WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ I ZARZĄDZANIA
pod auspicjami Polskiej Akademii Nauk

WYDZIAŁ INFORMATYKI

STUDIA I STOPNIA
(INŻYNIERSKIE)

PRACA DYPLOMOWA

Marek Marczak

TUNELOWANIE W WARSTWIE TRZECIEJ SIECI IP

Praca wykonana pod kierunkiem:

dr inż. Jerzy Królikowski

WARSZAWA, 2013 r.

Spis treści

WSTĘP

Tunelowanie to sposób tworzenia sieci wirtualnej na podstawie istniejącej sieci fizycznej.

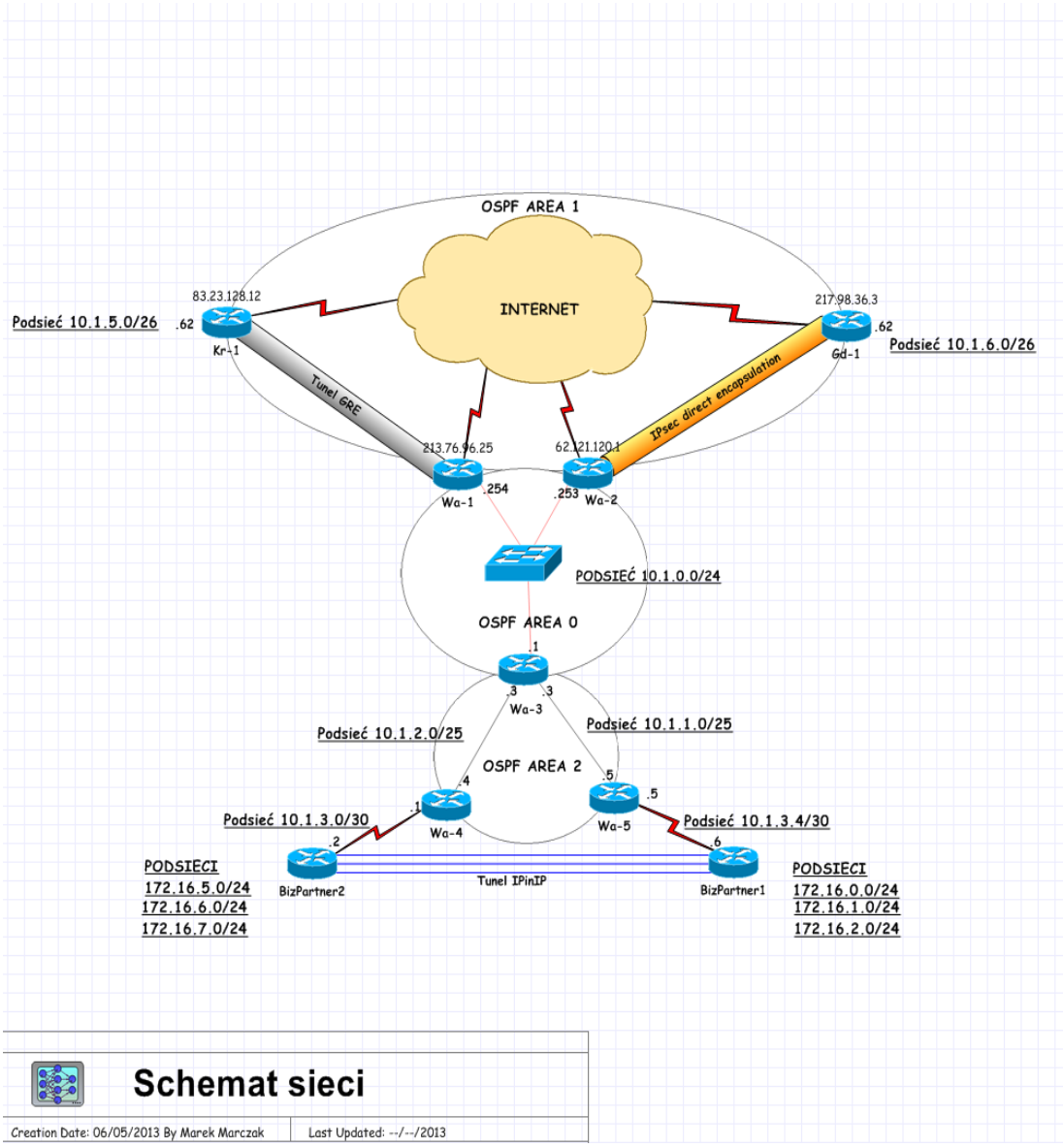
Tunel – zestawienie połączenia między dwoma odległymi hostami tak, by stworzyć wrażenie, że są połączone bezpośrednio. Stosując odpowiednią technikę tworzymy tunel wirtualny, który łączy oba hosty czy też sieci. W miarę jak sieci komputerowe stały się coraz bardziej skomplikowane i wykorzystywały różne protokoły do przesyłania danych nastąpiła potrzeba połączenia pomiędzy hostami oddzielonymi siecią opartą o inny protokół. Jako przykład można wymienić, komunikowanie się hostów w dwóch sieciach opartych na protokole IPv6 przedzielonych siecią Internet. Ponieważ sieć Internet opiera się na IPv4 nie jest możliwe bezpośrednie porozumiewanie się hostów w obu tych sieciach. Aby obejść ten problem stosuje się właśnie tunelowanie polegające na enkapsulacji protokołu Ipv6 w protokół IPv4. Ogólnie rzecz biorąc każdy tunel IP składa się z co najmniej dwóch komponentów:

protokołu pasażera – czyli protokołu, który chcemy przenieść. Należy zwrócić uwagę, iż protokół pasażer najczęściej należy do warstwy trzeciej modelu OSI lub wyższej od

Protokołu transportowego – czyli protokołu, który jest wykorzystany do przeniesienia protokołu pasażera.

PREZENTACJA SIECI

Prezentowana sieć składa się z 7 routerów oraz 2 routerów partnera handlowego. Ze względu na wymogi stałego dostępu do zasobów centrali w warstwie rdzenia znajdują się routery Cisco model 7200 (Wa-1, Wa-2). Została też wykupiona usługa u dwóch różnych dostawców Internetu (ISP). Dzięki temu zapewniona jest częściowa redundancja, a co za tym idzie większa niezawodność sieci. Na wyżej wymienionych routerach skonfigurowany został NAT, a raczej jego odmiana PAT (Port Address Translation). W celu podniesienia niezawodności na routerach Wa-1,Wa-2 uruchomiono protokół HSRP.W centrali znajduje się 5 routerów i oprócz wymienionych powyżej znajdują się również Wa-3, Wa-4, Wa-5 firmy Cisco model 2691. Pozostałe urządzenia aktywne oraz pasywne sieci ze względu na to, iż nie odgrywają roli w prezentacji zostaną pominięte.



Pomimo stosunkowo niewielkiej sieci zdecydowano o uruchomieniu pomiędzy routerami dynamicznego protokołu routingu. Wybór padł na OSPF ze względu na otwartość tego protokołu oraz fakt, iż jest on wspierany przez wszystkich większych dostawców urządzeń aktywnych sieci. Zdecydowano się na utworzenie 3 stref ospf, aby w ten sposób można było zmniejszyć tablice trasowania routerów. W celu ograniczenia domeny broadcastu zdecydowano się na podzielenie sieci na kilka podsieci. W centrali znajdują się następujące podsieci:

10.1.0.0/24
10.1.1.0/25
10.1.2.0/25
10.1.3.0/30
10.1.3.4/30

Istnieje potrzeba połączenia filli przedsiębiorstwa z centralą. Ponieważ łącza dzierżawione są drogie, a rozwój Internetu i nowoczesnych technologii stają się coraz bardziej godne zaufania, zdecydowano się na połączenie odległych biur przez Internet. Rodzi to pokreślone problemy z których największym jest to jak połączyć dwie podsieci o adresacji z zakresu sieci prywatnych, opisanych RFC 1918. Należy pamiętać że translacja adresów czyli NAT nie rozwiązuje sprawy, gdyż zamienia adresy nierutowalne na publiczne w danej sieci, ale nie istnieje możliwość komunikacji przez Internet pomiędzy dwiema sieciami prywatnymi. Po prostu każdy adres z zakresu tych sieci jest automatycznie blokowany przez ISP.

Podsieci w odległych oddziałach:
10.1.5.0/26
10.1.6.0/26

W centrali podjęto decyzję, iż mając sieć o dużej przepustowości udostępni się łącza do partnera handlowego, który ze względu na lokalizację skazany jest na łącza o małej przepustowości. Rodzi to problem zachowania przejrzystości i rozdzielności podsieci partnera od sieci centrali. W takim przypadku doskonale sprawdzi się tunel IPinIP przesyłające dane pomiędzy sieciami partnera transparentnie dla sieci centrali.

Podsieci partnera handlowego:

172.16.0.0/24
172.16.1.0/24
172.16.2.0/24
172.16.3.0/24
172.16.4.0/24

172.16.5.0/24

Aby zaprezentować jak w praktyce stosuje się idee tunelowania, zostaną utworzone 3 tunele.

1. Tunel IPsec direct encapsulation pomiędzy Gd-1 a centralą.
2. Tunel GRE pomiędzy Kr-1 a centralą.
3. Tunel IPinIP przechodzący przez sieć centrali, łączący sieci partnera handlowego.

Poniżej znajduje się konfiguracja interfejsów poszczególnych routerów, ich nazwy i lokalizacja:

			Interfejsy					
Nazwa	**Urządzenie**	**Lokalizacja**	**Serial 0/0**	**Serial 1/0**	**Fa 0/0**	**Fa 0/1**	**Fa 1/0**	**Fa 2/0**
Wa-1	7200	Centrala		213.76.96.25				10.1.0.254
Wa-2	7200	Centrala		62.121.120.1				10.1.0.253
Wa-3	2961	Centrala			10.1.0.1	10.1.2.3	10.1.1.3	
Wa-4	2961	Centrala	10.1.3.1		10.1.2.4			
Wa-5	2961	Centrala	10.1.3.5		10.1.1.5			
Kr - 1	2961	Filia		80.238.120.254	10.1.5.62			
Gd-1	2961	Filia	217.98.36.3		10.1.6.62			
BizPartner1	2962		10.1.3.6					
BizPartner2	2963		10.1.3.2					

W prezentowanej sieci przyjęto konwencję, iż wszystkie tunele mają podsieci z zakresu 10.2.0.0 255.255.0.0, a numer tunelu składa się połączonych ostatnich oktetów adresu hosta. Więc na przykład tunel pomiędzy Krakowem i Warszawa będzie miał numer 1225, gdyż 83.23.128.**12** oraz 213.76.96.**25**

KONFIGURACJA URZĄDZEŃ

Konfiguracja tunelu IPSec

Konfiguracja IKE ze współdzielonymi kluczami

Pomiędzy Gd-1 oraz centralą zostanie ustanowiony tunel Ipsec z bezpośrednim niekapsulowaniem. Aby to uczynić należy najpierw skonfigurować IKE(Internet Key Exchange) . IKE funkcjonalnie składa się z dwóch części: specyfikacji metod kryptograficznych uwierzytelnienia i negocjacji kluczy Oakley (IKE: RFC 2409, IKEv1: RFC 4109, IKEv2: RFC 4306) oraz specyfikacji formatów pakietów i stanów protokołu ISAKMP. W praktyce nazwy ISAKMP używa się zamiennie z IKE. Jako metodę uwierzytelniającą wybrano klucz współdzielony. Ten typ konfiguracji uwierzytelnień komunikujących się z sobą urządzeń nie wykorzystuje kryptografii klucza publicznego. Pomimo iż jego skalowalność jest stosunkowo ograniczona, a bezpieczeństwo zależne od przyjętej metody dystrybucji kluczy współdzielonych dla IKE, w przypadku gdy nie zalecane jest obciążanie routerów często się ją stosuje. Dwa urządzenia są uwierzytelniane tylko wtedy, gdy posiadają manualnie wprowadzony do ich konfiguracji tajny klucz. Jeżeli klucze te będą się różnić nie uda się nawiązać logicznego połączenia security association (SA) IKE. Jak wspomniano o tym wyżej w ramach SA IKE dochodzi do uzgodnienia mechanizmów kontroli integralności danych, enkrypcji, czasu ważności sesji IKE oraz grupy liczb Diffie-Hellmana. Liczby DH są wykorzystywane do ustalenia wspólnego tajnego klucza szyfrującego ważnego na czas trwania sesji, bez konieczności jego przesyłania łączami sieci. SA IKE pełni rolę zarządzającą dla później zestawianych SA IPSec. Oczywiste jest, że niemożność zestawienia SA IKE skutkuje brakiem połączeń IPSec.

Po wydaniu polecenia włączającego mechanizm IKE należy stworzyć propozycje parametrów IKE (transform set). Transform set dla IKE przyjmują postać reguł polityki IKE. Możliwe jest stworzenie wielu transform sets i dlatego każdy identyfikowany jest numerem decydującym o kolejności negocjacji, gdy dwa urządzenia próbują uzgodnić wspólną regułę polityki IKE. Czym wyższy priorytet zbioru reguł tym niższy jego numer.

```
Gd1(config)#crypto isakmp policy 10
Gd1(config-isakmp)#authentication pre-share
Gd1(config-isakmp)#hash sha
Gd1(config-isakmp)#encryption aes 256
Gd1(config-isakmp)#lifetime 180
```

Pierwsze z poleceń ustawia numer polityki IKE, a następne umożliwia uwierzytelnianie urządzeń wspólnym, tajnym kluczem, który będziemy musieli podać dla obydwu routerów. Do kontroli integralności informacji wymienianych w ramach SA IKE będzie wykorzystywana funkcja Secure Hash Algorithm SHA-1. Poufność transmisji zapewni szyfrowanie algorytmem AES z 256-bitową długością klucza. Czas trwania SA IKE wynosi 180 sekund, czyli 3 minuty. Po upływie tego czasu urządzenia będą musiały wynegocjować i zestawić nowe SA IKE. Wydaje się to bardzo mało, lecz należy

pamiętać, że jest to tylko faza negocjacji, jeżeli zestawiony jest tunel SA IKE jest niepotrzebne i fakt iż obecny SA IKE wygasł nie przeszkadza w funkcjonowaniu tunelu. W praktyce należy ustawiać czas trwania SA IKE gdyż po upływie tego czasu zasoby routera nie są wykorzystywane do negocjacji. W przypadku jednego tunelu nie ma problemu, lecz gdyby takich tuneli było dużo 1000 lub więcej (np. bankomaty), to wydajność routera spadała by drastycznie.

Do dopełnienia konfiguracji IKE pozostało jeszcze z poziomu globalnego trybu konfiguracyjnego IOS routera podanie klucza pre-shared:

```
Gd1(config)#crypto isakmp key 0 klucz address 62.121.120.1
```

Podany klucz musi być oczywiście identyczny dla obu urządzeń. Urządzenia identyfikowane są przez adresy IP. Ten sposób identyfikacji jest domyślnie przyjęty. Możliwa jest także identyfikacja przez nazwy mapowane statycznie w konfiguracji routera lub zarządzane przez DNS.

Pojawia się pytanie, kiedy stosować identyfikację urządzeń przez adres, a kiedy przez nazwę. Identyfikacja przez adres jest na ogół wykorzystywana wtedy, gdy negocjacja parametrów IKE odbywa się tylko przez jeden interfejs routera. Identyfikacja przez nazwę jest bardziej odpowiednia dla SA IKE nawiązywanych do różnych urządzeń przez więcej niż jeden interfejs lub gdy adres IP urządzenia nie jest z góry znany lub często się zmienia (w przypadku adresów z TP s.a).

Konfiguracja IPSec

Pierwszym etapem konfiguracji IPSec jest zazwyczaj zdefiniowanie zbiorów protokołów i funkcji kryptograficznych tworzących razem tzw. transform sets. Każdy taki zbiór jest identyfikowany poprzez nazwę. Zanim będzie możliwe nawiązanie SA IPSec, którym transmitowane będą dane, urządzenia negocjują wspólny dla nich transform set. Jeśli takowego nie da się ustalić na etapie negocjacji nie będzie możliwe utworzenie SA IPSec.

```
Gd1(config)#crypto ipsec transform-set TS esp-aes esp-sha-hmac
Gd1(cfg-crypto-trans)# mode tunel
```

Każdy zbiór protokołów IPSec, algorytmów i funkcji kryptograficznych może zawierać tylko jedną pozycję z AH (Authentication Header) i jedną lub dwie z ESP (Encapsulating Security Payload). W naszym przykładzie zbiór o nazwie TS określa, że do szyfrowania danych będzie wykorzystany algorytm AES w ramach SA IPSec w protokole ESP (tylko ESP oferuje funkcjonalność szyfrowania), kontrola integralności i uwierzytelnienia pochodzenia danych będą przeprowadzone funkcją SHA-1 także w ramach SA IPSec w protokole ESP.

Kolejną czynnością wymaganą dla uruchomienia IPSec jest zdefiniowanie jakie pakiety zostaną zabezpieczone przez IPSec. Odbywa się to przy pomocy rozszerzonych list dostępu. Należy pamiętać aby tworząc ACL nie używać słowa any, czy to w odniesieniu do celu czy też źródła. Listy określające pakiety podlegające zabezpieczeniom na routerach z obu stron tunelu IPsec powinny być swym lustrzanym odbiciem. Listy te, po pierwsze określają jakie pakiety pojawiające się na interfejsie

wyjściowym routera mają być zabezpieczone IPSec, po drugie, pozwalają routerowi określić ruch zwrotny (pojawiający się na wejściu tego samego interfejsu), który także powinien być zabezpieczony (lustrzane odbicie warunków ACL). Jeśli, z powodu błędów konfiguracji ACL część ruchu zwrotnego nie jest zabezpieczona, router będzie odrzucał takie pakiety IP. Warunki permit crypto ACL określają zabezpieczony ruch pakietów IP, natomiast jawne lub niejawne (działające na końcu każdej listy dostępu) warunki deny określają ruch, który nie będzie podlegał zabezpieczeniom.

```
Gd1(config)#access-list 101 permit ip 10.1.6.0 0.0.0.63 10.1.0.0 0.0.3.255
Wa2(config)#access-list 101 permit ip 10.1.0.0 0.0.3.255 10.1.6.0 0.0.0.255
```

Następny etap konfiguracji IPSec będzie polegał na utworzeniu tzw. crypto mapy. Jej zadaniem będzie związanie transform set oraz crypto ACL z konkretnym adresem urządzenia sąsiada oraz określenie czy w zestawieniu i zarządzaniu SA IPSec uczestniczy IKE.

```
Gd1(config)#crypto map CMAP 10 ipsec-isakmp
Gd1(config-crypto-map)# set peer 62.121.120.1
Gd1(config-crypto-map)# set transform-set TS
Gd1(config-crypto-map)# match address 101
```

Poleceniem tym utworzono jedną sekwencję crypto mapy CMAP. Crypto mapa może obejmować jeden lub więcej sekwencji, a wtedy każda z nich identyfikowana jest numerem podawanym po nazwie. Wielosekwencyjną crypto mapę definiuje się wtedy, gdy przez interfejs, z którym ją później związemy będziemy nawiązywać połączenia IPSec do więcej niż jednego urządzenia lub tylko do jednego urządzenia, ale celem naszym będzie zastosowanie innych reguł zabezpieczeń dla różnych kategorii ruchu. Po numerze sekwencji crypto mapy pojawia się określenie typu IPsec. Tunel może być dynamicznie zestawiany (ipsec-isakmp) lub też manualnie (ipsec-manual). Polecenie match address pozwala na związanie listy crypto ACL z crypto mapą a tym samym wskazaniem jaki ruch będzie szyfrowany, set peer określa adres urządzenia sąsiada dla tworzonego połączenia, natomiast set transform-set powołuje się na wcześniej utworzony zbiór protokołów IPSec (ESP, AH), tryb działania SA (tunel, transport) i ewentualnie rodzaj funkcji do kontroli integralności i uwierzytelnień pochodzenia danych (MD-5, SHA-1). Innymi słowy określa rodzaj i poziom zabezpieczeń.
Końcową czynnością jest podpięcie crypto mapy na określony interfejs(wychodzący do ISP):

```
Gd1(config)#interfaces0/0
Gd1(config-if)# crypto map CMAP
```

Teraz wystarczy jeszcze dodać trasowanie do tej sieci:

```
Gd1(config-if)#ip route 10.1.6.0 255.255.255.192 217.98.36.3
```

Po drugiej stronie konfiguracja jest symetryczna, a więc nie zostanie tutaj przedstawiona.
Ponieważ w centrali skonfigurowany jest NAT należy tak go zmodyfikować, aby nie próbował tłumaczyć ruchu skierowanego do sieci 10.1.6.0/26 na adres publiczny. W tym celu na routerze Wa-2

zmieniamy listę dostępu powiązaną z NAT:

```
Wa2(config)#access-list 111 deny ip any 10.1.6.0 0.0.0.63
Wa2(config)#access-list 111 permit ip 10.1.0.0 0.0.3.255 any
```

Pierwszy wpis zabrania tłumaczenia jakiegokolwiek adresu próbującego dostać się do sieci 10.1.6.0/26, natomiast drugi pozwala na translację adresów o zakresie 10.1.0.0.

Weryfikacja połączenia:

W celach testowych należy wysłać pakiet icmp do odległej sieci:

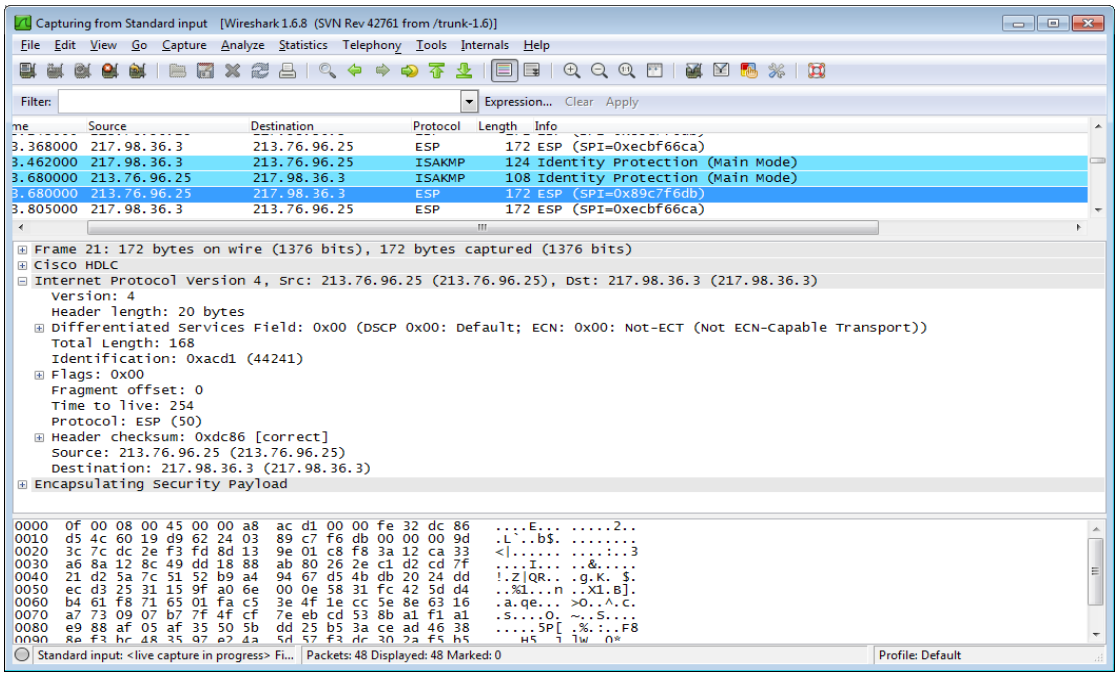
```
Gd1#ping 10.1.0.1 source fa0/0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.6.62
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 88/156/216 ms
```

Stan sesji można sprawdzić poleceniem:

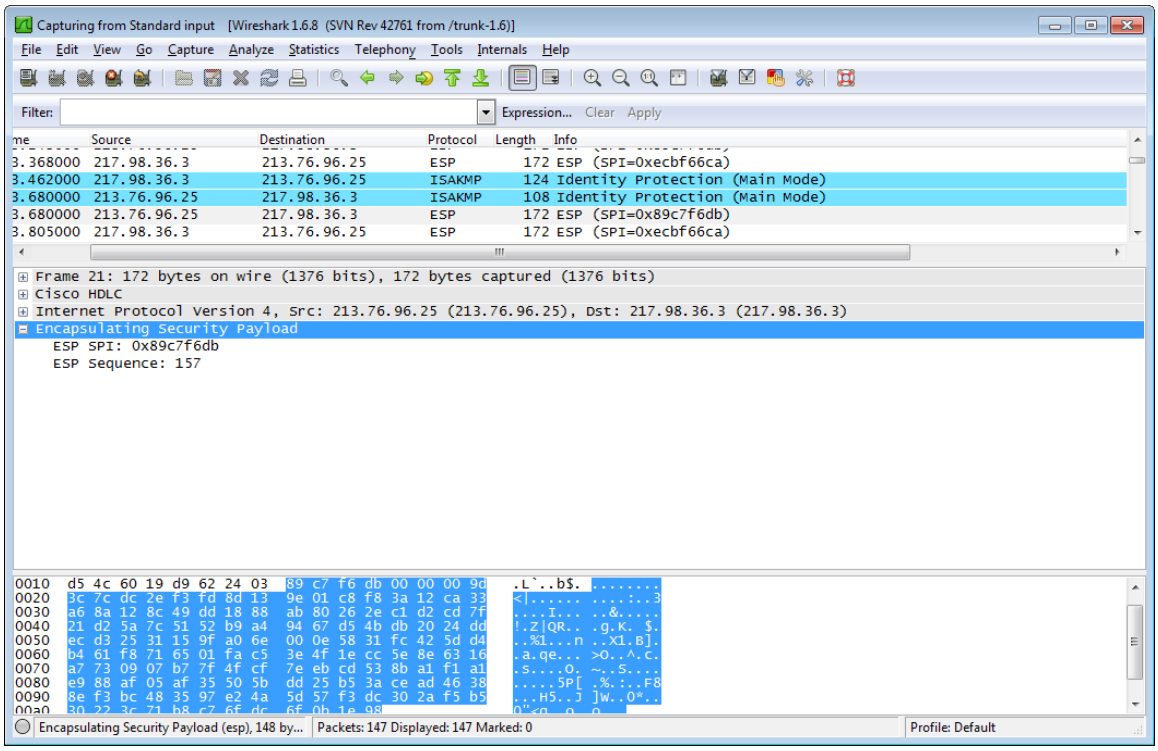
```
Gd1#show crypto session
Crypto session current status

Interface: Serial0/0
Session status: UP-ACTIVE
Peer: 62.121.120.1 port 500
  IKE SA: local 217.98.36.3/500 remote 62.121.120.1/500 Active
  IPSEC FLOW: permit ip 10.1.6.0/255.255.255.192 10.1.0.0/255.255.252.0
        Active SAs: 2, origin: crypto map

Gd1#

Gd1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
62.121.120.1    217.98.36.3     QM_IDLE           1001    0 ACTIVE

IPv6 Crypto ISAKMP SA

Gd1#
Gd1#show crypto ipsec sa

interface: Serial0/0
    Crypto map tag: CMAP, local addr 217.98.36.3

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.1.6.0/255.255.255.192/0/0)
   remote ident (addr/mask/prot/port): (10.1.0.0/255.255.252.0/0/0)
   current_peer 62.121.120.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
```

```
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 1, #recv errors 0

 local crypto endpt.: 217.98.36.3, remote crypto endpt.: 62.121.120.1
 path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0
 current outbound spi: 0x23B26BA5(598895525)

 inbound esp sas:
  spi: 0x652F8459(1697612889)
    transform: esp-aes esp-sha-hmac ,
    in use settings ={Tunnel, }
    conn id: 1, flow_id: SW:1, crypto map: CMAP
    sa timing: remaining key lifetime (k/sec): (4493043/3544)
    IV size: 16 bytes
    replay detection support: Y
    Status: ACTIVE

 inbound ah sas:

 inbound pcp sas:

 outbound esp sas:
  spi: 0x23B26BA5(598895525)
    transform: esp-aes esp-sha-hmac ,
    in use settings ={Tunnel, }
    conn id: 2, flow_id: SW:2, crypto map: CMAP
    sa timing: remaining key lifetime (k/sec): (4493043/3540)
    IV size: 16 bytes
    replay detection support: Y
    Status: ACTIVE

 outbound ah sas:

 outbound pcp sas:
```

Ponieważ jest to połączenie przesyłające pakiety przez Internet, rodzi to niebezpieczeństwo podsłuchania przez potencjalnego agresora wysyłanych danych. W przypadku IPSec wszystkie dane są oczywiście szyfrowane, co wyraźnie widać w popularnym snifferze Wireshark:



W nagłówku IP znajdują się adresy routowalne, a w polu protokół znajduje się ESP. Natomiast po przejściu do nagłówka ESP można odczytać tylko SPI (Security Parameters Index) oraz numer sekwencyjny. Wartość SPI jest stała, najczęściej jest generowana losowo i ma kluczowe znaczenie dla stron połączenia, a żadnego dla atakującego. Strony połączenia bowiem na podstawie SPI rozpoznają do jakiej sesji (tunelu) IPsec przynależy dany pakiet i jakie w związku z tym zastosować szyfry oraz klucze do jego rozszyfrowania.



Przenoszone dane są zaszyfrowane, a w związku z tym ich analiza jest niemożliwa.

Ponieważ bardzo istotne jest to, aby dostęp do centrali był zawsze, należy wprowadzić redundancję. Aby to zrobić trzeba przede wszystkim w crypto mapie dodać następnego peera, aby na wypadek niemożności połączenia się z routerem głównym, można było zestawić tunel IPSec z drugim routerem.

```
Gd1(config)#crypto map CMAP 10 ipsec-isakmp
Gd1(config-crypto-map)# set peer 213.76.96.25
```

Oraz dodać klucz dla nowego peera:

```
Gd1(config)#crypto isakmp key klucz address 213.76.96.25
```

Aby możliwe było zestawienie tunelu z routerem posiadającym interfej 213.76.96.25 router ten powinien również mieć skonfigurowany IKE oraz IPSec. Ponieważ konfiguracja przebiega tak samo jak wyżej nie zostanie tutaj przedstawiona. Dzięki tym poleceniom osiągnięta została częściowa redundancja polegająca na tym, że jeżeli dla Gd1 nie uda się zestawić tunelu z Wa-2, wtedy zestawi tunel IPSec z Wa-1.

Takie rozwiązanie nie zabezpiecza jednak przed sytuacją, gdy mając zestawiony tunel z przeciwną stroną, interfejs po przeciwnej stronie przechodzi w położenie down. Aby wykryć takie sytuacje powstał zdefiniowany w RFC 3706 **Dead Peer Detection**.

Jest to mechanizm używany do wykrywania nieaktywnych peerów korzystający z IPSEC. DPD ma dwie opcje: na żądanie oraz okresową. Domyślnie włączona jest opcja na żądanie. Przy włączonej tej opcji wiadomości DPD wysyłane są jeżeli zajdzie taka potrzeba. Na przykład, jeżeli router musi wysłać pakiety lecz wątpliwe jest to, czy peer jest dostępny, router wysyła zapytanie DPD. Jeżeli nie nic nie wysyła, to nigdy nie pyta się o to ,czy peer jest dostępny, a więc nie dowie się tego, aż do czasu wygaśnięcia IPSec SA. Opcja okresowa pozwala wymusić na routerze, aby co jakiś określony czas wysyłał zapytania DPD, tym samym sprawdzając dostępność hosta po drugiej stronie tunelu. Aby skonfigurować DPD należy wydać polecenie na wszystkich 3 routerach:

```
Gd1(config)#crypto isakmp keepalive 10
```

Najczęściej zapytania DPD można wysyłać co 10 sekund.
Dzięki dopisaniu do crypto mapy zapasowego peera, oraz dzięki Dead Peer Detection udało się wprowadzić

Reverse Route Injection to kolejna technologia, dzięki której staje się możliwe stworzenie wysoce niezawodnego tunelu(High Availability). RRI wstrzykuje statyczną trasę do tablicy routingu dla adresu sieciowego w oparciu o crypto ACL odległego routera. Następnie można taką trasę redystrybuować używając protokołu dynamicznego routingu. Statyczna trasa jest tylko wtedy aktywna, gdy IPSec SA jest aktywne. Należy zaznaczyć, że RRI działa w jedną stronę, to znaczy jeżeli nie ma żadnej aktywnej sesji, to host z centrali nie połączy się z odległą siecią właśnie z powodu tego, iż nie

ma trasy do tej sieci.

```
Wa1(config)#crypto map CMAP 10 ipsec-isakmp
Wa1(config-crypto-map)#reverse-route


Gd1#show crypto session
Crypto session current status

Interface: Serial0/0
Session status: UP-ACTIVE
Peer: 62.121.120.1 port 500
  IKE SA: local 217.98.36.3/500 remote 62.121.120.1/500 Active
  IPSEC FLOW: permit ip 10.1.6.0/255.255.255.192 10.1.0.0/255.255.252.0
        Active SAs: 2, origin: crypto map


Gd1#ping 10.1.0.1 source f0/0 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.6.62
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.............!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 87 percent (87/100), round-trip min/avg/max = 60/156/308 ms
Gd1#show crypto session
Crypto session current status

Interface: Serial0/0
Session status: UP-ACTIVE
Peer: 213.76.96.25 port 500
  IKE SA: local 217.98.36.3/500 remote 213.76.96.25/500 Active
  IPSEC FLOW: permit ip 10.1.6.0/255.255.255.192 10.1.0.0/255.255.252.0
        Active SAs: 2, origin: crypto map

Interface: Serial0/0
Session status: DOWN-NEGOTIATING
Peer: 62.121.120.1 port 500
  IKE SA: local 217.98.36.3/500 remote 62.121.120.1/500 Inactive
  IKE SA: local 217.98.36.3/500 remote 62.121.120.1/500 Inactive
```

Jak widać na listingu powyżej, po wykryciu, iż peer jest niedostępny została zestawiona sesja z drugim peera. W tym czasie zostało straconych 13 pakietów icmp. Dzięki powyższym mechanizmom i poleceniom uzyskano znaczną niezawodność połączenia.

Konfiguracja tunelu GRE

Aby utworzyć tunel na urządzeniu Cisco należy w pierwszej kolejności stworzyć interfejs tunel, następnie przypisać mu adres i maskę. Po wykonaniu tych kroków należy wskazać jaki interfejs jest źródłem, a który celem, a na koniec wybrać rodzaj tworzonego tunelu. W tym przypadku będą to kolejno polecenia:

```
Kr1(config)#interface Tunnel1225
Kr1(config-if)# ip address 10.2.5.12 255.255.255.0
Kr1(config-if)# tunnel source Serial0/0
Kr1(config-if)# tunnel destination 213.76.96.25
Kr1(config-if)#exit
```

Na urządzeniach Cisco każdy tworzony tunel domyślnie jest tunelem GRE więc nie trzeba wpisywać w tym przypadku polecenia:

```
Kr1(config-if)#tunnel mode gre ip
```

I analogicznie to drugiej stronie tunelu:

```
Wa1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Wa1(config)#interface Tunnel1225
Wa1(config-if)# ip address 10.2.5.25 255.255.255.0
Wa1(config-if)# tunnel source Serial1/0
Wa1(config-if)# tunnel destination 83.23.128.12
Wa1(config-if)#exit
```

Dzięki tym poleceniom stworzony został wirtualny tunel pomiędzy dwoma hostami 83.23.128.12 i 213.76.96.25. Tunel ten jednak nie będzie działał jeżeli nie ma trasy do hosta po przeciwnej stronie, a więc należy przekazać routerowi, którym interfejsem ma wysyłać pakiety, aby dotarły do drugiego końca tunelu. W tym celu tworzona jest trasa statyczna domyślna po obu stronach tunelu prowadząca do ISP.

```
Wa-1(config)#ip route 0.0.0.0 0.0.0.0 Serial1/0
Kr-1(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

Ostatnim krokiem jest utworzenie trasy do sieci, której ruch ma być tunelowany, czyli w tym przypadku:

```
Wa-1(config)#ip route 10.1.5.0 255.255.255.0  10.2.5.12
Kr-1(config)#ip route 10.0.0.0 255.0.0.0 10.2.5.25
```

Na obecnym etapie tunel po obydwu stronach powinien być w stanie up/up.
Na routerach Wa-1 oraz Kr-1 zostanie uruchomiony protokół ospf.

```
Kr1(config)#router ospf 1
Kr1(config-router)# network 10.0.0.0 0.255.255.255 area 1
Kr1(config-router)#exit
```

Na routerze Wa-1 natomiast:

```
Kr1(config)#router ospf 1
Kr1(config-router)# network 10.1.5.0 0.0.0.63 area 1
Kr1(config-router)#exit
```

Weryfikacja tunelu:

W celu zweryfikowania, czy możliwa jest komunikacja pomiędzy dwoma sieciami opisanymi w RFC 1918 poprzez Internet wydajemy polecenie:

```
Kr1#show ip interface b
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            10.1.5.254      YES manual up
up
Serial0/0                  83.23.128.12    YES NVRAM  up
up
FastEthernet0/1            unassigned      YES NVRAM  administratively down
down
Serial0/1                  unassigned      YES NVRAM  administratively down
down
Tunnel1225                 10.2.5.12       YES NVRAM  up
up
```

Ostatnia linijka mówi nam, że tunnel1225 działa jak należy. Aby uzyskać bardziej szczegółowe informacje tyczące się tunelu wydajemy polecenie:

```
Kr1#show interfaces tunnel 1225
Tunnel1225 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 10.2.5.12/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 141/255, rxload 56/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 83.23.128.12 (Serial0/0), destination 213.76.96.25
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 2000 bits/sec, 2 packets/sec
  5 minute output rate 5000 bits/sec, 5 packets/sec
     6622 packets input, 690008 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     7199 packets output, 763768 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

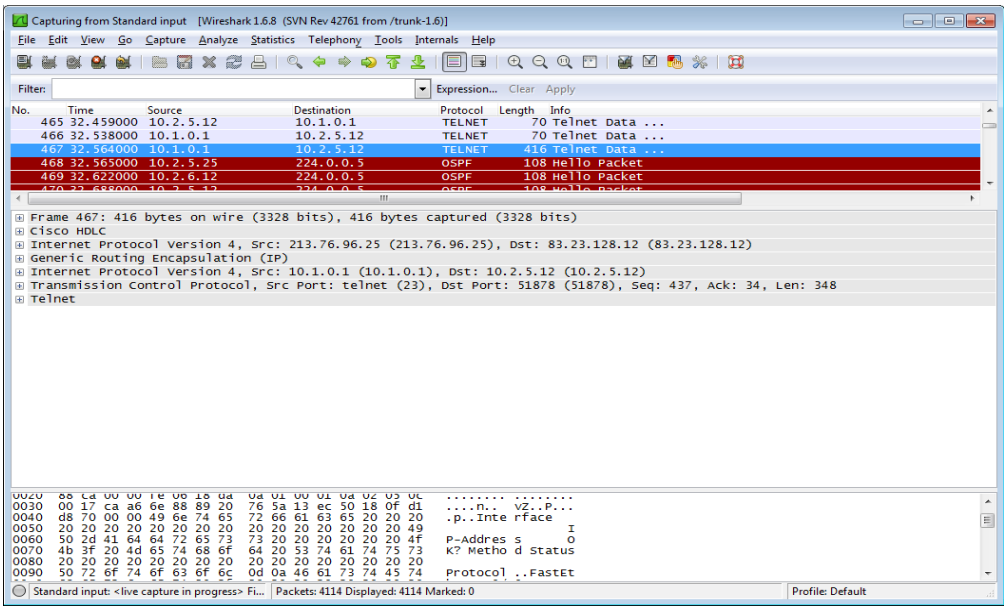
Aby przekonać się, czy ospf działa właściwie wydajemy polecenie:

```
Kr1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.2.5.25 to network 0.0.0.0

     83.0.0.0/24 is subnetted, 1 subnets
C       83.23.128.0 is directly connected, Serial0/0
     10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA    10.1.3.0/25 [110/11122] via 10.2.5.25, 00:00:11, Tunnel1225
O IA    10.1.2.0/24 [110/11177] via 10.2.5.25, 00:00:11, Tunnel1225
O IA    10.1.1.0/25 [110/11113] via 10.2.5.25, 00:00:11, Tunnel1225
O IA    10.1.0.0/21 [110/11112] via 10.2.5.25, 00:00:11, Tunnel1225
C       10.2.5.0/24 is directly connected, Tunnel1225
C       10.2.6.0/24 is directly connected, Tunnel121
O IA    10.1.4.0/24 [110/11186] via 10.2.5.25, 00:00:13, Tunnel1225
C       10.1.5.192/26 is directly connected, FastEthernet0/0
     213.76.96.0/32 is subnetted, 1 subnets
S       213.76.96.25 is directly connected, Serial0/0
O*E2 0.0.0.0/0 [110/1] via 10.2.5.25, 00:00:15, Tunnel1225
```

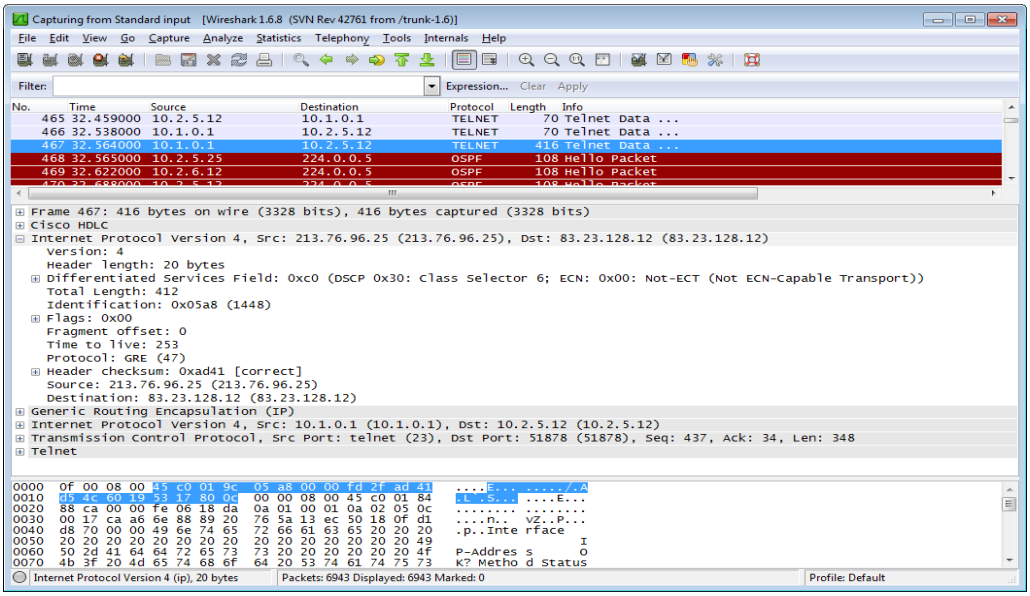
Ostatecznym sprawdzianem będzie nawiązanie połączenia telnetem do Wa-3 znajdującego się w biurze głównym. Dzięki temu będzie pewność, iż połączenie działa poprawnie, gdyż telnet działa w siódmej warstwie modelu ISO:

```
Kr1#telnet 10.1.0.1
Trying 10.1.0.1 ... Open
Unauthorized users prohibited

User Access Verification

Password:
Wa3>enable
Password:
Wa3#
```

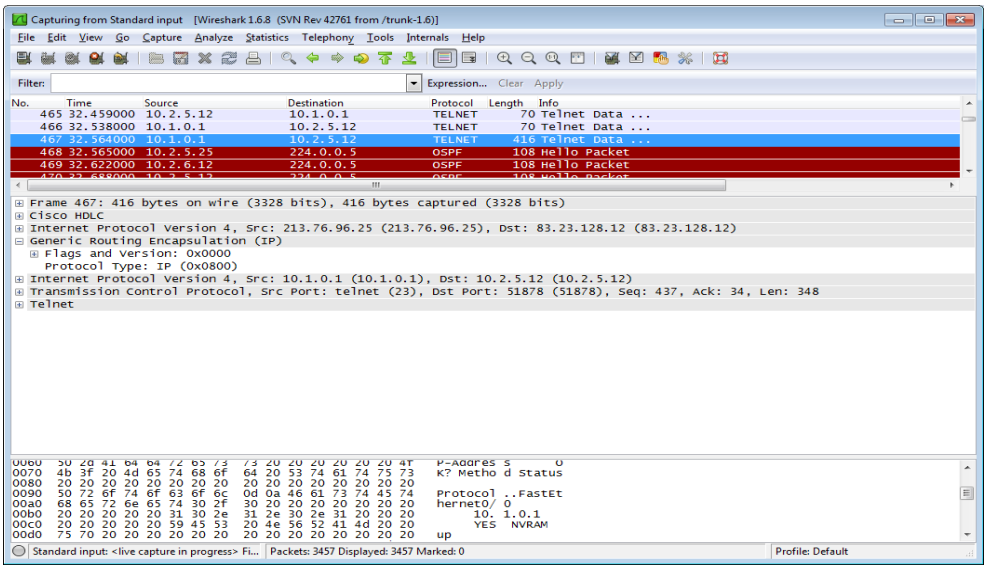
Oglądając przechwycone pakiety sesji w Wireshark wyraźnie widać jak wygląda pakiet.



Oryginalne dane (telnet) enkapsulowane są w TCP. Tak opakowane dane z kolei enkapsulowane są w protokół IP z adresami sieci prywatnych, aby następnie zostać ulec enkapsulacji przez protokół GRE. Do tego zostaje dodany jeszcze nagłówek IP.

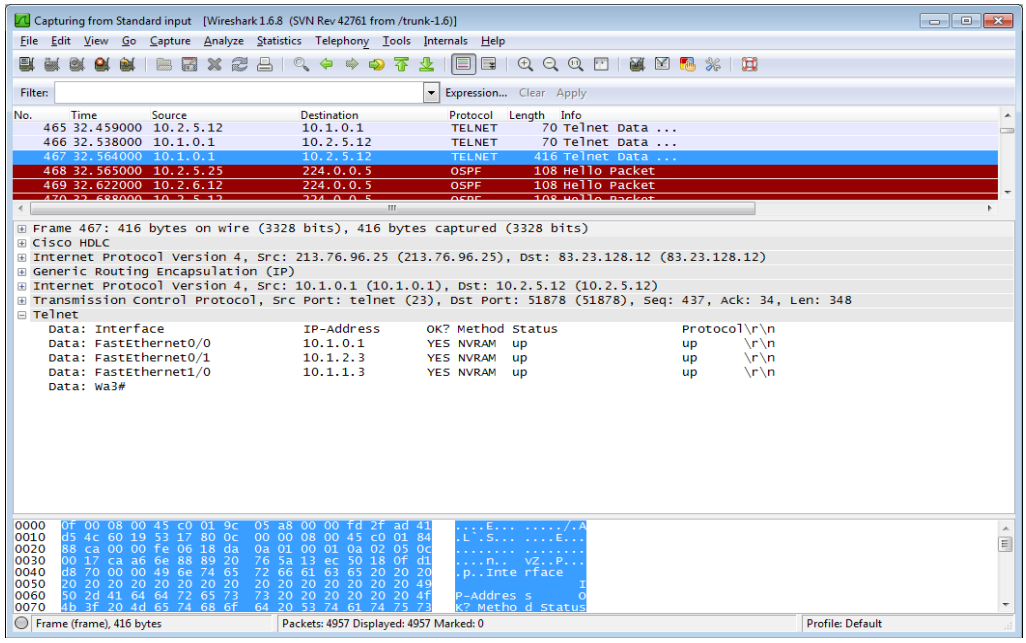


W tym nagłówku w polu protokoł znajduje się liczba 47 odpowiadająca kodowi protokołu GRE. Adres docelowy jak i źródłowy należą do klasy adresów routowalnych, gdyż w innym przypadku nie istniałaby możliwość przetransportowania pakietu przez Internet. Nagłówek protokołu GRE składa się z flag i wersji protokołu oraz jaki protokół przenosi – IP (0x0800). Następny nagłówek IP przenosi informacje o adresach prywatnych.



Następnie oczywiście jest już prawidłowa enkapsulacja. Router pozbywając się nagłówków IP oraz Gre dostrzega nowy adres IP i przesyła go w sieci. W Wireshark wyraźnie widać co zawiera telnet. Są to dane wyświetlone komendą:

```
Wa3#show ip interface brief
```



Po zweryfikowaniu konfiguracji tunelu1225 należy skonfigurować kolejny tunel, tym razem łączący Kr-1 z Wa-2. Konfiguracja jest analogiczna do przedstawionej wyżej, więc nie została zamieszczona. Po prawidłowym skonfigurowaniu tuneli w tabeli routing routera Kr-1 znajdują się wpisy:

```
Gateway of last resort is 10.2.5.25 to network 0.0.0.0

     83.0.0.0/24 is subnetted, 1 subnets
C       83.23.128.0 is directly connected, Serial0/0
     10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA    10.1.3.0/25 [110/11122] via 10.2.6.1, 00:00:03, Tunnel121
                    [110/11122] via 10.2.5.25, 00:16:11, Tunnel1225
O IA    10.1.2.0/24 [110/11177] via 10.2.6.1, 00:00:03, Tunnel121
                    [110/11177] via 10.2.5.25, 00:16:11, Tunnel1225
O IA    10.1.1.0/25 [110/11113] via 10.2.6.1, 00:00:03, Tunnel121
                    [110/11113] via 10.2.5.25, 00:16:14, Tunnel1225
O IA    10.1.0.0/21 [110/11112] via 10.2.6.1, 00:00:06, Tunnel121
                    [110/11112] via 10.2.5.25, 00:16:14, Tunnel1225
C       10.2.5.0/24 is directly connected, Tunnel1225
C       10.2.6.0/24 is directly connected, Tunnel121
O IA    10.1.4.0/24 [110/11186] via 10.2.6.1, 00:00:08, Tunnel121
                    [110/11186] via 10.2.5.25, 00:16:15, Tunnel1225
C       10.1.5.192/26 is directly connected, FastEthernet0/0
     62.0.0.0/32 is subnetted, 1 subnets
S       62.121.120.1 is directly connected, Serial0/0
     213.76.96.0/32 is subnetted, 1 subnets
S       213.76.96.25 is directly connected, Serial0/0
O*E2 0.0.0.0/0 [110/1] via 10.2.5.25, 00:16:15, Tunnel1225
```

Obydwa tunele działają i dodatkowo istnieje balansowanie ruchu pomiędzy tymi tunelami. Głównym celem istnienia 2 tuneli do biura w Warszawie jest redundancja. W dzisiejszych czasach straty spowodowane niedziałającym łączem są nie do zaakceptowania i należy im zapobiec. W chwili obecnej istnieje redundantny tunel do biura głównego i w przypadku awarii jednego z tuneli zadziała drugi.Niestety tzw. route balancing nie jest pożądany, gdyż sprawia trudności chociażby w działaniu firewalli (część sesji idzie przez jeden tunel, a część przez drugi). Aby to naprawić zostanie zmniejszony koszt jednego z tuneli w celu umieszczenia jego tras w tabeli routingu. Na obu routerach wykonujemy polecenia:

```
Kr1(config)#interface tunnel 1225
Kr1(config-if)#ip ospf cost 100
```

W rezultacie koszt ospf dla tras się zmniejszył i tylko trasy rozgłaszane przez tunel 1225 znajdują się w tabeli routingu:

```
     83.0.0.0/24 is subnetted, 1 subnets
C       83.23.128.0 is directly connected, Serial0/0
     10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA    10.1.3.0/25 [110/111] via 10.2.5.25, 00:04:40, Tunnel1225
O IA    10.1.2.0/24 [110/166] via 10.2.5.25, 00:04:40, Tunnel1225
O IA    10.1.1.0/25 [110/102] via 10.2.5.25, 00:04:40, Tunnel1225
O IA    10.1.0.0/21 [110/101] via 10.2.5.25, 00:04:40, Tunnel1225
C       10.2.5.0/24 is directly connected, Tunnel1225
C       10.2.6.0/24 is directly connected, Tunnel121
O IA    10.1.4.0/24 [110/175] via 10.2.5.25, 00:04:42, Tunnel1225
C       10.1.5.192/26 is directly connected, FastEthernet0/0
     62.0.0.0/32 is subnetted, 1 subnets
```

```
S       62.121.120.1 is directly connected, Serial0/0
     213.76.96.0/32 is subnetted, 1 subnets
S       213.76.96.25 is directly connected, Serial0/0
O*E2 0.0.0.0/0 [110/1] via 10.2.5.25, 00:33:16, Tunnel1225
```

Aby połączenie pomiędzy sieciami było pełne i redundantne pozostało tylko zmniejszyć interwały hello i dead w ospf po obu stronach tunelu:

```
Kr1(config)#interface tunnel 1225
Kr1(config-if)#ip ospf dead-interval minimal hello-multiplier 3
```

W celu sprawdzenia jak realizowane jest połączenie, gdy tunel główny zostanie zamknięty, wysłano 100 pakietów icmp zamykając jednocześnie interfejs Wa-1 prowadzący do Internetu.

```
Kr1#ping 10.1.4.4 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 10.1.4.4, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!.....!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 96 percent (96/100), round-trip min/avg/max = 44/118/212 ms
```

Podczas transmisji stracie uległy tylko 4 pakiety icmp, a więc wynik jest naprawdę dobry. Dzieje się tak dlatego, iż przez tunel gre istnieje możliwość przenoszenia protokołu dynamicznego routingu. To dzięki ospf możliwe była taka szybka reakcja, bez stosowania dodatkowych narzędzi. Warto podkreślić, że każdy z protokołów dynamicznego routingu znajduje tu zastosowanie, ale został wybrany ospf ze względu na swoją popularność oraz wsparcie na wszystkich liczących się producentów routerów. Dzięki ospf możliwe stało się jak najszybsze reagowanie na zmianę topologii w sieci i połączenie jest bardziej niezawodne. Oczywiście nie bez znaczenia jest też fakt, iż redundantnego tunelu nie trzeba było zestawiać dzięki czemu również udało się zyskać trochę na czasie.

Konfiguracja tunelu IPinIP

Recursive routing

Przy konfiguracji protokołu dynamicznego routingu w tunelach pojawia się niebezpieczeństwo rekurencyjnego routingu. Zjawisko to polega na tym, że po otrzymaniu tras dostępu, okazuje się, że router do tablicy routingu wstawia nową lepszą trasę prowadzącą na drugi koniec tunelu. Ta trasa prowadzi przez tunel. Jak wiadomo, status interfejsu tunelu zależy od tego, czy pakiety potrafią dotrzeć na jego drugi koniec. Kiedy router odkryje, że nastąpił routing rekursywny zamyka tunel.
Tabela routingu na BizPartner2 wygląda następująco:

```
BizPartner2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-
2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.5.0/24 is directly connected, Loopback2
C       172.16.6.0/24 is directly connected, Loopback0
C       172.16.7.0/24 is directly connected, Loopback1
S       172.16.0.0/22 is directly connected, Tunnel21
     10.0.0.0/24 is subnetted, 2 subnets
C       10.2.4.0 is directly connected, Tunnel21
C       10.1.4.0 is directly connected, Serial0/0
S*   0.0.0.0/0 is directly conn connected, Serial0/0
```

Jak widać jest trasa domyślna kierująca na interfejs połączony z Internetem, oraz trasa statyczna kierująca ruch 172.16.0.0 do tunelu która jest również niezbędna, aby ten ruch był przenoszony przez tunel. Po skonfigurowaniu protokołu ospf do tabeli routingu przybywa wpisów lecz interfejs tunelu zaczyna flapować, czyli zamykać się i otwierać co kilka chwil.

```
*Mar  1 01:04:00.631: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 21: Neighbor 10.2.4.2
(Tunnel21) is down: retry limit exceeded
BizPartner2#
*Mar  1 01:04:03.799: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 21: Neighbor 10.2.4.2
(Tunnel21) is up: new adjacency
BizPartner2#
*Mar  1 01:04:09.895: %TUN-5-RECURDOWN: Tunnel21 temporarily disabled due
to recursive routing
*Mar  1 01:04:10.891: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel21, changed state to down
*Mar  1 01:04:10.927: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 21: Neighbor 10.2.4.2
(Tunnel21) is down: interface down
```

Przyczyną tego stanu rzeczy jest właśnie recursive routing. Odpowiedzialna za to jest trasa, którą dla wyróżnienia zaznaczono czerwonym kolorem:

```
BizPartner2#show ip route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
D       172.16.4.0/22 [90/310044416] via 10.2.4.2, 00:00:01, Tunnel21
C       172.16.5.0/24 is directly connected, Loopback2
C       172.16.6.0/24 is directly connected, Loopback0
C       172.16.7.0/24 is directly connected, Loopback1
D       172.16.0.0/24 [90/297372416] via 10.2.4.2, 00:00:01, Tunnel21
S       172.16.0.0/22 is directly connected, Tunnel21
D       172.16.1.0/24 [90/297372416] via 10.2.4.2, 00:00:01, Tunnel21
D       172.16.2.0/24 [90/297372416] via 10.2.4.2, 00:00:03, Tunnel21
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.1.3.0/30 is directly connected, Serial0/0
C       10.2.4.0/24 is directly connected, Tunnel21
D EX    10.1.3.4/30 [170/297756416] via 10.2.4.2, 00:00:03, Tunnel21
S*   0.0.0.0/0 is directly connected, Serial0/0
```

Routery wybierają najbardziej optymalną trasę dla danego hosta kierując się wpisami tabeli routingu. Odbywa się to w ten sposób, że porównywane są długości pasujących do sieci masek. Router wybierze trasę z najdłuższą maską. Jeżeli przyjrzymy się wpisom staje się oczywiste, iż router kierując ruch do 10.1.2.1.wybierze trasę przez tunel, gdyż do domyślnej trasy pasuje każda maska nawet o długości 0, a do trasy przez tunel pasuje maska 24 bitowa. Router więc zamiast przesyłać ruch do routera na którym znajduje się cel tunelu (BizPartner1), przez interfejs serial 0/0 próbuje przesłać go przez tunel. System operacyjny wykrywając routing rekursywny zamyka interfejs.

Najprostszym rozwiązaniem tego problemu jest ustawienie statycznej trasy do hosta.

```
BizPartner2(config)#ip route 10.1.3.6 255.255.255.255 s0/0
```

Jeżeli jednak ważne jest to, aby tabela routingu była jak najkrótsza, wtedy stosujemy listę dystrybucyjną.

```
BizPartner1(config)#ip prefix-list Z_TUNELU seq 10 deny 10.1.3.4/30
BizPartner1(config)#ip prefix-list Z_TUNELU seq 20 permit 0.0.0.0/0 le 32
```

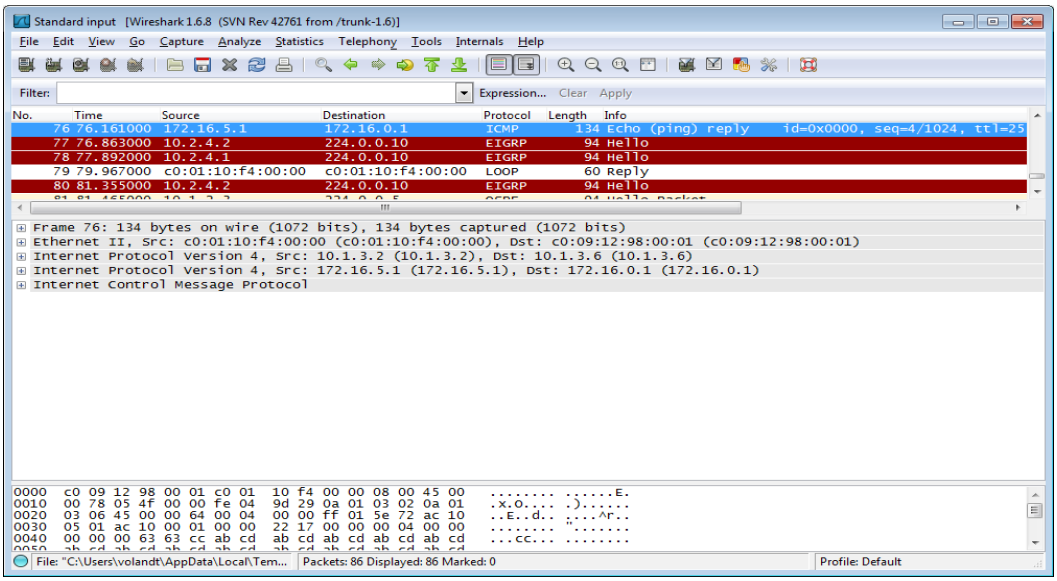
Aby lista dystrybucyjna zaczęła działać należy 'podpiąć' ją pod interfejs, w tym przypadku Tunelu 21 dla ruchu wychodzącego.

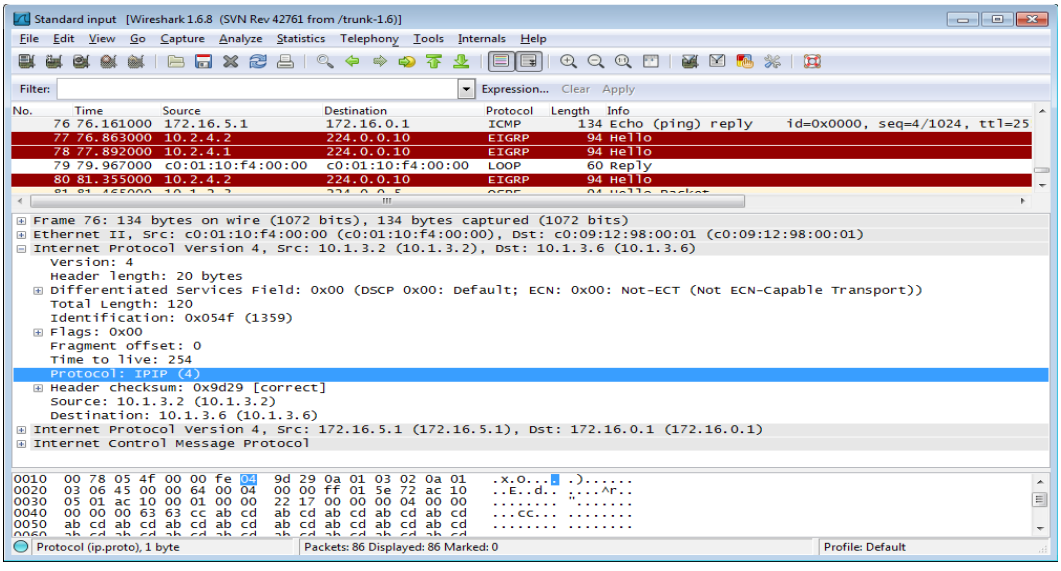
```
BizPartner1(config)#router eigrp 21
BizPartner1(config-router)#distribute-list prefix Z_TUNELU out Tunnel21
```

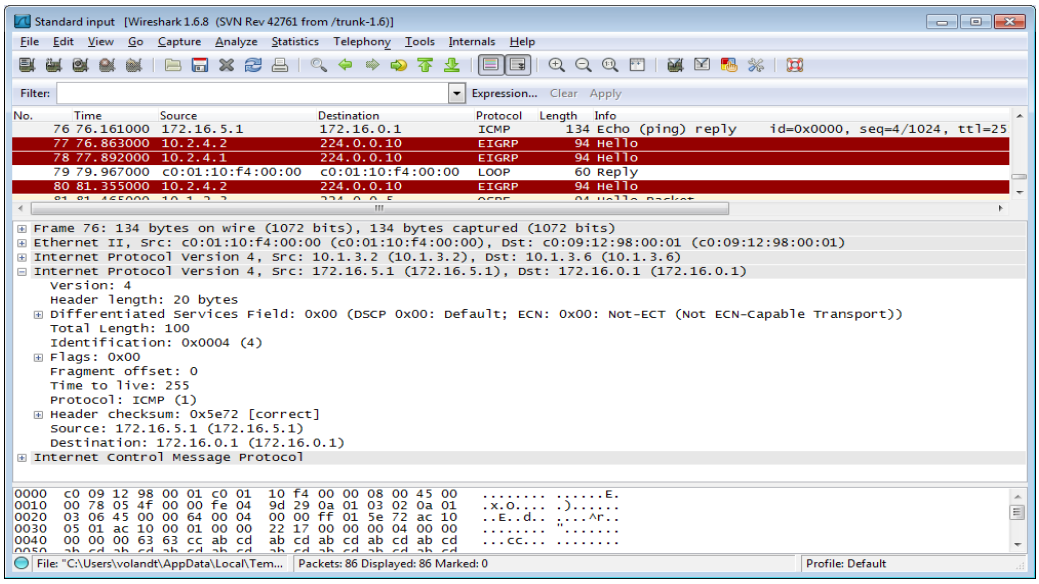
Po wykonaniu poleceń problem rekursywnego routingu zniknął, gdyż lista dystrybucyjna nie dopuszcza do rozgłoszenia trasy 10.1.3.4 z prefixem 30.

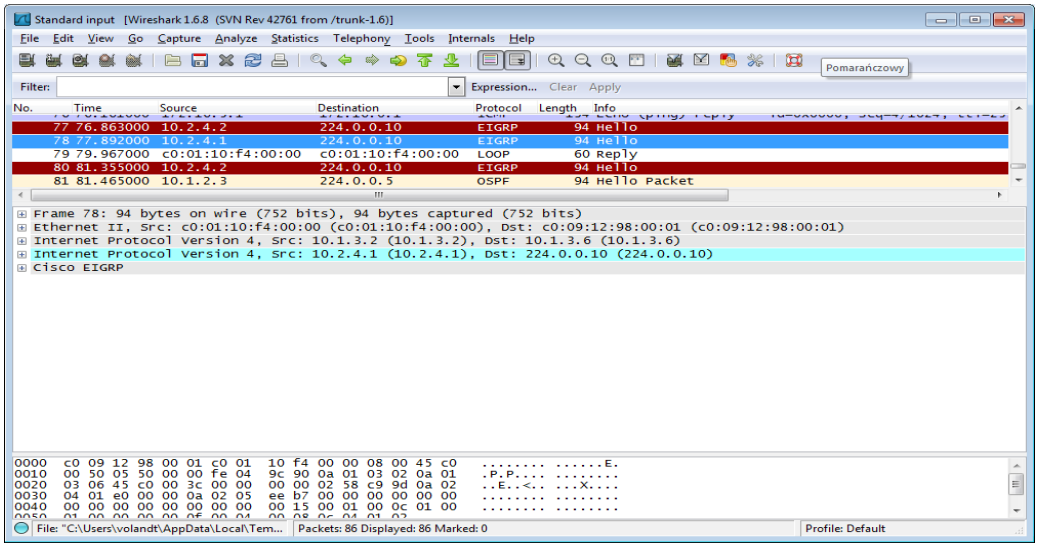
```
BizPartner2#show ip route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
      172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
D        172.16.4.0/22 [90/310044416] via 10.2.4.2, 00:00:16, Tunnel21
C        172.16.5.0/24 is directly connected, Loopback2
C        172.16.6.0/24 is directly connected, Loopback0
C        172.16.7.0/24 is directly connected, Loopback1
D        172.16.0.0/24 [90/297372416] via 10.2.4.2, 00:00:16, Tunnel21
S        172.16.0.0/22 is directly connected, Tunnel21
D        172.16.1.0/24 [90/297372416] via 10.2.4.2, 00:00:16, Tunnel21
D        172.16.2.0/24 [90/297372416] via 10.2.4.2, 00:00:19, Tunnel21
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.3.0/30 is directly connected, Serial0/0
C        10.2.4.0/24 is directly connected, Tunnel21
S*    0.0.0.0/0 is directly connected, Serial0/0
```






Standard input [Wireshark 1.6.8 (SVN Rev 42761 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Length Info
76 76.161000 172.16.5.1 172.16.0.1 ICMP 134 Echo (ping) reply id=0x0000, seq=4/1024, ttl=25
77 76.863000 10.2.4.2 224.0.0.10 EIGRP 94 Hello
78 77.892000 10.2.4.1 224.0.0.10 EIGRP 94 Hello
79 79.967000 c0:01:10:f4:00:00 c0:01:10:f4:00:00 LOOP 60 Reply
80 81.355000 10.2.4.2 224.0.0.10 EIGRP 94 Hello
Frame 76: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits)
Ethernet II, Src: c0:01:10:f4:00:00 (c0:01:10:f4:00:00), Dst: c0:09:12:98:00:01 (c0:09:12:98:00:01)
Internet Protocol Version 4, Src: 10.1.3.2 (10.1.3.2), Dst: 10.1.3.6 (10.1.3.6)
Internet Protocol Version 4, Src: 172.16.5.1 (172.16.5.1), Dst: 172.16.0.1 (172.16.0.1)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 100
Identification: 0x0004 (4)
Flags: 0x00
Fragment offset: 0
Time to live: 255
Protocol: ICMP (1)
Header checksum: 0x5e72 [correct]
Source: 172.16.5.1 (172.16.5.1)
Destination: 172.16.0.1 (172.16.0.1)
Internet Control Message Protocol
0000 c0 09 12 98 00 01 c0 01 10 f4 00 00 08 00 45 00
0010 00 78 05 4f 00 00 fe 04 9d 29 0a 01 03 02 0a 01
0020 03 06 45 00 00 64 00 04 00 00 ff 01 5e 72 ac 10
0030 05 01 ac 10 00 01 00 00 22 17 00 00 00 04 00 00
0040 00 00 00 63 63 cc ab cd ab cd ab cd ab cd ab cd
File: "C:\Users\volandt\AppData\Local\Tem... Packets: 86 Displayed: 86 Marked: 0 Profile: Default


Standard input [Wireshark 1.6.8 (SVN Rev 42761 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Pomarańczowy
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Length Info
77 76.863000 10.2.4.2 224.0.0.10 EIGRP 94 Hello
78 77.892000 10.2.4.1 224.0.0.10 EIGRP 94 Hello
79 79.967000 c0:01:10:f4:00:00 c0:01:10:f4:00:00 LOOP 60 Reply
80 81.355000 10.2.4.2 224.0.0.10 EIGRP 94 Hello
81 81.465000 10.1.2.3 224.0.0.5 OSPF 94 Hello Packet
Frame 78: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: c0:01:10:f4:00:00 (c0:01:10:f4:00:00), Dst: c0:09:12:98:00:01 (c0:09:12:98:00:01)
Internet Protocol Version 4, Src: 10.1.3.2 (10.1.3.2), Dst: 10.1.3.6 (10.1.3.6)
Internet Protocol Version 4, Src: 10.2.4.1 (10.2.4.1), Dst: 224.0.0.10 (224.0.0.10)
Cisco EIGRP
0000 c0 09 12 98 00 01 c0 01 10 f4 00 00 08 00 45 c0
0010 00 50 05 50 00 00 fe 04 9c 90 0a 01 03 02 0a 01
0020 03 06 45 c0 00 3c 00 00 00 00 02 58 c9 9d 0a 02
0030 04 01 e0 00 00 0a 02 05 ee b7 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 15 00 01 00 0c 01 00
File: "C:\Users\volandt\AppData\Local\Tem... Packets: 86 Displayed: 86 Marked: 0 Profile: Default