

Ataki ARP-Spoofing na sieć ethernetową w praktyce

Bartosz Chodorowski

wersja III, 17 października 2010r.

Spis treści

0	Wstęp	2
1	Czego potrzebuję, aby zrozumieć ten artykuł?	2
2	Czym jest protokół ARP? Podstawy sieci ethernetowych	2
3	NAT	4
4	ARP-Spoofing w teorii	6
4.1	„Spoofing symetryczny”	6
4.2	Zalety i wady „symetrycznego spoofingu”	9
4.3	„Spoofing z natowaniem”	9
5	Ataki w praktyce	10
5.1	„Spoofing symetryczny” w praktyce	10
5.2	„Spoofing z natowaniem” w praktyce	11
6	Zakończenie	12

0 Wstęp

Artykuł opisuje techniczną stronę przeprowadzenia ataków typu ARP-Spoofing. Użycie go może prowadzić do podsłuchiwania innych członków sieci. Jeżeli nie masz na to ich zgody, **jest to przestępstwem!** Mój artykuł przekazuje wiedzę na temat tych ataków, to jak ją spożytkujesz zależy od Ciebie. I pamiętaj: za użycie tej wiedzy niezgodnie z prawem odpowiadać będziesz tylko i wyłącznie Ty!

Adresy MAC użyte w tym artykule zostały wymyślane, wszelaka zbieżność jest przypadkowa. :)

1 Czego potrzebuję, aby zrozumieć ten artykuł?

Potrzebujesz dobrego systemu operacyjnego (innego niż Ms Windows), najlepiej opartego o jądro Linuksa, gdyż używać będę go przy omawianiu, jak można wykonać atak w praktyce.

Potrzebujesz również oprogramowania:

- iptables – filtr pakietów w Linuksie, może się przydać :)
- libpcap – podstawowa biblioteka, z której korzystają poniższe programy
- tcpdump – sniffer
- wireshark – sniffer z ładnym interfejsem graficznym (polecam!)
- ethutils – mój własny projekt, kilka przydatnych programów dostępnych pod adresem: <http://chomzee.gct-game.net/ethutils/>
- gcc, make – do kompilacji ethutils

2 Czym jest protokół ARP? Podstawy sieci ethernetowych

Rozpatrzmy przykładową, prostą sieć ethernetową. Niech będzie to kilka komputerów połączonych z przełącznikiem sieciowym (switchem). Każdy

komputer, a raczej każda karta sieciowa, posiada swój własny adres sprzętowy MAC. Tradycyjnie zapisuje się go jako sześć liczb, każda zapisana dwoma cyframi szesnastkowymi, oddzielonych dwukropkiem (np. 00:0E:6E:8C:89:17). Każdy pakiet wysłany w sieć musi być „zaadresowany” do konkretnej karty sieciowej, czyli musi zawierać MAC karty sieciowej, do której ma dotrzeć. To właśnie na podstawie adresów MAC switch decyduje, na który port wysłać pakiet. Czasami istnieje potrzeba wysłania pakietu do wszystkich komputerów w sieci – należy wówczas wysłać pakiet na adres FF:FF:FF:FF:FF:FF.

Wielu z Was pewnie powie: „Ale ja nigdy nie wysyłałem danych podając adres MAC. Chcąc połączyć się z jakimś komputerem w sieci, wystarczy znać jego adres IP”. Czym jest owy adres IP, z którym spotykamy się o wiele częściej? Adres IP jest tworem z wyższej warstwy, z innego protokołu. Również (przynajmniej w teorii) przypisywany jest jeden na jedną kartę sieciową.

Rodzi się kolejne pytanie: „Po co komu dwa protokoły, które odpowiadają za adresowanie komputerów? Nie wystarczy jeden?”. Otóż nie wystarczy. Trzeba mieć świadomość, że protokół IP był zaprojektowany tak, aby działał niezależnie od architektury sieci (tj. ethernetu). Idea IP zadziała więc tak samo dobrze w sieciach ethernet, token ring i innych dziadostwach, które strasznie rzadko można spotkać w dzisiejszych czasach.

Trzeba również mieć małe pojęcie o enkapsulacji protokołów sieciowych:

Ethernet

```
|
+ ARP
|
+ IP
|
+ TCP
|
+ UDP
|
+ ICMP
```

Jeżeli weźmiemy prosty pakiet, niech będzie to powiedzmy prosty pakiet UDP, zawierający dane. Dane są „opakowane” w nagłówek protokołu UDP. Całość opakowywana jest w protokół IP, a żeby pakiet IP mógł polecieć w sieć ethernetową, zostaje opakowany w protokół Ethernet. Proste i intuicyjne.

Popatrzmy, jak działa to w praktyce. Komputer o adresie IP 192.168.0.1 chce nawiązać połączenie z komputerem o adresie IP 192.168.0.2. Do ko-

munikacji z tym komputerem potrzebuje więc jego adresu IP oraz adresu MAC.

W tym momencie wkracza do akcji protokół ARP. Skąd 192.168.0.1 ma wiedzieć, jaki adres MAC ma karta sieciowa o adresie 192.168.0.2? Komputer chcący nawiązać połączenie wysyła pakiet ARP-Request (zapytanie) o treści mniej-więcej: „Tutaj 192.168.0.1 (00:0E:6E:8C:89:17). Kto ma adres 192.168.0.2?”. Takie dane zawiera w sobie pakiet już na poziomie protokołu ARP. Na poziomie Ethernet pakiet zostaje adresowany na adres rozgłoszeniowy, tj. FF:FF:FF:FF:FF:FF. Taki pakiet dostaje każdy komputer w sieci. Jeżeli 192.168.0.2 odbierze ten pakiet, odsyła odpowiedź (ARP-Reply), ale już nie na adres rozgłoszeniowy, tylko bezpośrednio do komputera, który pytał o niego. Coś na kształt: „Ja jestem 192.168.0.2 i mam adres MAC 00:F8:8B:8C:6E:01”. 192.168.0.1 zapisuje sobie w specjalnym miejscu (tablica ARP systemu), że adresowi IP 192.168.0.2 odpowiada adres MAC 00:F8:8B:8C:6E:01. Można wysłać już do niego pakiet zawierający więcej informacji.

Jeszcze jeden istotny wniosek: pakiet ARP **nie jest** pakietem IP.

Podsumowując – protokół ARP odpowiada za zamianę (kojarzenie) adresów IP na adresy MAC, aby możliwa była komunikacja w sieci ethernetowej.

Jeśli się gubisz, uruchom wireshark i poobserwuj, jak wygląda ruch w Twojej sieci ethernetowej. Możesz ustawić filtr na „arp”, aby skupić się na budowie pakietów ARP. Zwróć uwagę na to, jak różne protokoły współgrają ze sobą. Nie nauczysz się niczego, dopóki sam na własne oczy nie zobaczysz tych pakietów. :)

3 NAT

„Network Address Translation” to technika translacji adresów sieciowych. Jest powszechnie używana w sieciach osiedlowych, firmowych itp. Celem jest „dostarczenie Internetu” tak, aby komputery w sieci lokalnej mogły łączyć się z siecią globalną. Przyjmijmy, że mamy sieć złożoną z kilkunastu komputerów. Ale łączy do Internetu i adres IP „zewnętrzny” jest tylko jeden, czyli, bez użycia technologii NAT, tylko jedna maszyna ma połączenie z internetem.

Jak to się dzieje, że w sieciach osiedlowych każdy komputer może łączyć się z Internetem? Jeden z komputerów (zwany routerem) posiada dwie karty sieciowe – jedną połączoną z siecią LAN i drugą połączoną zazwyczaj

z modemem, dzięki któremu router ma dostęp do Internetu.

Rozważmy taki schemat:

ROUTER:	IP	MAC
eth0	192.168.0.1	00:12:34:56:78:9A
eth1	80.12.34.56	00:A5:87:20:BA:02

KLIENT:	IP	MAC
eth0	192.168.0.x	00:F8:8B:8C:6E:01

Przyjmujemy, że adres IP zewnętrzny routera to 80.12.34.56, a adres wewnętrzny to 192.168.0.1 (oczywiste jest, że router ma dwie karty sieciowe połączone do różnych sieci, więc ma dwa adresy IP). Wszystkie komputery w sieci lokalnej o adresach 192.168.0.x wiedzą, że 192.168.0.1 jest tzw. „bramą domyślną”. Tak więc, chcąc nawiązać połączenie z google.com (64.233.167.99), wysyłają pakiet o następujących parametrach:

MAC NADAWCY:	00:F8:8B:8C:6E:01
MAC ODBIORCY:	00:12:34:56:78:9A
IP NADAWCY:	192.168.0.x
IP ODBIORCY:	64.233.167.99

Router jest w miarę inteligentny, i domyśla się, że powinien przeroutować ten pakiet. Stosuje więc NAT (zamienia adres IP nadawcy na swój własny (80.12.34.56)), wysyłając pakiet do Sieci Globalnej.

MAC NADAWCY:	00:A5:87:20:BA:02
MAC ODBIORCY:	(MAC modemu)
IP NADAWCY:	80.12.34.56
IP ODBIORCY:	64.233.167.99

Mało tego – router jest na tyle inteligentny, że pamięta wszystkie połączenia, które utrzymuje. Prawdopodobnie więc, router otrzyma odpowiedź od google, która jest kierowana do niego, czyli odpowiedź można opisać tak:

MAC NADAWCY:	(MAC modemu)
MAC ODBIORCY:	00:A5:87:20:BA:02
IP NADAWCY:	64.233.167.99
IP ODBIORCY:	80.12.34.56

Router wie jednak, że ten pakiet powinien odesłać do 192.168.0.x do sieci lokalnej tak, aby klient w ogóle nie miał świadomości, jakie jaja się dzieją:

```
MAC NADAWCY:    00:12:34:56:78:9A
MAC ODBIORCY:    00:F8:8B:8C:6E:01
IP NADAWCY:      64.233.167.99
IP ODBIORCY:     192.168.0.x
```

4 ARP-Spoofing w teorii

Protokoły Ethernet i ARP były projektowane z myślą o tym, że każdy komputer w sieci będzie „grzeczny”, to znaczy będzie zachowywał się zgodnie ze standardem. Pojawiają się problemy, gdy ktoś chce celowo zamieszać w sieci. Udaje mu się to zrobić bez trudu – może wysłać do sieci dowolny pakiet, sfałszować zapytania lub odpowiedzi ARP, kontrolując tym samym ruch w całej sieci.

Nie jest bowiem problemem wysłać **dowolny** pakiet w sieć ethernetową (mamy kontrolę nad wszystkimi protokołami, łącznie z Ethernetem). Możemy podszyć się pod inny adres IP robiąc mniej lub bardziej szkodliwe i złe rzeczy.

Niech 192.168.0.1 będzie routerem, 192.168.0.2 naszą ofiarą, na której chcemy przeprowadzić atak. My istniejemy w sieci pod adresem 192.168.0.3. Switch uniemożliwia nam podsłuchiwanie pakietów, które przesyłane są pomiędzy ofiarą a routerem.

4.1 „Spoofing symetryczny”

Każdy z komputerów, jak już wspomniałem, posiada miejsce w pamięci, w którym zapisuje adresy ARP komputerów, z jakimi się komunikuje. Jeżeli 192.168.0.2 oraz 192.168.0.3 korzystają z internetu poprzez bramę 192.168.0.1, to ich „tablice ARP” przedstawmy w taki symboliczny sposób:

```
# Router
192.168.0.1 (00:12:34:56:78:9A)
- 192.168.0.2 is at 00:F8:8B:8C:6E:01
- 192.168.0.3 is at 00:16:05:A1:66:73
```

```
# Ofiara
192.168.0.2 (00:F8:8B:8C:6E:01)
- 192.168.0.1 is at 00:12:34:56:78:9A
```

```
# Atakujący
192.168.0.3 (00:16:05:A1:66:73)
- 192.168.0.1 is at 00:12:34:56:78:9A
```

Teraz chcemy tak ładnie zamieszać ofercie i routerowi w głowach (tablicach ARP), aby pakiety przechodziły sobie przez maszynę atakującego. Jak to zrobić?

Pierwszą rzeczą, która powinna wpaść Ci na myśl jest mniej więcej taka: „Trzeba poczekać, aż 192.168.0.2 zapyta o adres 192.168.0.1. Wtedy wyślemy sfalszowaną odpowiedź, podając jakoby 192.168.0.1 miało naszego MACa. System 192.168.0.2 będzie do nas wysyłał pakiety.”

Masz rację, możesz tak zrobić. Ale można to zrobić prościej. Okazuje się, że większości (o ile nie wszystkim) systemom można zmienić MACa w tablicy ARPów, wysyłając po prostu ARP-Reply, nawet jeżeli ten nie wysyłał ARP-Request. Przyznam się, że nie wiem jak formalnie powinna wyglądać implementacja protokołu ARP, jednak z doświadczenia wiem, że niektóre systemy czasem nie są do końca potulne, aby ot tak zmienić im ARPa. Jednak bombardowanie (wysyłanie chociaż 1 ARP-Reply na sekundę) na pewno prędzej czy później zmieni wpis w tabeli ARP systemu. Wniosek: wysyłając sfalszowane ARP-Request mamy władzę nad dynamicznymi tabelami ARP wszystkich komputerów w sieci.

Dobra, teraz już możemy przeprowadzić atak według planu:

- Wysyłamy do ofiary ARP-Reply: 192.168.0.1 is at 00:16:05:A1:66:73
- Wysyłamy do routera ARP-Reply: 192.168.0.2 is at 00:16:05:A1:66:73

Sytuacja wygląda teraz tak:

```
# Router
192.168.0.1 (00:12:34:56:78:9A)
- 192.168.0.2 is at 00:16:05:A1:66:73
- 192.168.0.3 is at 00:16:05:A1:66:73
```

```
# Ofiara
192.168.0.2 (00:F8:8B:8C:6E:01)
```

- 192.168.0.1 is at 00:16:05:A1:66:73

Atakujący

192.168.0.3 (00:16:05:A1:66:73)

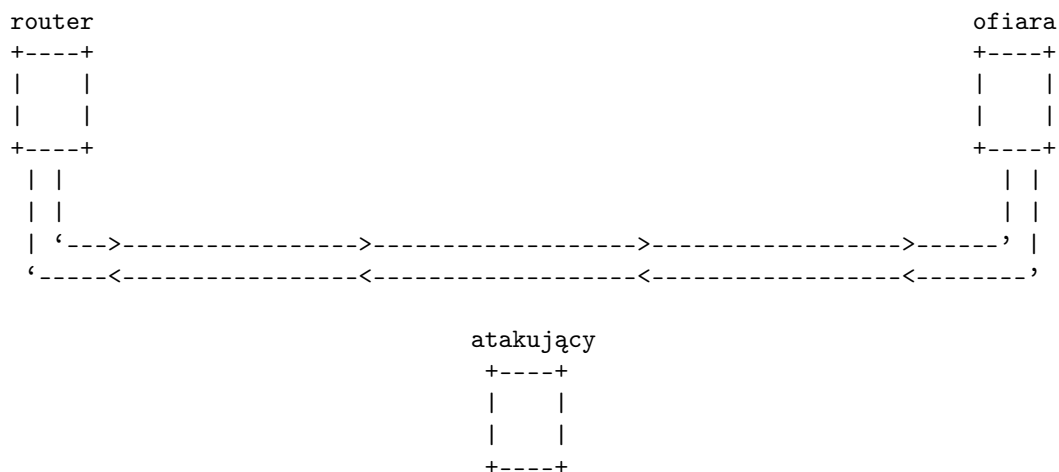
- 192.168.0.1 is at 00:12:34:56:78:9A

Wszystkie pakiety, które ofiara będzie kierować do Internetu polecą nie do routera lecz do komputera atakującego i tam... zginą. Potrzebujemy jeszcze pewnego programu, który pakiety od 192.168.0.2 (00:F8:8B:8C:6E:01) do 192.168.0.1 (00:16:05:A1:66:73) odbije z powrotem w sieć, zmieniając MAC adresata i odbiorcy na: 00:16:05:A1:66:73 do 00:12:34:56:78:9A. Taki pakiet doleci do routera. U niego wszystko będzie się zgadzać, bo router myśli, że 192.168.0.2 to 00:16:05:A1:66:73. Pakiet zostanie więc preroutowany do Internetu. Router wyśle odpowiedź od 192.168.0.1 (00:12:34:56:78:9A) do 192.168.0.2 (00:16:05:A1:66:73). Nasz programik powinien również te pakiety wylapywać, odbijać z powrotem w sieć, zmieniając adresata i odbiorcę na: 192.168.0.1 (00:16:05:A1:66:73) do 192.168.0.2 (00:F8:8B:8C:6E:01).

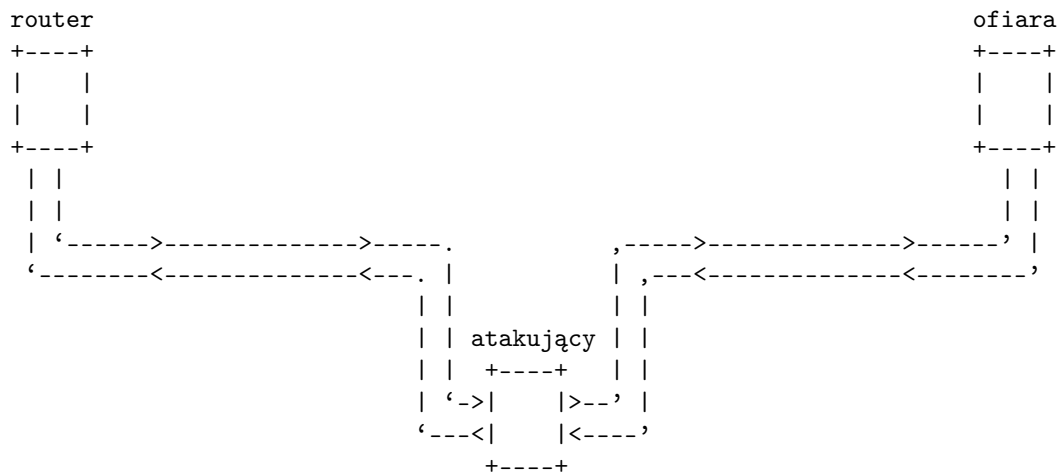
Podsumowując, router myśli, że my to ofiara, a ofiara myśli, że my to router. Odpowiednio odbijając odpowiednie pakiety – internet będzie działał, a co najważniejsze, atakujący będzie w stanie podsłuchać pakiety, które przez niego przechodzą.

Może kilka rysunków dla podsumowania:

Sytuacja normalna:



Sytuacja podczas symetrycznego ataku ARP-Spoofing:



UWAGA:

Wykonując symetryczny atak ARP-Spoofing nie musimy podawać SWOJEGO adresu MAC. Możemy użyć jakiegoś zupełnie innego, wymyślnego adresu – switche w sieci będą pilnowały, żeby pakiety kierowane na sfalszowany adres leciały do Ciebie, jednak zarówno ofiara jak i router nie będą miały wówczas pojęcia, skąd wykonano atak. Wykrycie takiego ataku może przysporzyć wiele trudności.

4.2 Zalety i wady „symetrycznego spoofingu”

Podstawową zaletą dobrze przeprowadzonego ataku (używanie sfałszowanego MACa do odbijania pakietów w sieć) jest fakt, że wykrycie która maszyna w sieci wykonuje ten atak jest bardzo trudne.

Wadą jest fakt, że w prosty sposób można ustrzec się przed atakiem tego typu. Wystarczy, że na routerze zastosuje się „statyczną tablicę ARP” – wystarczy, że router w swojej bazie danych będzie miał przypisane na sztywno adresy MAC z adresami IP. Jeżeli coś się nie zgadza, pakiet nie zostanie puszczony dalej.

4.3 „Spoofing z natowaniem”

Wydawałoby się, że statyczne MACi na routerze rzeczywiście popsują nam zabawę. Jednak pomimo takiego zabezpieczenia, atak ARP-Spoofing jest

możliwy do wykonania.

Statyczna tablica ARP jest często stosowana w routerach. Jednak praktycznie nigdzie nie myśli się o tym, aby takie rozwiązanie wprowadzić dla klientów. Nie mam nawet pojęcia, jak w systemach Windows można takie coś zaimplementować. Czy możemy więc wykonać ARP-Spoofing modyfikując jedynie tablicę ARP ofiary? Oczywiście że tak!

Wystarczy, że atakujący na swojej maszynie postawi NAT. Teraz tylko wystarczy powiedzieć ofierze, że 192.168.0.1 jest pod naszym adresem MAC. Ofiara będzie łączyła się z nami, myśląc, że my to 192.168.0.1. Pakiety te będą przez nas NATowane do prawdziwego 192.168.0.1 i dalej do internetu.

Wadą takiego rozwiązania jest fakt, że możliwość wykrycia znacząco wzrasta - pakiety są jawnie routowane przez naszego własnego MACa (nie możemy go zmienić, bo router zablokuje nasze pakiety). No, chyba że użyjemy pary MAC/IP jeszcze jakiegoś innego, postronnego komputera, który w chwili ataku jest wyłączony.

5 Ataki w praktyce

Aby wykonać oba ataki, należy zainstalować programy arpsnd z pakietu ethutls (link podany na początku artykułu). Do wykonania „symetrycznego spoofingu” wykorzystamy jeszcze programu ippong oraz skryptu spoofset (również z pakietu ethutls). Do wykonania „spoofingu z natowaniem” potrzebujemy natomiast działające iptables. Jeżeli nie pisałeś sam nigdy firewalli w iptables, naucz się więcej na ten temat, zanim będziesz próbował wykonać ten atak.

5.1 „Spoofing symetryczny” w praktyce

Należy zainstalować programy arpsnd oraz ippong (za pomocą `make install` z poziomu `root'a`).

Następnie, rozpakowujemy `spoofset-x.x.tar.gz`, gdzie `x.x` to numer wersji. W środku znajduje się pobieżny plik README, który opisuje właśnie ten atak. Oto najważniejszy fragment pliku README z wersji 1.1:

```
Krokiem pierwszym jest znalezienie danych koniecznych do przeprowadzenia
ataku. Potrzebujemy:
- nasz własny adres MAC
```

- adres IP hosta (najczęściej bramy domyślnej), z którym łączy się ofiara
- adres MAC hosta (najczęściej bramy domyślnej), z którym łączy się ofiara
- adres IP ofiary
- adres MAC ofiary

Wszystkie te dane wprowadzamy do pliku konfiguracyjnego 'definicje', korzystając ze swojego ulubionego edytora.

Krokiem kolejnym jest uruchomienie skryptów skrypt_pong1 oraz skrypt_pong2. Oba programy muszą działać przez cały czas trwania ataku arp spoofing. Teraz wystarczy uruchomić skrypt_arp, aby wysłać sfałszowane arpy. Jeżeli skrypty pong1 i pong2 wypisują na standardowe wyjście kropki, znaczy, że pakiety lecą przez naszą maszynę. Jeżeli kropek nie zauważamy, należy kilkakrotnie uruchomić skrypt_arp, aby atakowane hosty dokonały zmian w swoich tablicach arp. Skrypt o nazwie skrypt_arp_sleep wysyła automatycznie sfałszowane arpy co 5 sekund - ten skrypt może okazać się przydatny.

Gdy już pakiety lecą przez naszą maszynę, a skrypt pong1 i pong2 wypływają kropki, wystarczy uruchomić nasz ulubiony sniffer (tcpdump, wireshark etc.) i przechwytywać pakiety.

Aby posprzątać zamieszanie związane z atakiem arp spoofing, należy uruchomić skrypt skrypt_napraw. Wysyła on do obu zainteresowanych stron poprawne adresy MAC, przez co pakiety nie lecą już przez naszą maszynę. Kilukrotne uruchomienie skryptu napraw może okazać koniecznością (do czasu, gdy skryptu pong1 i pong2 przestaną wypływać kropki na standardowe wyjście). Teraz już można wyłączyć skrypty pong1 i pong2.

5.2 „Spoofing z natowaniem” w praktyce

Jako atakujący, przygotujmy najpierw firewall na naszym komputerze. Oto przykładowy firewall, który dobrze wykonuje swoją pracę:

```
iptables -F
iptables -F -t nat
iptables -X -t nat
iptables -F -t filter
iptables -X -t filter
iptables -F -t mangle
iptables -X -t mangle

iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```

iptables -A INPUT -i lo -j ACCEPT

echo "1" > /proc/sys/net/ipv4/ip_forward

iffrom="eth0" # zamień to na interfejs, którym uzyskujesz dostęp do sieci
from="192.168.0.0/255.255.255.0" # zamień to na adres Twojej sieci

ifto="eth0" # zamień to na interfejs, którym uzyskujesz dostęp do sieci
            # (powinien być taki sam co w $iffrom)
to="0/0"

iptables -A FORWARD -i $iffrom -o $ifto -s $from -d $to -j ACCEPT
iptables -A FORWARD -i $ifto -o $iffrom -s $to -d $from -j ACCEPT

# maskarada
iptables -t nat -A POSTROUTING -s $from -d $to -j MASQUERADE

# stealth nat (aby zatrzeć najbardziej widoczne ślady)
iptables -t mangle -A PREROUTING -j TTL --ttl-inc 1

```

I tyle... Teraz już możesz być routerem dla każdej osoby z Twojej sieci. Wystarczy tylko programem arpsnd wysłać ARP-Reply do ofiary z informacją, jakoby IP prawdziwego routera było skojarzone z naszym MACiem. Jeżeli nie działa, wysyłamy ARPy, aż zadziała. :)

6 Zakończenie

Bardzo gorąco zachęcam do przestudiowania źródeł programików z pakietu ethutils. Na podstawie ethkilla lub arpsnd można zbudować własne programy, które wysyłają własne pakiety. Wgłębiając się bardziej w programy ippong lub sniffclients uzyskamy szkielet prostego sniffera.

Mam nadzieję, że ten dokument przydał się komukolwiek.

Bartosz Chodorowski <chomzee@gct-game.net>