiędzy klientem i serwerem jest wykrywalne za pomocą programu *traceroute* o ile atakujący nie przedsięwziął specjalnych środków. Komputer pośredniczący aby pozostać niewidzialny dla programu *traceroute* nie może:

- zmieniać wartości **TTL** w przekazywanych pakietach
- wysyłać komunikatów **ICMP destination unreachable**.

Ubocznym działaniem tego ataku - o ile atakujący nie jest w stanie usuwać z łącza pakietów generowanych przez klienta oraz serwer - jest burza pakietów ACK. Liczba ich będzie rosła lawinowo w zależności od tego, jak dużo danych klient będzie chciał wymienić z serwerem. Dzieje się tak ze wspomnianego wcześniej powodu - każdy pakiet odrzucony przez którąś ze stron z powodu błędnego numeru sekwencyjnego powoduje wygenerowanie pakietu ACK zawierającego informację o spodziewanym numerze sekwencyjnym. Ponieważ jednak druga strona po otrzymaniu takiego pakietu stwierdzi, że nie zawiera on oczekiwanego numeru sekwencyjnego sama wygeneruje pakiet ACK tworząc zamkniętą pętlę. Pętla ta nie jest jednak nieskończona bowiem nie przenoszące informacji segmenty po zagubieniu nie są retransmitowane. Dlatego jeśli któryś z pakietów ACK zagubi się (TCP w warstwie sieciowej wykorzystuje zawodny protokół IP), to nie zostanie on ponownie stworzony. Pętle pakietów ACK mają tendencje do samoregulacji. Im więcej pakietów ACK będzie krążyło po sieci tym więcej z nich będzie gubionych z powodu przeciążenia sieci.

Jeśli połączenie będzie znajdować się w stanie rozsynchronizowanym i nie będzie atakującego do potwierdzania odbioru pakietów, to dane będą retransmitowane powodując burzę pakietów. Brak odpowiedzi doprowadzi też w końcu do zerwania połączenia.

Wczesna desynchronizacja - metoda ta polega na przerwaniu połączenia we wczesnej fazie i stworzeniu drugiego z innym numerem sekwencyjnym. Przebieg procesu pokazano poniżej.

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.

Desynchronizacja za pomocą pustych danych - w tej metodzie atakujący wysyła duże ilości danych do serwera oraz klienta. Dane te nie powinny być widoczne w warstwie aplikacji i jednocześnie powodować rozsynchronizowanie połączenia. Poniżej pokazano scenariusz ataku w przypadku połączenia przy pomocy protokołu *telnet*

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* mogą to być bajty zawierające 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 różny od tego, którego spodziewa się klient.
3. Atakujący postępuje w ten sam sposób z klientem.

Obrona przed atakiem

Atak na sesję TCP wywołuje skutki uboczne, które mogą posłużyć do jego wykrycia:

- 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 niemal wszystkie pakiety 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.

Najlepszym chyba jednak sposobem sposobem zabezpieczenia się przed porywaniem sesji TCP jest korzystanie z protokołów kryptograficznych szyfrujących całość komunikacji lub używanie programów tunelujących nie szyfrowane protokoły. Do kategorii tej należą między innymi wirtualne sieci prywatne oraz różnego rodzaju programy oferujące usługę tunelowaną.

1. Wprowadzenie do DOS

Ataki typu *Denial Of Service* mają na celu uniemożliwienie skorzystania z zasobów lub usługi systemów komputerowych przez uprawnionych użytkowników. Istnieje bardzo dużo odmian tego ataku, włącznie z jego rozproszoną wersją. Do kategorii DoS zaliczane są tak podstawowe sprawy jak przecięcie kabla zasilającego w celu uniemożliwienia skorzystania z komputera lub tak skomplikowane ataki jak skoordynowane uderzenie sieciowe z tysięcy stacji jednocześnie. W zależności od celu ataku można wprowadzić następującą klasyfikację:

Zużycie limitowanych lub nie odnawialnych zasobów - Aby sieć działała sprawnie urządzenia do niej przyłączone muszą posiadać kilka podstawowych zasobów, takich jak wolny czas procesora, powierzchnię dyskową, pamięć RAM, dostępną przepustowość łącza. Ataki DoS na te zasoby mają na celu zużycie tych zasobów lub ich zablokowanie.

- Blokowanie interfejsu - Ataki tego typu mają na celu niedopuszczenie do komunikacji pomiędzy klientem i serwerem. Do tej grupy zalicza się ataki **SYN Flooding**, **Teardrop** i inne.
- Wykorzystanie zasobów serwera przeciw niemu samemu - Ataki tego typu wykorzystują luki w bezpieczeństwie szeroko stosowanych protokołów. Przykładem jest atak **SMURF** lub **Snork**.
- Zużycie przepustowości sieci - Jest to atak brutalny, którego głównym zadaniem jest zużycie całej przepustowości sieci. Do przeprowadzenia tego rodzaju ataku potrzebne są duże zasoby - zwłaszcza szybkie łącza. Atak ten praktykowany jest głównie w wersji rozproszonej - *Distributed DoS*. Uległy mu swego czasu takie witryny jak Yahoo, CNN czy Ebay. Atak DDoS wymaga skoordynowanego działania wielu hostów w celu skutecznego zablokowania celu. Przygotowanie takiego ataku jest nietrywialnie i wymaga wiele przygotowań.
- Zużycie innych zasobów - Do tej kategorii zalicza się ataki nastawione na wyczerpanie zasobów takich jak pamięć operacyjna, pamięć dyskowa. Można tego dokonać na bardzo wiele sposobów - wysyłając dużą liczbę e-mailów, sprawić że system zapisze dużo powtarzających się komunikatów do logów, wysyłając dużo plików do publicznych serwerów FTP i wiele innych. Możliwe jest również wysłanie do procesu sieciowego nietypowych danych, których ten nie będzie w stanie przetworzyć powodując zużycie 100% czasu procesora lub restart maszyny.

Można również utworzyć bardzo dużą liczbę małych plików w systemie, co np.: w systemie Linux może doprowadzić do wyczerpania się węzłów trzymających informacje o plikach - INODE i uniemożliwieniu stworzenia jakiegokolwiek następnego pliku.

Zniszczenie lub zmiana informacji konfiguracyjnej

Żle skonfigurowany program może działać niepoprawnie lub w ogóle nie działać. Intruz może zdalnie lub lokalnie zniszczyć pliki konfiguracyjne blokując dostęp uprawnionym użytkownikom do danego usługi/programu. Na przykład modyfikacja tablicy *routingu* może doprowadzić do zablokowania całej sieci. Z tego też powodu protokoły *routingu* dynamicznego używane są zwykle na łączach, do których użytkownicy nie mają bezpośredniego dostępu. Zmiany w rejestrze systemu Windows również mogą prowadzić do blokady niektórych usług lub bardzo dziwnego zachowania się systemu uniemożliwiającego pracę z nim.

Fizyczne zniszczenie lub zmiana sprzętu

Ataki sieciowe na oprogramowanie nie są jedynym źródłem blokad dostępu do określonych zasobów lub usług. Bezpośrednie bezpieczeństwo fizycznego sprzętu sieciowego jest równie ważne. Zniszczenie kluczowego routera zablokuje całą sieć na długi czas. Dostęp do urządzeń powinni posiadać wyłącznie wybrani pracownicy po odpowiednim przeszkoleniu.

Sieciowe ataki DOS można podzielić na dwie grupy:

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

Ataki te wykorzystują luki w oprogramowaniu sieciowym. Aplikacje po otrzymaniu niepoprawnych, często specjalnie spreparowanych danych przerywają działanie lub powodują zajęcie 100% czasu procesora a nawet restart maszyny. Są to ataki niesymetryczne, które można przeprowadzić posiadając łącze nieproporcjonalnie wolniejsze niż atakowany system.

Niektóre ataki DoS - zwłaszcza *Distributed DoS* wykorzystują efekt skali zalewając atakowany serwer większą ilością danych niż ten jest w stanie przetworzyć. Są to ataki mniej wymyślne, które do przeprowadzenia wymagają jednak dużo większej ilości zasobów, np.: szybkiego łącza.

Bardzo często źródłem i bazą dla sieciowych ataków typu *Denial of Service* są niewłaściwe implementacje protokołów w systemach operacyjnych, np.: tworzenie niewłaściwej tablicy ARP, nie sprawdzanie długości pakietów przed ich przetworzeniem, czy zła obsługa składania pofragmentowanych pakietów IP. Wszystkie te niedociągnięcia, nawet mimo braku dostępu do kodu źródłowego są wychwytywane przez programistów i administratorów sieciowych, a stąd już tylko krok do wykorzystania tej wiedzy w celu ataku.

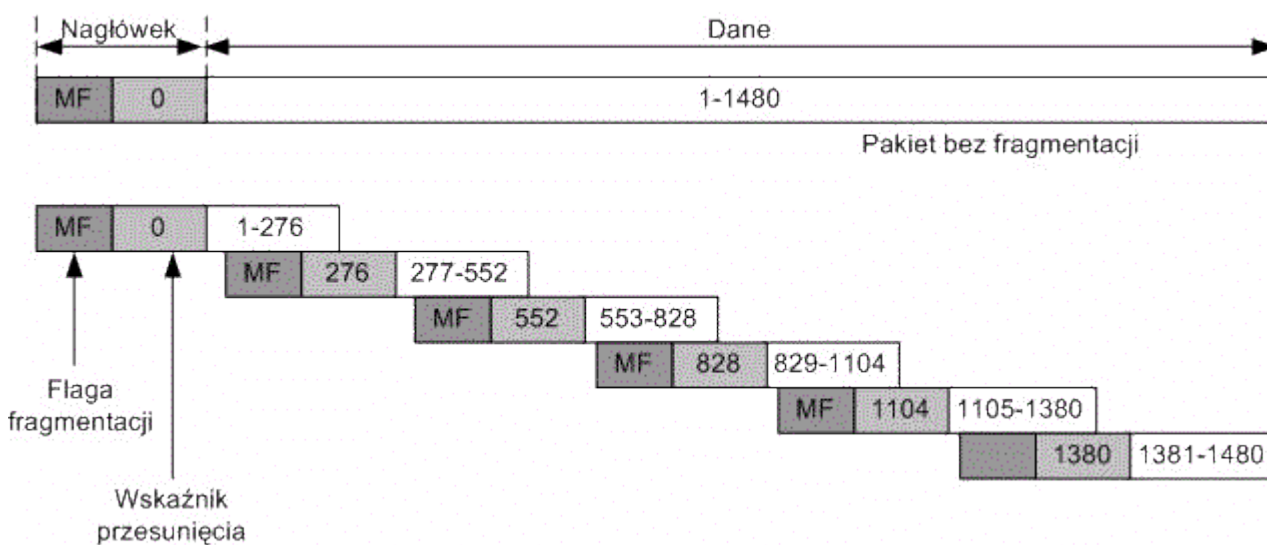
2. Wykorzystanie fragmentacji pakietów

Fragmentacja to proces dzielenia pakietu na mniejsze części celem transmisji przez różnego rodzaju sieci. Proces ten opisany jest szczegółowo w RFC 791. Jedną z wielkości charakteryzujących każdą sieć komputerową jest MTU (*Maximum Transmission Unit*). Określa ona maksymalną wielkość pakietu, który może przez daną sieć zostać przesłany. Dla sieci Ethernet MTU wynosi 1500 bajtów, dla połączeń PPP zwykle 296 bajtów. Pakiet przemierzający Internet przechodzi przez wiele sieci o różnych wielkościach MTU. Urządzenia łączące te sieci (najczęściej *routery*) odpowiadają za fragmentację pakietów gdy jest ona potrzebna. Każdy fragment niesie w sobie następujące informacje:

- identyfikator pakietu, który uległ fragmentacji (fragment ID),
- informację o ilości przesyłanych 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.

Przykład fragmentacji przedstawiono na rys. 1.

Większa liczba pakietów oznacza, że istnieje większa szansa na ich zagubienie i konieczność retransmisji. Dlatego większość systemów operacyjnych (między innymi: Linux) stara się uniknąć fragmentacji kiedy tylko jest to możliwe. Służą do tego flagi w nagłówku IP - **Nie Fragmentuj** (*Don't Fragment - DF*). Urządzenie sieciowe, które stwierdzi potrzebę podzielenia pakietu z zapaloną flagą DF na mniejsze części, odrzuci pakiet generując komunikat *ICMP Destination Unreachable* z wyszczególnionym powodem: **potrzebna fragmentacja a flaga DF jest ustawiona** (*fragmentation needed and DF set*). Podana zostanie również wartość MTU jaka wymagana jest na łączu wymuszającym fragmentację. Kiedy źródło odbierze komunikat ICMP retransmitowany zostanie pakiet o mniejszej długości spełniający warunek MTU. Proces ten nazywa się **odkrywaniem ścieżki MTU** (*Path MTU discovery*). Implementacja fragmentacji pakietów w stosie protokołów TCP/IP zawierała i zawiera wiele luk i niedociągnięć, które po ich odkryciu są wykorzystywane do przeprowadzania ataków. Powstało wiele wariantów takich ataków.



Rys. 1. Przykład fragmentacji

Ping of Death

Atak ten polega na wysłaniu do zdalnej maszyny fragmentów datagramu *ICMP Echo request* o łącznym rozmiarze przekraczającym 65535 bajtów. Jest to maksymalna wielość jaką może przyjąć pole **długość pakietu** w nagłówku IP. Niektóre systemy operacyjne nie są w stanie poprawnie przetworzyć takiego pakietu co powoduje zwykle zawieszenie lub restart maszyny. Atak ten do przeprowadzenia nie wymaga specjalnych narzędzi. Wystarczy standardowe polecenie *ping*. Datagram wysyłany jest w częściach, które złożone razem są większe niż 65535 bajtów. Niektóre maszyny podejmą próbę odtworzenia takiego datagramu i ulegną atakowi. W tej chwili jest to atak raczej nieskuteczny.

Teardrop

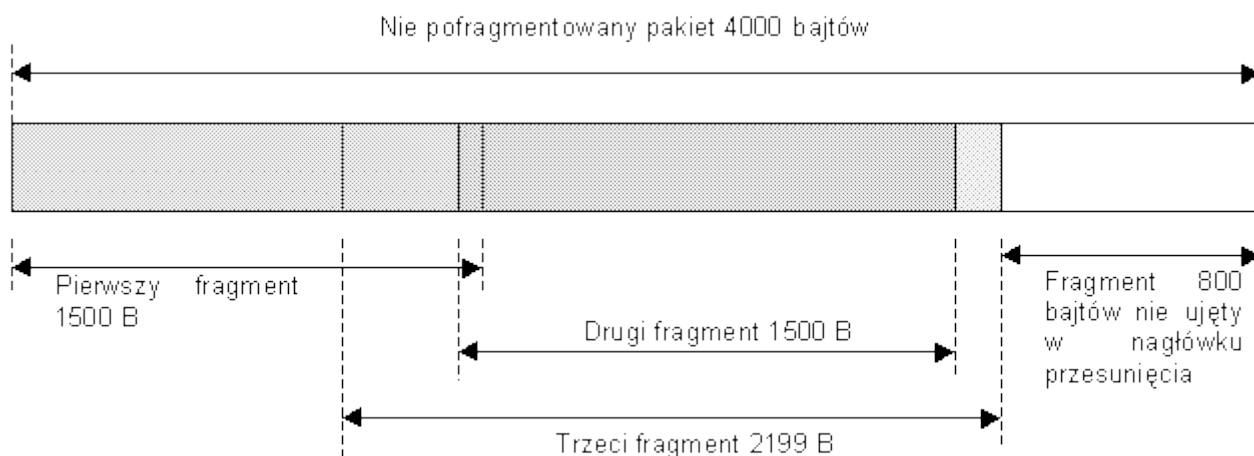
Teardrop atakuje zdalny system poprzez ustawianie nakładających się znaczników przesunięcia w nagłówkach IP. Pakiet o długości 4000 bajtów prawidłowo pofragmentowany na trzy części zawierałby następujące wartości znacznika przesunięcia:

- Pierwszy pakiet: bajty od 1 do 1500
- Drugi pakiet: bajty 1501 do 3000
- Trzeci pakiet bajty od 3001 do 4000

Teardrop ustawia wskaźnik przesunięcia tak, że kolejne kawałki pofragmentowanego pakietu zachodzą na siebie. Przykładowo dla 4000 bajtowego pakietu w trzech częściach wskaźniki mogłyby posiadać wartości:

- Pierwszy pakiet: bajty od 1 do 1500
- Drugi pakiet: bajty 1500 do 3000
- Trzeci pakiet bajty od 1001 do 3200

Innym wariant tego ataku przedstawiony jest na rys. 2. W przykładzie na rys. 2 pakiety zachodzą na siebie zakresami. Co więcej - cały pakiet nie zostanie odebrany bowiem ostatni kawałek pokazuje na wartość 3200, czyli nie pełną długość pierwotnego pakietu. Host po odebraniu takich pakietów nie jest w stanie ich poprawnie złożyć, co może prowadzić do większej zajętości procesora z powodu obsługi wielu błędów rekonstrukcji lub nawet restartu maszyny czy jej zawieszenia. Kolejne wersje ataku *Teardrop* znane były pod nazwami; *Bonk*, *Boinkoraz* *Newtear*.



Rys. 2. Wariant ataku Teardrop

Nakładanie fragmentów (*Fragment Overlapping*)

Atak ten posiada schemat podobny do *Teardrop*, ale jego działanie jest inne. Fragmentacja IP jest tutaj wykorzystywana w celu obejścia reguł filtrowania i przejścia przez zaporę ogniową lub inne urządzenie filtrujące. Atak ten próbuje nadpisać część nagłówka TCP z pierwszego fragmentu. Nagłówek ten oryginalnie zawiera 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. Można w ten sposób uzyskać poprzez zaporę połączenie np. z portem 23 (*telnet*) wysyłając pierwszy pakiet na port 80 (HTTP), a następnie nadpisując tą wartość w drugim datagramie. Sztuczka ta nie uda się z zaporami ogniowymi dokonującymi łączenia pakietów. Inną metodą jest ustalenie minimalnej dopuszczalnej wartości wskaźnika przesunięcia na routerze.

3. Ataki poprzez zalew pakietów

Przedstawione ataki bazują na błędach w oprogramowaniu i nie generują wzmoczonego ruchu w sieci. Druga grupa ataków DoS wykorzystuje efekt skali zalewając zaatakowaną maszynę bardzo dużą liczbą pakietów. Do ataku może posłużyć dowolny protokół stosowany w Internecie. Na odpowiednio dużą skalę są to ataki bardzo skuteczne o czym mogą świadczyć udane ataki na popularne witryny internetowe. Cechą szczególną tego typu ataków jest istnienie trzech stron: atakującego oraz dwóch ofiar: celu ataku oraz maszyny bądź maszyn generujących ruch (*reflector*). Cel ataku prawie nigdy nie zna prawdziwego źródła ataku, gdyż zalewany jest pakietami pochodzącymi od drugiej ofiary ataku, która może być niczego nieświadoma. Dopiero współpraca ofiar może pozwolić na zlokalizowanie źródła. Poszczególne ataki różnią się sposobem generowania ruchu. Mimo iż istnieje bardzo dużo możliwości przeprowadzenia tego typu ataku wszystkie one mają wspólne elementy.

Zalew UDP (UDP Flooding)

UDP Flooding wymaga najczęściej dwóch maszyn, które są atakowane. Polega on na stworzeniu pętli pomiędzy nimi. Aby atak był skuteczny komputery te muszą posiadać uruchomione usługi odpowiadające na datagramy UDP. Najczęściej wykorzystywanymi usługami są *echo* oraz *chargen* (*character generator protocol*). *Chargen* jest usługą, która na każdy otrzymany pakiet wysyła pakiet. Pętlę inicjuje się poprzez wysłanie do maszyny udostępniającej usługę *chargen*, pakietu ze sfałszowanym adresem nadawcy wskazującym na serwis *echo* innego (lub tego samego) komputera. *Chargen* zgodnie ze swoją funkcją odpowie pakietem, co z kolei zobliguje serwis *echo* do wysłania kolejnego pakietu. Jeśli usługi *echo* oraz *chargen* uruchomione są na jednym komputerze to pętlą taką można objąć tą konkretną maszynę. Jeśli jednak w ataku biorą udział dwa komputery, to poza ich zablokowaniem w sieci tworzy się burza kolizyjna, która czyni sieć bezużyteczną na pewien czas.

Bronić się przed tym atakiem można wiele sposobów. Najprostszy polega na wyłączeniu zbędnych usług (zwłaszcza *chargen*). Można również spowodować aby *chargen* nie odpowiadał na pakiety wysłane z portów niższych niż 1024, które to są zarezerwowane dla usług systemowych.

Zalew SYN (SYN Flooding)

Atak ten ma na celu wyczerpanie zasobów serwera poprzez zainicjowanie bardzo wielu połączeń TCP. Kiedy maszyna otrzymuje pakiet z zapaloną flagą SYN na otwartym porcie, to stos TCP/IP alokuje pamięć dla nowego połączenia wysyłając jednocześnie pakiet SYN/ACK. Jeśli inicjujący pakiet SYN został wysłany ze sfałszowanym adresem źródła to odpowiedź na pakiet SYN/ACK nie nadejdzie lub będzie to pakiet RST (jeśli sfałszowany adres źródła jest używany). Aby atak był skuteczny inicjujące pakiety SYN muszą nadchodzić szybciej niż zaatakowany system potrafi je przetworzyć. Atak SYN Flood można wykryć na szereg sposobów. Na zaatakowanej maszynie można skorzystać z programu *netstat*, tak jak to pokazano na rys. 3. Na rys. 4 przedstawiono obraz ruchu sieciowego w czasie ataku SYN Flooding.

```
C:\WINNT\System32\cmd.exe
Aktywne połączenia
Protokół Adres lokalny Obcy adres Stan
TCP 62.121.68.242:1033 195.159.0.91:7000 USTANOWIONO
TCP 62.121.68.242:1119 213.112.106.10:1176 USTANOWIONO

C:\>netstat -a -p tcp
Aktywne połączenia
Protokół Adres lokalny Obcy adres Stan
TCP inp:epnarp inp:0 NASEUCHIWANIE
TCP inp:microsoft-ds inp:0 NASEUCHIWANIE
TCP inp:1025 inp:0 NASEUCHIWANIE
TCP inp:1029 inp:0 NASEUCHIWANIE
TCP inp:1033 inp:0 NASEUCHIWANIE
TCP inp:1119 inp:0 NASEUCHIWANIE
TCP inp:1033 powertech.no.eu.dal.net:7000 USTANOWIONO
TCP inp:1119 c-0a6a70d5.07-4-73746f33.cust.bredbandsbolaget.s
e:1176 USTANOWIONO
TCP inp:1027 inp:0 NASEUCHIWANIE
TCP inp:1030 inp:0 NASEUCHIWANIE

C:\>
```

Rys. 3. Raport programu *netstat*, w czasie ataku SYN Flooding

No.	Time	Source	Destination	Protocol	Info
1	0.000000	193.27.11.211	192.168.1.1	TCP	26660 > 80 [SYN] Seq=1747189760 Ack=0 win=512 Len=0
2	0.000392	228.131.215.71	192.168.1.1	TCP	30725 > 80 [SYN] Seq=2013593600 Ack=0 win=512 Len=0
3	0.000751	129.153.198.182	192.168.1.1	TCP	14855 > 80 [SYN] Seq=973537280 Ack=0 win=512 Len=0
4	0.001060	50.139.253.45	192.168.1.1	TCP	39713 > 80 [SYN] Seq=2602631168 Ack=0 win=512 Len=0
5	0.001412	87.136.7.146	192.168.1.1	TCP	17920 > 80 [SYN] Seq=1174405120 Ack=0 win=512 Len=0
6	0.001724	86.254.66.6	192.168.1.1	TCP	26381 > 80 [SYN] Seq=1728905216 Ack=0 win=512 Len=0
7	0.002076	195.199.115.206	192.168.1.1	TCP	41751 > 80 [SYN] Seq=2736193536 Ack=0 win=512 Len=0
8	0.002390	97.178.30.57	192.168.1.1	TCP	1830 > 80 [SYN] Seq=119930880 Ack=0 win=512 Len=0
9	0.002749	66.185.146.9	192.168.1.1	TCP	5905 > 80 [SYN] Seq=386990080 Ack=0 win=512 Len=0
10	0.003056	193.59.251.190	192.168.1.1	TCP	48399 > 80 [SYN] Seq=3171876864 Ack=0 win=512 Len=0
11	0.003408	215.22.239.222	192.168.1.1	TCP	43033 > 80 [SYN] Seq=2820210888 Ack=0 win=512 Len=0
12	0.003724	240.33.167.51	192.168.1.1	TCP	52485 > 80 [SYN] Seq=3439656960 Ack=0 win=512 Len=0
13	0.004081	254.245.237.113	192.168.1.1	TCP	58653 > 80 [SYN] Seq=3843883008 Ack=0 win=512 Len=0
14	0.004421	136.210.118.166	192.168.1.1	TCP	39169 > 80 [SYN] Seq=2566979584 Ack=0 win=512 Len=0
15	0.004774	112.231.196.2	192.168.1.1	TCP	46116 > 80 [SYN] Seq=3022258176 Ack=0 win=512 Len=0
16	0.005088	52.194.43.47	192.168.1.1	TCP	47123 > 80 [SYN] Seq=3088252928 Ack=0 win=512 Len=0
17	0.005426	146.7.151.129	192.168.1.1	TCP	55586 > 80 [SYN] Seq=3642884096 Ack=0 win=512 Len=0

Rys. 4. Obraz ruchu sieciowego w czasie ataku SYN Flooding

Systemy IDS również zwykle wykrywają tego typu zdarzenia w sieci. Niektóre systemy operacyjne (np.: FreeBSD od wersji 4.5) posiadają specjalne mechanizmy wbudowane w stos TCP/IP zwane **SYN cookies** oraz **SYN cache**. Mają one na celu zwiększenie odporności systemu na atak SYN Flooding. Brytyjski *The Register* opublikował w sierpniu 2001 roku artykuł, w którym przedstawiono test odporności zapór ogniowych na ataki SYN flood. Niezabezpieczona maszyna z Linuxem RedHat 7.1 przestawała być użyteczna przy natężeniu 100 pakietów SYN na sekundę.

Obecność zapory ogniowej Cisco PIX Firewall nie zmieniła tych wyników. Checkpoint Firewall-1 z modułem SYNDefender był w stanie zmierzyć się ze strumieniem 500 pakietów SYN/sek. Zapora Netscreen-100 radziła sobie przy 14.000 pakietów na sekundę.

Land

Atak *land* jest odmianą *SYN Flooding*. Różnica polega na tym, że zarówno adres nadawcy jak i źródła ustawiany jest na adres atakowanego komputera. Tworzy to pętle, w którą wpada zaatakowany system próbujący sam sobie odpowiadać na otrzymane pakiety. Atak ten jest wychwytywany przez zapory ogniowe oraz przez same systemy operacyjne i nie stanowi obecnie realnego zagrożenia.

Smurf

Atak ten wykorzystuje niedopatrzenie w specyfikacji protokołu ICMP. Jedną z częściej wykorzystywanych funkcji jest żądanie echa: *ICMP echo request* sprawdzające istnienie komputera o określonym adresie w sieci. Komputer, który otrzyma taki pakiet zgodnie ze specyfikacją zobowiązany jest odesłać pakiet *ICMP echo replay*. Jeśli komputer nie jest dostępny, to informujący o tym datagram ICMP generowany jest przez router. Specyfikacja ICMP określa pożądaną reakcję systemu na otrzymany datagram *ICMP Echo Request* następująco:

"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".

Sytuację taką można łatwo wykorzystać wysyłając pakiet ICMP żądania echa ze sfałszowanym adresem źródła na adres rozgłoszeniowy sieci. Pakiet ten zostanie przetworzony przez wszystkie komputery, które odesłają odpowiedź do nadawcy pakietu. Przy odpowiednio dużej sieci może to 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

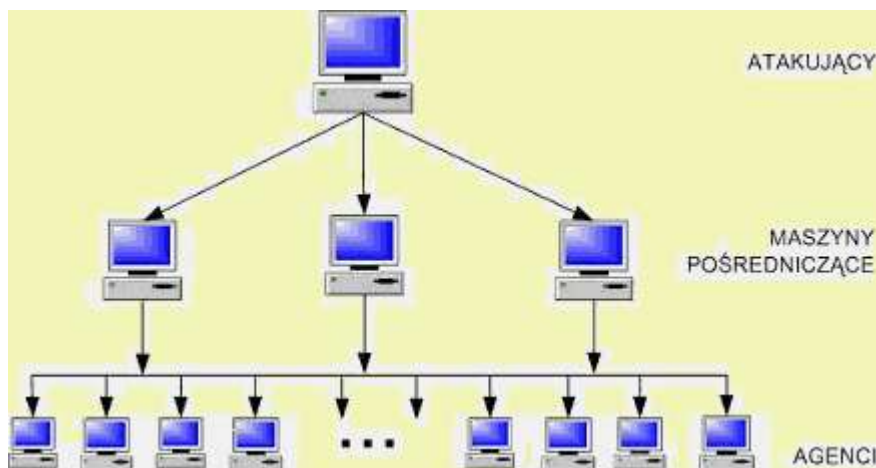
Atak ten ma duże szanse powodzenia w sieciach lokalnych, rzadziej gdy jego źródło znajduje się na zewnątrz sieci. Routery brzegowe są z reguły skonfigurowane tak, by nie przepuszczają pakietów z adresem rozgłoszeniowym do wnętrza sieci. Aby uniknąć tej sytuacji już w 1989 roku w dokumencie RFC 1122 dopuszczono możliwość, by żądania ICMP wysyłane na adres typu *broadcast* lub *multicast* były ignorowane.

Fraggle

Atak *fraggle*, jest bardzo podobny do ataku typu *smurf*, różni się tylko protokołem. Zamiast ICMP, wykorzystuje protokół UDP oraz typowo udostępnione usługi takie jak *echo* czy *chargen*. Aby wywołać burzę UDP, można wysłać sfałszowany pakiet UDP na port usługi *chargen* z rozgłoszeniowym adresem zwrotnym. Innym sposobem jest wywołanie zapętlenia między usługami *chargen* i *echo*, co skutkuje szybkim pomnażaniem wysyłanych pakietów.

4. Rozproszone ataki DoS

Rozproszone ataki DoS wykorzystują efekt skali jaki daje równoczesny atak z wielu maszyn jednocześnie. W tym przypadku nie ma znaczenia użyty protokół, nie trzeba też wyszukiwać luk w oprogramowaniu gdyż generowany jest tak duży potok pakietów, że maszyna/sieć będąca celem ataku nie jest w stanie ich przetworzyć. Atak DDoS wymaga organizacji oraz koordynacji działań. Schemat ataku przedstawiony jest na rys. 5.



Rys. 5. Schemat rozproszonego ataku DoS

Taka organizacja ataku wymusza konieczność koordynacji działań, która w dużej części musi odbywać się automatycznie. Na szczycie piramidy znajduje się komputer dyrygujący atakami. To tam znajduje się konsola, z której wychodzą polecenia, co i kiedy zaatakować. W warstwie pośredniej umieszczany jest specjalny program przekazujący polecenia konsoli do agentów, którzy wykonując polecenie przeprowadzają właściwy atak. Program pośredniczący oraz agenci instalowani są często automatycznie na komputerach posiadających luki w systemie zabezpieczeń. Obecnie obserwuje się działania zmierzające do utrudnienia wykrycia programów nadzorczy oraz agenta. Komunikacja pomiędzy nimi odbywa się z wykorzystaniem różnych protokołów, często jest też szyfrowana.

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

- skanowanie dużej liczby komputerów pod kątem posiadania znanej luki,
- przejęcie kontroli nad wrażliwymi hostami,
- zainstalowanie agenta,
- użycie zdobytego komputera do dalszego skanowania.

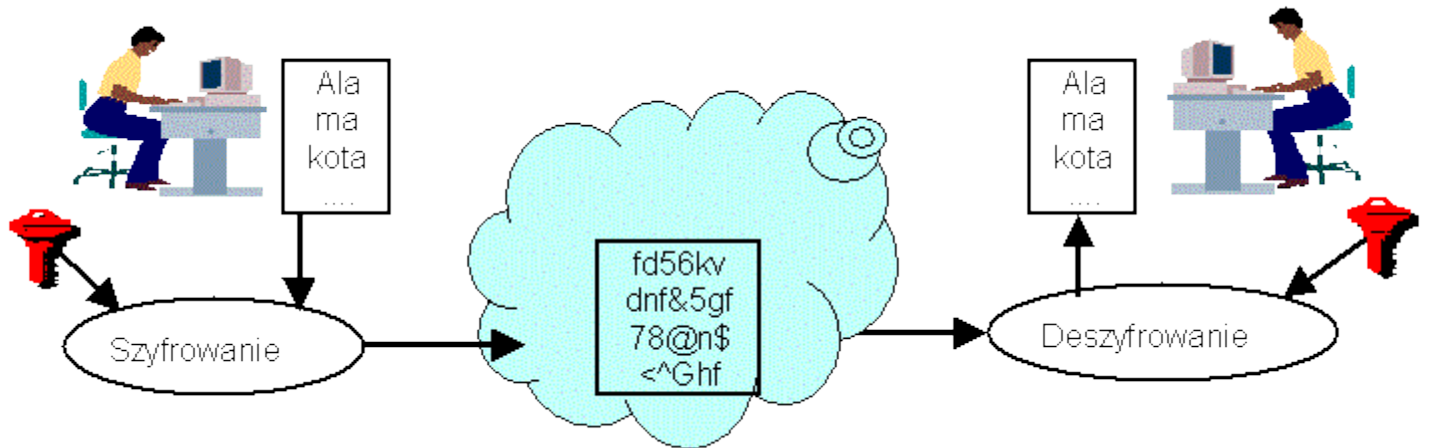
Do podstawowych metod ochrony przed atakami DoS należy zaliczyć:

- 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.

2. Algorytmy z kluczem tajnym

Algorytmy z kluczem tajnym (z *kluczem prywatnym lub symetrycznym, lub algorytmy symetryczne*) używają tego samego klucza do szyfrowania i deszyfrowania informacji.

Wspólną wadą algorytmów symetrycznych jest konieczność bezpiecznego uzgodnienia klucza szyfrującego przed samą transmisją - pojawia się tu klasyczny problem - jak uzgodnić klucz bezpiecznej komunikacji? Idea algorytmu z kluczem tajnym przedstawiona została na rys. 1.



Rys. 1. Idea algorytmu z kluczem tajnym

Do najpopularniejszych algorytmów symetrycznych należą:

- skipjack,
- IDEA,
- RC2,
- RC4,
- RC5,
- DES,
- 3DES.

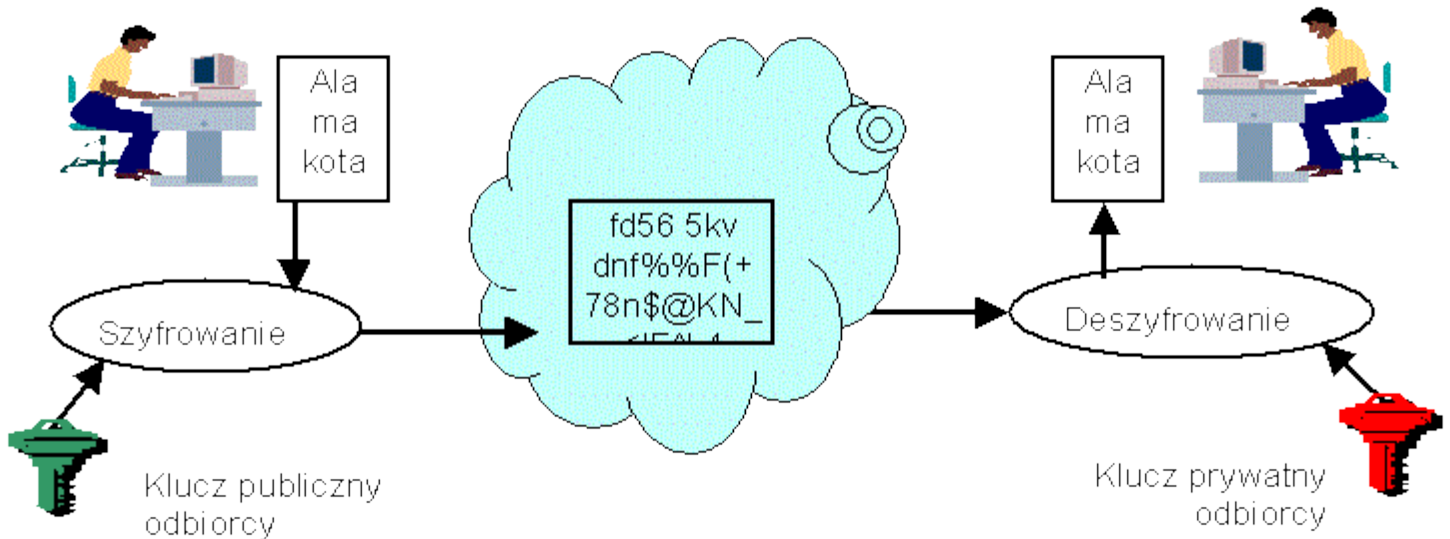
Niektóre algorytmy działają na 64-bitowych blokach danych. Konieczne jest zatem podzielenie większych wiadomości na odpowiednie bloki a następnie ustawienie tych bloków w kolejce. Mechanizmy kolejkowania mogą również zapewniać dodatkowe zabezpieczenie przy manipulowaniu przesyłanymi danymi. W tej chwili zdefiniowano 4 sposoby łączenia otwartego tekstu, klucza i szyfrogramu w celu uzyskania strumienia danych przesyłanego do odbiorcy:

- **ECB (Electronic Code Book) - elektroniczna książka kodów.** Każdy blok tekstu jawnego jest szyfrowany niezależnie, przy użyciu tego samego klucza. Dla jednakowych danych, wynik będzie taki sam.
- **CBC (Cipher Block Chaining) - wiązanie bloków zaszyfrowanych.** Danymi wejściowymi algorytmu szyfrującego jest wynik różnicy symetrycznej szyfrogramu poprzedniego bloku i tekstu jawnego następnego bloku.
- **CFB (Cipher FeedBack) - szyfrowanie ze sprzężeniem zwrotnym.** Dane wejściowe przetwarzane są w porcjach po J bitów. Jako dane wejściowe wykorzystywany jest poprzedzający tekst zaszyfrowany. Algorytm produkuje w ten sposób pseudolosowe dane wyjściowe. Odejmuje się je od tekstu jawnego bloku bieżącego dla stworzenia następnej jednostki tekstu zaszyfrowanego.
- **OFB (Output FeedBack) - szyfrowanie ze sprzężeniem zwrotnym wyjściowym.** Podobnie jak w CFB, z tym, że danymi wejściowymi jest poprzedni wynik działania algorytmu szyfrowania.

Poza ECB, pozostałe algorytmy generują dla tych samych danych różne szyfrogramy. Uzyskuje się to poprzez stosowanie dla różnych bloków różnych kluczy lub tzw. wektora inicjacji (*initialization vector*). Jest to losowy wybrany pierwszy blok szyfrowanych danych. Może być np. uzyskany ze znacznika czasu.

3. Algorytmy z kluczem publicznym

Algorytmy z kluczem publicznym (*lub z kluczem jawnym lub kluczem asymetrycznym lub algorytmy asymetryczne*) wykorzystują parę kluczy. Do szyfrowania używa się jednego z nich, a do deszyfrowania drugiego. Jeden z tych kluczy musi być tajny (*klucz prywatny*). Idea algorytmu z kluczem publicznym przedstawiona została na rys. 2.



Rys. 2. Idea algorytmu z kluczem publicznym

Do podstawowych algorytmów z kluczem publicznym należą:

- RSA,
- DSA,
- El Gamal.

4. Algorytmy skrótu

Skrót wiadomości (*kryptograficzna suma kontrolna - hasz*) jest specjalną liczbą będącą produktem funkcji, którą jest nieodwracalna. Aby algorytm takiej funkcji mógł być uznany za odpowiedni do zastosowań kryptograficznych, musi posiadać następujące właściwości:

- *Spójność* - dla takich samych danych wejściowych musi dawać takie same dane wyjściowe.
- *Losowość* - musi być losowy lub sprawiać takie wrażenie.
- *Unikalność* - niemal niemożliwe powinno być znalezienie dwóch różnych wiadomości dających ten sam skrót.
- *Jednokierunkowość* - niemal niemożliwe powinno być wydedukowanie wiadomości wejściowej na podstawie wiadomości wyjściowej.

Do najczęściej wykorzystywanych algorytmów skrótu należy zaliczyć:

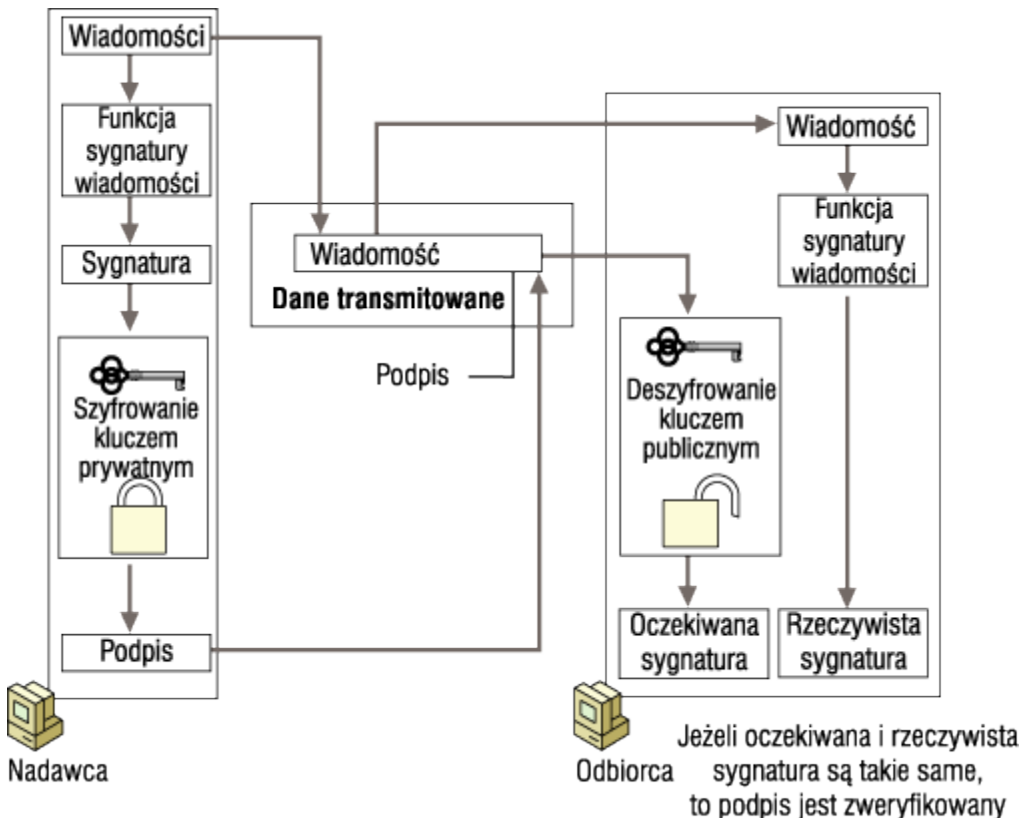
- MD5,
- SHA.

5. Podpis cyfrowy

Podpis cyfrowy to (zwykle) skrót wiadomości zaszyfrowany za pomocą osobistego klucza osoby podpisującej się - używany w celu potwierdzenia treści. W tej sytuacji proces szyfrowania nazywamy **podpisaniem**. Podpis cyfrowy:

- wskazuje, czy dana wiadomość została zmieniona (zapewnienie spójności),
- umożliwia weryfikację osoby podpisującej się (zapewnienie autentyczności),
- zapewnienie nie wypierania się podpisującego.

Każdy dokument można wyposażyć w skrót wiadomości i dołączyć go do przesyłanego dokumentu. Odbiorca może na nowo wyliczyć skrót i porównać go ze skrótem odebrany. Zgodność skrótów świadczy o autentyczności dokumentu. Idea procedury wykorzystania podpisu przedstawiona została na rys. 3.



Rys. 3. Idea podpisu cyfrowego

6. Dystrybucja kluczy kryptograficznych

Klucz sesyjny zwykle jest generowany przez jednego z użytkowników lub Centrum Dystrybucji Kluczy. Istnieje jednak zawsze problem ich dystrybucji.

Dystrybucja kurierska - metoda najstarsza, najczęściej stosowana i najsłabiej zabezpieczona. Wykorzystuje się przesyłki pocztowe polecone lub kurierów. Dla zwiększenia bezpieczeństwa dzieli się klucz na części i przekazuje je osobnymi kanałami.

Dystrybucja elektroniczna

W systemach symetrycznych wykorzystuje się serwis uwierzytelniający w postaci Centrum Dystrybucji Kluczy (*Key Distribution Center KDC*). Jego zadaniem jest generowanie kluczy sesyjnych. Dodatkowo poszczególni użytkownicy (abonenci) posiadają klucze do komunikowania się z KDC. Przykładowym algorytmem dystrybucji tego typu jest algorytm Cerbera.

Użytkownicy mogą nie korzystać z KDC. Samodzielnie generują klucze sesyjne i samodzielnie je przekazują. Przykładem takiego algorytmu jest EKE (*Encrypted Key Exchange*) lub algorytm *Diffie Hellmana*.

W przypadku dystrybucji kluczy wykorzystywanych w algorytmach asymetrycznych, problem pojawia się paradoksalnie w najmniej spodziewanym miejscu: klucz publiczny. Fakt, że klucze te są powszechnie dostępne nie zwalnia nas bowiem z dbania o to, żeby były one właścive! Przykład wyjaśniający potencjalne zagrożenie zaprezentowano poniżej:

Przypadek 1

O (odbiorca) publikuje swój klucz publiczny w sieci (np. na WWW).

N (nadawca) chce wysłać szyfrowaną wiadomość do **O**.

W tym czasie **H** (hacker) poprzez włamanie sprawia że klucz publiczny **O** pobrany przez **N** jest fałszywy, wie o tym tylko **H**.

N szyfruje wiadomość fałszywym kluczem i wysyła.

Przypadek 2

N celowo sfałszuje swój klucz publiczny, aby następnie zaprzeczyć, że wysłał wiadomość.

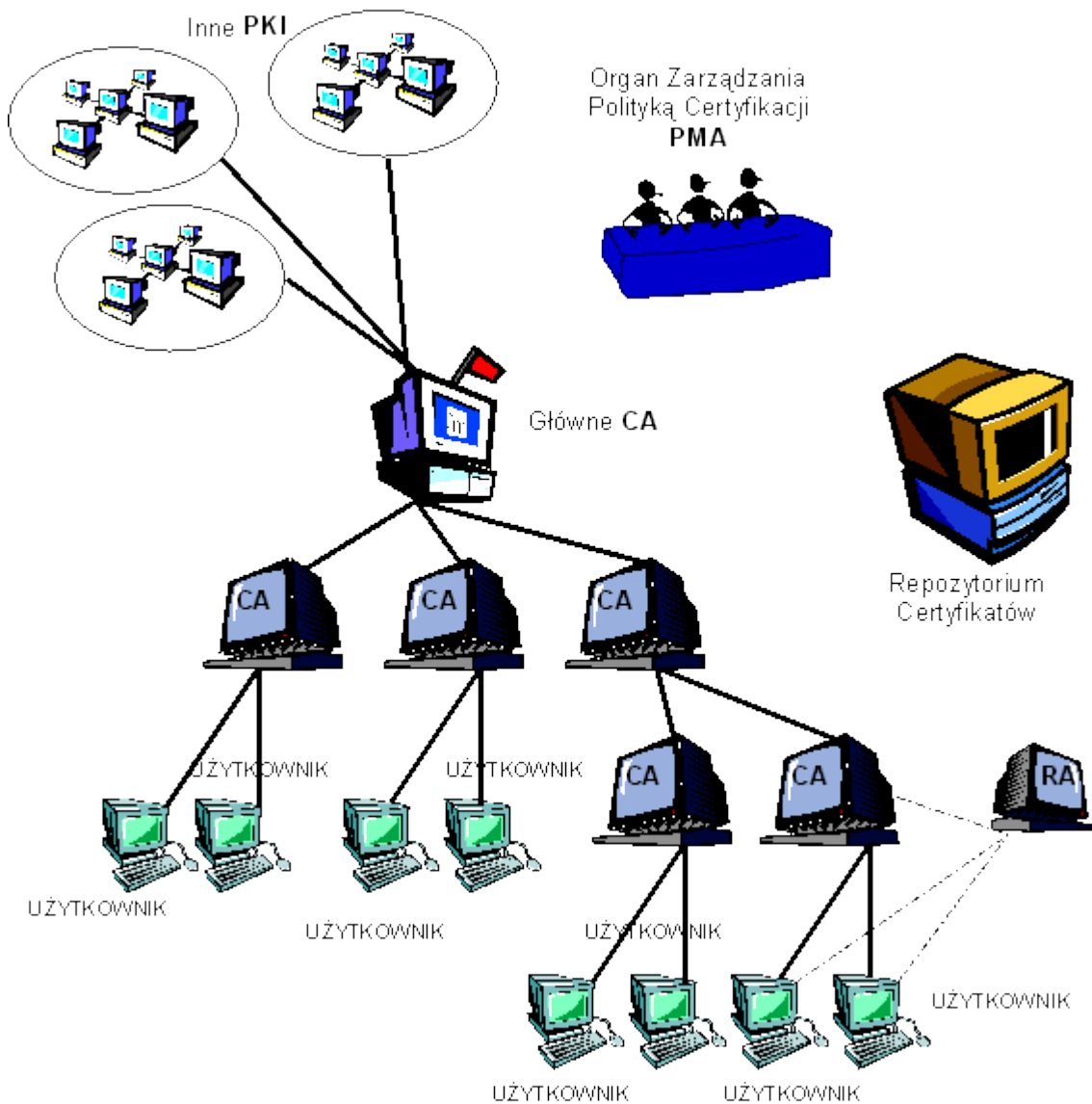
Zatem nie tylko nadawca ale i odbiorca musi mieć pewność, że klucze publiczne używane w bezpiecznej komunikacji są właściwe! Jak widać, bez dodatkowej pomocy nie da się zapobiec takim zagrożeniom. Można by co prawda założyć, że np. **N** i **O** wymieniają się kluczami "z ręki do ręki" przed samą transmisją, ale to wypacza całą ideę kryptografii asymetrycznej - równie dobrze można wtedy zastosować zwykłe szyfrowanie kluczem tajnym. Problem ten nie ma na razie idealnego rozwiązania "matematycznego" ale wymyślono sposób na jego obejście.

Rozwiązanie to polega na wprowadzeniu jeszcze jednej strony zwanej urzędem certyfikacyjnym (*Certificate Authority, CA*). Jest to zaufana osoba, lub instytucja przechowująca klucze publiczne osób chcących się wspólnie komunikować. Instytucja ta również posiada swoją parę kluczy publiczny/prywatny których używa do wszystkich operacji związanych z kluczami użytkowników. Klucz CA jest albo powszechnie znany ("powszechnie" - na tyle, że jego sfałszowanie będzie łatwo widoczne), albo jego weryfikacja jest możliwa przy użyciu klucza publicznego jakiejś "ważniejszej" instytucji. Schemat działania z wykorzystaniem urzędu certyfikacyjnego przedstawiono poniżej:

- **N** wysyła do **CA** zapytanie o klucz publiczny **O**.
- **CA** znajduje klucz **O** w bazie i wysyła go do **N**, podpisując wiadomość swoim kluczem prywatnym (nie musi być ona szyfrowana, w końcu jest to informacja dostępna dla każdego).
- **N** odbiera klucz, deszyfruje go kluczem publicznym **CA** (który ma skądinąd).
- W tym momencie jest pewny, że posiada właściwy klucz publiczny **O**. Może zatem wysłać wiadomość do **O** szyfrując ją tym kluczem (i np. podpisując swoim prywatnym).
- odbiera wiadomość - widzi, że jest podpisana przez kogoś, kto przedstawia się jako **N**.
- wysyła zatem zapytanie do **CA** z prośbą o przesłanie klucza publicznego **N** aby upewnić się, że autor przesyłki jest właściwy.
- Jeżeli tak, to deszyfruje wiadomość swoim kluczem prywatnym.

7. Infrastruktura klucza publicznego

Celem infrastruktury kluczy publicznych PKI (*Public Key Infrastructure*) jest zapewnienie zaufanego i wydajnego zarządzania kluczami oraz certyfikatami. PKI jest zdefiniowana w dokumencie *Internet X.509 Public Key Infrastructure*. Wg tego dokumentu, PKI to: *zbiór sprzętu, oprogramowania, ludzi, polityki oraz procedur niezbędnych do tworzenia, zarządzania, przechowywania, dystrybucji oraz odbierania certyfikatów opartych na kryptografii z kluczem publicznym.*



Rys. 4. Komponenty PKI

Główny CA jest "punktem zaufania" dla wszystkich jednostek w strukturze PKI. Jego klucz publiczny jest stosowany bezpośrednio lub pośrednio do podpisywania wszystkich certyfikatów w strukturze PKI. Główny CA podpisuje także certyfikaty, wydawane innym strukturom PKI (jest to zazwyczaj nazywane *cross-certification* lub certyfikacją skrośną). Liczba podległych CA jest teoretycznie nieograniczona.

Organ Zarządzania Polityką Certyfikacji (PMA - Policy Management Authority) jest ciałem, które ustala i administruje zbiorem polityk bezpieczeństwa, stosowanych w infrastrukturze, zatwierdza certyfikację innych Głównych CA i zewnętrznych CA i nadzoruje działanie Głównego CA.

Repozytoria certyfikatów, w których przechowywane są certyfikaty, listy CRL i listy organów unieważnionych (listy ARL) (ARL - Authority Revocation List). Repozytorium musi być dostępne dla wszystkich tych, (w pewnych przypadkach, również spoza struktury danej PKI) którzy chcą wykorzystywać usługi ochrony, dostępne w aplikacjach obsługujących strukturę PKI, a co za tym idzie, informacje dotyczące struktury PKI muszą być rozpowszechniane w jednolitej formie.

Organy Rejestracji (RA) mogą służyć do wymiany niezbędnych informacji pomiędzy użytkownikiem a Organami Certyfikacji, przejmując niektóre z funkcji administracyjnych CA.

Funkcje realizowane przez PKI:

- Rejestracja. Proces za pomocą którego dana jednostka przedstawia się CA. Może to robić bezpośrednio lub za pośrednictwem zarządu rejestracji (*Registration Authority - RA*).
- Inicjacja. Procedura wg której, zgłaszający się otrzymują wartości (dane) potrzebne do rozpoczęcia komunikacji z PKI. Może obejmować dostarczenie klientowi systemu z kluczem publicznym lub certyfikatem CA, lub wygenerowanie pary kluczy klienta.
- Certyfikowanie. Proces, w którym CA tworzy certyfikat i przekazuje go właścicielowi. Może go również upublicznić w magazynie.

- Odzyskiwanie par kluczy. Kopia zapasowa prywatnego klucza użytkownika może być przechowywana przez CA lub specjalnie powołany do tego system.
- Generowanie kluczy. Para kluczy może być generowana lokalnie przez użytkownika lub przez CA. Mogą one być dostarczone użytkownikowi w zaszyfowanym pliku lub w postaci fizycznej (inteligentna karta).
- Uaktualnianie kluczy. Wszystkie pary kluczy muszą być regularnie uaktualniane. Należy też wydawać nowe certyfikaty. Procedura taka jest realizowana po przekroczeniu terminu ważności lub po ujawnieniu klucza.
- Certyfikowanie przechodnie. Procedura wydawania certyfikatu innemu CA. Certyfikat przechodni używany jest w celu umożliwienia systemom klienckim jednej domeny administracyjnej bezpieczne komunikowanie się z systemami klienckimi innej domeny.
- Unieważnianie. Różne okoliczności mogą spowodować przedterminową utratę ważności certyfikatu. Może to być zmiana powiązania między klientem a CA, ujawnienie lub podejrzenie ujawnienia klucza prywatnego.

Użytkownicy w systemach wykorzystujących kryptografię klucza publicznego muszą być pewni, że klucz publiczny jednostki, z którą się komunikują, rzeczywiście należy do tej jednostki. Ta pewność jest uzyskiwana przez użycie certyfikatów kluczy publicznych. Są to struktury danych łączące klucze publiczne z jednostkami, do których one należą. Powiązanie to jest osiągnięte dzięki zweryfikowaniu przez zaufany CA tożsamości jednostki i podpisaniu certyfikatu. Certyfikat ma ograniczony czas życia. Struktura certyfikatu musi być jednolita w całym PKI. Certyfikat jest podpisany, więc może być rozprowadzany za pomocą niezauważanych systemów komunikacyjnych. Mogą być także buforowane w niezabezpieczonych pamięciach.

Podstawowe elementy występujące w certyfikacie

- Numer wersji - numer wersji formatu certyfikatu
- Numer seryjny - numer przydzielony certyfikatowi przez CA. Unikalny w obrębie funkcjonowania CA
- Identyfikator algorytmu - określa algorytm użyty do podpisania certyfikatu i jego parametry
- Identyfikator wystawcy - nazwa CA, który wydał i podpisał certyfikat
- Okres ważności - data początku i końca ważności certyfikatu
- Użytkownik certyfikatu - określa użytkownika
- Informacja o kluczu publicznym - klucz publiczny użytkownika oraz identyfikator algorytmu, który będzie ten klucz wykorzystywał
- Rozszerzenia - informacje dodatkowe
- Podpis cyfrowy - uwierzytelnia pochodzenie certyfikatu. Funkcja skrótu jest stosowana do wszystkich pól certyfikatu (oprócz pola podpisu). Wynik *haszowania* jest szyfrowany kluczem prywatnym CA.

Proces poświadczania certyfikatu składa się z czterech kroków:

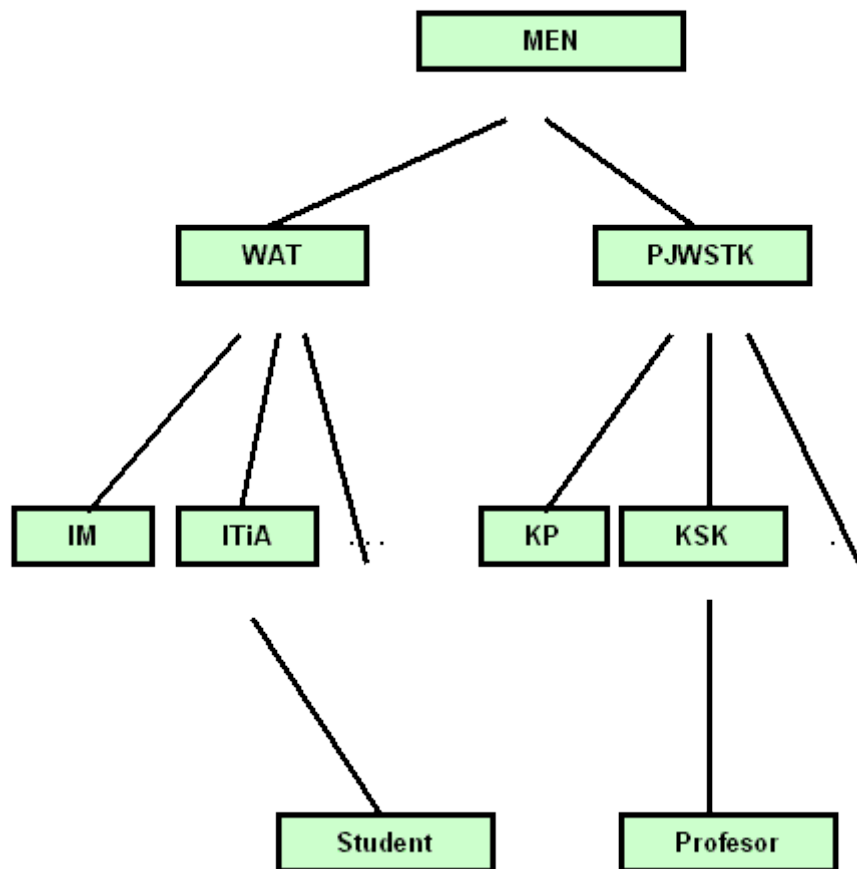
1. Sprawdzenie czy tożsamość nadawcy jest zgodna z opisem w certyfikacie.
2. Sprawdzenie czy żaden certyfikat na ścieżce uwierzytelnienia nie został unieważniony.
3. Sprawdzenie czy dane mają atrybuty, do których podpisujący nie jest upoważniony.
4. Sprawdzenie czy dane nie zostały zmienione od momentu ich podpisania.

Standard X.509 definiuje również metodę unieważniania certyfikatów. CA jest zobowiązany do okresowego wydawania podpisanej listy unieważnienia certyfikatu CRL (*certificate revocation list*). Lista ta powinna być udostępniana w publicznym miejscu. Każdy certyfikat w CRL jest identyfikowany poprzez numer seryjny.

Do dystrybucji certyfikatów można wykorzystywać różne protokoły. Mogą to być protokoły uniwersalne takie jak HTTP czy FTP. Mogą to być protokoły dedykowane takie jak LDAP. Protokół LDAP (*Lightweight Directory Access Protocol*) jest używany do uzyskania dostępu do usług katalogowych. Ma on ułatwiać dostęp do katalogów X.500. Jest on ukierunkowany na aplikacje zarządzające i przeglądarki. Protokół LDAP V2 jest zdefiniowany w RFC 1777.

Dlaczego hierarchia CA ?

Wszystkie opisywane dotąd mechanizmy są już od dawna stosowane w Sieci - ilość używanych certyfikatów można mierzyć w milionach. Oczywiście żadne pojedyncze centrum certyfikacji nie byłoby w stanie podjąć obsługę takiej liczby klientów, i to nie tylko z przyczyn technicznych (szybkość łącz, biurokracja itp.) ale również organizacyjnych. Naturalna staje się zatem potrzeba wprowadzenia pewnej hierarchii urzędów certyfikacyjnych. Przykładowa, hipotetyczna hierarchia urzędów certyfikacyjnych i związana z tym ścieżka certyfikacji została przedstawiona na rys. 5.



Rys. 5. Hipotetyczna hierarchia CA

Urząd CA MEN nie wydaje wtedy certyfikatów poszczególnym osobom w obrębie danej uczelni - zamiast tego wydaje certyfikat tylko dla CA PJWSTK, czy CA WAT. Te z kolei mogą wydawać certyfikaty dla CA jeszcze niższego poziomu, lub dla swoich pracowników. Takie podejście ma dwie zalety:

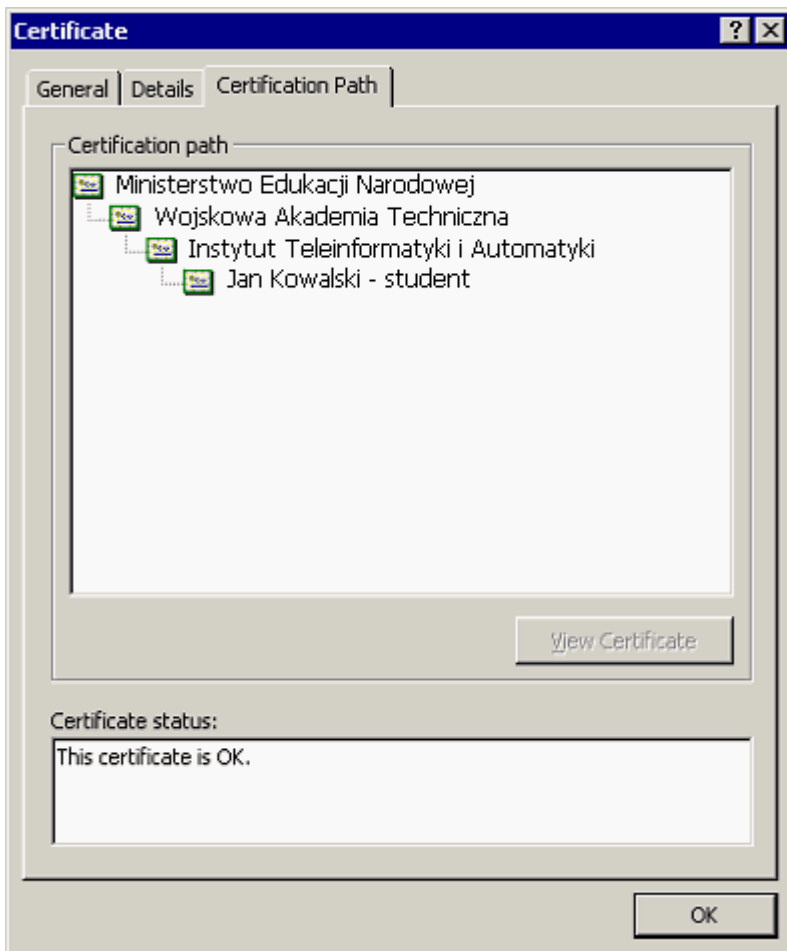
- zmniejszony ruch w sieci - do weryfikacji certyfikatu przy korespondencji w ramach jednej jednostki organizacyjnej nie jest potrzebna wiedza o CA wyższego poziomu,
- rozłożenie odpowiedzialności w sposób odpowiadający organizacji instytucji.

Jak jednak jest realizowana weryfikacja przy takiej "drzewiastej" strukturze urzędów? Załóżmy, że student Informatyki WAT chce korespondować z profesorem katedry sieci komputerowych PJWSTK.

Student wysyłając list podpisuje go swoim kluczem prywatnym, dołącza swój certyfikat oraz wszystkie certyfikaty urzędów nadrzędnych, czyli np. ITiA, WAT, MEN. Tworzy w ten sposób łańcuch certyfikatów (*certificate chain*) przedstawiony na rys.6. Natomiast tok myślenia profesora (a raczej jego oprogramowania:) przy weryfikacji przesyłki jest następujący:

- Student wykazał się certyfikatem wydanym przez CA ITiA - *nie wiem co to jest, nie ufam mu*
- Ale CA ITiA legitymuje się certyfikatem wydanym przez CA WAT (i podpis w certyfikacie się zgadza) - *ale ja nadal nie wiem co to za certyfikat - nie znam tej struktury (WAT)*
- Ale CA WAT legitymuje się certyfikatem wydanym przez CA MEN (i podpis się zgadza!) - *o, to co innego, sam należę do struktury MEN i skoro moje CA najwyższego poziomu to zatwierdziło, to ja ten certyfikat też uznaję*

Jeżeli na którymś etapie to rozumowanie zawiedzie (podpis nie będzie się zgadzał) to znaczy że gdzieś nastąpiło fałszerstwo.



Rys. 6. Łańcuch certyfikatów

1. Klasyfikacja metod uwierzytelniania

Użytkownicy mogą być uwierzytelniani na podstawie jednej lub kilku informacji pochodzących z następujących zbiorów:

- A. **Tego, co użytkownik wie** - tajny tekst np. hasło znane tylko użytkownikowi i systemowi. W procesie rejestracji jest ono wprowadzane przez użytkownika i sprawdzane przez system.
- B. **Tego, co użytkownik posiada** - klucz, plakietka, karta pomagające w weryfikacji użytkownika. W metodzie hasło-odzew (*challenge-response*) użytkownik dysponuje kartą wyświetlającą identyfikator liczbowy. Można stosować również metodę haseł jednorazowych.
- C. **Tego, kim użytkownik jest** - cechy fizyczne (odciski palców, odciski dłoni, wzorec siatkówki) lub behawioralne (wzorec głosu, podpis), które można zapamiętać i porównać. Weryfikacja polega na ponownym zbadaniu użytkownika i porównaniu wyników badań z zapamiętanymi w systemie. Ta metoda umożliwia również ewentualną identyfikację włamywacza. W przypadku metod behawioralnych istnieje możliwość odrzucenia prawidłowego użytkownika (*błąd ujemny*) i pozytywnej identyfikacji niewłaściwej osoby (*błąd dodatni*). W chwili obecnej techniki tego typu są zwykle stosowane jako dodatkowe (obok hasła). Dwustopniowa weryfikacja zwiększa bezpieczeństwo.

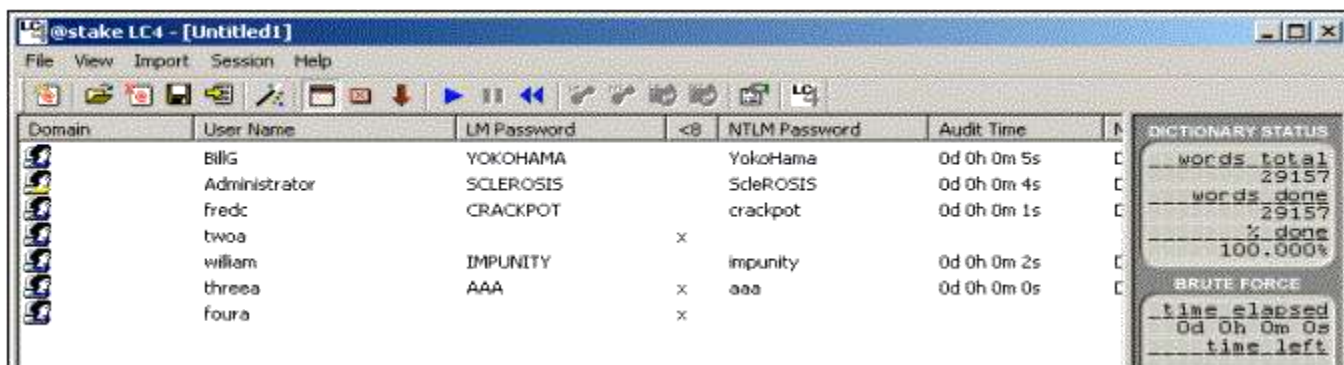
2. Słowniki haseł

Hasła mogą zostać ukradzione z bazy haseł lub przechwycone podczas przesyłania poprzez sieć. Do odgadnięcia hasła może być wykorzystywana metoda słownikowa lub łamania brutalnego. Metoda łamania brutalnego polega na pełnym przeglądzie, czyli wypróbowywaniu każdej kombinacji kodowej, która mogłaby być hasłem. Metoda słownikowa polega na sprawdzeniu każdego słowa występującego w pliku nazywanym słownikiem.

W latach 90-tych Klein prowadził analizę słabości haseł. Zgromadził duży zbiór haseł. Następnie budował słownik wg poniższego schematu:

- wykaz nazw użytkowników, ich inicjałów, nazw kont i innej informacji związanej z użytkownikiem,
- wykaz słów z różnych słowników: imiona i ich permutacje,
- nazwy miejsc, tytuły filmów i książek i postaci w nich występujących,
- różne przekształcenia słów z kroku poprzedniego,
- dowolne zamiany liter małych na duże i odwrotnie,
- słowa w obcych językach dla użytkowników będących obcokrajowcami.

Wyniki eksperymentu pokazały że łatwo jest odgadnąć hasło dysponując informacją o użytkowniku. Wynika z tego, że hasła powinny być trudne do odgadnięcia i ukradzenia. Rozwój komputerów umożliwia coraz szybsze łamanie haseł. Istnieje wiele narzędzi, które umożliwiają łamanie haseł. Jednym z nich jest program LC4, którego okno zamieszczono na rys. 1.



Rys. 1. Okno programu LC4

3. Ochrona haseł

Techniki ochrony haseł można podzielić następująco:

- **Nadzorowanie haseł (wybór, pielęgnacja)**
 - Komunikaty systemowe: wyświetlany jest komunikat przed i po rejestracji użytkownika określający dane systemu. Komunikaty takie mogą dostarczyć intruzowi pewnych wskazówek i dlatego powinny zostać wyłączone lub ograniczone.
 - Wprowadzanie hasła: hasła nie powinny być widoczne w momencie wprowadzania.
 - Ograniczanie ilości prób rejestracji: po osiągnięciu limitu nieudanych logowań konto użytkownika powinno zostać zablokowane (uniemożliwienie dalszych prób). Odblokowanie możliwe po weryfikacji przez administratora przy pomocy innej metody niż hasło. Uniemożliwia to stosowanie metody brutalnego łamania haseł. Informacja o nieudanych logowaniach powinna być zachowywana.
 - Starzenie się haseł: hasło powinno mieć określony czas życia, po którym musi zostać zmienione. Możliwa może być również zmiana hasła przed upływem ważności (w pewnych granicach). Wykorzystane hasła powinny być pamiętane (w określonym zakresie). Administrator, w przypadku zagrożenia, powinien mieć możliwość natychmiastowej zmiany hasła.
 - Systemy z dwoma hasłami: Drugie hasło jest zwykle wykorzystywane podczas próby dostępu do szczególnie chronionych zasobów.
 - Minimalna długość hasła: Krótkie hasła są łatwiejsze do odgadnięcia. Wymaga się aby miały co najmniej 6 lub 8 znaków i występowały w nich określone kombinacje grup znaków.

- Blokowanie konta użytkownika: blokować należy konta nie używane. Usunięcie blokady po weryfikacji przez administratora.
- Ochrona hasła administratora: Ze względu na znacznie większe uprawnienia, w porównaniu do innych użytkowników, jest częściej atakowane i powinno być lepiej chronione. Można wymagać aby było ciągiem znaków heksadecymalnych. Nie powinno być przesyłane przez sieć i powinno być często zmieniane.
- Generowanie hasła przez system: niektóre systemy oferują użytkownikowi kilka haseł do wyboru. Są to zwykle hasła trudne do zapamiętania, co powoduje że użytkownicy je zapisują. Hasła generowane powinny być łatwe do wymówienia.
- **Zabezpieczanie przed odgadnięciem poprzez odrzucanie zbyt łatwych haseł - sprawdzanie haseł**
 - Sprawdzanie bierne: Realizowane jest po wprowadzeniu haseł do użytku za pomocą programu uruchamianego w ustalonych odstępach czasu. Program taki porównuje istniejące hasła z listą haseł łatwych do odgadnięcia. Hasła łatwe są unieważniane a informacja o tym powinna zostać przesłana do użytkownika. Metoda bierna wymaga zużycia znacznych zasobów. Ponadto łatwe hasła funkcjonują w systemie do momentu ich wykrycia stwarzając potencjalne niebezpieczeństwo.
 - Sprawdzanie aktywne. Podczas zmiany hasła przez użytkownika, podane przez niego nowe hasło jest weryfikowane zgodnie z wbudowanym algorytmem. Hasło zbyt łatwe jest odrzucane a użytkownik jest proszony o podanie nowego. W tego typu algorytmie istotne jest zapewnienie równowagi pomiędzy użytecznością i bezpieczeństwem. Zbyt restrykcyjny algorytm będzie powodował niezadowolenie użytkowników. Zbyt liberalny algorytm obniży bezpieczeństwo systemu.
- **Bezpieczne przechowywanie haseł**

Hasła są najczęściej przechowywane w postaci zaszyfrowanej. Do przechowywania haseł wiele systemów używa jednokierunkowej funkcji skrótu. Zapamiętywany jest wyliczony skrót. Oznacza to, że oryginalnego hasła nie można uzyskać z jego postaci zaszyfrowanej. Sprawdzenie polega na wyliczeniu skrótu hasła podanego przez użytkownika podczas logowania i jego porównaniu ze skrótem przechowywanym w systemie.

4. Systemy haseł jednorazowych

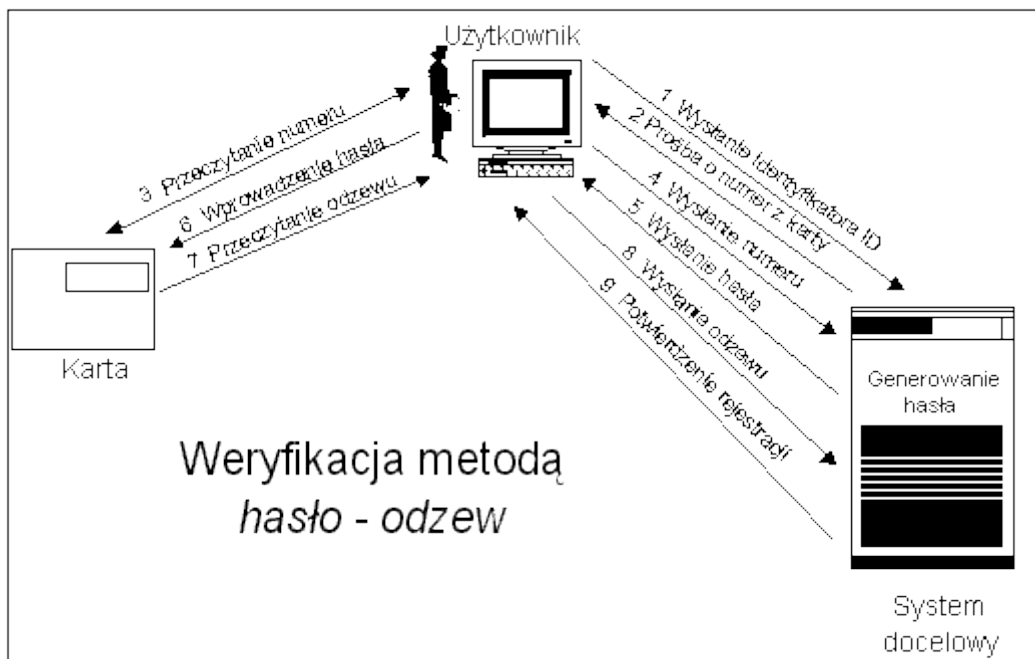
Metoda haseł jednorazowych (*one-time passwords*) polega na jednorazowym wykorzystaniu wygenerowanego hasła. Wobec tego kradzież hasła nie stanowi zagrożenia. Najczęściej są to liczby wygenerowane na stacji klienckiej i weryfikowane na serwerze. Mogą one być również generowane na specjalnym serwerze. Można również wyposażyć użytkownika w specjalną kartę. Przy pomocy klawiatury wprowadza on swój PIN. Procesor karty wylicza liczbę, która zostanie wyświetlona a następnie wprowadzona przez użytkownika jako hasło. Serwer na podstawie podanego identyfikatora użytkownika potrafi wygenerować taki sam kod i dzięki temu zweryfikować użytkownika.

System jednorazowego hasła S/Key zdefiniowany przez RFC 1760 oparty jest na funkcji MD4 i MD5. Protokół ten został zaprojektowany do przeciwdziałania atakowi metodą powtórzeń. Atak ten w kontekście logowania występuje wtedy, gdy ktoś podsłucha połączenie i zdobędzie legalny identyfikator oraz hasło a potem wykorzysta je do uzyskania dostępu do sieci lub hosta. Procedura przedstawia się następująco:

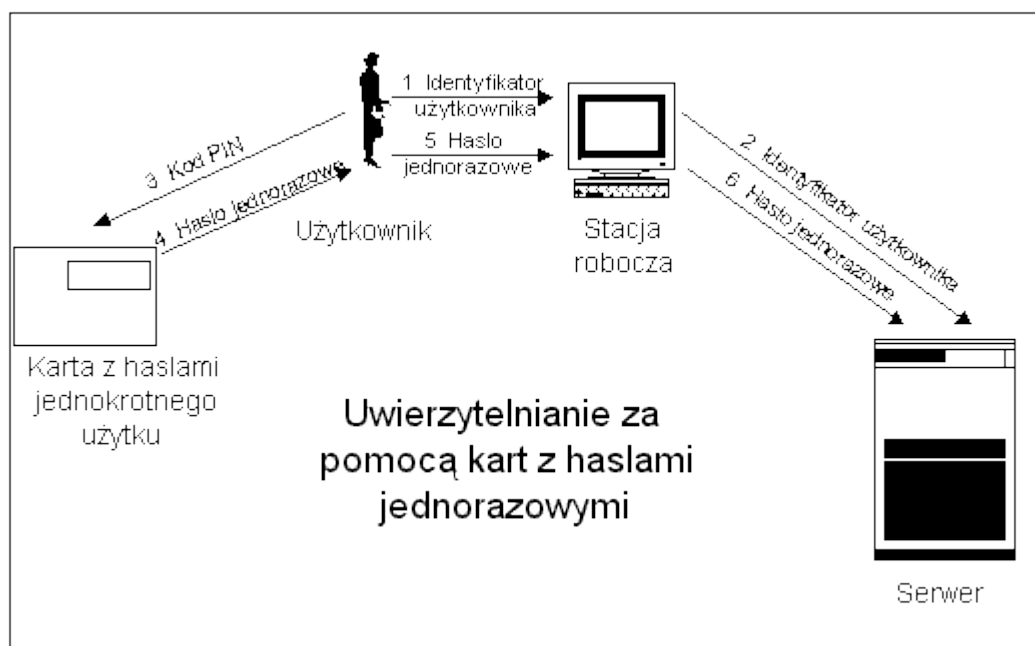
- Klient i serwer są wstępnie skonfigurowani tym samym hasłem oraz licznikiem iteracji. Licznik iteracji określa wymaganą ilość powtórzeń funkcji mieszającej. Przy każdym logowaniu licznik iteracji stronie klienta maleje.
- Klient inicjuje wymianę wysyłając pakiet inicjujący.
- Serwer odpowiada numerem sekwencji. Wysyła również tzw. *ziarno*.
- Po stronie klienta wyliczane jest hasło jednorazowe:
 - operator wprowadza tajne hasło, które jest łączone z *ziarnem*,
 - kilkakrotnie wykonywana jest funkcja mieszająca generująca dane wyjściowe (wg licznika powtórzeń),
 - dane wyjściowe przekształcane są do postaci czytelnej i prezentowane operatorowi.
- Klient przesyła jednorazowe hasło do serwera.
- W serwerze znajduje się plik zawierający dla każdego użytkownika jednorazowe hasło z poprzedniego pomyślnego logowania.
- Serwer jednokrotnie przepuszcza odebrane hasło przez funkcję mieszającą. Wynik powinien odpowiadać hasłu z poprzedniego logowania.

Po pewnym czasie klient musi znowu zainicjować system za pomocą specjalnego polecenia.

Inne rozwiązania to uwierzytelnianie hasła za pomocą znacznika. Wymagają użycia tzw. inteligentnej karty (*smart card*) lub karty znacznika (*token card*). Chroniony obiekt musi być wyposażony w oprogramowanie agenta. Chroniony obiekt musi być wyposażony w oprogramowanie agenta. Ten mechanizm uwierzytelniania oparty może być na systemie wyzwanie-odpowiedź (*challenge-response*) lub uwierzytelnienie zsynchronizowane z czasem (*time-synchronous authentication*). Przykłady takich mechanizmów przedstawiono na rys. 2 i 3.



Rys. 2. Weryfikacja metodą "hasło-odzew"



Rys. 3. Uwierzytelnienie z pomocą kart z hasłami jednorazowymi



Rys. 4. Przykładowe tokeny wykorzystywane przy uwierzytelnianiu transakcji bankowych

5. System Kerberos

W procesie uwierzytelniania może być wykorzystywana zaufana strona trzecia (*trusted third-party*), która poświadcza tożsamość klienta i serwera. Jest nazywana **serwerem bezpieczeństwa** (*security server*). Jego zadaniem jest przechowywanie haseł wykorzystywanych podczas weryfikacji użytkowników i serwerów. Jest to jedyne miejsce przechowywania haseł.

Wymagania:

- Zapewnienie dwustronnego uwierzytelnienia dwukierunkowego. Zadaniem trzeciej strony jest przechowywanie i pielęgnacja haseł.
- Hasła nie powinny być przesyłane poprzez sieć.
- Hasła nie powinny być przechowywane na stacji klienckiej.
- Zarejestrowany użytkownik powinien otrzymać tymczasowy klucz tajny. Jest on wykorzystywany przez klienta przy wszystkich dostęпах.
- System powinien pozwalać na bezpieczne przesyłanie kluczy szyfrowania pomiędzy klientami i serwerami.

Przykładem systemu uwierzytelniania z udziałem strony trzeciej jest system *Kerberos*.

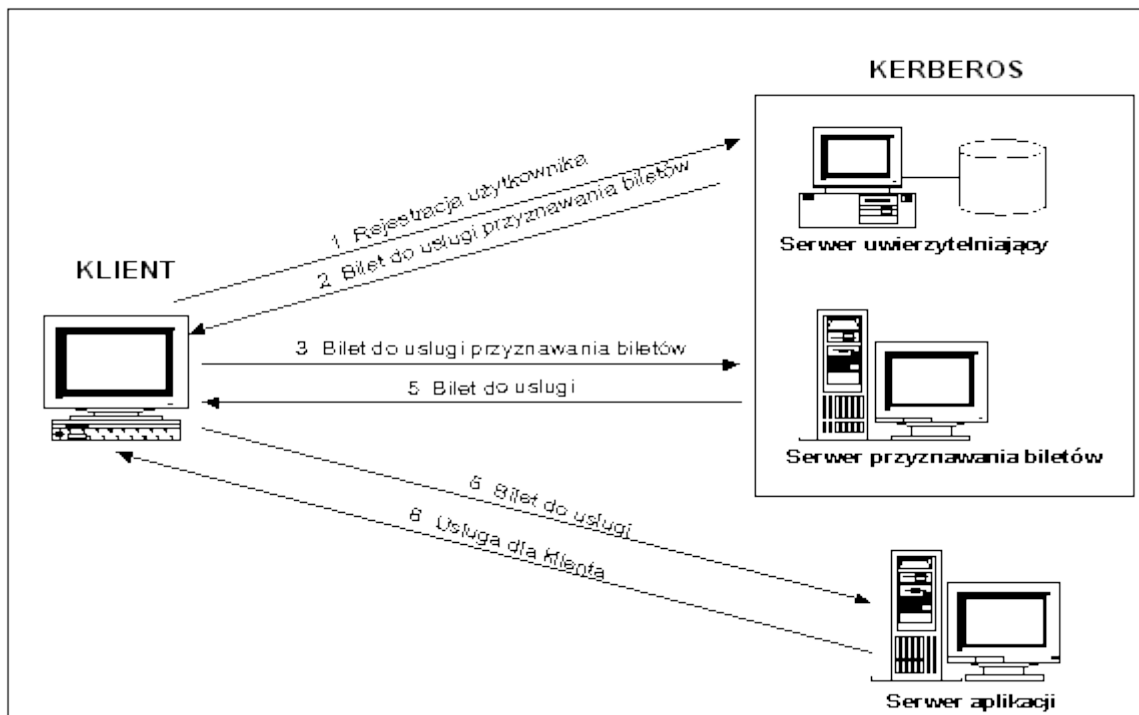
System **Kerberos** powstał w czasie realizacji projektu *Athena* na uniwersytecie MIT. Projekt miał na celu integrację komputerów uniwersyteckich. System weryfikacji autentyczności jest oparty na znajomości haseł zapisanych w serwerze *Kerberos*. W procesie uwierzytelniania wykorzystuje się tajny dzielony klucz (*shared secret*), który pozwala na identyfikację użytkowników bez ekspozycji informacji narażających bezpieczeństwo sieci. W systemie uwierzytelniania wyróżnić można cztery komponenty.

- Pierwszy komponent to **klient** czyli użytkownik lub aplikacja reprezentująca użytkownika. Jest to miejsce, z którego użytkownik prowadzi pracę, wprowadza identyfikator i hasło.
- Drugi komponent to **serwer uwierzytelniający** (*authentication server*) służący do przechowywania haseł i sprawdzania tożsamości użytkownika. W czasie wymiany informacji z klientem dostarcza on klientowi bilet uprawniający do korzystania z usługi przyznawania biletów (*ticket-granting ticket*).
- Komponent trzeci to **serwer przepustek** lub **serwer przyznawania biletów** (*ticket-granting server*), który dostarcza klientowi bilet uprawniający do skorzystania z serwera aplikacji.
- Komponent czwarty to **serwer aplikacji** (*application server*) czyli zasób, który chce się upewnić, że dany klient jest poprawny. Dostarcza klientowi żądanej przez niego usługi.

Konto w bazie zawiera dane dotyczące tożsamości oraz klucze główne (np. hasła) wszystkich klientów i serwerów z danego obszaru. Klucz główny serwera uwierzytelniającego służy do szyfrowania wszystkich kluczy głównych klientów udaremniając nieautoryzowany dostęp do serwera.

Poważną wadą systemu jest jego dostęp do zaszyfrowanych haseł użytkowników. Powoduje to, że zawarte są w nim dane krytyczne dla bezpieczeństwa i powinien być on chroniony w sposób szczególny.

Serwer *Kerberos* jest bezstanowy. Odpowiada po prostu na żądania użytkowników i wydaje przepustki (żetony, bilety). Ułatwia to tworzenie replikowanych serwerów zapasowych. Funkcjonowanie systemu z punktu widzenia użytkownika niczym nie różni się od systemu tradycyjnego. Idea systemu Kerberos została przedstawiona na rys. 5.



Rys. 5. Idea systemu Kerberos

Wymiana informacji między klientem (C1) i serwerem uwierzytelniającym (AS)

Celem tej wymiany jest weryfikacja tożsamości użytkownika i nadanie mu prawa do otrzymywania biletu dostępu do usługi

C1>AS: C1, TGS, T₂

Klient C1 wysyła niezaszyfrowany komunikat do AS prosząc o bilet (*ticket*) na komunikację z serwerem TGS. Komunikat zawiera identyfikator klienta (C1), identyfikator serwera przyznającego bilety i znacznik czasowy umożliwiający synchronizację zegarów C1 i AS. AS na podstawie przechowywanego u siebie hasła użytkownika tworzy klucz szyfrowania.

AS>C1: {TGS, K_{C1,TGS}, T₂, L₂, {TGT_{C1,TGS} K_{AS,TGS}} K_{C1}

AS odsyła do C1 komunikat zaszyfrowany kluczem wytworzonym na podstawie hasła klienta C1. Komunikat ten zawiera identyfikator serwera przyznającego bilety (TGS), klucz sesyjny (klucz znany klientowi C1 i TGS), znacznik czasu (T₂), okres ważności biletu (L₂).

Druga część komunikatu zawiera **przepustkę udzielającą przepustki** (*Ticket Granting Ticket - TGT*) zaszyfrowaną kluczem wspólnym AS i TGS. TGT nie jest odszyfrowywana przez klienta C1. Będzie wykorzystywana do otrzymywania zezwolenia uzyskania określonych usług wewnątrz obszaru odpowiedzialności Kerberos. TGT eliminuje potrzebę ponawiania procesu identyfikacji przy każdej następnej prośbie.

TGT zawiera: identyfikator i adres klienta, identyfikator serwera TGS, klucz sesyjny (K_{C1,TGS}), znacznik czasu (T₂), okres ważności biletu (L₂).

Klucze wykorzystywane przy szyfrowaniu mają długość 56 bitów i są uzyskiwane na podstawie hasła określonej jednostki.

Wymiana informacji między klientem (C1) i serwerem przepustek (TGS)

Po otrzymaniu *przepustki udzielającej przepustki* (TGT) użytkownik może wykonywać operacje wymagające uwierzytelnienia (np. dostęp do pliku). Kiedy użytkownik po raz pierwszy próbuje uzyskać dostęp do serwera aplikacji (S1) zabezpieczonego przez *Kerberos*, oprogramowanie jego stacji komunikuje się z **serwerem przepustek** (TGS) i prosi o przepustkę do serwera aplikacji.

C1>TGS: {S1, C1, T₃} K_{C1,TGS},
{TGT_{C1,TGS} K_{AS,TGS}

Prośba jest zakodowana kluczem sesyjnym ($K_{C1,TGS}$) otrzymanym poprzednio od AS. Zawiera nazwę serwera S1 i poświadczenie (*authenticator*) klienta (nazwę i adres klienta oraz znacznik czasu). Poświadczenie ma bardzo krótki okras ważności i nie może być wykorzystywane wielokrotnie. Klient przedstawia również TGT zakodowany kluczem wspólnym AS i TGS. Zawarty w TGT klucz sesyjny umożliwia rozszyfrowanie prośby.

TGS w odpowiedzi tworzy klucz sesyjny, którym będą się posługiwały C1 i S1 ($K_{C1,S1}$). Tworzy też specjalną przepustkę ($B_{C1,S1}$) uprawniającą do dostępu do serwera S1 i wysyła komunikat do klienta C1.

TGS>C1: $\{ K_{C1,S1}, S1, T_4, \{ B_{C1,S1} \}_{K_{S1,TGS}} \}_{K_{C1,TGS}}$

Uzyskany bilet jest przedstawiany serwerowi plików razem z żądaniem dostępu. Bilet ten zawiera sesyjny klucz szyfrowania $K_{C1,S1}$, identyfikator i adres klienta, identyfikator serwera S1, znacznik czasu i okres ważności biletu. Jest zaszyfrowany kluczem współdzielonym przez TGS i S1.

Wymiana informacji między klientem (C1) i serwerem aplikacji (S1)

C1>S1: $\{ B_{C1,S1} \}_{K_{S1,TGS}}, \{ C1, T_5 \}_{K_{C1,S1}}$

Bilet zawiera m.in. klucz sesyjny $K_{C1,S1}$. Jest on odszyfrowywany przez serwer plików. W ten sposób serwer uzyskuje klucz sesyjny i może odczytać (odszyfrować dane identyfikacyjne klienta i znacznik czasu).

S1>C1: $\{ T_5 + 1 \}_{K_{C1,S1}}$

Serwer koduje i wysyła oryginalny znacznik czasu klienta zwiększony o 1. C1 po odszyfrowaniu komunikatu uzyskuje pewność co do autentyczności serwera, gdyż tylko on mógł dokonać odszyfrowania biletu, uzyskać klucz sesyjny i użyć go do zakodowania znacznika czasu.

Wszystkie przesyłane żądania i przepustki są szyfrowane. Informacja o porze dnia (znacznik czasu) jest umieszczana w przepustce aby uniemożliwić atak przez powtarzanie (podśluchanie, przechwycenie, ponowienie żądania w późniejszym czasie).

Serwer przepustek jest w stanie ustalić tożsamość użytkownika gdyż:

- Żądanie przepustki do serwera plików jest zaszyfrowane kluczem sesyjnym $K_{C1,TGS}$.
- Użytkownik może poznać ten klucz sesyjny poprzez odszyfrowanie komunikatu odebranego z AS.
- Aby odszyfrować ten komunikat należy znać hasło użytkownika.

Serwer aplikacji może ustalić tożsamość użytkownika gdyż:

- Przepustka przedstawiana przez użytkownika jest szyfrowana za pomocą klucza dzielonego przez serwer aplikacji i TGS.
- Przepustka zawiera adres IP i nazwę użytkownika.
- W/w. dane są umieszczane w przepustce przez TGS, który jest pewny tożsamości użytkownika, a to wystarcza serwerowi plików.

System *Kerberos* to system weryfikacji autentyczności, który może być wykorzystywany z wieloma schematami RPC. Może być również wykorzystywany do samej wymiany kluczy. Istnieje taka wersja polecenia *telnet*. Istnieje również taka, zmodyfikowana przez MIT, wersja NFS.

1. IPSec

Opracowaniem protokołu IPSec zajęła się w 1992 roku samodzielna grupa robocza IETF (*Internet Engineering Task Force*). Pierwsze wersje zostały przedstawione w 1995 roku. Ipssec jest protokołem warstwy trzeciej (poziom protokołu IP) według modelu OSI zapewniającym:

- poufność (szyfrowanie),
- integralność (skrót),
- uwierzytelnienie (podpisywanie) użytkownika lub komputera,
- przezroczystość dla aplikacji.

IPSec jest integralną częścią protokołu IPv6 (adres 128 bitowy) lub Ipvng (*IP Next Generation*). Ponieważ znajduje się jeszcze w fazie rozwojowej, więc stworzono specyfikację IPSec dla obecnie używanego IPv4.

Przezroczystość dla użytkownika oznacza, że nie musi on nawet wiedzieć o szyfrowaniu ruchu. Nie musi pamiętać żadnych haseł, czy trzymać w bezpiecznym miejscu prywatnych kluczy. IPSec umożliwia zabezpieczenie standardowych protokołów takich jak POP3, SMTP, HTTP bez nanoszenia jakichkolwiek poprawek do kodu ich serwerów i klientów. Rozwiązuje to odwieczny problem przesyłania haseł w postaci jawnej (np. POP3) lub innych cennych informacji (numery kart kredytowych).

Składniki IPSec:

- Protokoły bezpieczeństwa:
 - uwierzytelniający (*Authentication Header - AH*)
 - zabezpieczenia zawartości pakietu (*Encapsulating Security Payload - ESP*) - Skojarzenia zabezpieczeń (*Security Associations*)
- Zarządzanie kluczami (*Internet Key Management - IKE*)
- Algorytmy uzgadniania parametrów (ISAKMP, Photuris), szyfrowania, kompresji danych (IPCOMP).

Protokół AH zapewnia usługi związane z uwierzytelnieniem pakietu. Robi to za pomocą algorytmów typu MAC (*Message Authentication Code*). Dodatkowo zapewnia to również integralność przesyłanych danych. Protokół ESP zapewnia poufność danych i funkcjonalność protokołu AH. Oprócz mechanizmów MAC stosuje on algorytmy szyfrujące dane.

Skojarzenia zabezpieczeń definiują jednokierunkowe połączenie, które zabezpiecza transmitowane dane. Komunikacja pomiędzy dwoma urządzeniami w sieci wymaga co najmniej dwóch takich połączeń. Każde połączenie SA jest identyfikowane przez trzy parametry:

- *Security Parameter Index (SPI)*,
- Adres IP przeznaczenia (DA),
- Protokół bezpieczeństwa (AH lub ESP).

Każdy pakiet IPSec zawiera w nagłówku numer SPI, pozwalający na określenie informacji potrzebnych do odszyfrowania treści pakietu, sprawdzenia jego integralności lub potwierdzenia tożsamości nadawcy. Pozwala on zlokalizować SA, które określa:

- informacje definiujące algorytm szyfrowania,
- informacje definiujące algorytm uwierzytelniania,
- informacje definiujące algorytm sprawdzania integralności,
- klucze szyfrujące i kodujące wykorzystywane w AH i ESP,
- okres ważności kluczy,
- okres ważności tunelu.

Każdy generowany w urządzeniu pakiet musi być konstruowany zgodnie z przyjętą wcześniej polityką bezpieczeństwa. Zalecenia IETF definiują dwie bazy danych, w których przechowywane są informacje na temat sposobu traktowania wszystkich pakietów IP:

- baza polityki bezpieczeństwa (*Security Policy Database - SPD*),
- baza połączeń bezpieczeństwa (*Security Association Database - SAD*).

Zalecenia nie definiują budowy tych baz, wskazują jednak na konieczność implementowania ich dla każdego interfejsu sieciowego osobno (jeśli urządzenie posiada więcej niż jeden interfejs) i wskazują ogólne założenia, jakie powinny być przez nie spełnione.

Baza SPD musi określać sposób traktowania każdego pakietu IP - czy należy go odrzucić, przepuścić omijając mechanizm IPSec czy zastosować ochronę IPSec. W tym trzecim przypadku baza SPD powinna zawierać informacje o usługach bezpieczeństwa, jakie tego pakietu dotyczą (protokoły, algorytmy itp.). SPD musi posiadać interfejs zarządzający, umożliwiający tworzenie, modyfikację i usuwanie rekordów z bazy oraz pola służące do selekcji rekordów (pola selekcyjne).

Baza SAD jest powiązana z SPD dla ruchu wychodzącego - rekord SAD jest wyznaczany poprzez rekord SPD. W przypadku ruchu przychodzącego rekordy SAD identyfikowane są przez adres IP przeznaczenia, typ protokołu IPSec i SPI. Rekordy SAD tworzone są w chwili zestawiania połączenia bezpieczeństwa (SA) i zawierają parametry wynegocjowane dla tego połączenia:

- licznik numerów sekwencyjnych - do generacji numerów sekwencyjnych w nagłówkach protokołów AH i ESP,
- znacznik przekroczenia zakresu licznika numerów sekwencyjnych,
- dla AH - algorytm uwierzytelnienia, klucze itp.,
- dla ESP - algorytm szyfrowania, klucze itp. oraz jeśli ESP umożliwia uwierzytelnienie - odpowiedni algorytm i klucze,
- czas życia danego SA, po którym musi być stworzone nowe połączenie lub dane połączenie musi zostać zamknięte,
- tryb pracy - transportowy lub tunelowany.

Tryb transportowy: jest charakterystyczny dla bezpośrednich połączeń *komputer - komputer*. Polega na dodaniu za nagłówkiem IP (IPv4), a przed nagłówkiem warstwy transportowej lub za podstawowym nagłówkiem IP (IPv6) nagłówka protokołu bezpieczeństwa (AH lub ESP). Nagłówek IP nie jest więc ukrywany. Z tego powodu można go stosować tylko do transmisji w sieciach LAN (w WAN - problemy z fragmentacją i routingiem). Tryb transportowy stosuje się do komunikacji między komputerami i komunikacji komputerów z bramkami IPSec. W pakiecie szyfrowane są tylko dane. Oryginalny nagłówek IP pozostaje niezmieniony, ale może być podpisany. Zaletą tego rozwiązania jest to, że do każdego pakietu dodawanych jest tylko kilka bajtów. Tryb ten umożliwia urządzeniom sieci publicznej określanie adresu źródłowego i docelowego każdego pakietu. Pozostawienie nieszyfrowanego nagłówka umożliwia obcym prowadzenie analizy ruchu pomiędzy węzłami. Przesyłane dane mogą być jednak szyfrowane.

Tryb tunelowy: jest charakterystyczny dla połączeń *sieć-sieć*. Oryginalny datagram IP jest w całości szyfrowany stając się zawartością w nowym pakiecie IP. Funkcje szyfrowania, deszyfrowania, sprawdzania integralności i uwierzytelnienia realizują bramy (*gateway*) rozpoznające protokół IPSec. Źródłowa stacja kliencka wysyła pakiety do sieci odległej w takiej samej (niezaszyfrowanej) postaci jak do innych hostów swojej sieci lokalnej. Całą pracę związaną z zapewnieniem bezpiecznego przesyłania danych wykonują bramy na obu końcach zestawionego tunelu. Główną zaletą tego rozwiązania jest fakt, że systemy docelowe nie muszą być modyfikowane aby korzystać z IPSec. Ten tryb uniemożliwia analizę ruchu. Ukrywane jest prawdziwe źródło i miejsce przeznaczenia pakietu. Jedynym nie szyfrowanym (ale podpisywanym) elementem pakietu jest jego zewnętrzny nagłówek IP. Z zewnątrz możliwe jest więc określenie jedynie końców tunelu.

IPSec zapewnia mechanizmy manualnej i automatycznej wymiany kluczy. Pierwszy z nich polega na wpisywaniu przez administratora wszystkich kluczy używanych w zabezpieczanych połączeniach. W praktyce jest on używany tylko w małych, statycznych środowiskach. Nie jest to rozwiązanie skalowalne. Drugi umożliwia tworzenie większych rozwiązań i jest dobrze skalowalny. Umożliwia implementację wielu różnych systemów dystrybucji klucza. Negocjacje podlegają wymagania odnośnie poziomów zabezpieczeń, które będą stosowane przez obie strony:

- algorytmy szyfrujące,
- algorytmy uwierzytelnienia,
- algorytmy kompresji,
- kombinacje w/w w poszczególnych kanałach SA,
- parametry szczegółowe algorytmów i kluczy kryptograficznych.

Podstawowe protokoły negocjacji i dystrybucji:

- **ISAKMP** - protokół wymiany parametrów kanałów SA.
- **OAKLEY** - protokół wymiany kluczy oparty o algorytm *Diffie-Hellmana*.
- **IKE** - protokół będący połączeniem ISAKMP i OAKLEY.
- **PHOTURIS** - protokół negocjacji parametrów kluczy oparty o algorytm *Diffie-Hellmana*.
- **SKIP** - jw. - produkcji *Sun Microsystems*.

IETF zaleca system IKE (*Internet Key Exchange*). W protokole IKE budowa połączenia przebiega w dwóch etapach. Najpierw ustalana jest tożsamość węzłów i budowany jest bezpieczny, uwierzytelniony tunel między dwoma hostami, a później następuje negocjowanie SA. Uwierzytelnienie w tym procesie może być realizowane przez systemy takie jak Kerberos, klucze preinstalowane lub certyfikaty. Okresowo realizowana może być renegotjacja parametrów połączenia.

2. L2TP i PPTP

Sieć VPN to bezpieczny tunel pomiędzy komputerem zdalnego użytkownika a prywatną siecią organizacji przechodzący poprzez Internet. W tej chwili podstawowymi protokołami wykorzystywanymi do budowy VPN są:

- L2TP (*Layer 2 Tunneling Protocol*) - RFC 2661
- PPTP (*Point-to-Point Tunneling Protocol*) - RFC 2637.

Oba protokoły bazują na protokole PPP. PPP jest podstawowym elementem w obu protokołach oraz jedynym, który kapsułkuje przekazywane dane (tj. ładunki) w sieciach prywatnych. PPTP i L2TP dodają po prostu kolejną warstwę kapsułkowania do tunelowanych ładunków w sieci publicznej.

Protokół PPP w warstwie łącza danych modelu OSI został pierwotnie opracowany do kapsułkowania danych i przenoszenia ich pomiędzy dwoma punktami. Protokół PPP ma wiele zalet - m.in. uwierzytelnianie i kompresja - których nie zapewnia jego starszy kuzyn, czyli protokół SLIP (*Serial Line Internet Protocol*). Nadzbiór protokołu PPP zajmuje się obsługą połączeń: PPP LCP (*Link Control Protocol*) nawiązuje, konfiguruje, obsługuje i zakańcza połączenia pomiędzy dwoma punktami, PPP NCP (*Network Control Protocol*) nawiązuje i konfiguruje różne protokoły warstwy sieciowej w połączeniach PPP.

Uwierzytelnianie w połączeniach VPN przybiera dwie formy:

- Uwierzytelnianie użytkowników. Aby połączenie VPN mogło zostać nawiązane, serwer VPN uwierzytelnia klienta VPN próbującego je ustanowić i weryfikuje, czy klient posiada odpowiednie uprawnienia. Jeżeli używane jest uwierzytelnianie wzajemne, klient VPN dodatkowo uwierzytelnia serwer VPN, chroniąc się w ten sposób przed podszywającymi się serwerami.
- Uwierzytelnianie danych i kontrola ich integralności. Aby zapewnić, że dane przesyłane przez połączenie VPN pochodzą z drugiego jego zakończenia, i że nie zostały zmodyfikowane podczas przesyłania, zawierają one kryptograficzną sumę kontrolną obliczoną w oparciu o klucz szyfrowania, znany jedynie nadawcy i odbiorcy.

PPTP oraz L2TP kapsułkują i standardowo szyfrują dane przed ich przeniesieniem. Jednak przed rozpoczęciem kapsułkowania związanego z tunelowaniem występuje kapsułkowanie przez PPP. W kapsułkowaniu PPP pojedyncza jednostka danych jest umieszczana wewnątrz innej jednostki podczas przechodzenia przez kolejne warstwy modelu OSI. Dla przykładu, protokół TCP (w warstwie transportowej) jest kapsułkowany poprzez protokół IP (w warstwie sieciowej), który następnie jest kapsułkowany przez protokół PPP (w warstwie łącza danych).

Protokoły tunelowania są protokołami warstwy wyższej, które transportują zakapsułkowany ładunek. Protokół VPN kapsułkuje już wcześniej zakapsułkowany ładunek i przesyła go pomiędzy krańcami tunelu.

Protokół PPTP kapsułkuje ramki protokołu PPP w datagramach IP przed ich transmisją w sieci opartej na protokole IP. Klienci PPTP stosują port docelowy TCP 1723 do utworzenia połączenia sterującego protokołu PPTP dla tunelu.

Protokół PPTP używa połączenia TCP zwanego połączeniem kontroli PPTP do tworzenia, utrzymywania i kończenia tunelu. Pakiety kontroli PPTP składają się z nagłówka IP, nagłówka TCP oraz komunikatu kontroli PPTP. Pakiet kontroli zawiera ponadto nagłówek i końcówkę warstwy łącza danych.

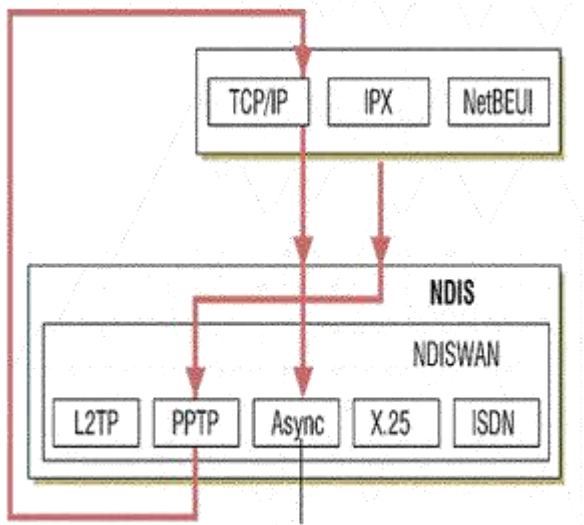
Oprócz połączenia sterującego PPTP, które protokół używa do utrzymania tunelu, PPTP korzysta również z łącza do tunelowania danych. Do kapsułkowania ramek PPP jako tunelowanych danych wykorzystywana jest zmodyfikowana wersja protokołu GRE (*Generic Routing Encapsulation*). Tunelowanie danych odbywa się podczas dwóch etapów kapsułkowania. Podczas tworzenia ładunku PPP dane przechodzą w dół poprzez kolejne warstwy modelu OSI, poczynając od warstwy aplikacji, a kończąc na warstwie łącza danych. Po utworzeniu ładunku dane przekazywane są z powrotem w modelu OSI i kapsułkowane przez protokoły wyższych warstw.

Po dotarciu danych do warstwy transportowej, protokół ten nie może przesłać ładunku, bowiem jest to zadanie, za które odpowiada warstwa łącza danych. PPTP zarządza zadaniami warstwy 2, które zwykle należą do protokołu PPP oraz dodaje nagłówek PPP i końcówkę (*trailer*) do struktury danych PPTP. PPTP szyfruje ładunek, następnie kapsułkuje go z nagłówkiem PPP, aby utworzyć ramkę warstwy łącza danych. PPTP w kolejnej fazie kapsułkuje ramkę PPP do postaci pakietu GRE (*Generic Routing Encapsulation*), który operuje na poziomie warstwy sieci. GRE udostępnia sposób kapsułkowania protokołów warstwy 3, takich jak IPX, AppleTalk i DECnet dla sieci IP. Brakuje w nim jednak możliwości konfigurowania sesji oraz zabezpieczeń. Dlatego stosowane jest połączenie sterujące PPTP, dzięki któremu można konfigurować oraz zabezpieczać sesje. Zastosowanie GRE jako metody kapsułkowania ogranicza wykorzystanie PPTP do sieci IP.

Po zakapsułkowaniu ramki PPP z nagłówkiem GRE protokół PPTP kapsułkuje ramkę z nagłówkiem IP. Zawiera on adres źródłowy i docelowy pakietu. Na zakończenie PPTP dodaje nagłówek PPP i końcówkę. System źródłowy przesyła następnie dane poprzez tunel. System docelowy usuwa z danych wszystkie nagłówki i końcówki aż dotrze do ładunku PPP. Na rys. 1. Przedstawiono ramkę protokołu PPTP, a na rys. 2. schemat procedury tworzenia tej ramki.

nagłówek łącza danych	nagłówek IP	nagłówek GRE	nagłówek PPP	zaszyfrowany ładunek PPP	końcówka łącza danych
-----------------------	-------------	--------------	--------------	--------------------------	-----------------------

Rys. 1 Ramka protokołu PPTP



Rys. 2 Schemat procedury tworzenia ramki PPTP

Protokół L2TP to połączenie PPTP i protokołu L2F (*Layer 2 Forwarding*). PPTP zapewnia tunel dla protokołu PPP, zaś L2F tuneluje protokoły SLIP i PPP. Po zaprojektowaniu L2F przez *Cisco Systems* organizacja IETF zaleciła połączenie PPTP i L2F w jeden protokół, by uniknąć pomyłek i problemów ze współpracą. L2TP zawiera według wielu opinii najlepsze właściwości dostępne w PPTP oraz w L2F.

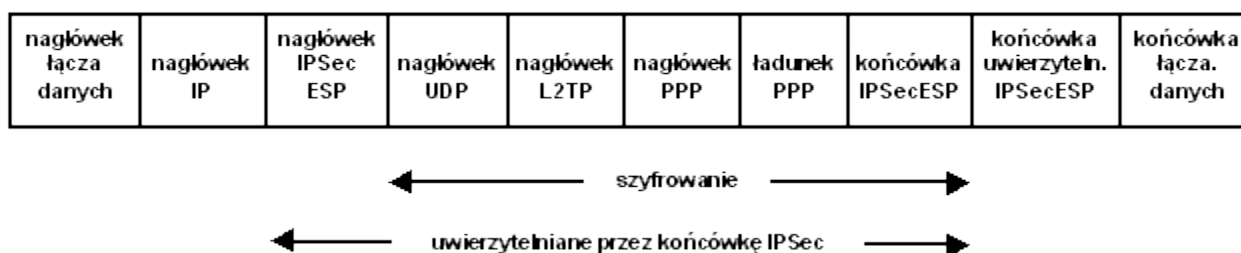
Jednym z istotniejszych udoskonaleń w L2TP jest możliwość uruchamiania tego protokołu nie tylko w sieciach opartych na protokole IP, lecz również w sieciach ATM, X.25 oraz Frame Relay. W Win2K obsługiwany jest wyłącznie protokół IP.

L2TP wykorzystuje do obsługi tunelowania ten sam format komunikatów jak w przypadku tunelowania danych. UDP to preferowany protokół warstwy transportowej dla L2TP. W implementacji L2TP Microsoftu komunikaty sterujące są szyfrowane ładunkiem PPP przesyłanym przez IP jako komunikaty UDP. Komunikaty L2TP zawierają pole *Next-Received* oraz pole *Next-Sent*, które można porównać odpowiednio z polami TCP o nazwie *Acknowledgement Number* oraz *Sequence Number*.

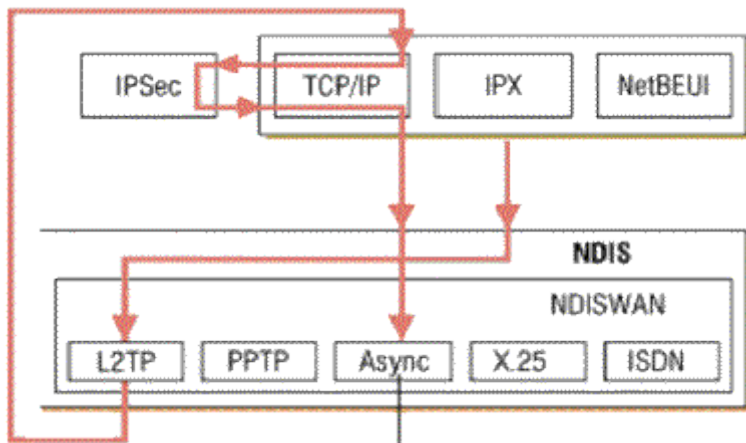
Tak samo jak w przypadku tunelowania danych w ramach PPTP, tunelowanie danych L2TP rozpoczyna się od ładunku PPP. L2TP kapsułkuje ładunek PPP dodając nagłówek PPP i L2TP, w wyniku czego otrzymujemy pakiet zakapsułkowany przez L2TP. L2TP wykorzystuje port UDP 1701 zarówno dla portu źródłowego, jak i docelowego. W zależności od wybranej zasady IP Security, L2TP może zaszyfrować komunikat UDP oraz dodać nagłówek i końcówkę ESP protokołu IPsec. L2TP następnie kapsułkuje pakiet IPsec dodając nagłówek IP, który zawiera adres źródłowy i docelowy. Na końcu L2TP wykonuje drugie kapsułkowanie PPP w celu przygotowania danych do transmisji.

Kiedy komputer docelowy otrzymuje dane, przetwarzany jest nagłówek i końcówka PPP, a następnie usuwany jest nagłówek IP. Komputer korzysta z końcówki IPsec Authentication do uwierzytelnienia ładunku IP, a następnie przy użyciu nagłówka ESP IPsec deszyfruje pakiet.

W dalszej kolejności system przetwarza nagłówek UDP, po czym używa nagłówka L2TP do identyfikacji tunelu. Po tej operacji pozostanie tylko ładunek PPP, serwer zaś przetwarza pozostałe dane lub przekazuje je do właściwego systemu docelowego. Na rys. 3. Przedstawiono ramkę protokołu L2TP, a na rys. 4. schemat procedury tworzenia tej ramki.



Rys. 3 Ramka protokołu L2TP



Rys. 4 Schemat procedury tworzenia ramki L2TP

Zabezpieczenia PPTP i L2TP

Do uwierzytelniania w ramach PPTP stosowane są protokoły uwierzytelniania oparte na PPP, włączając w to EAP (*Extensible Authentication Protocol*), MSCHAP (*Microsoft Challenge Handshake Authentication Protocol*), CHAP (*Challenge Handshake Authentication Protocol*), SPAP (*Shiva Password Authentication Protocol*) oraz PAP (*Password Authentication Protocol*). MSCHAP v. 2 oraz EAP-TLS (*Transport Layer Security*) to najbezpieczniejsze z wymienionych protokołów.

MPPE (*Microsoft Point-to-Point Encryption*) negocjuje szyfrowanie w połączeniu PPTP i może zostać wykorzystany wyłącznie z MSCHAP (w wersja 1 i 2) oraz EAP-TLS. MPPE pozwala na zastosowanie szyfrowania w oparciu o jeden z trzech kluczy o sile 40, 56 lub 128 bitów.

PPTP zmienia klucze szyfrowania wraz z każdym otrzymanym pakietem. MPPE został zaprojektowany pod kontem połączeń typu punkt-punkt, w których pakiety danych przychodzą w odpowiedniej kolejności oraz w których niektóre pakiety danych są tracone. W tego typu środowisku klucz szyfrowania jednego pakietu jest uzależniony od szyfrowania poprzedniego pakietu. W takiej konfiguracji środowisko VPN nie zadziała, gdyż pakiety często przychodzą nie po kolei. Dlatego PPTP szyfruje pakiety niezależnie od innych i stosuje numer sekwencji do zmiany klucza szyfrowania, dzięki czemu proces szyfrowania działa mimo braku informacji o poprzednich pakietach. Chociaż PPTP jest względnie bezpiecznym protokołem, nie jest on tak bezpieczny jak L2TP z IPSec, który zapewnia uwierzytelnianie na poziomie użytkownika i komputera, a także uwierzytelnianie oraz szyfrowanie.

Do uwierzytelnienia klienta, jak i serwera VPN, L2TP z IPSec używa certyfikatów lokalnych komputerów, uzyskanych z odpowiedniego urzędu certyfikacji (CA).

Kiedy L2TP z IPSec zakończy uwierzytelnianie komputerów rozpoczyna uwierzytelnianie na poziomie użytkownika. Możemy wybrać dowolny protokół uwierzytelniania PPP - nawet PAP, który przesyła nazwę użytkownika i hasło jawnym tekstem - a mimo to proces dalej jest bezpieczny, gdyż L2TP z IPSec szyfruje całą sesję. Możemy jednak spowodować, że uwierzytelnianie użytkownika będzie bezpieczniejsze dzięki zastosowaniu MS CHAP, który stosuje klucze szyfrowania oddzielne od szyfrowania na poziomie komputera.

L2TP z IPSec stosuje algorytm 3DES, więc szyfrowanie danych odbywa się na dużo wyższym poziomie niż w PPTP. Jeżeli wystarczy nam niższy poziom zabezpieczeń (który pozwala także zmniejszyć obciążenie komputera dodatkowymi operacjami), możemy wdrożyć DES.

L2TP z IPSec zapewnia również uwierzytelnianie danych. Do uwierzytelniania danych L2TP z IPSec wykorzystujemy HMAC (*Hash Message Authentication Code*) MD5. Jest to algorytm haszujący, który generuje 128-bitowy kod do uwierzytelniania danych.

Kluczowe różnice między protokołem PPTP i L2TP są następujące:

- Protokół PPTP wymaga, by warstwa transportowa sieci oparta była na IP, podczas gdy L2TP wymaga jedynie, by sieć, w której jest wykorzystywany, umożliwiała zestawienie połączeń punkt-punkt. W związku z tym protokół L2TP może być używany bezpośrednio w sieciach IP, Frame Relay, X.25 czy ATM. PPTP nie może pracować bezpośrednio w sieciach nie stosujących IP.
- PPTP wspiera jedynie jeden tunel między serwerem VPN a klientem. Użycie protokołu L2TP zapewnia wykorzystanie wielu tuneli między punktami końcowymi. W efekcie w L2TP możliwe jest stworzenie wielu tuneli dla różnych poziomów jakości usługi (OoS) czy dla różnych poziomów zabezpieczeń.
- Protokół L2TP dostarcza mechanizmów umożliwiających kompresję nagłówka. Jeśli funkcja ta zostaje aktywowana, wówczas nagłówek L2TP jest mniejszy od nagłówka PPTP, dzięki czemu przepustowość łączy wykorzystywanych w trakcie tunelowania jest lepiej wykorzystywana.

3. SSL i TLS

Protokół SSL (*Secure Socket Layer*) został opracowany przez firmę *Netscape*. Projektowany był jako protokół otwarty, czyli charakteryzujący się brakiem przywiązania do jednego algorytmu szyfrowania. Realizuje uwierzytelnienie (autoryzację), szyfrowanie oraz zapewnia integralność wiadomości. Posiada wbudowany mechanizm uwierzytelniania serwera i opcjonalnie klienta. Współpracuje z zaporami sieciowymi i połączeniami tunelowanymi. Bazuje na protokole zapewniającym niezawodną komunikację (np. TCP). Jest niezależny od aplikacji warstw wyższych. Dzięki temu może być wykorzystywany do zabezpieczania takich protokołów jak TELNET, FTP, HTTP. Na stosie protokołów TCP/IP SSL leży pomiędzy warstwą aplikacji zawierającą HTTP, SMTP, Telnet, FTP i inne a warstwą transportową, która zawiera protokół TCP.

SSL wykorzystuje dwa rodzaje kryptografii: symetryczną (z pojedynczym kluczem) oraz niesymetryczną (z kluczem prywatnym i publicznym).

Przebieg sesji:

1. Nawiązanie połączenia poprzez zwykłe TCP.
2. Wymiana informacji o obsługiwanych algorytmach szyfrowania, certyfikatów i innych danych.
3. Uzgodnienie wspólnego zbioru algorytmów.
4. Potwierdzenie tożsamości serwera i opcjonalnie klienta.
5. Wymiana kluczy sesyjnych.
6. Przesyłanie danych.

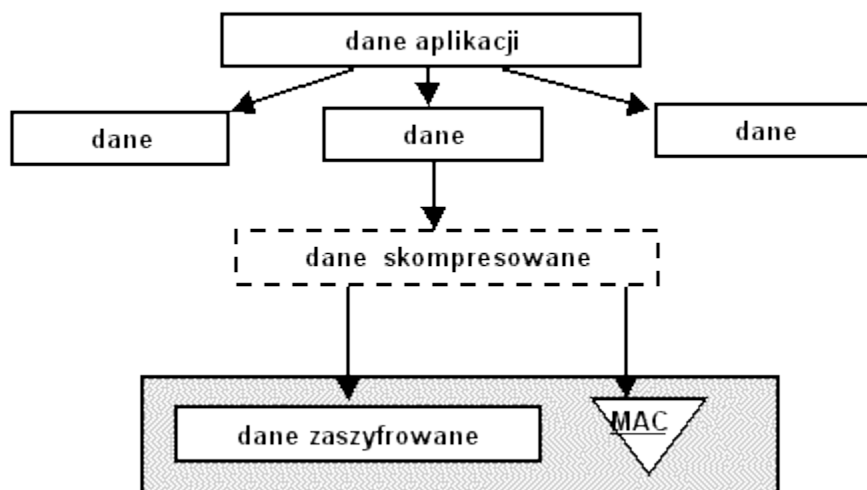
Protokół SSL jest właściwie zestawem protokołów:

- **SSL Record Protocol** - używany do bezpiecznego przesyłania wiadomości,
- **SSL Handshake Protocol** - używany do negocjowania parametrów bezpiecznego połączenia,

SSL Change Cipher Spec Protocol - używany do zmiany szyfrowania,

- **SSL Alert Protocol** - używany do alarmowania.

Protokół rekordu służy do zapewnienia bezpieczeństwa i integralności danych i w tych celach jest stosowany przez "wyższe" protokoły wchodzące w skład SSL. Celem protokołu rekordu jest podzielenie danych przesyłanych przez aplikację, opakowanie ich w odpowiednie nagłówki i stworzenie obiektu zwanego właśnie rekordem, który zostaje zaszyfrowany i może zostać przekazany do przesłania poprzez protokół TCP. Schemat procedury realizowanej przez ten protokół przedstawiono na rys. 5.



Rys. 5 Schemat procedury realizowanej przez *SSL Record Protocol*

Pierwszym krokiem realizowanym podczas przygotowania danych aplikacji do przesłania jest podzielenie ich na jednostki po 16 kB lub mniejsze. Taka porcja danych może zostać poddana dodatkowo kompresji, jednak w specyfikacji protokołu SSL 3.0 nadal nie został ustalony żaden protokół kompresji, więc w chwili obecnej kompresja danych nie jest stosowana.

Dla każdego fragmentu rozpoczyna się budowanie rekordu, poprzez dodanie:

- nagłówka,
- ewentualnych informacji uzupełniających do wymaganego rozmiaru danych,
- znacznika MAC.

Nagłówek rekordu, dodawany do każdej porcji danych, zawiera dwie elementarne informacje, a mianowicie długość rekordu i długość bloku danych dołożonych do danych podstawowych.

Kolejnym krokiem po stworzeniu nagłówka rekordu jest zbudowanie danych rekordu, które składają się z następujących elementów:

- danych podstawowych,
- danych uzupełniających pakiet do wymaganego rozmiaru,
- znacznika MAC.

Znacznik MAC ma za zadanie zapewnić możliwość sprawdzenia integralności danych przesyłanych w rekordzie. Jest on wynikiem działania funkcji mieszającej według określonego w sesji algorytmu mieszania, na przykład MD5lub SHA-1. Schemat wywołania funkcji mieszającej:

MAC = Funkcja mieszająca (hasło, dane podstawowe, dane uzupełniające, numer sekwencji).

Jako hasło w procesie tworzenia MAC stosowane jest odpowiednio hasło zapisu MAC klienta lub serwera, w zależności od strony przygotowującej pakiet. Po odebraniu pakietu, strona odbierająca dokonuje własnego wyliczenia wartości znacznika MAC i porównuje go z przesłaną wartością. Długość uzyskanego znacznika zależy od metody jego uzyskania.

Dane połączone ze znacznikiem MAC poddawane są następnie szyfrowaniu przy użyciu ustalonego algorytmu szyfrowania symetrycznego, na przykład DES lub 3DES. Szyfrowaniu poddane są zarówno dane, jak i znacznik MAC. Do tak przygotowanych danych dodawane są pola nagłówka, a pomiędzy nimi:

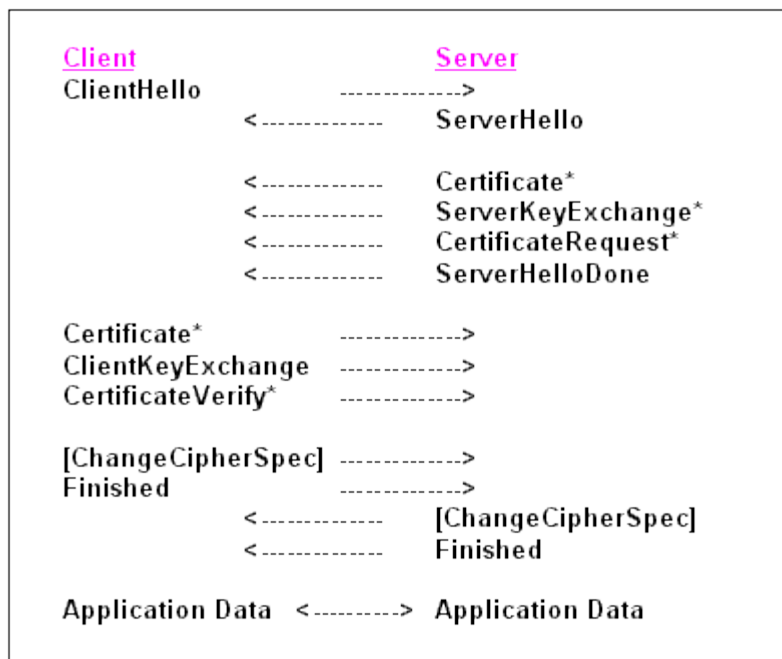
- **Content type:** określa, jaką zawartość niesie pakiet, wskazującą, który z wyższych protokołów ma zostać użyty do przetwarzania zawartych w pakiecie danych. Możliwe wartości to *change - cipher - spec*, *alert*, *hand- shake*, i *application - data*, wskazujące odpowiednie protokoły.
- **Major version:** określa główną część stosowanej wersji protokołu. Dla SSL 3.0 pole to ma wartość 3.
- **Minor version:** określa dodatkową część stosowanej wersji protokołu. Dla SSL 3.0 pole to ma wartość 0.

Rekord jest następnie przesyłany do punktu docelowego. Protokół rekordu SSL jest używany do przesyłania wszelkich danych w ramach sesji, zarówno komunikatów pozostałych protokołów SSL (na przykład *handshake*), jak i do wszelkich przesyłanych danych aplikacji.

Protokół alarmu (SSL Alert Protocol) służy do przesyłania komunikatów związanych z wymianą danych i działaniem protokołu. Każdy komunikat wysyłany przez protokół alarmu składa się z dwóch bajtów. Pierwszy wskazuje stopień ważności przesyłanego komunikatu. Drugi zawiera jeden ze zdefiniowanych kodów błędu, które mogą wystąpić w trakcie komunikacji SSL.

Protokół SSL ChangeCipher Spec składa się tylko z jednego komunikatu, w którym przekazywana jest jedynie wartość 1. Celem przesłania tego komunikatu jest spowodowanie, aby ustanawiany właśnie stan sesji został uznany za ustalony. Komunikat taki musi zostać przesłany przez klienta do serwera oraz przez serwer do klienta. Po wymianie tych komunikatów stan sesji uznaje się za ustalony.

Protokół wymiany (SSL Handshake Protocol) jest najbardziej skomplikowaną częścią protokołu SSL. Służy on do nawiązania sesji pomiędzy serwerem a klientem. Umożliwia ustalenie takich elementów, jak na przykład algorytmy i klucze używane do szyfrowania danych. Umożliwia też uwierzytelnienie stron połączenia i wynegocjowanie odpowiednich parametrów sesji. Schemat procedury realizowanej przez ten protokół przedstawiono na rys. 6.



Rys. 6. Schemat procedury realizowanej przez protokół wymiany

Proces przebiegu negocjacji może zostać podzielony na 4 fazy. W pierwszej fazie pomiędzy klientem a serwerem musi zostać nawiązane logiczne połączenie i rozpoczęta zostaje negocjacja parametrów tegoż połączenia. Klient wysyła do serwera komunikat **client - hello**, zawierający takie dane, jak:

- Wersja: numer najwyższej wersji protokołu SSL obsługiwanej przez klienta.
- Dane losowe: dane składające się z 32-bitowego znacznika czasu i 28 bajtów losowo wygenerowanych danych. Dane te stosowane będą do zabezpieczenia wymiany kluczy pomiędzy stronami połączenia.
- Identyfikator sesji (Session ID): jeżeli pole to posiada wartość różną od zera, oznacza to, że klient chce uaktualnić dane istniejącego połączenia lub uzyskać nowe połączenie w ramach istniejącej sesji. Wartość zero w tym polu oznacza chęć nawiązania nowego połączenia.
- CipherSuite: zestaw algorytmów szyfrowania i metod wymiany kluczy obsługiwanych przez klienta.

Serwer w odpowiedzi na komunikat **client - hello** przesyła komunikat **server - hello**, zawierający identyczny zestaw pól, jak komunikat przesłany przez klienta, wypełniając je następującymi danymi:

- Wersja: najniższy numer wersji protokołu SSL wspomagana przez serwer,
- Dane losowe: analogiczne do danych przesłanych przez klienta, jednak wygenerowane całkowicie niezależnie,
- Identyfikator sesji: w przypadku, gdy klient przesłał wartość różną od zera, zwracana jest ta sama wartość, w innym przypadku zwracany jest wygenerowany przez serwer identyfikator sesji,
- CipherSuite: w polu tym serwer przesyła pojedynczy zestaw protokołów wybranych z możliwych zestawów przesłanych przez klienta. Zestaw ten określa trzy elementy:
 - metodę wymiany kluczy pomiędzy serwerem i klientem,
 - algorytm szyfrowania na potrzeby szyfrowania danych,
 - funkcję wykorzystywaną do uzyskania wartości MAC.

Następna faza negocjacji rozpoczyna się od wysłania przez serwer swojego **certyfikatu** w celu uwierzytelnienia u klienta. Przesyłana do klienta wiadomość zawiera jeden lub więcej certyfikatów X509 koniecznych do uwierzytelnienia serwera i ścieżki certyfikacji prowadzącej do zaufanego urzędu certyfikacji, który wystawił certyfikat serwera. Krok ten nie jest obowiązkowy i może być pominięty, jeżeli wynegocjowana metoda wymiany kluczy nie wymaga przesłania certyfikatu (w przypadku anonimowej metody *Diffie-Hellmana*). W zależności od wynegocjowanej metody wymiany kluczy, serwer może wysłać dodatkowy komunikat **server_key_exchange**, który nie jest jednak wymagany

w przypadku, gdy wynegocjowana została metoda *Diffie- Hellmana* lub *RSA key exchange*. Dodatkowo serwer może zażądać od klienta przesłania jego certyfikatu. Ostatnim krokiem w drugiej fazie negocjacji jest przesłanie przez serwer komunikatu **server_done**, który nie niesie ze sobą żadnych parametrów, lecz jest jedynie sygnałem dla klienta, że serwer zakończył przysyłanie komunikatów i oczekuje na odpowiedź.

Po otrzymaniu tego komunikatu klient powinien zweryfikować certyfikat serwera, jego ważność i ścieżkę certyfikacji, jak również wszelkie inne parametry przesłane przez serwer w komunikacie *server_hello*. Potwierdzenie autentyczności i poprawności certyfikatu serwera po stronie klienta polega na:

- Sprawdzeniu daty ważności certyfikatu.
- Sprawdzeniu, czy urząd certyfikacyjny, który wystawił certyfikat serwera, znajduje się na liście zaufanych urzędów certyfikacji po stronie klienta. Jeżeli CA, który wystawił certyfikat serwera nie znajduje się na tej liście, klient rozpoczyna procedurę weryfikacji CA, który jest wystawcą tego certyfikatu. Jeżeli informacje o tym urzędzie nie mogą być uzyskane, klient przerwie procedurę identyfikacji, zwracając błąd lub zwróci się o rozstrzygnięcie tego problemu do użytkownika.
- Potwierdzeniu poprawności klucza publicznego urzędu certyfikacyjnego, który wydał certyfikat: jeżeli urząd certyfikacyjny znajduje się na liście zaufanych CA po stronie klienta. (porównywany jest klucz publiczny CA. zawarty w certyfikacie serwera z kluczem publicznym udostępnionym na tej liście. Pozwala to potwierdzić autentyczność urzędu wystawiającego] certyfikat.
- Sprawdzeniu, czy nazwa domeny wymieniona w certyfikacie odpowiada : nazwie serwera. który przedstawił się tym certyfikatem.

Po prawidłowym przejściu wszystkich tych kroków, serwer uznawany jest za uwiarygodniony. Wówczas klient odpowiada serwerowi jednym lub wieloma komunikatami. Jedynym koniecznym komunikatem jest przesłanie do serwera kluczy klienta w komunikacie **client_key_exchange**, którego zawartość zależy od wynegocjowanej metody wymiany kluczy. Dodatkowo, jeżeli serwer tego zażądał, przesyłany jest certyfikat klienta i komunikat umożliwiający weryfikację tego certyfikatu. Wysłanie tych komunikatów kończy trzecią fazę negocjacji.

Czwarta faza stanowi potwierdzenie wysłanych wcześniej wiadomości i weryfikację poprawności wynegocjowanych danych. Rozpoczyna ją klient wysyłając komunikat **change_cipher_spec** (protokół *SSL ChangeCipher Spec*), po czym ustanawia wynegocjowany zestaw parametrów algorytmów i kluczy jako obowiązujący. Następnie wysyła dodatkowy komunikat **finished**, zaszyfrowany z wykorzystaniem wynegocjowanych algorytmów i uzyskanych kluczy. Komunikat ten stanowi potwierdzenie poprawności uzyskanych w trakcie negocjacji parametrów i danych. Serwer odpowiada klientowi identyczną sekwencją komunikatów. Jeżeli komunikat **finished** został prawidłowo odczytany przez każdą ze stron, potwierdza to prawidłowość przesłanych danych i wynegocjowanych algorytmów i klucza sesji, co oznacza, że negocjacja została zakończona i możliwe jest rozpoczęcie przesyłania danych aplikacji pomiędzy serwerem a klientem z wykorzystaniem SSL. Po zakończeniu negocjacji połączenie TCP pomiędzy klientem a serwerem zostaje zakończone, jednakże po obu stronach zachowany zostaje stan sesji pozwalający na nawiązanie dalszych połączeń w ramach sesji, z wykorzystaniem wynegocjowanych parametrów.

Na podstawie wersji 3 SSL został opracowany protokół TLS (*Transport Layer Security*). Jest on nieco ulepszony i firmowany przez IETF. W 1996 roku w ramach IETF rozpoczęła prace grupa, która ma za zadanie opracowanie i wypromowanie jako standardu internetowego, protokołu zabezpieczenia transmisji w sieci. Grupa ta jako punkt wyjścia do dalszych prac przyjęła protokół SSL w wersji 3.0, a jako wynik prac, w 1999 roku opublikowała ona dokument RFC 2246, definiujący nowy protokół Transport Security Layer (TLS) w wersji 1.0.

Podstawowe zadania były podobne jak założenia protokołu SSL, a więc zapewnienie prywatności i integralności danych wymienianych pomiędzy dwoma aplikacjami w sieci. Dodatkowo tworzący go zespół przyjął dla niego wytyczne uzupełniające:

- **Interoperacyjność** (ang. *interoperability*): umożliwienie budowy aplikacji przez jedną ze stron, bez konieczności znajomości szczegółów implementacji TLS po drugiej stronie połączenia.
- **Łatwość rozszerzenia**: TLS ma stanowić w założeniu ramy, w które w przyszłości w razie potrzeby łatwo będzie wbudować nowe technologie kryptograficzne. Ma to w założeniu zapobiec konieczności definiowania nowych protokołów w miarę zmian zachodzących w metodach kryptograficznych.

Protokół TLS również składa się z dwóch podstawowych warstw protokołów:

- **TLS Record Protocol**: spełniający rolę identyczną jak protokół rekordu SSL, czyli zapewnienie bezpieczeństwa i integralności przesyłanych danych.
- **TLS Handshake Protocol**: służący do negocjacji warunków połączenia.

Specyfikacja zawarta w RFC 2246 nie określa jednak sposobu w jaki protokoły warstw wyższych mają wykorzystywać TLS do zabezpieczenia transmisji. Zespół pracujący nad TLS opracował dwa dodatkowe dokumenty RFC:

- RFC 2817 "*Upgrading to TLS Within HTTP/1.1*",
- RFC 2818 "*HTTP Over TLS*".

Przedstawiają one sposób implementacji TLS dla protokołu HTTP, zastępując nim wykorzystywany dotychczas protokół SSL. Dokumenty te przedstawiają między innymi sposób wykorzystania połączenia HTTP zabezpieczonego TLS, bez potrzeby używania dodatkowego portu dla połączenia szyfrowanego, tak jak to ma miejsce w przypadku wykorzystania protokołu SSL. Połączenie szyfrowane poprzez TLS nawiązywane jest z wykorzystaniem standardowego portu HTTP.

Kolejnym przykładem specyfikacji wykorzystania bezpiecznego połączenia z wykorzystaniem TLS jest dokument RFC 2487 "*SMTP Service Extension for Secure SMTP over TLS*", definiujący sposób nawiązania bezpiecznego połączenia dla protokołu SMTP z wykorzystaniem standardowego portu i rozszerzeń protokołu.

Protokół TLS jest już w chwili obecnej wspomagany przez część oprogramowania. Jest on wspomagany między innymi przez przeglądarkę *Internet Explorer*. Pomimo tego, że protokół SSL jest nadal najbardziej rozpowszechnioną metodą zabezpieczenia transmisji wykorzystywaną w Internecie, należy spodziewać się, że w przyszłości zostanie on zastąpiony właśnie przez TLS uznany za standard zabezpieczenia transmisji dla usług internetowych.

1. Co to jest firewall?

Zapora sieciowa jest umieszczana między siecią wewnętrzną organizacji i siecią zewnętrzną. Dostarcza ona prostego mechanizmu kontroli ilości i rodzaju ruchu sieciowego między obydwojema sieciami. Podstawowym jej zadaniem jest ograniczenie przepływu danych między tymi sieciami. Przed postawieniem zapory trzeba określić, jakie rodzaje danych mają być przez nią przepuszczane, a jakie nie. Czyli trzeba zdefiniować **politykę zapory**. Następnie należy skonstruować **mechanizmy**, które umożliwią wprowadzenie tej polityki w życie.

Podstawowe strategie definiowania zapory:

Domyślne przepuszczanie polega na określeniu zbioru warunków, których spełnienie będzie powodowało blokowanie danych. Każdy *host* lub protokół nie objęty tą polityką będzie przepuszczany.

Domyślne blokowanie polega na określeniu protokołów, które będą mogły przechodzić przez zaporę oraz *hostów*, które będą mogły przesyłać przez nią dane i z którymi komputery w sieci wewnętrznej będą się mogły komunikować. Wszystkie elementy nie objęte definicjami będą blokowane.

Do podstawowych działań realizowanych przez zaporę należy zaliczyć:

- Blokowanie dostępu do całej sieci z określonych miejsc w sieci zewnętrznej.
- Blokowanie dostępu do wybranych serwerów i usług określonym użytkownikom.
- Monitorowanie komunikacji między sieciami. Przykładowo rejestracja końcowych punktów połączeń i ilość przesyłanych danych.
- Podsluchiwanie i rejestrowanie komunikacji między sieciami w celu badania przypadków penetracji sieci, wykrywania intruzów i wewnętrznych sabotażystów.
- Tunelowanie. Automatyczne szyfrowanie i deszyfrowanie pakietów. Sieć zewnętrzna staje się własnym połączeniem typu WAN (prywatna sieć wirtualna - *Virtual Private Network*).
- Uwierzytelnianie. Użytkownicy sieci zewnętrznej muszą uwierzytelnić się wobec zapory.

Podstawowe mechanizmy spotykane w implementacjach zapór to:

- Filtrowanie pakietów (*packet filtering*)- odrzucanie pakietów nie spełniających reguł zapory.
- Translacja adresów (*network address translations*) - dokonywanie zamiany adresu hosta wewnętrznego w celu ukrycia go przed zewnętrznym monitorowaniem.
- Usługi proxy - dokonywanie połączenia na poziomie aplikacji w imieniu hosta wewnętrznego. Przerzywa połączenie na poziomie warstwy sieciowej.

2. Filtrowanie pakietów

Filtrowanie pakietów oparte jest na pomysłach aby badać nagłówki pakietów i odrzucać te pakiety, które nie odpowiadają określonej specyfikacji. Istnieją dwa podstawowe typy filtrów pakietów:

- filtry bezstanowe (*stateless*),
- filtry z badaniem stanów (*statefull inspection*).

Filtry bezstanowe

Teoretycznie w takich filtrach decyzja o akceptacji lub odrzuceniu pakietu mogłaby brać pod uwagę każdy element składowy nagłówka określonego protokołu. Najczęściej jednak filtrowanie oparte jest na takich polach jak:

- adres IP,
- typ protokołu (UDP, TCP, ICMP) - ta informacja jest jednak na tyle ogólna, że zwykle dopuszcza się wszystkie protokoły,
- port TCP/UDP,
- informacja o wyborze trasy,
- znacznik fragmentu.

Filtrowanie adresów IP może mieć sens, jeżeli dopuścimy tylko połączenia z zaufanych hostów. Może to być lista komputerów własnych, komputerów klientów i pracowników zdalnych. Zwykle nie jest możliwe powiązanie listy adresów IP z listą portów (protokołów) - a byłoby to bardzo dobre rozwiązanie. Taki filtr ogranicza ruch na podstawie pola adresowego. Jednak podany w nagłówku adres nie musi być prawdziwy (mógł zostać sfalszowany). Taka sytuacja może wystąpić podczas ataku, gdy nie jest potrzebna informacja zwrotna, np. atak typu DoS. Informacja zwrotna nie jest potrzebna również wtedy, gdy adres zwrotny jest zawarty dodatkowo w polu danych (np. w FTP).

Filtrowanie portów nazywane jest również filtrowaniem protokołów wyższych warstw. Do najważniejszych protokołów, które należy zablokować należą:

- Telnet,
- NetBIOS Session (usługi Windows i SMB) - możliwe jest podłączenie się do serwera plików w charakterze lokalnego klienta,
- POP - jawne hasła dostępne do poczty,
- NFS - podobnie jak w przypadku NetBIOS możliwe jest podłączenie się do serwera plików w charakterze lokalnego klienta,
- X Windows.

Inne porty (np. DNS mogą być wykorzystane do uszkodzenia pewnej informacji, ale nie mogą służyć do przejęcia pełnej kontroli nad komputerem). Blokować się również powinno porty obsługujące każde oprogramowanie pozwalające na zdalny dostęp lub zdalne nadzorowanie sieci (np. PC Anywhere). Występują problemy z określeniem ruchu do zwrotnych portów połączeń nawiązanych z wnętrza sieci.

Routing źródłowy (*source routing*) - opcja ta umożliwia określenie przez nadawcę trasy pakietu, czyli wykazu hostów, które powinny kolejno uczestniczyć w przekazywaniu pakietu. Pierwotnie była wykorzystywana do testowania i usuwania usterek. Są dwa sposoby wyboru trasy przez nadawcę:

- swobodny (*loose source routing*) - pakiet musi przejść przynajmniej przez jeden z hostów z listy,
- rygorystyczny (*strict source routing*) - pakiet musi przejść dokładnie zadaną trasą.

W tej chwili nie istnieje żaden protokół wymagający ustawienia tej opcji. Czyli można ją zablokować.

Fragmentacja - umożliwia przesyłanie dużych pakietów IP (przekraczających dozwolone wymiary ramek). Pakiet jest dzielony i przesyłany po kawałku. System odbierający składa je w całość. Najbardziej użyteczne do filtrowania dane (numery portów TCP/UDP) są na początku - we fragmencie zerowym. Z tego powodu dalsze fragmenty nie mogą być filtrowane w oparciu o numery portów i przechodzą przez filtry mimo, że fragment zerowy został odrzucony. Niektóre błędne implementacje TCP/IP składają te fragmenty zamiast je odrzucić. Oznacza to, że jeżeli wysyłane fragmenty będą numerowane od 1, a nie od 0, to filtrowanie takie zostanie oszukane.

Większość współczesnych systemów operacyjnych udostępnia funkcje filtrowania. Technika taka nazywana jest filtrowaniem pakietów w systemie końcowym, ponieważ ostatni komputer na trasie pakietu wykonuje filtrowanie. Włączając pomocnicze filtrowanie bezpośrednio na serwerach, zapewniamy dodatkową ochronę, przydatną w przypadku, gdy awarii ulegnie graniczny punkt kontroli lub atak przyjdzie z wnętrza sieci.

Problemy filtrowania bezstanowego:

Brak możliwości dokładnego sprawdzania ładunków danych. Decyzja jest podejmowana jedynie na podstawie zawartości nagłówka. Wobec tego nie ma możliwości wykrycia np. koni trojańskich umieszczonych w kontrolkach *ActiveX*, niewłaściwego sformatowania danych dla serwera pocztowego.

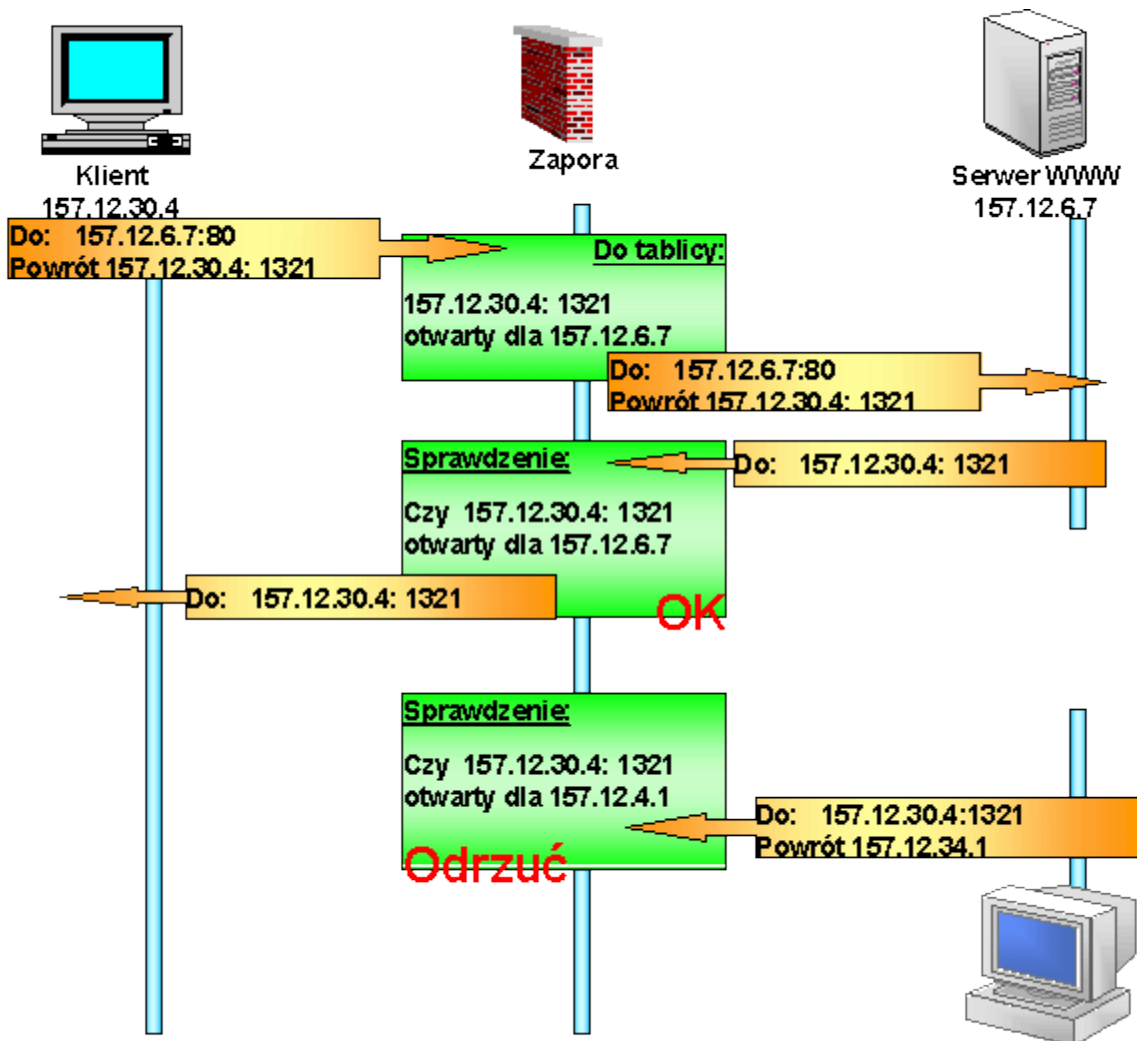
Brak pamiętania stanu połączenia. Istnieje wobec tego problem określenia ruchu zwrotnego do portów, z których nawiązywano połączenie. Faktycznie nie istnieje możliwość filtrowania ruchu do portów wysokich (powyżej 1024).

Filtry z badaniem stanu

Przechowują informację o stanie całego ruchu przechodzącego przez filtr. Wykorzystują ją do określania czy pojedynczy pakiet powinien być odrzucony. Filtry takie działają na poziomie warstwy sieciowej oraz sesji. Informacja jest pobierana z pakietów przepływających podczas nawiązywania sesji. Gdy host wewnętrzny łączy się z hostem zewnętrznym, w pakiecie inicjującym umieszczony jest adres gniazda zwrotnego (adres IP i numer portu) na którym oczekuje na odpowiedź. Informacje te są zapamiętywane przez filtr. Umożliwiają potem odróżnienie poprawnych pakietów zwrotnych od niepoprawnych prób połączeń lub włamań. Kiedy przychodzi odpowiedź, sprawdzane są zapisy w tablicy filtra w celu sprawdzenia, czy pakiet ma zostać przepuszczony. Jeżeli z zewnątrz przychodzi pakiet, który nie ma pozycji w tablicy stanów, to jest odrzucany. Zapisy w tablicy stanów są usuwane gdy przesyłane są pakiety związane z zamknięciem sesji lub po upływie określonego czasu. Wyjątki określają pakiety:

- które zawsze będą odrzucane,
- których nigdy nie należy odrzucać,
- usługi z zewnątrz do określonych hostów.

Filtry z badaniem stanu nadal nie rozwiązują problemu wewnętrznej analizy protokołów wyższych warstw. Ideę funkcjonowania tego typu filtra przedstawia rys. 1.



Rys. 1 Idea funkcjonowania filtra z badaniem stanu

3. Translacja adresów

Translacja adresów (Network Address Translation - NAT) umożliwia przydzielenie komputerom z sieci wewnętrznej adresów z puli adresów nie rejestrowanych w sieci Internet (pula adresów prywatnych) oraz zapewnienie tym komputerom możliwości dwustronnego komunikowania się z komputerami w sieci Internet. Początkowo miało to być sposobem na zwiększenie ilości hostów podłączonych do Internetu. Okazało się, że jest również mechanizmem bezpieczeństwa, gdyż umożliwia ukrywanie wewnętrznych hostów. Sprawia wrażenie, że ruch pochodzi z pojedynczego adresu IP. Jest to adres zapory. NAT jest implementowany tylko w warstwie transportowej. Czyli aby zapobiec naruszeniom bezpieczeństwa w warstwach wyższych, trzeba używać np. *proxy*.

Stacja kliencka powinna traktować bramę NAT jako swój *gateway*. Jeżeli tak nie jest, to brama NAT powinna funkcjonować jako *serwer proxy arp*. Pakiet pochodzący od klienta otrzymuje nowy numer portu źródłowego i adres źródłowy. W takiej postaci jest wysyłany. Brama zapamiętuje zrealizowane przekształcenie. Gdy przychodzi odpowiedź, to musi zostać rozpoznana i pakiet przekształcony. Przywracany jest adres klienta i pakiet trafia do klienta.

Zakresy adresów prywatnych:

10.0.0.0 , 10.255.255.255 - pojedynczy numer sieci klasy A;
 172.16.0.0 , 172.31.255.255 - 16 ciągłych numer sieci klasy B;
 192.168.0.0 , 192.168.255.255 - 255 ciągłych numer sieci klasy C.

Tryby translacji

Translacja statyczna (static translation) inaczej przekierowanie portów (*port forwarding*) - określony zasób ma przypisane stałe przekształcenie. Stosujemy gdy udostępniamy hosty wewnętrzne dla połączeń z hostów zewnętrznych. Np. serwer pocztowy.

Translacja dynamiczna (dynamic translation) inaczej automatyczna (*automatic*), tryb ukrywania (*hide mode*), maskowanie IP (*IP masquerade*) - stosujemy gdy duża grupa klientów wewnętrznych współużytkuje adres lub grupę adresów wewnętrznych. Adresy tych klientów zastępowane

są adresem zapory. Klienci są identyfikowani numerem portu połączenia przechodzącego przez zaporę. Host zewnętrzny nie ma możliwości zainicjowania połączenia z wewnętrznym. Niektóre protokoły nie pracują poprawnie po zmianie portu.

Translacja ze zrównoważonym obciążeniem (*load balancing translation*) - pojedynczy adres i numer portu są przekształcane na jeden z adresów identycznie skonfigurowanych serwerów. W efekcie jeden adres jest obsługiwany przez kilka serwerów. Stosowany jest wówczas zwykle algorytm cykliczny (*round robin*) lub zrównoważonego obciążenia (*balanced load*). Z puli dostępnych serwerów zapora za każdym razem wybiera jeden. Jeżeli miarą jest obciążenie, to serwery powinny informować zaporę o swoim obciążeniu. Nie ma w tym zakresie standardów. Jest to dobre rozwiązanie dla bezstanowych serwerów WWW, ale nie dla poczty. Zrównoważone obciążenie jest szczególnie ważne dla serwerów obsługujących handel elektroniczny.

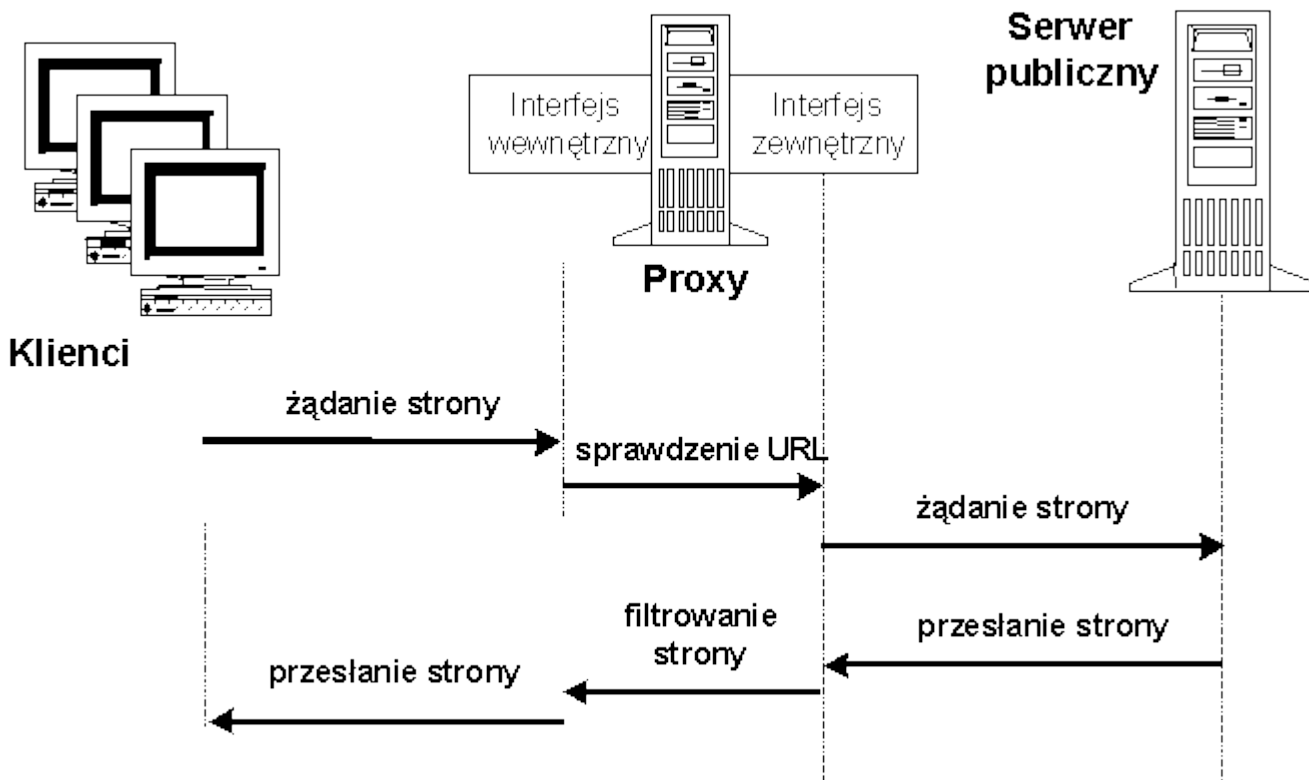
Translacja ze zwielokrotnionymi połączeniami (*network redundancy translation*) - zwielokrotnione połączenia z Internetem są przyłączone do pojedynczej zapory NAT i wykorzystywane w oparciu o obciążenie i dostępność. Za każdym razem gdy host wewnętrzny łączy się z zewnętrznym, podejmowana jest decyzja, którą drogą skierować jego pakiety.

4. Usługi proxy

Pierwotnym przeznaczeniem serwerów *proxy* było przechowywanie w pamięci podręcznej często przeglądanych stron WWW. Znacznie przyspieszało to dostęp do informacji. W chwili obecnej pełnią one raczej funkcje ściany ogniowej.

Serwery *proxy* pośredniczą w przekazywaniu żądań klientów sieci wewnętrznej do sieci zewnętrznej. Pozwala to ukrywać klientów przed badaniem z zewnątrz. Wiele aktualnie dostępnych pakietów tego typu realizuje również usługi filtrowania pakietów i NAT. Połączenie tych wszystkich technologii pozwala wyeliminować pewne ataki, którym prawdopodobnie "czyste" *proxynie* sprostałoby. Serwery *proxy* najczęściej związane są z WWW (ze względów historycznych) ale podobnie działają dla innych usług.

Proxy nasłuchuje zleceń usługi od klientów wewnętrznych i przesyła je w ich imieniu do sieci zewnętrznej. Po otrzymaniu odpowiedzi z zewnątrz zwraca ją do rzeczywistego klienta. Ideę funkcjonowania serwera *proxy* przedstawiono na rys. 2.



Rys. 2. Idea funkcjonowania serwera *proxy*

Zalety proxy

Ukrywanie klienta jest możliwe ponieważ *proxy* generują żądania warstwy usługowej w imieniu swoich klientów. Nie jest to jedynie zmiana nagłówek. Klienci zwracają się do *proxy* jak do serwera docelowego. Przez *proxy* przesyłane są tylko komunikaty warstwy usługowej (np. HTTP) a nie pakiety TCP czy IP. Pakiety protokołów warstw niższych są ponownie generowane przez *proxy*. Jest to więc zupełnie inny mechanizm niż NAT, chociaż skutek jest podobny.

Blokowanie URL jest realizowane poprzez porównywanie URL z listą adresów zakazanych. Można to jednak ominąć poprzez użycie adresów liczbowych a nie nazw, gdyż zwykle sprawdzeniu podlega pełny tekst URL. Wymagane korzystanie z WWW łatwiej jest wymusić poprzez informowanie pracowników o śledzeniu ich dostępu a nie poprzez blokowanie. Zakazane strony często migrują.

Filtrowanie zawartości może polegać na usuwaniu określonych elementów z ładunku danych. Mogą to być kontrolki ActiveX, aplety Javy, duże pliki graficzne, wykonywalne pliki binarne przesyłane jako załączniki w poczcie.

Badanie spójności to sprawdzanie zgodności z protokołem. Eliminuje się w ten sposób, lub przynajmniej ogranicza użycie nieprawidłowo sformatowanych danych do wykorzystywania luk w systemie. Np. we wczesnych wersjach demona *sendmail* występował problem przepełnienia bufora. Przydzielał on bufor o wielkości określonej w komunikacie. Jeżeli danych jest więcej (przed znacznikiem końca pliku) to zwykle umieszczany tam był kod wykonywalny umożliwiający dostęp super-użytkownika do serwera pocztowego. Podobny błąd wykryto w IIS4. Problem przekroczenia bufora występował również we wczesnych przeglądarkach gdy wpisywano adresy URL dłuższe niż 256 znaków. O takich lukach dowiadujemy się jednak dopiero po ich wykorzystaniu.

Blokowanie routingu jest jedną z najważniejszych zalet. Żaden pakiet TCP/IP nie jest przesyłany pomiędzy sieciami, więc eliminuje się wiele ataków typu DoS oraz wykorzystujących luki TCP/IP. Niestety nie dla wszystkich usług dostępne są dobre *proxy* i nie można wobec tego całkowicie wyeliminować routingu. Można wykorzystać *proxy* TCP, ale nie będą one wówczas filtrowały zawartości.

Proxy mogą podnosić wydajność przetwarzania gdyż:

- mogą przechowywać często żądane dane - eliminacja nadmiarowego ruchu pomiędzy sieciami,
- mogą równomiernie rozkładać obciążenie usług pomiędzy pewną liczbę serwerów wewnętrznych.

Pierwsze zastosowanie nabierze ponownie znaczenia gdy rozwine się handel elektroniczny. Drugie zastosowanie to tzw. *proxy* odwrotne (*reverse proxy*) obsługujące klientów zewnętrznych. Jeden serwer nie jest w stanie obsłużyć dużej liczby klientów (np. w handlu elektronicznym).

Wady proxy

Pojedynczy punkt kontroli - pojedynczy punkt awarii powoduje duże zagrożenie w przypadku jego awarii. Dlatego powinien on współpracować z filtrami. Sam *proxy* powinien być dodatkowo chroniony przez filtry lub ścianę ogniową.

Klienci muszą współpracować z proxy. Klienci niewłaściwie skonfigurowani nie będą korzystać z *proxy*. Jeżeli *proxy* jest elementem zapory sieciowej z translacją adresów, to ten problem nie istnieje. Ponadto niektóre pakiety klienckie nie pozwalają na współpracę z *proxy*. Zwłaszcza dotyczy to standardowych pakietów dostarczanych z systemem operacyjnym (np. FTP). Można jednak stosować tzw. *proxy* transparentne.

Oddzielne proxy dla każdej usługi. Dla niektórych usług trudno jest zbudować *proxy*, ponieważ wymagają one kanału zwrotnego. Dla niektórych usług nie ma skutecznych filtrów zawartości. Np. usługi strumieniowe typu RealVideo lub RealAudio wymagają przepływu skompresowanego strumienia w czasie rzeczywistym. Jego przerwanie uniemożliwia zwykle dalsze dekodowanie. Niemożność filtrowania powinna implikować konieczność blokowania.

Ochrona systemu operacyjnego ma decydujące znaczenie dla funkcjonowania *proxy*. Włamanie na serwer, na którym zainstalowano *proxy* może spowodować unieszkodliwienie tego mechanizmu. Zaleca się stosowanie jak najmocniejszych zabezpieczeń opartych na prawach dostępu oraz filtrowanie portów i protokołów na poziomie systemu operacyjnego. Nie należy gromadzić usług publicznych na tym samym serwerze co *proxy*. Jeśli usługa FTP lub SMTP pozwoli na dostęp do serwera *proxy*, to możliwe stanie się usunięcie zabezpieczeń *proxy* i dostęp do reszty sieci.

Zatory to wada powodująca spadek wydajności. Należy dodać więcej serwerów *proxy*.

5. Etapy budowy zapory

Planowanie konfiguracji zapory sieciowej

Jest to bardzo ważny etap, ponieważ błędy popełnione przy planowaniu konfiguracji wpłyną na wszystkie dalsze etapy i w rezultacie końcowy efekt działania zapory może być całkowicie odmienny od oczekiwanego. W pierwszej kolejności należy odpowiedzieć na pytanie: *co chronić?* Jeżeli ochronie mają podlegać dwa lub trzy komputery to prawdopodobnie zamiast budować zaporę sieciową wystarczy zastosować zabezpieczenia na poziomie pojedynczych hostów. Zapora sieciowa należy do mechanizmów *cięższego kalibru*.

Kolejny etap to rozpoznanie topologii sieci oraz potrzeb w zakresie aplikacji i protokołów. Polegać on będzie na analizie topologii sieci pod kątem bezpieczeństwa, na zidentyfikowaniu systemów operacyjnych i aplikacji działających w sieci, czego efektem może być np.: konieczność skorzystania z usług ekspertów w dziedzinie bezpieczeństwa poszczególnych aplikacji.

Należy również dokonać analizy zależności służbowych. Polegać ona będzie na analizie kompetencji decyzyjnych i potrzeb dostępu do zasobów poszczególnych użytkowników czy grup użytkowników. Konieczne jest przy tym uświadomienie użytkownikom wszystkich potrzebnych zmian w konfiguracji sieci oraz wszelkich ich wrażliwości. Od użytkowników w dużym stopniu będzie zależało bezpieczeństwo sieci i ostatnią rzeczą jaką jest nam potrzebna to niezadowoleni użytkownicy.

Kolejna decyzja dotyczy konfiguracji zapory. Przede wszystkim należy określić czy wystarczy filtrowanie pakietów, czy też należy zastosować serwery proxy, a jeżeli tak to jakie.

Wreszcie powinniśmy rozpatrzyć czy skonstruować własną czy też kupić pakiet gotowej zapory. Samodzielnie można wykonać całkiem dobrą zaporę, lecz jeden błąd przy jej konfiguracji może spowodować katastrofę. Z drugiej strony najlepsza, źle skonfigurowana kupiona zaporą również może być przyczyną poważnych kłopotów.

Rozwiązanie alternatywne polega na wykupieniu usługi monitorowania zapory. Jeżeli nie mamy możliwości monitorowania zapory sieciowej 24 godziny na dobę przez 7 dni w tygodniu, to może lepiej taką usługę wykupić?

Zdefiniowanie reguł dostępu do zasobów sieciowych

W oparciu o poczynione obserwacje i wykonane analizy opracowujemy zasady korzystania z zasobów sieci. W tym etapie określamy: kto i w jaki sposób ma dostęp do sieci i jej zasobów. Reguły musimy odpowiednio dostosować do posiadanej infrastruktury. Oznacza to uwzględnienie stosowanych platform sprzętowych, czy protokołów sieciowych.

Znalezienie zapory odpowiedniej dla naszych potrzeb

W oparciu o zdobyte informacje, wykonane analizy i ustalone reguły dostępu do zasobów możemy właściwie wybrać potrzebną nam zaporę sieciową.

Właściwa instalacja i konfiguracja zapory

Drobiazgowe przetestowanie zapory

Testowanie zapory powinno odbyć się w dwóch etapach. W pierwszym należy przeprowadzić testowanie zasad korzystania z sieci prywatnej przez użytkowników zewnętrznych. W drugim testujemy wewnętrzne reguły korzystania z sieci. Oba etapy należy wykonać dokładnie, ponieważ jest to ostatnia czynność przed włączeniem zapory do sieci.

Mimo, że zaporą sieciową to bardzo skuteczny środek ochrony sieci prywatnej, należy być świadomym jej wad. Jedną z nich jest fakt, że zaporą, której konfigurację zorientowano na maksymalne bezpieczeństwo sieci, będzie jednocześnie upośledzać jej działanie. Inną wadą jest zgromadzenie w jednym miejscu wszystkich składników zapory, ponieważ ich pokonanie daje intruzowi pełny dostęp do sieci prywatnej.

Istnieje kilka zagrożeń dla bezpieczeństwa sieci, które nie są eliminowane przez zastosowanie zapory:

- naruszenie bezpieczeństwa od wewnątrz, ponieważ zaporą sieciową nie chroni zasobów przed atakiem od strony użytkowników wewnętrznych,
- bezpośrednie połączenie z Internetem - jeśli użytkownik wewnętrzny połączy się z Internetem z pominięciem zapory np.: poprzez łącze telefoniczne, to stanowi to poważną lukę w bezpieczeństwie,
- niektóre skanery (np.: *stealth skaner*) potrafią skanować aktywne porty komputerów nawet za zaporą.

2. Klasyfikacja IDS według źródeł informacji

Obecnie najczęściej stosowaną metodą klasyfikacji systemów IDS jest podział według tak zwanych źródeł informacji. Mogą one być bardzo różne, od porcji danych dostarczanych przez aplikacje lub system operacyjny pracujący na komputerze, aż po tysiące pakietów krążących po sieci. Z tego powodu trzy główne kategorie systemów detekcji intruzów to:

- IDS hostowy (*Host IDS*),
- IDS sieciowy (*Network IDS*),
- IDS węzłowy (*Network Node IDS*).

Istnieje jeszcze kategoria określana mianem IDS Aplikacyjnych (*Application-Based IDS*), która jest jednak podgrupą systemów hostowych.

W rozwiązaniach hostowych oprogramowanie rezyduje na wszystkich monitorowanych *hostach*. Oprogramowanie to analizuje logi zdarzeń, kluczowe pliki systemu i inne możliwe do sprawdzenia zasoby, w poszukiwaniu nieautoryzowanych zmian lub podejrzanego aktywności.

Przykładowo monitorowane mogą być próby logowania do systemu i używanie błędnego hasła. Inną metodą jest monitorowanie plików systemowych i plików aplikacji oraz rejestrów systemu Windows. Można to osiągnąć metodą tak zwanej "fotografii stanów", zapamiętując w świeżo zainstalowanym systemie stany ważnych plików. Jeżeli napastnikowi uda się włamać do systemu i wprowadzić do niego pewne zmiany, zostanie to zauważone (ale zwykle nie w czasie rzeczywistym).

Podstawowe zalety i wady rozwiązań typu HIDS:

- Dzięki swojej obecności bezpośrednio na komputerze i stałemu monitorowaniu lokalnych zasobów mogą wykryć ataki niewidoczne dla sieciowych IDS.
- Niezależne od topologii sieciowej.
- Dzięki integracji z systemem operacyjnym mogą skutecznie działać nawet w oparciu o zaszyfrowane dane.
- Mogą wykrywać różne rodzaje "koni trojańskich" lub pewne rodzaje ataków powodujące naruszenie integralności oprogramowania (kasowanie plików etc.).
- Trudne do zarządzania.
- Mogą zostać wyłączone przy użyciu pewnych typów ataków DoS.
- Wymagające często dużej przestrzeni dyskowej.
- Obciążające (zmniejszające wydajność) systemu produkcyjnego.

HIDS aplikacyjny jest to tak naprawdę podzbiór Hostowych IDS, który analizuje zdarzenia zachodzące w obrębie aplikacji. Najbardziej popularnym źródłem informacji dla tego typu systemu IDS są logi tworzone przez aplikacje.

- Monitoruje interakcję użytkownika z aplikacją, co umożliwia dopasowanie niedozwolonych działań do konkretnej osoby.
- Ma dostęp do zaszyfrowanych danych po odszyfrowaniu przez aplikację.
- Może być bardziej podatny na ataki niż zwykły HIDS.
- Dostosowany zwykle monitorowania zdarzeń na poziomie użytkownika, więc może nie wykryć ataku dokonanego przez konia trojańskiego lub innych ataków szkodliwych dla aplikacji.

Rozwiązania typu NIDS monitorują ruch sieciowy w czasie rzeczywistym, sprawdzając szczegółowo pakiety w celu wykrycia niebezpiecznej zawartości, bądź też rozmaitych typów ataków, zanim osiągną one miejsce przeznaczenia. Mechanizm działania jest bardzo prosty: opiera się na porównywaniu pakietów z sygnaturami ataków (*attack signatures*), przechowywanymi w bazie danych IDS, lub na analizie użytych protokołów, mającej na celu wyszukiwanie w nich wszelkich anomalii. Bazy danych sygnatur cały czas uaktualniane są przez dostawców systemów IDS, w miarę wykrywania coraz to nowych form ataków.

Skuteczność całego systemu zależy w dużym stopniu od użytej metody detekcji - każda ma jakieś zalety i wady. Niezależnie jaką metodę wykrywania ataków wykorzystuje system NIDS, jedną z podstawowych czynności instalacyjnych jest jego dostrojenie i dopasowanie do konkretnych warunków pracy. Polega to zwykle na aktywacji wybranych sygnatur i zablokowaniu innych, "nauczeniu" systemu topologii sieci oraz prawidłowym skonfigurowaniu wielu innych specyficznych parametrów. Nie można pominąć tej fazy, gdyż spowoduje to nieefektywne działanie systemu oraz dużą liczbę fałszywych alarmów.

Rozwiązania oparte na metodzie sieciowej działają w tak zwanym trybie bezładnym (*promiscuous mode*), który polega na przeglądaniu każdego pakietu w kontrolowanym przez nas segmencie sieci, niezależnie od adresu docelowego takiego pakietu. Mając na uwadze stosunkowo duże obciążenie, jakie niesie ze sobą przeglądanie każdego pakietu, rozwiązania te są przeważnie stosowane na dedykowanych hostach. Na każdy segment sieci wymagany jest jeden system NIDS, ponieważ nie mogą one "widzieć" sieci poza routerami i przełącznikami.

Podstawowe zalety i wady rozwiązań typu NIDS:

- Kilka dobrze umiejscowionych sieciowych systemów wykrywania intruzów może monitorować rozległą sieć.
- Rozmieszczenie takich systemów nie wpływa na aktualną topologię sieci. Systemy NIDS są przeważnie pasywne w swoich działaniach i nasłuchując w danym segmencie nie zakłócają jednocześnie pracy sieci.
- Są odporne na ataki, mogą nawet zostać skonfigurowane jako niewidzialne dla potencjalnego włamywacza.
- Problemy z analizą wszystkich pakietów w rozległej i ruchliwej sieci.
- Kłopoty ze znalezieniem optymalnej lokalizacji.
- Brak możliwości analizy zaszyfrowanych danych.
- Brak możliwości określenia rzeczywistej skuteczności ataki.

System typu NNIDS to dość typ agenta IDS, w którym udało się wyeliminować niektóre ograniczenia sieciowych IDS. Agent taki pracuje w sposób bardzo podobny do systemów NIDS - pakiety przechwycone w sieci są porównywane ze znanymi sygnaturami ataków z bazy danych - jednak interesuje się tylko pakietami adresowanymi do hosta, na którym rezyduje (stąd czasem nazywany jest węzłowym: *Stack-Based IDS*).

Systemy takie są także niekiedy określane mianem "hostowe", jednak termin ten dotyczy raczej systemów skupiających się na monitorowaniu logów i analizie zachowań, natomiast sieciowe i węzłowe IDS skupiają się na śledzeniu i analizie ruchu TCP - z tą jednak różnicą, że NIDS pracuje w trybie "beźładnym" (*promiscuous mode*), podczas gdy NNIDS ograniczają się do wybranych pakietów krążących w sieci.

Podstawowe zalety i wady rozwiązań typu NNIDS:

- Nie zajmują się wszystkimi pakietami krążącymi w sieci co powoduje, iż pracują one znacznie szybciej i wydajniej, co z kolei pozwala na instalowanie ich na istniejących serwerach bez obawy ich przeciążenia.
- Niezależne od topologii sieciowej.
- Zaszyfrowane dane nie stanowią dla nich przeszkody - dzięki obecności bezpośrednio na komputerze mogą mieć dostęp do danych po ich deszyfracji.
- Obciążają w pewnym stopniu procesor maszyny, lecz nie dotyczy go problem skalowalności systemów NIDS, które do skutecznego działania potrzebują wydzielonych maszyn o dużej mocy obliczeniowej.
- Konieczne jest instalowanie całego szeregu pakietów - po jednym na każdym chronionym komputerze - a każdy z nich musi przekazywać raporty do centralnej konsoli.
- Mogą zostać wyłączone w przypadku udanego ataku na komputer.
- Zużywają zasoby monitorowanego serwera, co prowadzi zwykle do obniżenia wydajności.

3. Klasyfikacja IDS według metod analizy

W obecnej chwili istnieją dwa główne podejścia do analizy zdarzeń w celu wykrycia ataku: wykrywanie nadużyć oraz wykrywanie anomalii. Podczas **wykrywania nadużyć** stosuje się analizę znanych "złych" zachowań (użytkownika, systemu etc.). Jest to technika stosowana przez większość systemów komercyjnych. Natomiast wykrywanie anomalii wiąże się z wyszukiwaniem i analizą "dziwnych" schematów aktywności. Wykrywanie anomalii jest używane w ograniczonym zakresie przez niektóre systemy IDS. Z każdym z tych podejść wiążą się pewne wady i zalety, ale okazuje się, iż najbardziej efektywne systemy IDS stosują głównie metodę wykrywania nadużyć z niewielką domieszką wykrywania anomalii.

Wykrywanie nadużyć (*Misuse Detection*)

W metodzie tej analizowana jest wszelka aktywność systemowa, w celu odnalezienia zdarzenia lub ciągu zdarzeń pasujących do znanego schematu ataku. Ponieważ schematy pasujące do określonych typów ataków są często nazywane sygnaturami, wykrywanie nadużyć bywa czasem nazywane wykrywaniem ataków opartym na sygnaturach (*signature-based detection*). W takiej detekcji każdy wzór zdarzeń odpowiadających znanemu atakowi traktowany jest jako osobna sygnatura. Jednakże są spotykane bardziej zaawansowane podejścia do wykrywania nadużyć (oparte na analizie stanów), które potrafią używać pojedynczej sygnatury do wykrywania wielu ataków.

Podstawowe zalety i wady:

- Wykrywanie nadużyć jest niezwykle skuteczną metodą wykrywania włamań wolną od generowania ogromnej ilości fałszywych alarmów.
- W tej metodzie możliwe jest szybkie i trafne wykrycie zastosowanej metody lub narzędzia ataku, co pozwala osobie odpowiedzialnej za bezpieczeństwo podjąć odpowiednich środków zaradczych.
- Pozwala na łatwe wysledzenia problemów związanych z bezpieczeństwem nawet mniej zaawansowanym administratorom.
- Dzięki metodzie wykrywania nadużyć można wykryć tylko znane wcześniej typy ataków, dlatego też należy dbać o aktualność naszej bazy wzorców ataków (bazy sygnatur).
- Ścisłe porównywanie sygnatur w celu wykrycia włamania - co czasami powoduje niemożność detekcji nieco zmodyfikowanej wersji ataku na system.

Detekcja anomalii (*Anomaly Detection*)

Detekcja anomalii polega na wykrywaniu niezwykłych zachowań (anomalii) na komputerze lub w sieci. Metoda ta opiera się na fakcie, iż ataki znacząco różnią się od "zwykłej" (dozwolonej) aktywności i dzięki temu mogą być wykrywane przez systemy wychytujące te różnice. Przy użyciu detektorów anomalii konstruuje się profile dopuszczalnej działalności użytkowników, komputerów bądź połączeń sieciowych opierając się na danych zebranych podczas normalnej działalności systemu. Następnie detektory te kontrolują system zbierając dane i używając różnych testów by stwierdzić, czy monitorowana aktywność znacząco odbiega od normy.

Podstawowe zalety i wady:

- Potrafią wykryć "dziwne" zachowania i w związku z tym mają możliwość wykrycia symptomów ataku bez specyficznej wiedzy o nim samym.
- Detektory anomalii mogą generować dane wykorzystywane później w celu definiowania sygnatur dla detektorów nadużyć.
- Detektory anomalii zwykle generują dużą liczbę fałszywych alarmów w związku z nie zawsze możliwymi do przewidzenia zachowaniami użytkowników bądź sieci.
- Metoda często wymaga rozległych "zbiorów treningowych" wydarzeń systemowych w celu zdefiniowania "normalnego" zachowania systemu.

4. Klasyfikacja IDS według typów odpowiedzi

Gdy systemy IDS zdobędą już informacje o wydarzeniach w systemie i przeanalizują ją w celu znalezienia symptomów ataku, przychodzi czas na wygenerowanie odpowiedzi.

Odpowiedzi aktywne (*Active Responses*)

Aktywne odpowiedzi systemów IDS są to zautomatyzowane działania, podejmowane gdy wykryte zostaną określone typy włamań. Wyróżniamy trzy kategorie aktywnych odpowiedzi.

- **Zbieranie dodatkowych informacji.** Najbardziej łagodna, ale czasami najbardziej efektywna odpowiedź polega na zebraniu dodatkowych informacji o prawdopodobnym ataku. System zaczyna "nasłuchiwać" bardziej uważnie w poszukiwaniu informacji, która może przesądzić o podjęciu dalszej akcji. W przypadku systemów IDS może to oznaczać zwiększenie wrażliwości źródeł informacji, na przykład poprzez powiększenie liczby zapisywanych zdarzeń systemowych lub ustawienie monitoringu sieciowego na przechwytywanie wszystkich pakietów, a nie tylko tych kierowanych do określonej maszyny lub portu.
- **Zmiana środowiska.** Kolejna aktywna odpowiedź polega na zatrzymaniu ataku i zablokowaniu kolejnych działań ze strony atakującego. Systemy IDS nie mogą w sposób całkowicie skuteczny odciąć danej osobie dostęp do zasobów. Jednakże stosują blokadę adresu IP, z którego atak wydaje się przychodzić. Zablokowanie zdeterminowanego i wyedukowanego włamywacza jest niezwykle trudne, ale systemy IDS mogą zniechęcić zaawansowanych włamywaczy lub powstrzymać początkujących stosując następujące metody:
 - Dołączanie pakietów TCP resetujących połączenie do danych wysyłanych przez atakującego, przerywając w ten sposób połączenie.
 - Konfiguracja routerów i zapór ogniowych na ignorowanie pakietów przychodzących z określonego adresu IP.
 - Konfiguracja routerów i zapór ogniowych na zablokowanie portów, protokołów lub usług używanych przez atakującego.
 - W ekstremalnych sytuacjach konfiguracja routerów i zapór ogniowych na odcięcie wszystkich połączeń używających określonych interfejsów sieciowych.
- **Podjęcie akcji przeciwko intruzowi.** Niektórzy są zdania, że pierwszą czynnością w aktywnej odpowiedzi powinno być podjęcie stosownej akcji wymierzonej w intruza. Najbardziej agresywna forma tej odpowiedzi wiąże się z przypuszczeniem kontrataku lub aktywnym zbieraniem informacji o komputerze atakującego. Jakkolwiek wydaje się to być bardzo kuszące podejście, zdecydowanie nie jest jednak zalecane. Z powodu wątpliwej legalności takich działań ta metoda może wyrządzić nam więcej szkód, niż sam atak. Pierwszym powodem bardzo ostrożnego podejścia do tej opcji są wątpliwości natury prawnej. Co więcej, ponieważ wielu atakujących używa fałszywych adresów IP w celu dokonania ataku na system, niesie to ze sobą ryzyko uszkodzenia zupełnie przypadkowych i niezaangażowanych komputerów w sieci. W końcu odwet taki może spowodować eskalację "przemocy", prowokując atakującego (który przykładowo pierwotnie miał jedynie zamiar obejrzenia naszej listy plików) do bardziej agresywnych zadań.

Odpowiedzi pasywne (Passive Responses)

Pasywne odpowiedzi w systemach IDS mają za zadanie dostarczać informacje operatorom systemu, zostawiając podejmowanie akcji na podstawie przekazanych informacji ludziami za to odpowiedzialnym. Wiele komercyjnych systemów IDS opiera się wyłącznie na metodzie odpowiedzi pasywnych.

- **Alarmy i powiadomienia.** Alarmy i powiadomienia są generowane przez systemy IDS w celu poinformowania operatorów o wykryciu ataku. Większość komercyjnych systemów IDS pozwala na bardzo swobodny wybór w określaniu jak i kiedy tworzone są alarmy oraz kto jest o nich informowany. Najbardziej popularnym rodzajem alarmu jest alarm na ekranie monitora lub wyskakujące okienko. Jest ono przedstawiane na konsoli IDS lub na innym systemie, który wybraliśmy podczas konfiguracji IDS. Informacja dostarczona w takiej wiadomości może być różnorodna - od suchego faktu, iż atak miał miejsce, aż po niezwykle szczegółowość, z opisem adresu IP atakującego, zastosowanej metody ataku i wyrządzonych szkód. Kolejna opcja doceniana głównie przez duże i rozproszone organizacje polega na zdalnym powiadamianiu o zaistniałych alarmach. To pozwala organizacjom na takie skonfigurowanie systemu IDS, by alarmy wysyłane były na telefony komórkowe lub pagery noszone przez personel odpowiedzialny za bezpieczeństwo sieci. Niektóre produkty oferują również opcję powiadamiania o wydarzeniach pocztą elektroniczną. Nie jest to zalecane, gdyż atakujący monitorując czasem wiadomości e-mail i mogą nawet zablokować taką wiadomość.
- **Pałapki SNMP.** Niektóre komercyjne systemy IDS są zaprojektowane do przesyłania wygenerowanych alarmów do systemu zarządzającego siecią. Używają one pałapek SNMP (*Simple Network Management Protocol*) do wysyłania alarmów do centralnej konsoli zarządzającej, gdzie mogą one być następnie obsługiwane przez personel nadzorujący. Powiązanych jest z tym wiele korzyści, przykładowo możliwość przystosowania całej infrastruktury sieciowej do odpowiedzi na wykryty atak, możliwość przeniesienia obciążenia systemu (związanego z wygenerowaniem aktywnej odpowiedzi) na system inny od atakowanego oraz możliwość użycia wspólnych kanałów komunikacyjnych.

5. Typowe symptomy działania intruzów

- **Powtarzanie się podejrzanego działania.** Jest to jedna z najlepszych metod wykrywania włamań. Zwykle intruz nie wie dokładnie jak za pierwszym razem uzyskać dostęp. Posługuje się wobec tego techniką prób i błędów. Problemem jest rozpoznanie tego powtarzania oraz określenie ile powtórzeń traktować jako potencjalne włamanie. Kolejne próby może dzielić duży odstęp czasowy i to znacznie utrudnia lub wręcz uniemożliwia wykrycie włamania. Metoda ta umożliwia wykrywanie włamań bez znajomości ich szczegółów.
- **Omyłkowe polecenia lub odpowiedzi pojawiające się podczas wykonywania sekwencji automatycznych.** Będą to niezwykle komunikaty o błędach od programów pocztowych i demonów usług systemowych. Trudno jest scharakteryzować rodzaj omyłkowych informacji. Można założyć, że polecenia lub odpowiedzi redagowane przez procesy systemowe lub użytkowe nie redagują ich błędnie. Można wobec tego przyjąć, że prawdopodobnie są redagowane przez człowieka podszywającego się pod określony proces. Przykładem mogą być komunikaty redagowane przez proces *sendmail*. Obserwowane błędy to np. próby usunięcia lub poprawienia błędnie wprowadzonych poleceń, tzw. literówki, lub błędy występujące w jednej próbie połączenia z danej lokalizacji a w pozostałych już nie.
- **Wykorzystanie znanych słabych punktów.** W każdym systemie istnieją słabe punkty, które są dobrze znane i opisane. Dostępne są narzędzia darmowe i komercyjne, które umożliwiają skanowanie integralności. Należą do nich m.in. NNESSUS, Tripwire, SAFEsuite, NetSonar. Są one niekiedy bezcenne podczas oceny bezpieczeństwa systemu. Posługują się nimi jednak nie tylko administratorzy lecz również potencjalni włamywacze. Wobec tego skuteczną techniką wykrywania włamań jest monitorowanie użycia skanerów integralności oraz wykorzystywania znanych słabych punktów systemu. Znane skanery przejawiają pewne regularności w działaniu i wobec tego można utworzyć ich model. Można następnie zaimplementować narzędzia, które będą wskazywały wystąpienie wyszukiwania słabego punktu. Problem polega jednak na tym, że nowe słabe punkty i ataki są ciągle odkrywane, testowane, wykorzystywane i rozpowszechniane. Wymaga to ciągłego śledzenia najnowszych technik hakerskich aby wiedzieć co należy monitorować.

- Niespójności kierunkowe w pakietach przychodzących lub wychodzących. Za symptomy włamania można uważać:
 - pojawienie się zewnętrznych pakietów wejściowych z wewnętrznym adresem źródłowym IP,
 - pojawienie się pakietów wyjściowych z zewnętrznym adresem źródłowym,
 - pojawienie się pakietów z nieoczekiwanymi portami źródła lub przeznaczenia, tzn. niezgodnymi z żądaniem usługi,
 - pojawienie się pakietów z niespodziewanymi potwierdzeniami (ustawiony ACK), tzn. takimi, z którymi nie można związać poprzedniego żądania.
- Niespodziewane atrybuty pewnego żądania usługi lub pakietu. W danym środowisku pewne typy zdarzeń mogą występować regularnie w określonych momentach czasowych lub nie występują w określonych porach (np. transakcje finansowe - EDI). Odstępstwo od takiej reguły może być traktowane jako włamanie. Niektóre włamania mogą być wykryte na podstawie wykorzystywanego unikatowego zestawu zasobów systemowych, np. wzrastanie wykorzystanie procesora przy blokowaniu usługi. Dotyczyć to może również innych zasobów jak określone procesy, usługi, rozmiary systemu plików, natężenie ruchu sieciowego, w itd.
- Niewyjaśnione problemy z pewnym żądaniem usługi, z systemem lub środowiskiem. Mogą one dotyczyć problemów ze sprzętem, zasobami systemowymi, wydajnością systemu, zachowaniem użytkowników, dziennikiem audytu (np. maleje zamiast rosnać).
- Zewnętrzna wiedza o włamaniu. Czasopisma, książki, grupy dyskusyjne, konferencje, spotkania.
- Pojawianie się podejrzanych objawów w ruchu pakietów w sieci. Może to być podejrzana treść ze względu na miejsce przeznaczenia lub źródła.

1. Co to jest polityka bezpieczeństwa?

Polityka bezpieczeństwa jest zbiorem zasad i procedur obowiązujących przy zbieraniu, przetwarzaniu i wykorzystaniu informacji w organizacji. Dotyczy ona całego procesu korzystania z informacji, niezależnie od sposobu jej gromadzenia i przetwarzania.

Procedury zabezpieczania obejmują:

- systemy klasyczne (papierowe) - znane, wypracowane i stosowane regulaminy
- systemy komputerowe - nie opracowane lub niewystarczające zasady. W tej dziedzinie mamy do czynienia z permanentną ewolucją technologii informatycznych. Trendy w TI zmieniają się bardzo często i systemy same niosą pewne zagrożenia wynikające z ich awaryjności i przestarzałości.

Podstawowe zasady tworzenia polityki bezpieczeństwa:

- Inicjatywa w zakresie bezpieczeństwa informacji musi wyjść ze strony kierownictwa organizacji.
- Ostateczną odpowiedzialność za bezpieczeństwo informacji ponosi kierownictwo.
- Tylko, gdy kierownictwo jest zainteresowane bezpieczeństwem, zadania w tym zakresie są traktowane poważnie.
- Wszystkie strategie i procedury powinny odzwierciedlać potrzeby ochrony danych niezależnie od przyjmowanej przez nie formy - dane powinny być chronione niezależnie od nośnika, na którym występują.
- W skład zespołu d/s zarządzania bezpieczeństwem muszą wejść przedstawiciele praktycznie wszystkich komórek organizacyjnych.
- Każdy powinien sobie uświadomić sobie własną odpowiedzialność za utrzymywanie bezpieczeństwa.

2. Zadania zespołu d/s bezpieczeństwa

Dla opracowania koncepcji bezpieczeństwa, organizacji przedsięwzięć związanych z bezpieczeństwem, określenia zakresów odpowiedzialności za bezpieczeństwo, kontroli stanu bezpieczeństwa bezwzględnie należy powołać osobną komórkę organizacyjną: zespół d/s zarządzania bezpieczeństwem. Zespół d/s zarządzania bezpieczeństwem powinien opracować, koordynować i kontrolować proces osiągnięcia wymaganego poziomu bezpieczeństwa. Do podstawowych zadań takiego zespołu należy zaliczyć:

- Ustalenie celów bezpieczeństwa oraz opracowanie polityki gwarantującej osiągnięcie założonych celów.
- Ustalenie zakresu obowiązków osób odpowiedzialnych za bezpieczeństwo.
- Doradzanie i kontrola w zakresie osiągnięcia celów bezpieczeństwa przy opracowywaniu koncepcji bezpieczeństwa.
- Opracowanie planu wdrażania przedsięwzięć bezpieczeństwa określonych w koncepcji bezpieczeństwa.
- Nadzór nad realizacją planu wdrażania przedsięwzięć bezpieczeństwa.
- Nadzór nad informowaniem pracowników o działaniach w zakresie bezpieczeństwa.
- Kontrola efektywności przedsięwzięć bezpieczeństwa w toku pracy organizacji.
- Określanie zasobów niezbędnych do realizacji założonych procesów wdrażania bezpieczeństwa.

Niezwykle ważną osobą dla realizacji koncepcji bezpieczeństwa jest pełnomocnik d/s bezpieczeństwa. Ze względu na wagę problematyki bezpieczeństwa dla niezakłóconej pracy organizacji, jak najbardziej uzasadnione jest stworzenie osobnego etatu. Do jego podstawowych obowiązków można zaliczyć:

- Współdziałanie w ustalaniu koncepcji bezpieczeństwa.
- Informowanie kierownictwa i zespołu d/s zarządzania bezpieczeństwem o przebiegu procesów bezpieczeństwa.
- Ustalenie dróg przepływu informacji od lokalnych pełnomocników i przetworzenie tej informacji.
- Odpowiedzialność i nadzór nad realizacją wybranych przedsięwzięć bezpieczeństwa.
- Planowanie i koordynacja szkoleń w zakresie bezpieczeństwa.
- Utrzymywanie założonego stopnia bezpieczeństwa (kontrole) w trakcie działania organizacji.
- Analizowanie i reagowanie na zdarzenia naruszające bezpieczeństwo.

3. Procedura tworzenia polityki bezpieczeństwa

Osiągnięcie wymaganego poziomu bezpieczeństwa można osiągnąć postępując wg następującego schematu:

- Planowanie
 - Zaprojektowanie polityki bezpieczeństwa.
 - Określenie pożądanego poziomu bezpieczeństwa.
 - Powołanie zespołu d/s zarządzania bezpieczeństwem.
 - Opracowanie polityki bezpieczeństwa.
 - Opracowanie koncepcji bezpieczeństwa.
- Realizacja
 - Wdrożenie przedsięwzięć bezpieczeństwa.
 - Szkolenie personelu w zakresie bezpieczeństwa.
- Eksploatacja - utrzymywanie bezpieczeństwa w toku pracy organizacji.

Wstępem do projektowania *Polityki Bezpieczeństwa Informacji* jest audyt bezpieczeństwa:

- Rozpoznanie problemu przetwarzania informacji w organizacji (przede wszystkim określenie podstawy prawnej).
- Rozpoznanie rodzajów informacji przetwarzanych w organizacji.
- Analiza obiegu poszczególnych grup informacji i rozpoznanie systemów przetwarzania informacji.
- Analiza zagrożeń i ryzyka przetwarzania informacji.
- Ocena stosowanych systemów zabezpieczeń.

Pojęcie ryzyka jest wieloznaczne. W różnych opracowaniach (słownikach, normach) występują różne jego definicje. Wg normy IEC 61508 pod pojęciem ryzyka należy rozumieć miarę stopnia zagrożenia dla tajności, integralności i dostępności informacji wyrażona jako iloczyn prawdopodobieństwa wystąpienia sytuacji stwarzającej takie zagrożenie i stopnia szkodliwości jej skutków.

Analiza ryzyka jest procesem składającym się z następujących etapów:

- Szacowanie ryzyka:
 - Co chronić?
 - Przed czym chronić?
 - Jakie jest prawdopodobieństwo wystąpienia zagrożenia lub skutków zagrożenia?
- Ocena akceptowalności ryzyka:
 - Jaki jest stopień szkodliwości skutków zagrożenia?
 - Jakie są koszty zabezpieczeń?
 - Czy warto chronić?

Szacowanie ryzyka odgrywa bardzo ważną rolę podczas opracowywania strategii zapewnienia bezpieczeństwa. Do szacowania ryzyka czasami wykorzystuje się specjalne programy, lub zatrudnia firmę konsultingową. Można również przeprowadzić w firmie szereg zajęć warsztatowych. Wspólnie tworzona jest wówczas lista zasobów i zagrożeń. Dodatkowym efektem jest wtedy wzrost świadomości zagrożeń wśród uczestników warsztatów. Można powiedzieć, że szacowanie ryzyka polega na ocenie wszystkich negatywnych skutków zagrożeń i określeniu prawdopodobieństw ich wystąpienia.

Lista chronionych zasobów powinna być oparta na odpowiednim planie biznesowym i zdrowym rozsądku. Lista powinna zawierać wszystko co przedstawia pewną wartość z punktu widzenia ewentualnych strat wynikających z nieosiągniętych zysków, straconego czasu, wartości napraw i wymiany uszkodzonych elementów. Przy tworzeniu może okazać się niezbędna:

- znajomość prawa,
- zrozumienie mechanizmów funkcjonowania firmy,
- znajomość zakresu polis ubezpieczeniowych.

Po zdefiniowaniu listy zagrożeń należy wyznaczyć wymiar zagrożeń. Należy ocenić prawdopodobieństwo wystąpienia każdego ze zdarzeń (np. w układzie rocznym). Zwykle jest to zadanie bardzo trudne. Wykonane szacunki trzeba okresowo weryfikować. Trzeba to również robić przy każdej zmianie organizacyjnej (zmiana w działaniu lub strukturze organizacji). Szacowanie ryzyka dotyczy zwykle zdarzeń, dla których częstotliwości występowania nie są określone ani bezpośrednio wyznaczalne na odpowiednim poziomie ufności. Dlatego przy szacowaniu ryzyka używa się metod modelowania przystosowanych do szacowania małych prawdopodobieństw - drzew zdarzeń i drzew błędów. W każdej z tych metod, zadanie złożone dekomponuje się na mniejsze części, które po starannej analizie łączy się ponownie. Uzyskujemy w ten sposób lepsze zrozumienie całego zadania oraz możliwość określania występujących w nim prawdopodobieństw zdarzeń składowych.

Drzewo zdarzeń jest modelem zależności przyczynowo-skutkowych występujących w rozpatrywanym problemie. Zakłada się, że skutek jest wynikiem wystąpienia ciągu zdarzeń. Drzewo zdarzeń rozpoczyna się wobec tego pewnym zdarzeniem inicjującym i przedstawia wszystkie możliwe ciągi zdarzeń będące następstwami zdarzenia inicjującego. W różnych miejscach drzewa zdarzeń znajdują się punkty rozgałęzień ilustrujące fakt, że po pewnych zdarzeniach istnieje możliwość wystąpienia różnych innych ciągów zdarzeń. Prawdopodobieństwo wystąpienia określonego skutku otrzymuje się mnożąc przez siebie prawdopodobieństwo wystąpienia wszystkich zdarzeń występujących na ścieżce od zdarzenia inicjującego do rozważanego skutku.

Drzewo błędów budowane jest w kierunku przeciwnym niż drzewo zdarzeń. Rozpoczyna się od określonego skutku i rozwija w kierunku zdarzeń poprzedzających, pokazując wszystkie możliwe kombinacje zdarzeń niepożądanych.

Ocena akceptowalności ryzyka

Po oszacowaniu ryzyka, do każdego zagrożenia należy przypisać odpowiedni koszt i zestawić to wyliczenie z kosztami ochrony. Takie postępowanie nazywamy *analizą kosztów i zysków* lub *oceną akceptowalności ryzyka*. W większości przypadków nie ma potrzeby przypisywania dokładnych wartości kosztów do poszczególnych zagrożeń. Czasami wystarczy przedział. Należy również wyliczyć koszty działań prewencyjnych, które odpowiadają poszczególnym pozycjom strat. Np. działaniem prewencyjnym, zapobiegającym stratom wynikłym z zaniku zasilania jest zakup i instalacja UPS'a. Należy pamiętać o amortyzacji kosztów w określonym czasie.

Akceptowalność ryzyka napotyka na poważne trudności gdy w grę wchodzi ludzkie życie. Taka sytuacja będzie miała miejsce w systemach wspomagających intensywną terapię, nawigację w samolotach, nadzorowanie pracy reaktorów jądrowych. Wynika to z braku powszechnie akceptowanych metod oceniania ludzkiego życia.

Ostatnim krokiem jest sporządzenie wielowymiarowej tablicy określającej zasoby, koszty i ewentualne straty. Dla każdej straty określa się prawdopodobieństwo jej wystąpienia, przewidywaną wartość straty i koszt prewencji. Określenie, czy zastosowana forma prewencji jest adekwatna - polega na pomnożeniu wartości straty przez prawdopodobieństwo wystąpienia, posortowaniu wyników malejąco i porównaniu kosztów wystąpienia strat z kosztami prewencji.

Taka tabela może dostarczyć również listy zadań priorytetowych. Czasami wyniki są zaskakujące. Celem powinno być unikanie dużych strat. Zwykle pożar i utrata kluczowych osób z personelu są bardziej prawdopodobne i brzemienne w skutkach niż wirusy i włamania poprzez sieć. Natomiast uwagę przykuwają zwykle te ostatnie.

Bezpieczeństwa nie uzyskuje się za darmo. Im bardziej zaawansowane metody jego uzyskania, tym droższe. Zwykle systemy bezpieczniejsze są również trudniejsze w eksploatacji.

Szacowanie ryzyka pomaga w umotywowaniu potrzeby przeznaczenia określonych środków finansowych na zapewnienie bezpieczeństwa. Większość kadry menedżerskiej niewiele wie na temat komputerów, ale rozumie analizę kosztów i zysków.

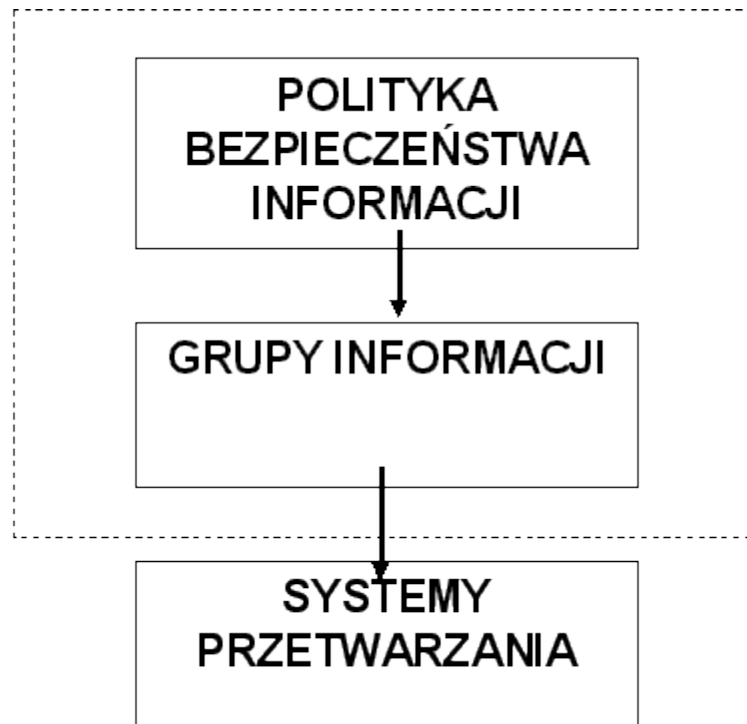
Chyba jedną z najlepszych metodyk opracowywania polityki bezpieczeństwa jest TISM (*Total Information Security Management*) zaproponowana przez Europejski Instytut Bezpieczeństwa Sieciowego.

Wg metodologii TISM przyjmuje się trzy podstawowe poziomy w hierarchii polityki bezpieczeństwa:

- Polityka Bezpieczeństwa Informacji - dokument główny
- Grupa Informacji
- System Przetwarzania

Poziom głównego dokumentu Polityki Bezpieczeństwa Informacji jest poziomem, na którym ustala się podstawowe zasady ochrony informacji w organizacji. Na poziomie grupy informacji ustala się specyficzne wymagania ochrony dla danej grupy informacji. Poziom systemu przetwarzania jest natomiast poziomem, na którym określa się spełnienie wymagań wyższych poziomów przez system przetwarzania, w którym informacje z danej grupy się znajdują. Schemat TISM przedstawiono na rys. 1.

**Poziomy określenia
wymagań**



**Poziom spełnienia
wymagań**

Rys. 1. Idea metodologii TISM