W A R S Z A W S K A

W Y Ż S Z A S Z K O Ł A I N F O R M A T Y K I

P R A C A D Y P L O M O W A

S T U D I A P O D Y P L O M O W E

mgr inż. Leszek IGNATOWICZ

Numer albumu 110/EFS

Analizator Wireshark w procesie wykrywania zagrożeń bezpieczeństwa sieci teleinformatycznej

Promotor:

dr inż. Krzysztof RÓŻANOWSKI

W A R S Z A W A 2008

Spis treści:

1. Wprowadzenie

Zapewnienie bezpieczeństwa sieci teleinformatycznej jest jednym z najważniejszych zadań administratora sieci. Zadowalające wywiązanie się z tego zadania wymaga potraktowania bezpieczeństwa jako procesu, a nie jako ustalonego, niezmiennego stanu. Proces ten składa z kilku podprocesów, miedzy innymi z monitorowania bezpieczeństwa sieci teleinformatycznej.

1.1. Proces bezpieczeństwa sieci teleinformatycznej

Proces utrzymania bezpieczeństwa sieci teleinformatycznej, przy założonym akceptowalnym poziomie ryzyka składa się z czterech podprocesów (procesów składowych):

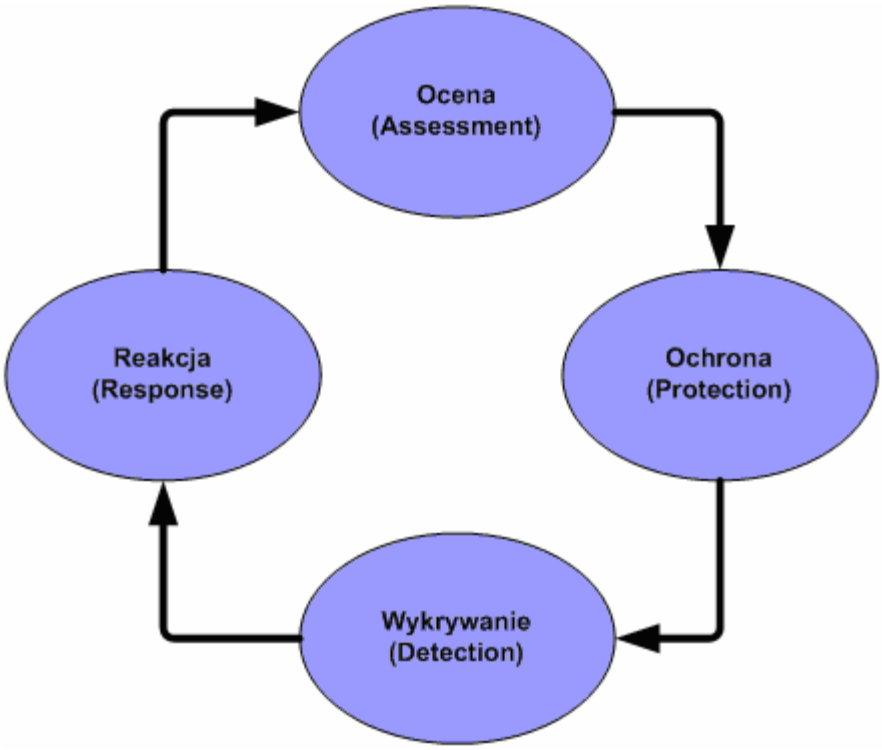
A. Ocena (ang. Assessment) – proces ten obejmuje odniesienie do polityki bezpieczeństwa, ustalenie obowiązujących zasad bezpieczeństwa (ang. policies), procedur, regulacji prawnych oraz innych funkcji dotyczących pozostałych podprocesów.

B. Ochrona (ang. Protection) – traktowana nie tylko jako pożądany stan, ale proces stosowania odpowiednich środków zabezpieczających przed skompromitowaniem sieci teleinformatycznej.

C. Wykrywanie (ang. Detection) – proces identyfikacji (monitorowania) naruszeń zasad bezpieczeństwa oraz innych nieautoryzowanych działań w sieci teleinformatycznej – temat ten będzie rozwinięty w dalszej części niniejszej pracy.

D. Reakcja (ang. Response) – obejmuje działania wynikające z informacji uzyskanych w poprzednim podprocesie, które dotyczą utrzymania skutecznego działania środków ochrony zastosowanych w procesie B oraz likwidację skutków naruszenia bezpieczeństwa sieci [2].

Proces bezpieczeństwa sieci teleinformatycznej można zilustrować graficznie w sposób pokazany na rysunku 1. Warto zwrócić uwagę, że podprocesy pokazane na tym rysunku tworzą zamknięty cykl, co obrazuje ciągłość procesu utrzymania bezpieczeństwa sieci teleinformatycznej (po Reakcji następuje powrót do Oceny [6]).



Rys. 1. Proces bezpieczeństwa sieci teleinformatycznej [2].

1.2. Monitorowanie bezpieczeństwa sieci teleinformatycznej

Monitorowanie bezpieczeństwa sieci teleinformatycznej (ang. network security monitoring – NSM) obejmuje gromadzenie różnorodnych danych o zdarzeniach w sieci, a następnie ich filtrowanie, korelowanie i analizowanie w celu wykrycia i odpowiedniej reakcji na incydenty naruszenia ochrony sieci. Proces monitorowania – wykrywania naruszeń i zagrożeń bezpieczeństwa sieci wykracza poza to, co oferują systemy wykrywania włamań (ang. Intrusion Detection System – IDS), które na ogół poprawnie wykrywają uprzednio zdefiniowane (na przykład przez odpowiednie sygnatury) ataki [6]. Przy użyciu odpowiedniego oprogramowania można wykrywać bardzo różne anomalie ruchu sieciowego, które podane dogłębnej analizie przez administratora sieci umożliwiają wyrycie nieznanych ataków i zagrożeń bezpieczeństwa. Mogą to być ataki wykorzystujące nie wykryte podatności systemów operacyjnych, oprogramowania lub sprzętu sieciowego (na przykład tzw. zero-day exploits). Wykorzystanie monitorowania sieci jest istotnym elementem procesu utrzymania bezpieczeństwa sieci teleinformatycznej.

1.3. Wykrywanie zagrożeń bezpieczeństwa sieci teleinformatycznej

Wykrywanie zagrożeń bezpieczeństwa jest procesem przechwytywania ruchu sieciowego, jego identyfikacji, walidacji oraz eskalacji zdarzeń odbiegających od normy. Przechwytywanie polega na użyciu odpowiednich sensorów (może to być karta sieciowa oraz tzw. sterownik przechwytywania – ang. capture driver) w celu wychwycenia i zapisania pakietów w sieci dla celów ich analizy i archiwizacji. W fazie identyfikacji administrator dzieli przechwycony ruch sieciowy na trzy grupy: normalny, podejrzany oraz wskazujący na działania złośliwe. Potem następuje walidacja w celu określenia kategorii incydentu, z czego wynikają wskazania i ostrzeżenia. Ostatnią fazą jest eskalacja incydentu, w celu podjęcia decyzji mających na celu likwidację jego skutków [6].

2. Charakterystyka analizatorów sieci (snifferów)

2.1. Analiza sieci i sniffing

Analiza sieci (ang. network analysis) polega na przechwyceniu ruchu w sieci oraz określeniu na podstawie analizy zapisanych pakietów tworzących ten ruch jakie zdarzenie zaistniało w sieci. Używa się w tym celu analizatorów sieci (ang. network analyzers), które potrafią nie tylko przechwycić ruch sieciowy, ale także zdekodować pakiety typowych protokołów sieciowych i wyświetlić je w formie czytelnej dla człowieka. Następnym krokiem jest szczegółowa analiza tak uzyskanych wyników, która umożliwia doświadczonemu administratorowi sieci wykrycie zdarzeń stanowiących zagrożenie bezpieczeństwa sieci teleinformatycznej.

Termin analiza sieci jest używany zamiennie z kilkoma innymi:

- sniffing, co oznacza podsłuch ruchu (pakietów) w sieci – w przeszłości miał negatywne konotacje, bowiem oznaczał podsłuch w sieci wykorzystywany przez napastników w złośliwych celach
- analiza ruchu sieciowego (ang. traffic analysis)
- analiza pakietów (ang. packet analysis)
- analiza protokołów (ang. protocol analysis)

Warto zaznaczyć, że terminy analiza sieci, a zwłaszcza sniffing, w szerokim sensie, są używane obecnie najczęściej [4]. W niniejszej pracy oba wyżej wymienione terminy używane są jako synonimy.

2.2. Charakterystyka najczęściej spotykanych analizatorów sieci (snifferów)

Analizator sieci może być samodzielnym urządzeniem wyposażonym w wyspecjalizowane oprogramowanie lub programowym analizatorem zainstalowanym na komputerze stacjonarnym, czy też laptopie. Dostępne są programowe analizatory sieci (sniffery) komercyjne oraz bezpłatne, często jako Open Source Software, GNU GPL.

„Sniffer jest to program komputerowy, którego zadaniem jest przechwytywanie i ewentualne analizowanie danych przepływających w sieci. Wspólną cechą wielu takich analizatorów jest przełączenie karty sieciowej w tryb promiscuous, w którym urządzenie odbiera wszystkie ramki z sieci, także te nie adresowane bezpośrednio do niego; sniffery mogą jednak być uruchamiane także na routerze lub na komputerze będącym jedną ze stron komunikacji sieciowej - i w tych przypadkach, tryb promiscuous nie jest konieczny”. [1]

Sniffery różnią się miedzy sobą takimi cechami jak liczba obsługiwanych protokółów, interfejsem, możliwościami graficznej i statystycznej prezentacji przechwyconych pakietów, a także jakością dekodowania pakietów i zawansowanymi możliwościami ich analizy [4].

2.3. Zastosowanie analizatorów sieci jako narzędzi administratorskich

Analizatory sieci są używane jako narzędzia diagnostyczne przez administratorów sieci oraz administratorów bezpieczeństwa, a nawet czasami przez programistów. Są one bardzo użyteczne w procesie usuwania problemów z wydajnym funkcjonowaniem sieci, a także umożliwiają analizę zjawisk wskazujących na zagrożenia bezpieczeństwa sieci. Przykładowo służą one do:

- konwersji binarnych pakietów do formy czytelnej dla człowieka
- rozwiązywania problemów działania sieci
- analizy wydajności sieci w celu wykrycia wąskich gardeł
- wykrywania ataków sieciowych
- analizy działania aplikacji
- wykrywania uszkodzonych kart sieciowych
- wykrywania źródła ataków typu Denial of Service (DoS)
- wykrywania złośliwego oprogramowania
- wykrywania skompromitowanych stacji roboczych w sieci
- edukacji w dziedzinie protokółów sieciowych [4].

[1] http://pl.wikipedia.org/wiki/Sniffer

2.4. Sniffer jako narzędzie wykorzystywane do ataków sieciowych

Sniffery wykorzystywane we wrogich celach stanowią poważne zagrożenie bezpieczeństwa sieci. Napastnicy sieciowi używają ich w celu przechwycenia wartościowych, poufnych informacji. Jest to nazywane pasywnym atakiem, bowiem nie występuje tu bezpośrednie oddziaływanie na systemy sieciowe. Sniffer jest zwykle instalowany na skompromitowanym, w wyniku aktywnego ataku, komputerze działającym w sieci. Może być programem samodzielnym, jak również częścią złośliwych programów typu koń trojański używanych w celu przechwycenia specyficznych informacji, takich jak na przykład hasła, które następnie mogą być przesłane przez sieć do napastnika.

Sniffery są najczęściej wykorzystywane przez napastników sieciowych w celu:

- przechwycenia identyfikatorów i haseł użytkowników przesyłanych w postaci otwartego tekstu
- pasywnego wykrywania systemu operacyjnego hosta (ang. passive OS fingerprinting)
- uzyskiwania informacji o topologii sieci i konfiguracji urządzeń dostępowych
- przechwytywania i odtwarzania rozmów VoIP (ang. Voice over IP) [4].

2.5. Ochrona przed pasywnymi atakami sieciowymi wykorzystującymi sniffery

Pasywne ataki przeprowadzane przy pomocy snifferów są trudne do wykrycia, bowiem nie wchodzą one w jakiekolwiek interakcje z urządzeniami działającymi w sieci, ani nie generują żadnego ruchu sieciowego [4]. Mimo tego wykrywanie snifferów jest możliwe. Najprostsza metodą jest sprawdzenie, czy w sieci nie ma interfejsów działających w trybie niewybiórczym (ang. promiscuous mode). W systemach UNIX-owych wykorzystujemy do tego celu polecenie **ifconfig –a**. Ustawiony znacznik PROMISC zdradza tryb niewybiórczy wylistowanego interfejsu, jak w poniższym przykładzie:

[root@localhost root]# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:02:B3:06:5F:5A
inet addr:192.168.1.2 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
RX packets:204 errors:0 dropped:0 overruns:0 frame:0
TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
…[2]

[2] [4]

Można użyć również polecenia **ip link**, jak w poniższym przykładzie:

[root@localhost root]# ip link

1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: eth0: <BROADCAST,MULTICAST,PROMISC,UP> mtu 1500 qdisc pfifo_fast qlen 100 link/ether 00:02:b3:06:5f:5a brd ff:ff:ff:ff:ff:ff [3]

Wykrywanie trybu niewybiórczego w systemach Windowsowych jest dużo trudniejsze, bowiem nie ma standardowych poleceń, które zwracają informację o takim trybie pracy karty sieciowej. Można jednak użyć w tym celu bezpłatnego programu narzędziowego o nazwie PromiscDetect autorstwa Arne Vidstrom'a, które wykrywa tryb niewybiórczy karty sieciowej w systemach Windows NT, 2000 i XP. Można go pobrać pod adresem: http://ntsecurity.nu/toolbox/promiscdetect. Poniższy przykład pokazuje wynik działania tego programu:

C:\>promiscdetect

PromiscDetect 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)

- http://ntsecurity.nu/toolbox/promiscdetect/

Adapter name:

- D-Link DWL-650 11Mbps WLAN Card

Active filter for the adapter:

- Directed (capture packets directed to this computer)

- Multicast (capture multicast packets for groups the computer is a member of)

- Broadcast (capture broadcast packets)

Adapter name:

- Intel(R) PRO/100 SP Mobile Combo Adapter

Active filter for the adapter:

- Directed (capture packets directed to this computer)

- Multicast (capture multicast packets for groups the computer is a member of)

- Broadcast (capture broadcast packets)

- Promiscuous (capture all packets on the network)

WARNING: Since this adapter is in promiscuous mode there could be a sniffer running on this computer! [4]

---

[3] [4]

[4] [4]

Niektóre sniffery mogą jednak ukrywać znacznik trybu niewybiórczego, jak również instalować się w systemie przy pomocy rootkita, który może zamienić polecenia, takie jak ifconfig, tak aby nie wskazywały trybu niewybiórczego [4]. Można więc użyć innych metod wykrywania snifferów. Nie jest to jednak przedmiotem mniejszej pracy. Więcej informacji na ten temat można znaleźć w poz. [4] oraz w zasobach Internetu.

Jak to już powiedziano wyżej, wykrywanie snifferów jest trudne i nie zawsze skuteczne. Można jednak chronić się przed negatywnymi skutkami ich działania. Już samo używanie w sieci przełączników zamiast koncentratorów mocno utrudni sniffing, aczkolwiek nie wyeliminuje go całkowicie. Najlepszą ochroną przed skutkami sniffingu jest zastosowanie szyfrowania ruchu sieciowego, bowiem z przechwyconych pakietów nie da się uzyskać informacji, które są celem napastnika. Niektóre metody szyfrowania pozostawiają nagłówki w formie otwartego tekstu, co pozwala na odczytanie adresów pakietów, jednak przesyłane dane są zaszyfrowane, a przez to niedostępne dla napastnika [4].

Inną metodą ochrony przed podsłuchem w sieci jest zastosowanie technologii VLAN (ang. Virtual LAN). Polega ona na tworzeniu wirtualnych (logicznych) sieci poprzez przypisanie do nich określonych portów przełączników, dla wydzielonych grup użytkowników. W efekcie następuje podział sieci na rozdzielne domeny broadcastowe [3].

Podział sieci na VLAN'y znacząco redukuje narażenie na podsłuch, a także umożliwia wydzielenie szczególnie chronionych wirtualnych sieci. Można utworzyć oddzielny VLAN dla serwerów, oddzielny dla stacji administratorskich i ograniczyć dostęp do nich przy pomocy odpowiednio zdefiniowanych, restrykcyjnych list dostępu (Access Control Lists)[5].

3. Analizator sieci Wireshark

Analizator Wireshark wcześniej znany jako „Ethereal jest snifferem i analizatorem pakietów z możliwością dekodowania wielu protokołów. Funkcjonalność Ethereal'a jest bardzo podobna do tcpdump'a, lecz Ethereal posiada GUI i dużo więcej opcji sortowania i filtrowania. Pozwala użytkownikowi zobaczyć cały ruch w sieci przez przełączenie karty sieciowej w tryb promiscuous.

Ethereal rozpowszechniany jest jako FOSS (ang. Free Libre/Open Source Software, także FOSS, F/OSS), jest dostępny na następujące platformy: Windows, Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Mac OS X".[6]

[5] http://www.securitydocs.com/library/2829

[6] http://pl.wikipedia.org/wiki/Wireshark

3.1. Krótka historia analizatora Wireshark

Wireshark ma ciekawą historię, która zaczęła się w 1998 roku. Wtedy to Gerald Combs, absolwent informatyki University of Missouri w Kansas City stworzył oryginalną wersję oprogramowania do analizy sieci pod nazwą Ethereal, które udostępnił publicznie na zasadach licencji GNU General Public License (GNU GPL). Osiem lat później Gerald Combs zmienił pracę w poszukiwaniu rozwoju swojej kariery zawodowej, natomiast prawa do nazwy Ethereal pozostały własnością poprzedniego pracodawcy, z którym w tej sprawie nie mógł dojść do porozumienia. Dlatego też Combs z resztą zespołu w połowie 2006 roku uruchomił kontynuację projektu rozwoju analizatora Ethereal pod nazwą Wireshark. Od tego czasu popularność Wiresharka bardzo wzrosła i aktualnie jest on nadal rozwijany przez zespół około 500 współpracowników, natomiast analizator Ethereal nie jest już rozwijany. [1]

3.2. Zalety analizatora Wireshark

Analizator Wireshark charakteryzuje się kilkoma zaletami, które umieszczają go w gronie najlepszych snifferów, zarówno bezpłatnych jak i komercyjnych. Najważniejsze z nich wymieniano poniżej:

Ilość obsługiwanych protokołów – 879 (Wireshark w wersji 0.99.6a), począwszy od podstawowych, takich jak TCP/IP i DHCP do specyficznych jak BitTorrent i VNC. Watro zauważyć, że ponieważ Wireshark jest rozwijany jako Open Source, w każdej jego nowej wersji są dołączane kolejne protokoły.

Łatwość obsługi – Wireshark jest wyposażony w interfejs graficzny (GUI), bardzo jasno zaprojektowane menu kontekstowe, a także czytelne zaznaczenie protokołów kolorami oraz szczegółowe graficzne odwzorowanie przechwytywanych danych [1].

Wireshark jest dostępny bezpłatnie zarówno do osobistego jak i komercyjnego użytku na zasadach licencji GNU General Public License version 2.

3.3. Instalacja Wiresharka na platformie Windows

Wireshark może być zainstalowany w systemach Windowsowych począwszy od Windows 2000 i nie stawia szczególnych wymagań sprzętowych (zależą one od ilości przechwytywanego ruchu sieciowego). Wymaga użycia sterownika przechwytującego pakiety WinPcap capture driver – jest on zwarty w pakiecie instalacyjnym Wiresharka.

Instalacja przebiega typowo za pomocą kreatora, jasno objaśniającego dostępne opcje i ułatwiającego ich wybór (proponuje instalację drivera WinPcap, jeśli nie wykryje go w systemie).

3.4. Instalacja Wiresharka w systemach Linuxowych

Pierwszym krokiem przy instalacji Wiresharka w systemach Linuxowych jest uzyskanie właściwej dla danej dystrybucji Linuxa wersji pakietu instalacyjnego (może być niedostępny dla niektórych wersji Linuxa). Następnie w zależności od typu systemu Linux należy użyć polecenia :

Dla systemów opartych na RPM, takich jak RedHat

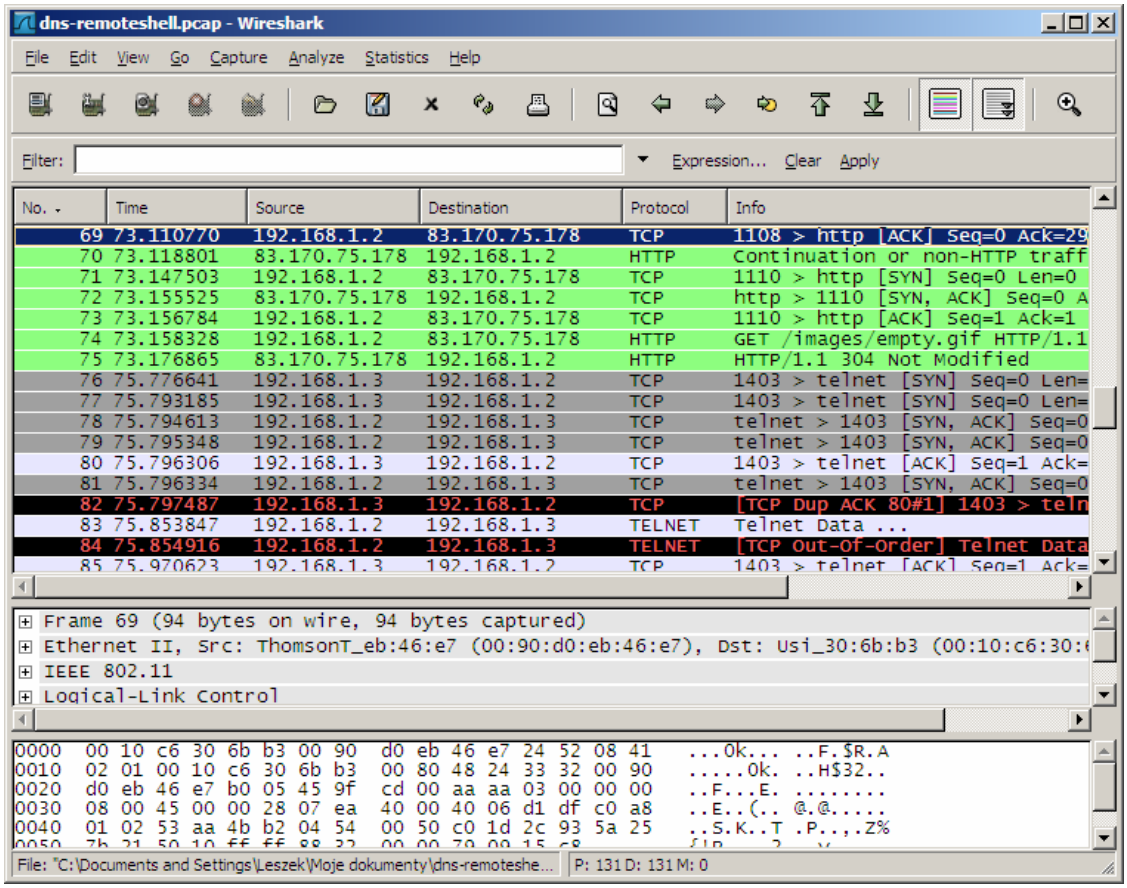
rpm -ivh wireshark-0.99.3.i386.rpm[7]

Dla systemów opartych na DEB, takich jak Debian lub Ubuntu

apt-get install wireshark[8]

3.5. Charakterystyka analizatora Wireshark

Główne okno aplikacji Wireshark zwierające wszystkie elementy służące do przechwycenia i analizy pakietów przedstawiono na rysunku 2.



Rys. 2. Główne okno analizatora Wireshark w wersji 0.99.6a

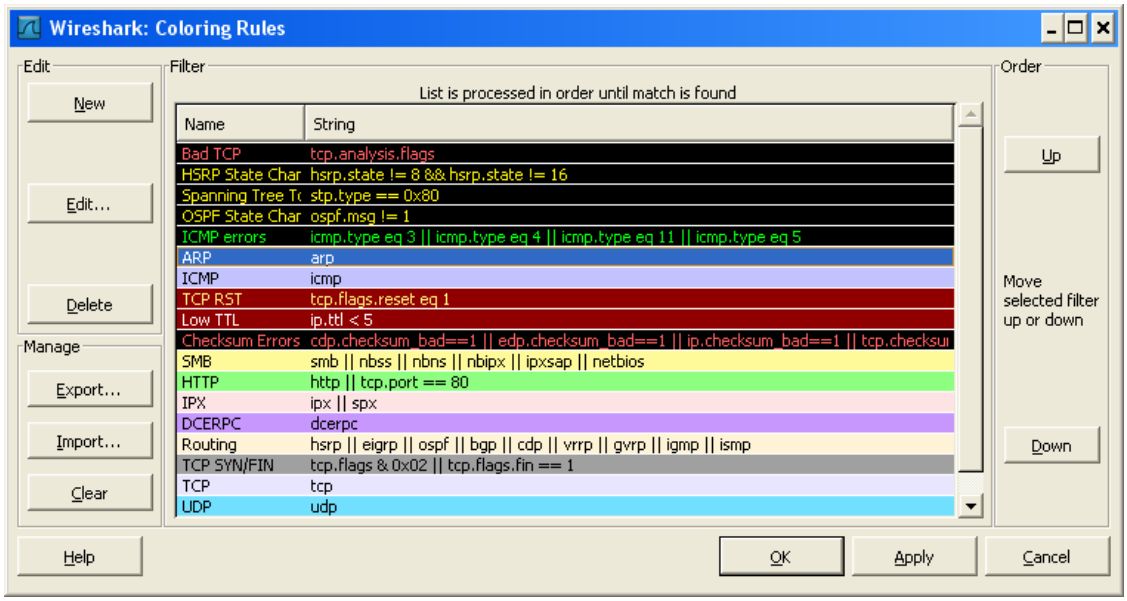
[7] [1] strona 31

[8] [1] strona 31

Interfejs użytkownika określa sposób prezentacji danych przez Wiresharka. Większość jego opcji może być konfigurowana w zależności od potrzeb lub upodobań.

Preferencje przechwytywania pozwalają na ustawienie domyślnego trybu działania karty sieciowej – czy ma być ustawiona w tryb niewybiórczy (domyślnie zaznaczone).

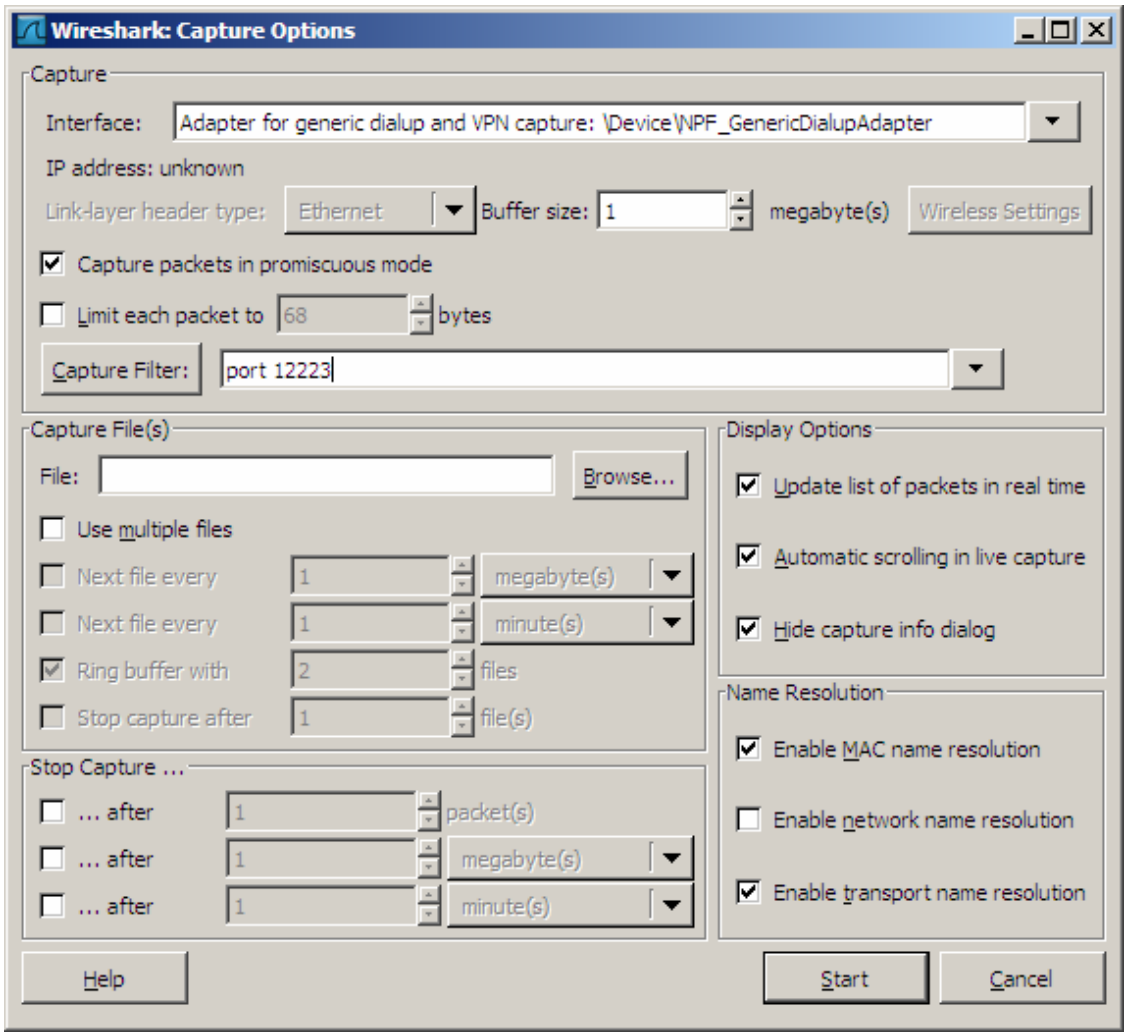
Preferencje protokołów określają sposób ich przechwytywania i wyświetlania. Szczególną zaletą Wiresharka jest oznaczanie każdego protokołu określonym kolorem, co bardzo ułatwia ich identyfikację i analizę. Predefiniowane kolory można modyfikować, jeśli zachodzi taka potrzeba - rysunek 3.



Rys. 3. Okno dialogowe Reguł Oznaczania Kolorami

Podstawową funkcją analizatora sieci jest przechwytywanie pakietów oraz przedstawienie ich w formie dogodnej do analizy. Ilość przechwytywanych danych może być bardzo duża, co utrudnia analizę. W celu rozwiązania tego problemu w Wiresharku zastosowano filtrowanie pakietów. Dostępne są dwa rodzaje filtrów: filtry przechwytywania oraz filtry wyświetlania. Zastosowanie pierwszego rodzaju filtra ogranicza ilość przechwytywanych pakietów, co może być czasami korzystne, choćby ze względu na ograniczone zasoby stacji roboczej, na której jest zainstalowany Wireshark. Wadą jest to, że wykluczona przy pomocy filtra przechwytywania część ruchu sieciowego może zwierać przydatne do analizy pakiety, które jednak nie zostały zapisane. Takiej wady nie wykazują filtry wyświetlania, bowiem można je dowolnie modyfikować, czy nawet usuwać uzyskując dostęp do wszystkich przechwyconych pakietów. Filtry wyświetlania, połączone z możliwością wyszukiwania i zaznaczania pakietów bardzo ułatwiają analizę ruchu sieciowego i wykrywanie wszelkich jego anomalii.

Filtry przechwytywania pozwalają na podsłuch tylko określonego ruchu sieciowego, na przykład ruchu na wybranym interfejsie oraz na wybranym porcie. Na rysunku 4 pokazano okno dialogowe pozwalające na ustawienie filtra w celu przechwycenia ruchu na porcie, przykładowo 12223 (Hack'99 KeyLogger). W polu obok przycisku Capture Filter można wpisać wyrażenie definiujące warunki filtrowania lub użyć w tym celu kreatora wywoływanego tym przyciskiem.

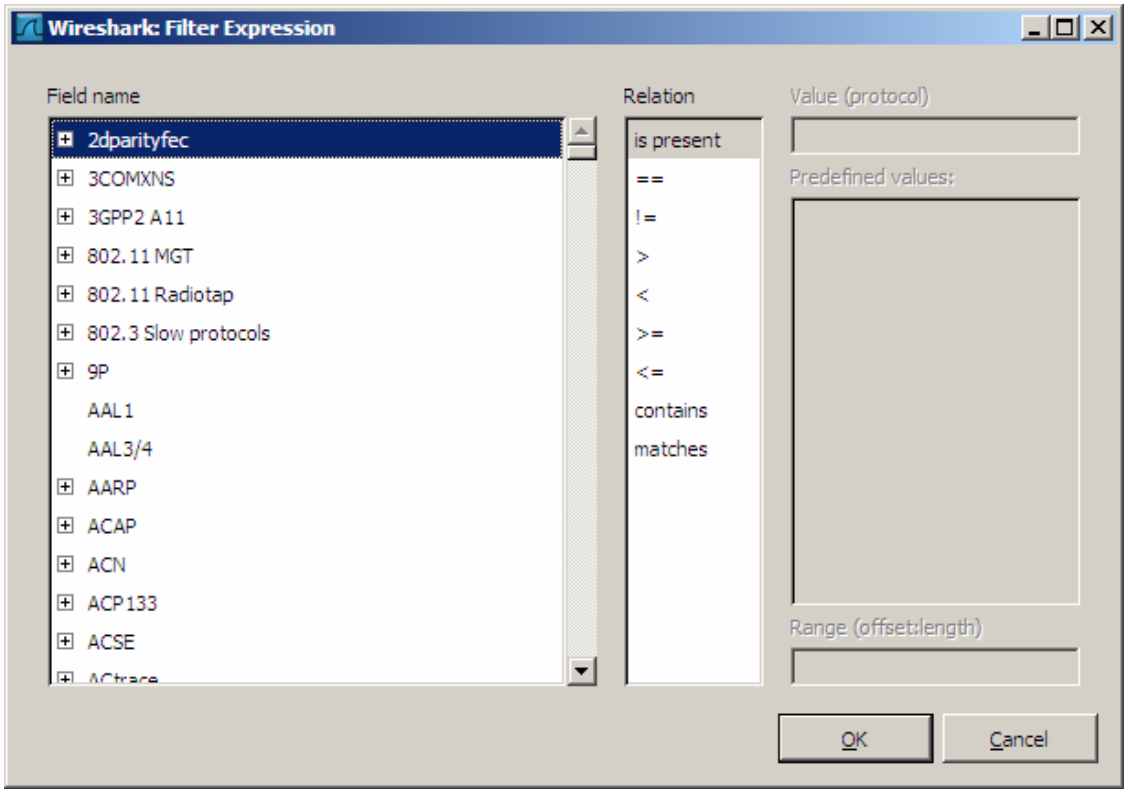


Rys. 4. Okno dialogowe Opcje Przechwytywania umożliwiające utworzenie filtra.

Filtry wyświetlania są często używane w celu wyeliminowania z analizy nie relewantnych pakietów, na przykład broadcastu ARP. Oczywiście, te pakiety mogą być przydatne w dalszej analizie. Nie ma problemu z ich przeanalizowaniem, ponieważ są przechwycone – wystarczy zmienić ustawienie filtra wyświetlania i staną się widoczne.

Na rysunku 5 pokazano okno dialogowe Wyrażenia Filtrującego, umożliwiające łatwe i szybkie ustawienie filtra. O wiele bardziej elastyczną, aczkolwiek trudniejszą metodą

ustawiania filtra wyświetlania jest ręczne wpisanie wyrażenia filtrującego, tak jak dla filtra przechwytywania. Przykładowe wyrażenia filtrujące zostały zamieszczone w tabeli 1.



Rys. 5. Okno dialogowe Wyrażenie Filtrujące pozwalające na łatwe utworzenie filtra.

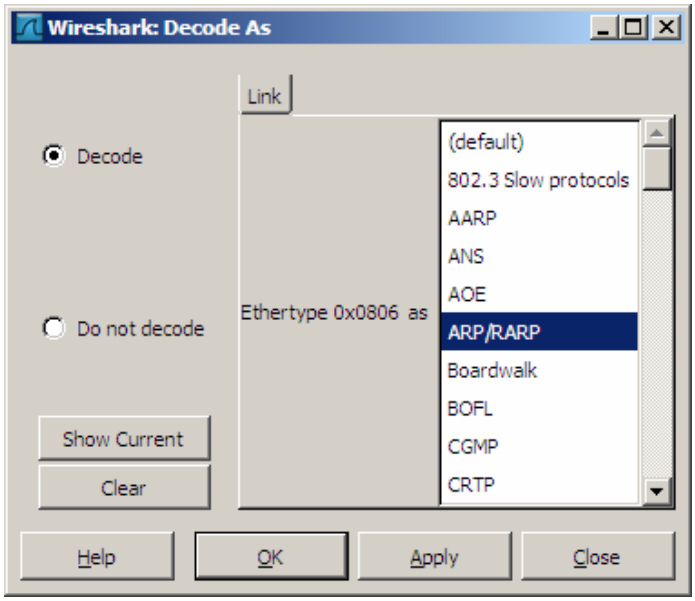
Tabela 1.

Przykładowe wyrażenia filtrujące dla filtrów przechwytywania i filtrów wyświetlania [1].

Wyrażenie filtrujące	Opis
host www.example.com	Wyświetla/przechwytuje cały ruch sieciowy z hosta www.example.com
host www.example.com and not (port 80)	Wyświetla/przechwytuje cały ruch sieciowy z hosta www.example.com za wyjątkiem WWW
!dns	Pokazuje/przechwytuje cały ruch za wyjątkiem DNS
not broadcast and not multicast	Pokazuje/przechwytuje tylko ruch unicastowy
ip.dst==192.168.0.1	Pokazuje/przechwytuje cały ruch wysyłany na adres 192.168.0.1

Wireshark posiada wbudowane dekodery protokołów (ang. protocol dissectors), które identyfikują i formatują przechwycone pakiety w dogodny dla celów analizy sposób. Na ogół identyfikacja protokołu jest poprawna, ponieważ Wireshark używa w tym celu kilku

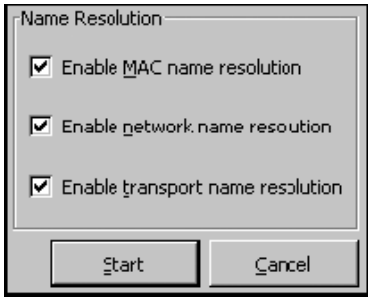
dekoderów jednocześnie, aczkolwiek zdarzają się błędne rozpoznania. W takim przypadku można wymusić żądany sposób dekodowania pakietów – rysunek 6.



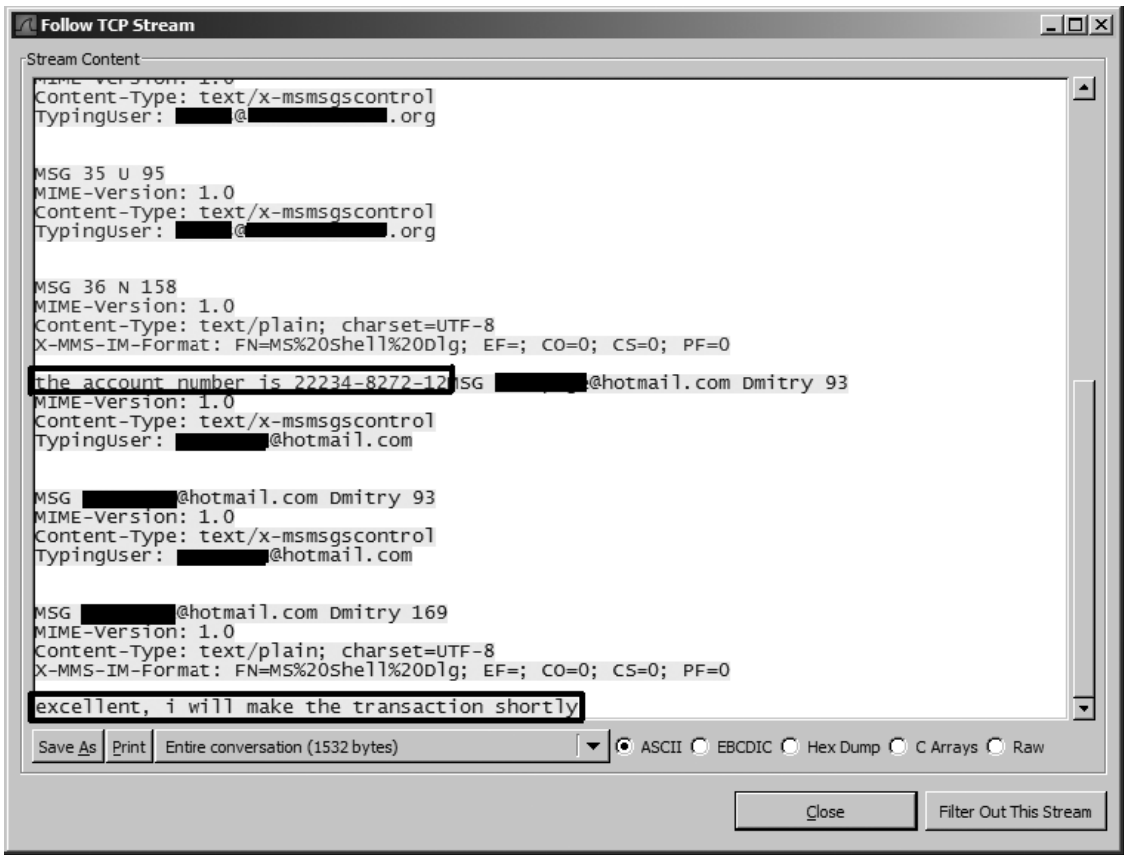
Rys. 6. Okno dialogowe Dekoduj Jako umożliwiające zmianę dekodowania protokołu.

Jedną z najbardziej użytecznych cech analizatora Wireshark jest opcja Śledzenia Strumienia TCP (ang. Following TCP Stream), która umożliwia zobrazowanie ruchu TCP w taki sposób, jaki jest adekwatny dla warstwy aplikacji. Cecha to pozwala na połączenie wszystkich informacji zawartych w pakietach i ukazanie danych zawartych w tych pakietach w takiej formie, w jakiej są widoczne w aplikacjach przez użytkownika. Umożliwia to, na przykład przechwycenie i odczytanie informacji przekazywanych przez komunikatory, co widać na rysunku 8.

Na zakończenie charakterystyki sniffera Wireshark warto wspomnieć o jego możliwościach rozwiązywania nazw, co nie rzadko może bardzo ułatwić analizę incydentu. Wireshark obsługuje trzy typy rozwiązywania nazw: ARP, DNS i portu warstwy transportowej – rysunek 7.



Rys. 7. Okno dialogowe ustawienia Rozwiązywania Nazw [1].



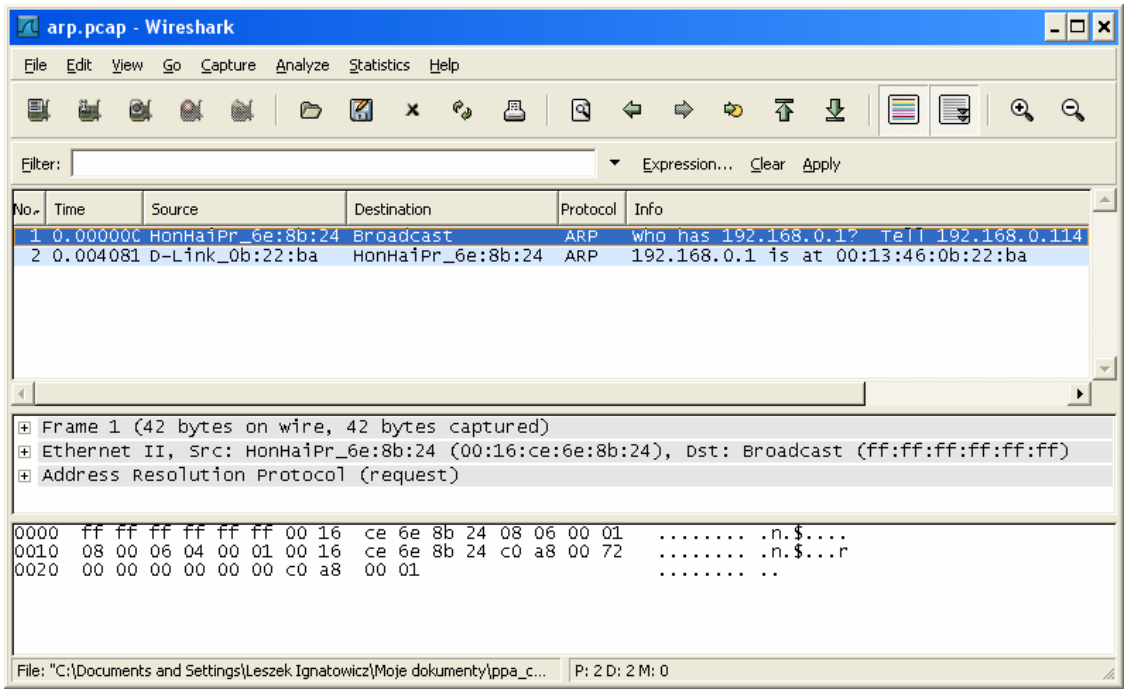
Rys. 8. Okno Śledzenia Strumienia TCP z przechwyconymi informacjami [1].

3.6. Przegląd niektórych protokołów przechwytywanych przez Wiresharka - przykłady

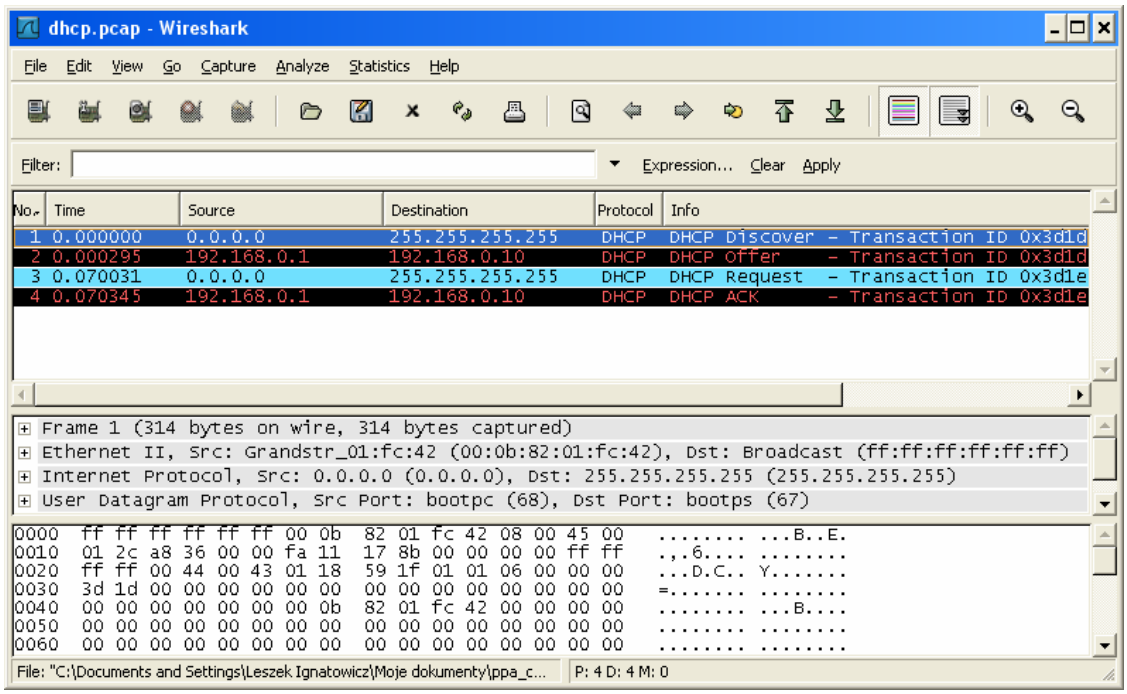
Wireshark ma wbudowane dekodery protokołów, które bardzo przejrzyście, w formie graficznej pokazują zawartość przechwyconych pakietów. Poniżej przedstawiono przykłady kilku podstawowych protokołów (pliki z zapisami przechwyconych pakietów pobrano z http://www.nostarch.com/frameset.php?startat=packet) oraz dwóch bardziej złożonych VNC i BitTorrent (pliki z zapisami przechwyconych pakietów pobrano z http://wiki.wireshark.org/SampleCaptures).

Jednym z najprostszych protokołów jest ARP (Address Resolution Protocol), którego zadaniem jest konwersja adresów IP Warstwy 3 na adresy fizyczne Warstwy 2 (adresy MAC). Działanie tego protokółu ilustruje rysunek 9. Jak widać, występują tu tylko dwa rodzaje pakietów: żądanie ARP, rozgłoszeniowy (broadcast ff:ff:ff:ff:ff:ff) oraz odpowiedź na adres wysyłającego żądanie.

Kolejny rysunek 10 obrazuje działanie protokołu DHCP. Widoczne są pakiety: rozgłoszeniowy DHCP Discover („odkrycie DHCP”), unicastowy DHCP Offer („oferta DHCP”), rozgłoszeniowy DHCP Request („żądanie DHCP”) oraz unicastowy DHCP ACK („potwierdzenie DHCP”).

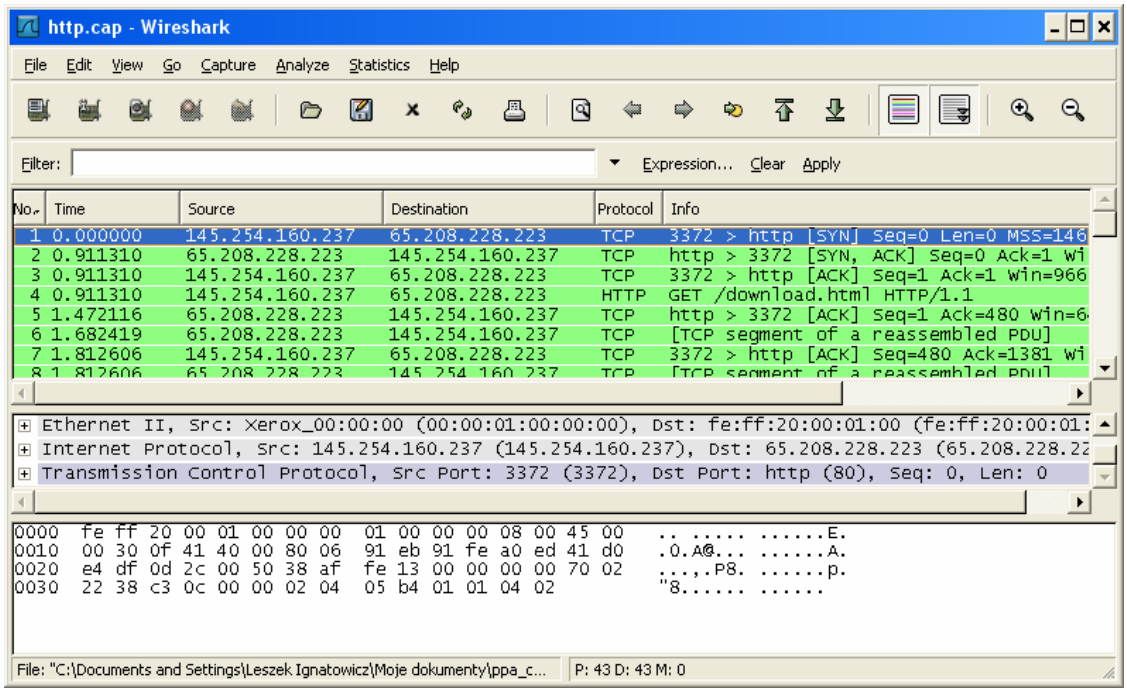


Rys. 9. Pakiety protokołu ARP.

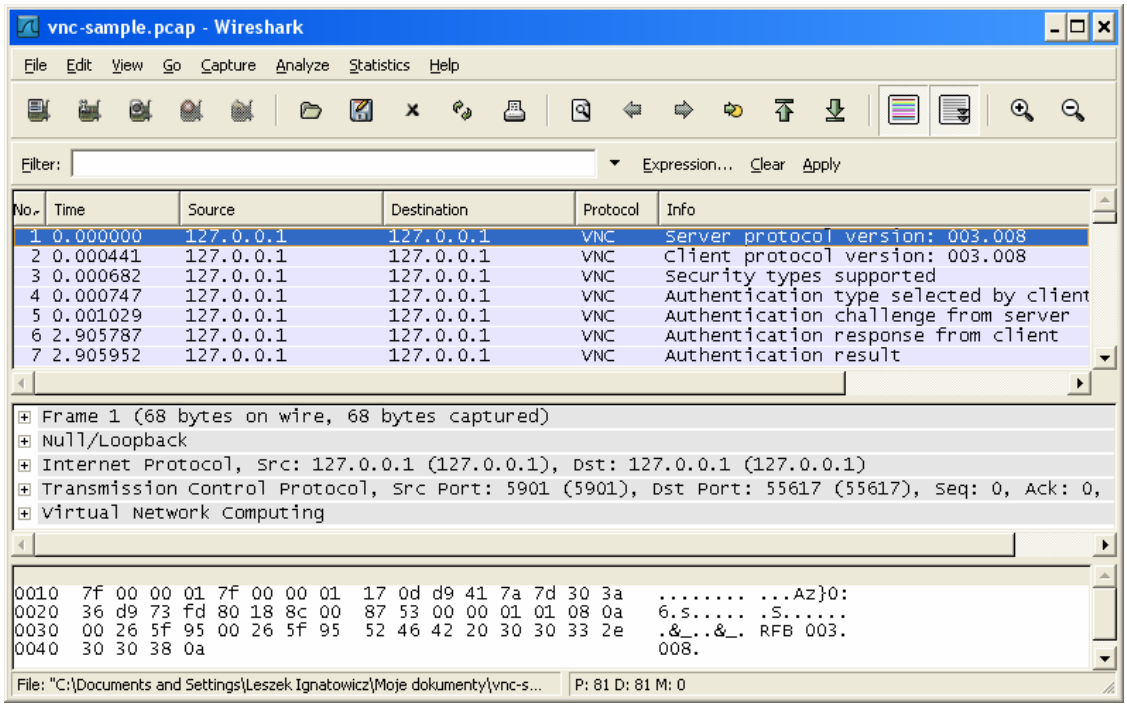


Rys. 10. Pakiety protokołu DHCP.

Przykład na rysunku 11 obrazuje protokoły TCP/IP oraz HTTP. W przechwyconym ruchu widać zestawioną sesję TCP/IP (standardowy 3-etapowy handshake), następnie żądanie GET transmisji danych HTTP. Warto zwrócić uwagę na okienko Szczegółów Pakietu (w środku okna głównego) – widać w nim, że docelowym portem jest Dst Port: http (80).



Rys. 11. Protokóły TCP/IP oraz DHCP.

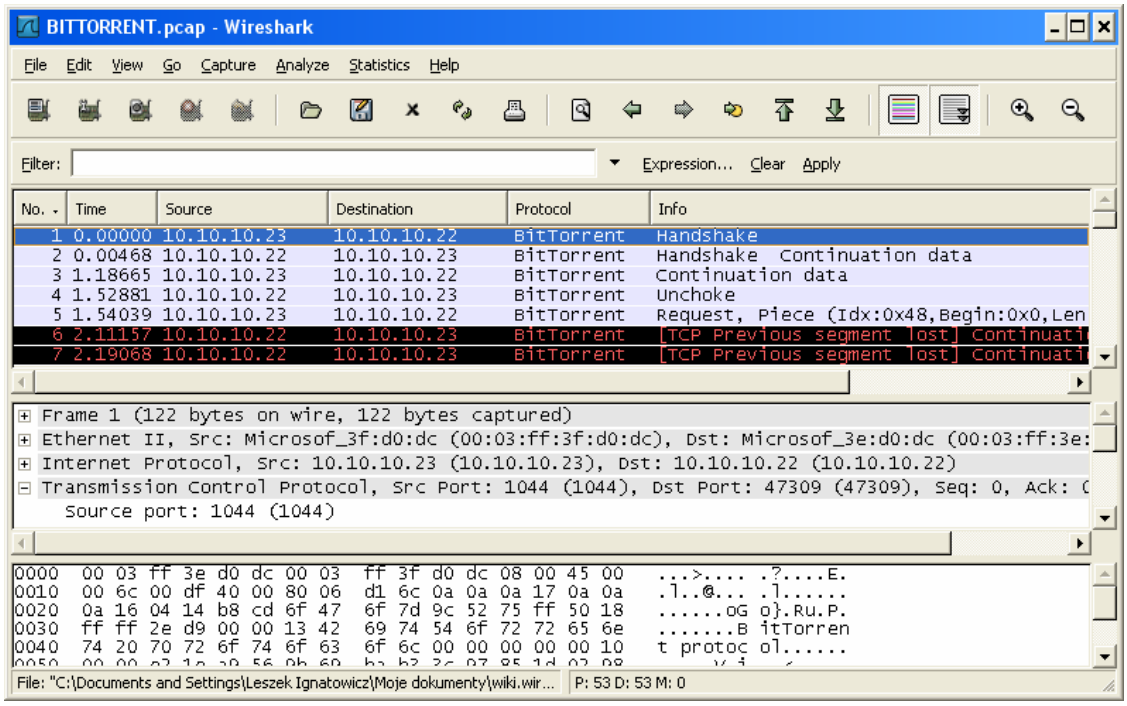


Rys. 12. Zobrazowanie ruchu sieciowego związanego z protokołem VNC.

Na zakończenie przeglądu wybranych protokołów dekodowanych przez Wiresharka pokazano dwa bardziej złożone protokoły VNC – rysunek 12 i BitTorrent – rysunek 13. Są to protokoły, których wykrycie w sieci może oznaczać zagrożenie jej bezpieczeństwa. Dlatego poniżej podano ich krótkie charakterystyki:

„VNC (ang. Virtual Network Computing) - system przekazywania obrazu z wirtualnego, bądź fizycznego środowiska graficznego. Prosty pakiet serwer+klient jest dostępny pod najpopularniejsze systemy operacyjne z trybem graficznym. Domyślnie VNC korzysta z portów TCP 5900 - 5906, gdzie każdy z portów oznacza odrębną sesję (:0 do :6), lecz zarówno klient jak i serwer mogą zostać przekonfigurowane.”[9]

„BitTorrent – protokół wymiany i dystrybucji plików przez Internet, którego celem jest odciążenie łączy serwera udostępniającego pliki. Jego największą zaletą w porównaniu do protokołu HTTP jest podział pasma pomiędzy osoby, które w tym samym czasie pobierają dany plik. Oznacza to, że użytkownik w czasie pobierania wysyła fragmenty pliku innym użytkownikom.”[10]



Rys. 13. Protokół BitTorrent zdekodowany w Wiresharku.

Powyższy przegląd protokołów ilustruje sposób graficznego przedstawienia dekodowanych pakietów, a także daje możliwość zaobserwowania jak wygląda normalny, prawidłowy ruch sieciowy. Jest to niezwykle istotne dla zbudowania w administrowanej sieci tzw. linii bazowej (ang. baseline), która umożliwi zauważenie wszelkich anomalii (odstępstw od znanego, ustalonego stanu) ruchu sieciowego, co jest podstawą wykrywania incydentów stanowiących zagrożenie bezpieczeństwa sieci.

[9] http://pl.wikipedia.org/wiki/VNC

[10] http://pl.wikipedia.org/wiki/Bittorrent

4. Wykrywanie i analizowanie zagrożeń bezpieczeństwa

Skuteczne wykrywanie i usuwanie zagrożeń bezpieczeństwa sieci teleinformatycznej wymaga znajomości podstawowych scenariuszy ataków stosowanych przez napastników sieciowych, nazywanych potocznie, lecz niepoprawnie hackerami. Faktycznym zagrożeniem są ci, których określamy mianem crackers lub Black Hats, wykorzystujący swoją wiedzę o systemach komputerowych w celu osiągnięcia korzyści, w sposób nie tylko nieetyczny, ale też najczęściej niezgodny z prawem.

Ataki na sieci teleinformatyczne, aczkolwiek wykorzystujące bardzo różne techniki mają w ogólnych zarysach powtarzalny scenariusz, obejmujący niżej wymienione podstawowe fazy:

Rozpoznanie – jest to faza początkowa przed właściwym atakiem, podczas której napastnik zbiera jak najwięcej informacji o interesującym go celu. Informacje te wykorzystuje podczas właściwego ataku. Zbieranie informacji ma często pasywny charakter, stąd też takie działania są trudne do wykrycia. Z drugiej strony wykrycie napastnika na tym etapie z reguły gwarantuje jego skuteczne unieszkodliwienie. Stąd też w punkcie 4.1 niniejszej pracy omówiono wykrywanie systemu operacyjnego, które z reguły jest etapem wstępnym ataku.

Skanowanie – jest to aktywna metoda uzyskania informacji o urządzeniach i usługach dostępnych w sieci będącej przedmiotem ataku. Wykrycie napastnika w tej fazie również daje duże szanse jego zneutralizowania, zanim poczyni jakieś szkody.

Uzyskanie dostępu – jest to kluczowa faza ataku. Jeśli się powiedzie, napastnik może narazić zasoby sieci na poważne szkody, w zależności od tego jakie uzyska uprawnienia w systemie.

Utrzymanie dostępu – w tym celu napastnik może wykorzystać rootkita, bądź też trojana lub backdoora. Im dłużej napastnik utrzyma dostęp do systemu, tym większa groźba penetracji zasobów. Jak najszybsze wykrycie i zneutralizowanie intruza jest więc niezwykle istotne dla zminimalizowania strat.

Zacieranie śladów – jest to działanie napastnika mające zapobiec wykryciu jego działań w systemie, co daje mu spore szanse pozostania bezkarnym.

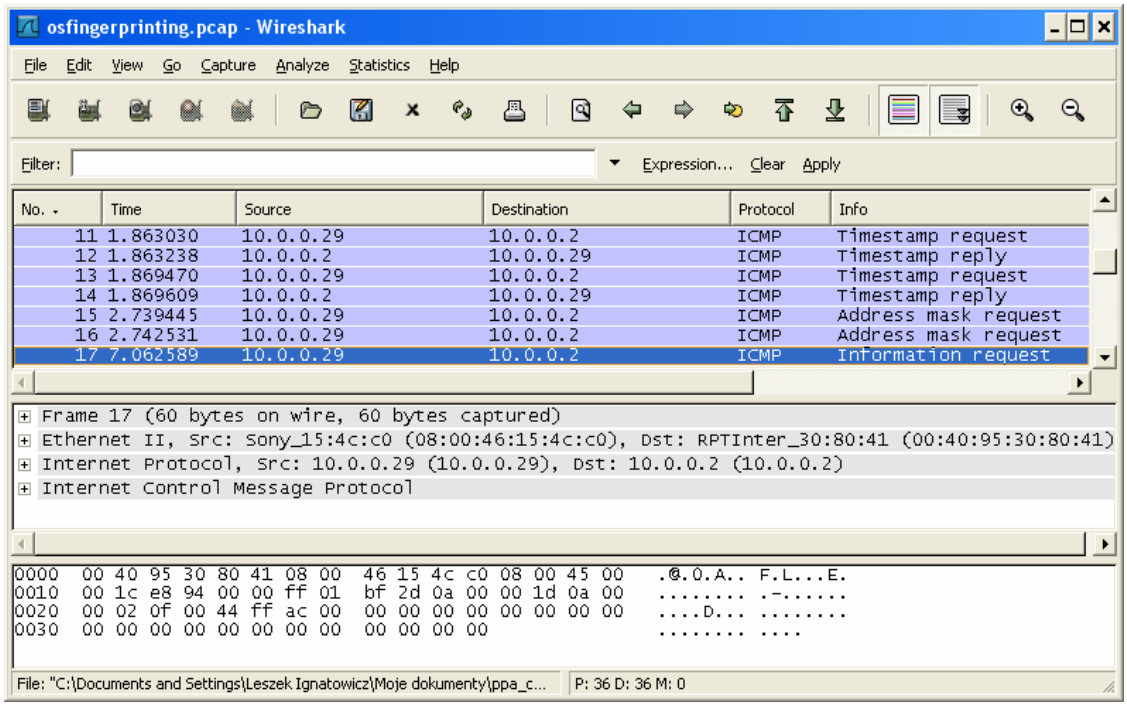
Z powyższego opisu wynika, że atak sieciowy jest procesem rozwijającym się w pewnym okresie czasu. Daje to administratorowi sieci szansę na wdrożenie skutecznych metod wykrywania, analizowania i neutralizowania napastników zanim uda im się uzyskać dostęp do chronionych zasobów, pod warunkiem, że dysponuje on wiedzą oraz dobrym oprogramowaniem monitorującym sieć. Jednym z takich programów jest Open Source'owy sniffer Wireshark, który daje szerokie możliwości analizowania różnych protokołów w sieci.

4.1. Wykrywanie systemu operacyjnego (ang. OS fingerprinting)

Wykrywanie systemu operacyjnego zdalnego komputera może być przeprowadzone następującymi metodami:

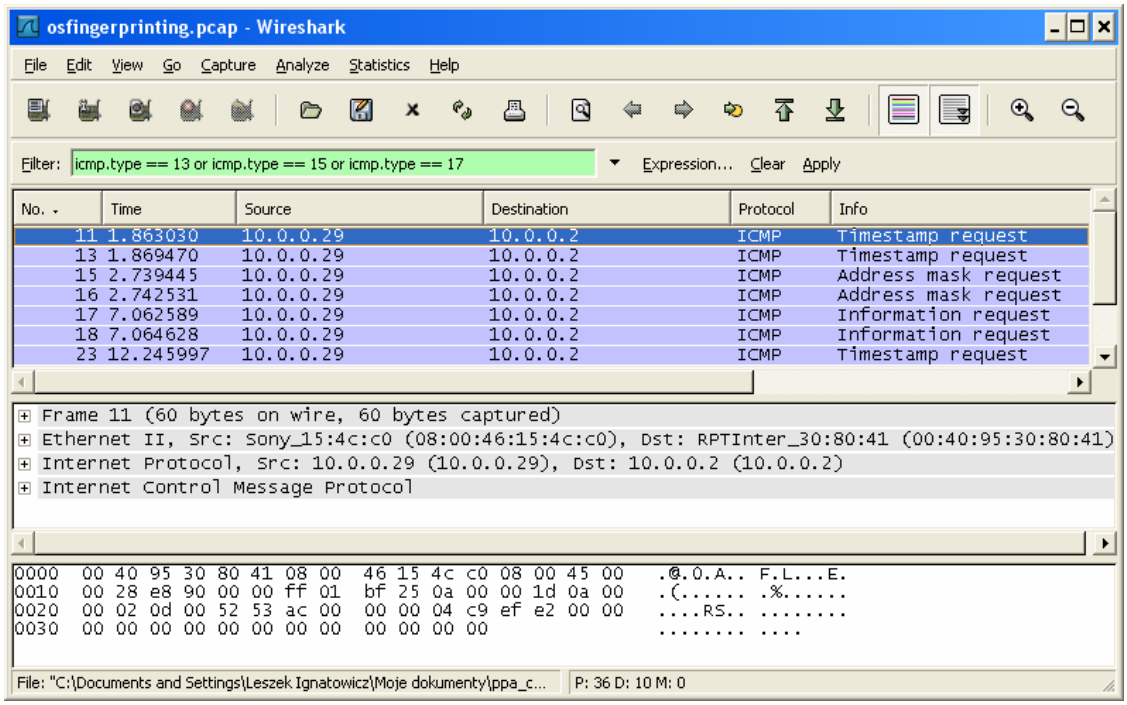
Pasywną – w której nie wysyła się do maszyny docelowej żadnych pakietów sondujących, za wyjątkiem, zupełnie poprawnego połączenia z usługą sieciową uruchomioną na zdalnym hoście. Polega ona na pasywnym analizowaniu stosu TCP/IP oraz identyfikacji wykorzystującej warstwę 7 aplikacji modelu OSI [3].

Aktywną – polegającą na wysyłaniu ze zdalnego hosta do docelowego komputera odpowiednio spreparowanych pakietów i analizowaniu odpowiedzi na nie w oparciu o określone wzorce. Na tej podstawie można wyciągnąć wnioski, co do systemu operacyjnego zainstalowanego na docelowym komputerze [1].



Rys. 14. Pakiety ICMP sodujące komputer w celu rozpoznania jego systemu operacyjnego.

Rysunek 14 przedstawia przechwycone pakiety ICMP, które nie występują w prawidłowym ruchu sieciowym. Wskazują one na rozpoznawanie systemu operacyjnego docelowego komputera metodą skanowania pakietami ICMP. Napastnik wysyła takie pakiety po to, aby na podstawie reakcji (bądź braku reakcji) docelowego komputera określić jego system operacyjny. W celu łatwiejszego zauważenia pakietów sondujących ICMP można zastosować filtr, co zostało pokazane na rysunku nr 15.



Rys. 15. Pakiety ICMP sondujące komputer wyselekcjonowane przy pomocy filtra.

Sondowanie ICMP (ang. ICMP probing) wykorzystuje następujące typy pakietów: Type 8 (echo request), Type 13 (timestamp request), Type 15 (information request) oraz Type 17 (subnet address mask request) [5].

W tabeli 2 przedstawiono odpowiedź różnych systemów operacyjnych na wyżej wymienione typy pakietów wysłanych jako bezpośrednie, nie rozgłoszeniowe. Natomiast w tabeli 3 przedstawiono odpowiedź tych samych systemów na te same typy pakietów ICMP, ale wysłanych jako nie bezpośrednie, rozgłoszeniowe.

Tabela 2.

Sondowanie pakietami ICMP wysłanymi jako bezpośrednie, nie rozgłoszeniowe [5].

System operacyjny	**ICMP bezpośrednie, nie rozgłoszeniowe**			
	8	**13**	**15**	**17**
Linux	Tak	Tak	Nie	Nie
BSD	Tak	Tak	Nie	Nie
Solaris	Tak	Tak	Nie	Tak
HP-UX	Tak	Tak	Tak	Nie
AIX	Tak	Tak	Tak	Nie
Ultrix	Tak	Tak	Tak	Tak
Windows 95, 98 i ME	Tak	Tak	Nie	Tak
Windows NT 4.0	Tak	Nie	Nie	Nie
Windows 2000	Tak	Tak	Nie	Nie
Cisco IOS	Tak	Tak	Tak	Nie

Tabela 3.

Sondowanie pakietami ICMP wysłanymi jako nie bezpośrednie, rozgłoszeniowe [5].

System operacyjny	ICMP nie bezpośrednie, rozgłoszeniowe			
	8	**13**	**15**	**17**
Linux	Tak	Tak	Nie	Nie
BSD	Nie	Nie	Nie	Nie
Solaris	Tak	Tak	No	Nie
HP-UX	Tak	Tak	Tak	Nie
AIX	Nie	Nie	Nie	Nie
Ultrix	Nie	Nie	No	Nie
Windows95,98 i ME	Nie	Nie	Nie	Nie
Windows NT 4.0	Nie	Nie	Nie	Nie
Windows 2000	Nie	Nie	Nie	Nie
Cisco IOS	Nie	Nie	Tak	Nie

4.2. Skanowanie portów

Skanowanie jest zwykle kolejną po rozpoznaniu fazą ataku. Ma ono na celu uzyskanie informacji o usługach dostępnych na wykrytych wcześniej maszynach, najczęściej poprzez skanowanie portów. Odnalezienie otwartych portów umożliwia identyfikację dostępnych usług, które mogą być celem ataku.

Najpopularniejszym skanerem jest nmap[11], dostępny na wielu platformach, w tym na najpopularniejszych: Unix/Linux i Windows.

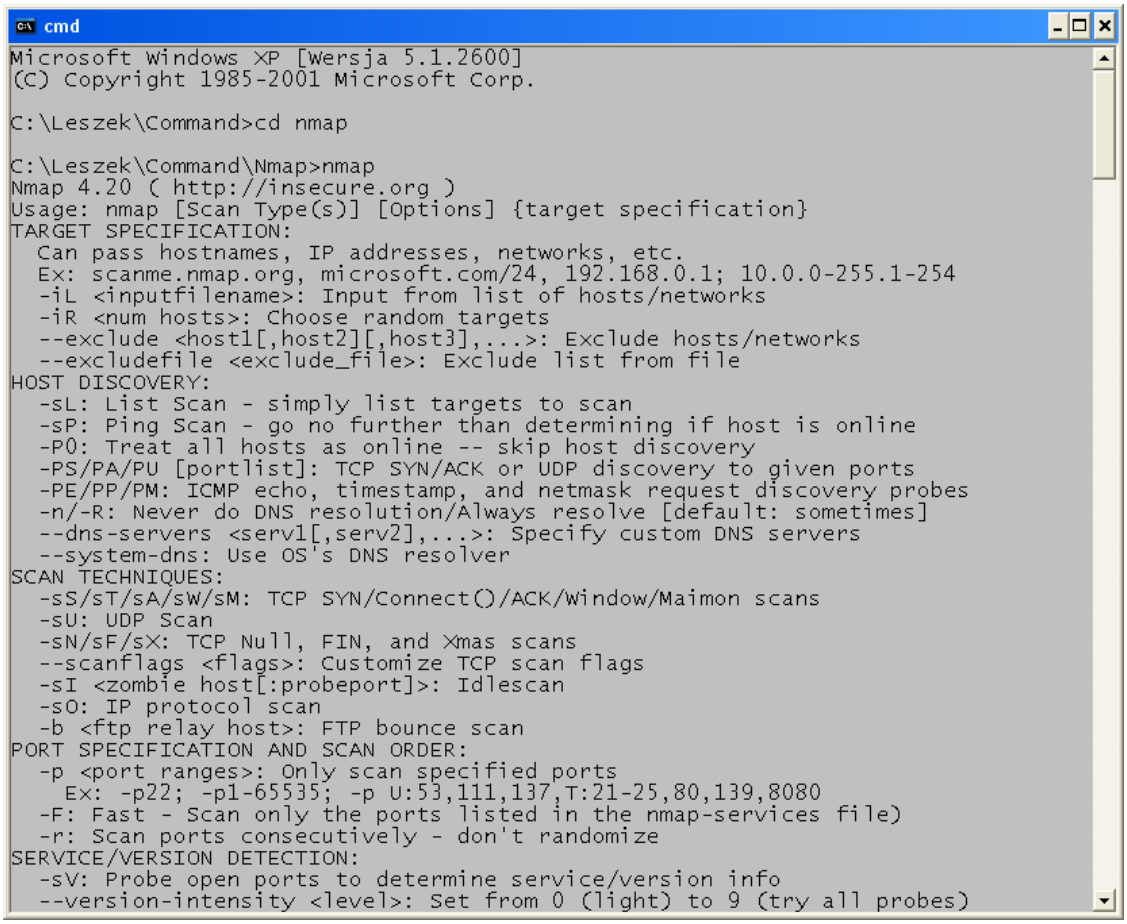
„Nmap (z ang. Network mapper), program komputerowy autorstwa Fyodora (Gordon Lyon), służący do skanowania portów. Program implementuje wiele różnych technik testowania portów TCP, w tym niestandardowe podejścia wynikające ze specyfiki implementacji stosów sieciowych, które potencjalnie mogą omijać zapory sieciowe lub platformy Intrusion Detection System.” [12]

W wersji podstawowej nmap jest obsługiwany z linii poleceń – rysunek 16. Umożliwia wybieranie trybu skanowania, oraz ustawianie wielu opcji. Potrafi nie tylko stwierdzić jakie komputery działają w sieci, ale także z dużym prawdopodobieństwem określić ich system operacyjny, a nawet podać wersje usług uruchomionych na poszczególnych portach. Nmap jest aktywnym skanerem, a przez to stosunkowo łatwo wykrywalnym.

[11] www.insecure.org/nmap/

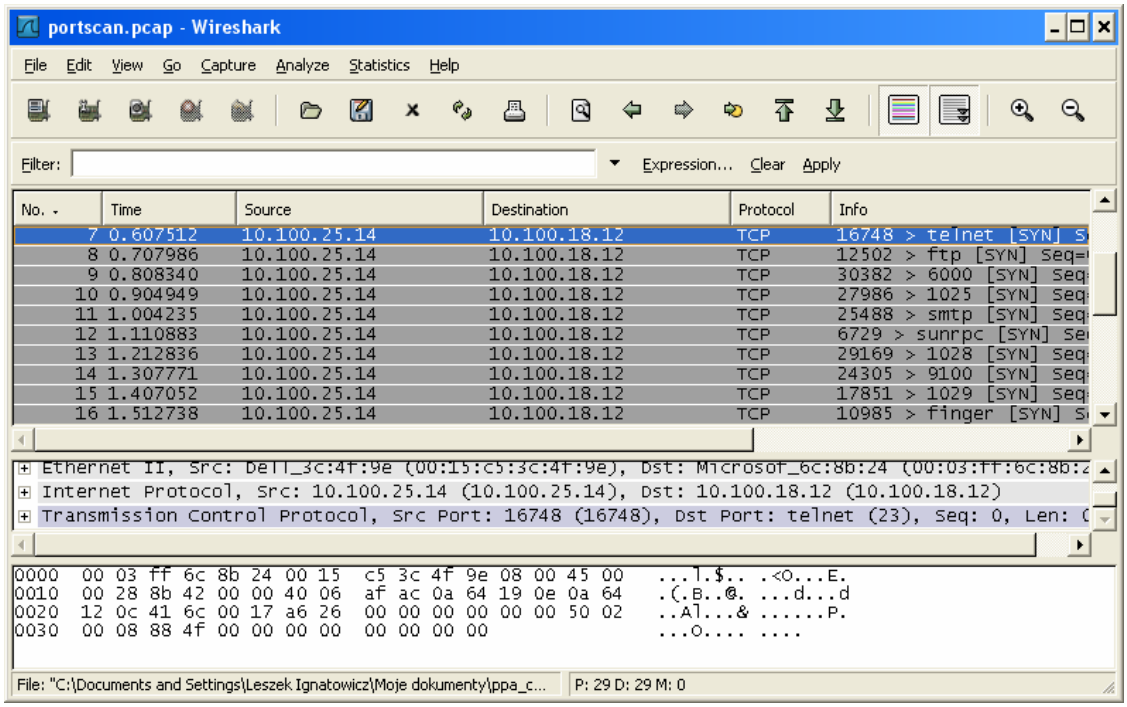
[12] http://pl.wikipedia.org/wiki/Nmap

Podstawowe techniki skanowania portów polegają na wysyłaniu pakietów TCP lub UDP do badanego hosta. Skanowanie TCP-connect, skanowanie TCP SYN, skanowanie TCP FIN, skanowanie TCP ACK, skanowanie TCP NULL, skanowanie TCP XMAS oraz skanowanie TCP-bounce, a także skanowanie UDP są szczegółowo omówione w pozycji [3].



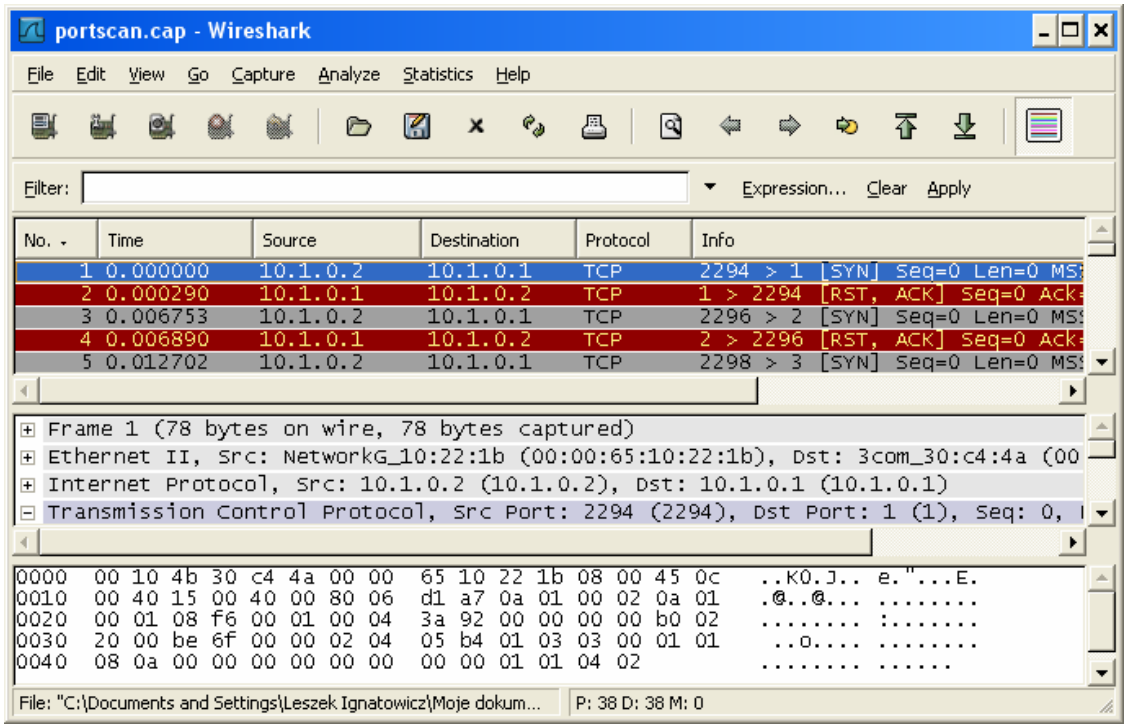
Rys. 16. Nmap 4.20 uruchomiony z linii poleceń Windows.

Na rysunku 17 przedstawiono ruch sieciowy przechwycony przez Wiresharka. Widoczne są pakiety przesyłane miedzy lokalnym komputerem o adresie 10.100.25.14 oraz zdalnym hostem o adresie 10.100.18.12. Warto zauważyć, że każdy pakiet wysłany ze zdalnego hosta jest przeznaczony dla innego portu lokalnego komputera, na przykład porty telnet, ftp, smtp, sunprc, finger i inne określone tylko numerami. Można z dużym prawdopodobieństwem podejrzewać, że jest to skanowanie portów jako przygotowanie do ataku. Przedstawiony przykład to dość proste, łatwe do zauważenie skanowanie portów. Bardziej wyrafinowane metody są trudniejsze do wykrycia. Bardzo pomocne może być zastosowanie filtrów, jednak ostateczny rezultat zależy to od wiedzy i doświadczenia administratora analizującego podejrzany ruch w sieci.



Rys. 17. Przykład skanowania portów.

Pliki z zapisem przechwyconego ruchu sieciowego (ang. trace files), obrazującego różne metody skanowania portów można wyszukać w Internecie i poprzez ich analizę doskonalić swoje umiejętności. Na rysunku 18 przedstawiono plik **portscan.cap** pobrany z http://www.packet-level.com/traces/ załadowany do analizatora Wireshark.



Rys. 18. Przykład skanowania TCP SYN.

4.3. Ataki sieciowe

Najczęściej wykorzystywanym rodzajem ataków sieciowych są różne odmiany ataków odmowy usługi (ang. Denial of Service, DoS). Podstawowym celem takich ataków jest zablokowanie dostępu do określonej usługi lub zasobu. Ataki odmowy usługi można podzielić na dwa podstawowe rodzaje, a mianowicie na ataki powodujące zawieszanie się usług oraz ataki przepełnienia. Ataki DoD powodujące zawieszanie się usług zwykle są ściśle powiązane z konkretnymi programami. Przypominają one bardziej próby włamania do programów niż typowe ataki sieciowe. Odmienne są ataki przepełnienia bufora, bowiem ich celem jest przekierowanie pracy systemu do własnego fragmentu kodu. Nie zawsze się to udaje i taki atak również często się kończy zawieszeniem całego programu.

Wikipedia podaje następującą definicję: „DoS, czyli Denial of Service (ang. odmowa usługi) - atak na system komputerowy lub usługę sieciową w celu uniemożliwienia działania poprzez zajęcie wszystkich wolnych zasobów. Atak polega zwykle na przeciążeniu aplikacji serwującej określone dane czy obsługującej danych klientów (np. wyczerpanie limitu wolnych gniazd dla serwerów FTP czy WWW), zapełnienie całego systemu plików tak, by dogrywanie kolejnych informacji nie było możliwe (w szczególności serwery FTP), czy po prostu wykorzystanie błędu powodującego załamanie się pracy aplikacji.”[13]

Ataki DoS wykorzystują słabości specyfikacji TCP/IP określonego systemu operacyjnego (Ping of Death, Teardrop), słabości standardu TCP/IP – Land, czy też tzw. „brutalną siłę” (ang. brute force) – Smurf [3].

4.3.1. Ping of Death

Atak Ping of Death polega na wykorzystaniu protokołu ICMP niezgodne z jego specyfikacją. Komunikaty echa ICMP mogą zawierać maksymalnie 2^{16}, czyli 65 536 bajtów w obszarze danych pakietu. Okazuję się jednak, że specjalnie spreparowane pakiety o długości przekraczającej wyżej przytoczony limit mogą spowodować zawieszenie się wielu systemów operacyjnych. Proces wysyłania komunikatów echa ICMP (znany jako ping) o bardzo dużej wielkości został nazwany atakiem Ping of Death. Jest to bardzo prosta technika ataku, obecnie często już nieskuteczna, bowiem większość systemów operacyjnych jest na nią uodporniona. W przypadku, kiedy z jakichś powodów nie można zaktualizować systemu, zapewniając jego niewrażliwość na tego typu ataki, należy ustawić odpowiednią regułę na zaporze (ang. firewall) odrzucającą zbyt długie pakiety ICMP.

[13] http://pl.wikipedia.org/wiki/DoS

4.3.2. Teardrop Attack

Teardrop jest popularnym typem ataku DoS, który powoduje zawieszenie się usługi. Wykorzystuje on luki w implementacji procedur łączenia pofragmentowanych pakietów, jakie są wykorzystywane przez wielu dostawców oprogramowania. Dane przesyłane w sieci publicznej przechodzą przez wiele routerów i różnych podsieci, czasami takich w których maksymalny rozmiar ramki może być mniejszy niż rozmiar przesyłanego pakietu. W takiej sytuacji konieczne jest jego podzielenie na mniejsze fragmenty. Przesunięcia fragmentów pakietu, zapisywane są w przesyłanych nagłówkach (offset field), nie mogą się nakładać, co umożliwia odtworzenie pakietu.

Podczas ataku Teardrop wysyłane są fragmenty pakietów z nakładającymi się przesunięciami, co powoduje zawieszanie sie programów, które nie sprawdzają odbieranych pakietów pod tym względem. Zaktualizowane systemy operacyjne radzą sobie z tym typem ataku. Można również zastosować odpowiednią regułę w systemie typu IDS (ang. Intrusion Detection System) potrafiącym wykrywać tego typu atak. Może to być darmowy Snort.

4.3.3. Land Attack

Atak Land należy do klasycznej kategorii ataków DoS polegającej na masowym wysyłaniu pakietów pod adres ofiary (ang. packet flooding). W swojej najprostszej postaci ataki tego typu polegają na zwiększaniu ruchu sieciowego do momentu, w którym atakowane łącze lub serwer nie są w stanie go obsłużyć – oznacza to faktyczne odcięcie dostępu do zasobów legalnym użytkownikom.

Land jest modyfikacją ataku TCP SYN (SYN flood) wykorzystującego mechanizm standardowego połączenia TCP. Rozpoczyna je host inicjujący połączenie wysłaniem do hosta docelowego pakietu SYN, który odpowiada pakietem SYN ACK i oczekuje na pakiet ACK od hosta, który to połączenie zainicjował. Parametry do tego połączenia są zapisywane w kolejce. Ma ona z góry określoną długość i w normalnej sytuacji szybko się opróżnia, bowiem odpowiedź ACK powinna nadejść w przeciągu kilku milisekund. Jednak tak się nie stanie, jeśli w pakiecie inicjującym SYN zostanie umieszczony sfałszowany, losowo wygenerowany adres źródłowy, jak to ma miejsce w ataku TCP SYN. Oczekiwana odpowiedź ACK nigdy nie nadejdzie i proces nawiązywania połączenia, nie zostanie nigdy ukończony. Wpis w kolejce połączeń oczekujących pozostaje około minuty, co jak łatwo przewidzieć doprowadzi do przepełnienia kolejki. W takiej sytuacji kolejne nadchodzące połączenia nie będą obsługiwane, również te pochodzące od legalnych użytkowników.

W przypadku ataku Land adres źródłowy jest również sfałszowany, ale w szczególny sposób. Mianowicie adres źródłowy jest identyczny z adresem docelowym. Wygląda to tak, jakby atak pochodził z wnętrza sieci. Zarówno atak TCP SYN, jak i Land jest skuteczny, jeżeli napastnik znajdzie otwarte porty, na przykład port 13 (daytime) lub 37(time), jak to się dzieje w wielu dystrybucjach Linuxa.

Najlepszym zabezpieczeniem przed atakami tego typu jest uaktualnienie sytemu operacyjnego. Można również skonfigurować filtrację pakietów z poziomu ścian ogniowych lub routerów.

4.3.4. Smurf

Atak Smurf stanowi odmianę ataku ICMP flood, który polega na wysłaniu masowej liczby pakietów echo request (ping) do atakowanego hosta. Wykorzystuje on wysyłanie pakietów ICMP (echo request) na adres broadcastowy sieci pośredniczącej w ataku, ze sfałszowanym adresem źródłowym – jest to adres ofiary. Pakiet ICMP (echo request) wysłany na adres broadcastowy dotrze go każdego komputera sieci pośredniczącej i wywoła odpowiedź echo replay wysłaną na adres ofiary (nie dotyczy to systemów Windows, które nie odpowiadają na broadcastowe ICMP echo request). Rezultatem takiej masowej odpowiedzi może być zator, albo przerwa w działaniu sieci. Oczywiście host będący celem ataku zostanie przeciążony zlewem pakietów, co z reguły zblokuje dostęp do niego. Możliwe jest wzmocnienie ataku Smurf przez synchroniczne wysłanie broadcastowych pakietów ICMP do wielu sieci pośredniczących.

4.3.5. ICMP Destination Unreachable

Atak ICMP Destination Unreachable wykorzystuje właściwości protokołu ICMP, który niestety nie ma wbudowanych żadnych procedur sprawdzania wiarygodności otrzymanych pakietów i dość łatwo może być zmanipulowany. Brak takich procedur stwarza możliwości resetowania połączeń TCP oraz zmniejszania MTU (Maximum Transmission Unit) – maksymalnego rozmiaru pakietów do drastycznie małych wartości.

Komunikaty ICMP mogą być generowane zarówno przez routery pośredniczące jak i hosty. Kiedy router wykrywa problem wysyła do nadawcy pakietu komunikat ICMP, który zostaje obsłużony przez odpowiedzialny za pakiet protokół, który może próbować naprawić problem, zignorować go lub zerwać połączenie.

Komunikaty ICMP dzielą się na tzw. soft errors i hard errors. Po otrzymaniu hard error TCP musi zerwać połączenie, natomiast po soft error komunikacja może być

kontynuowana (RFC 1222). Rodzaj błędu jest zapisany w nagłówku ICMP, w opcjonalnym kodzie, który określa w czym dokładnie tkwi problem.

Kody błędów dla Destination Unreachable to:

– 0 net unreachable: (brak trasy do sieci) soft error

– 1 host unreachable: (brak trasy do hosta) soft error

– 2 protocol unreachable: (nieobsługiwany protokół) hard error

– 3 port unreachable: (nieobsługiwany port) hard error

– 4 fragmentation needed and DF bit set: (pakiet jest zbyt duży, żeby go przetworzyć i ma ustawioną flagę zabraniającą fragmentacji) hard error

– 5 source route failed: (nie udało się ustawić ręcznej trasy do hosta) soft terror

Do zerwania połączenia TCP przy pomocy komunikatu Destination Unreachable wystarczy znajomość źródłowego i docelowego IP oraz portu jednego z hostów. Aby zerwać połączenie atakujący musi wysłać do którejkolwiek ze stron pasujący do numerów portów komunikat ICMP rodzaju hard error.

Przeciwdziałanie atakowi polega na zmianie reakcji TCP na hard error. Mianowicie na traktowaniu hard errors jako soft errors dla nawiązanego połączenia i poleganiu na mechanizmach warstwy wyższej do wykrywania problemów z połączeniem. Tak się zachuje większość aktualnych systemów operacyjnych.

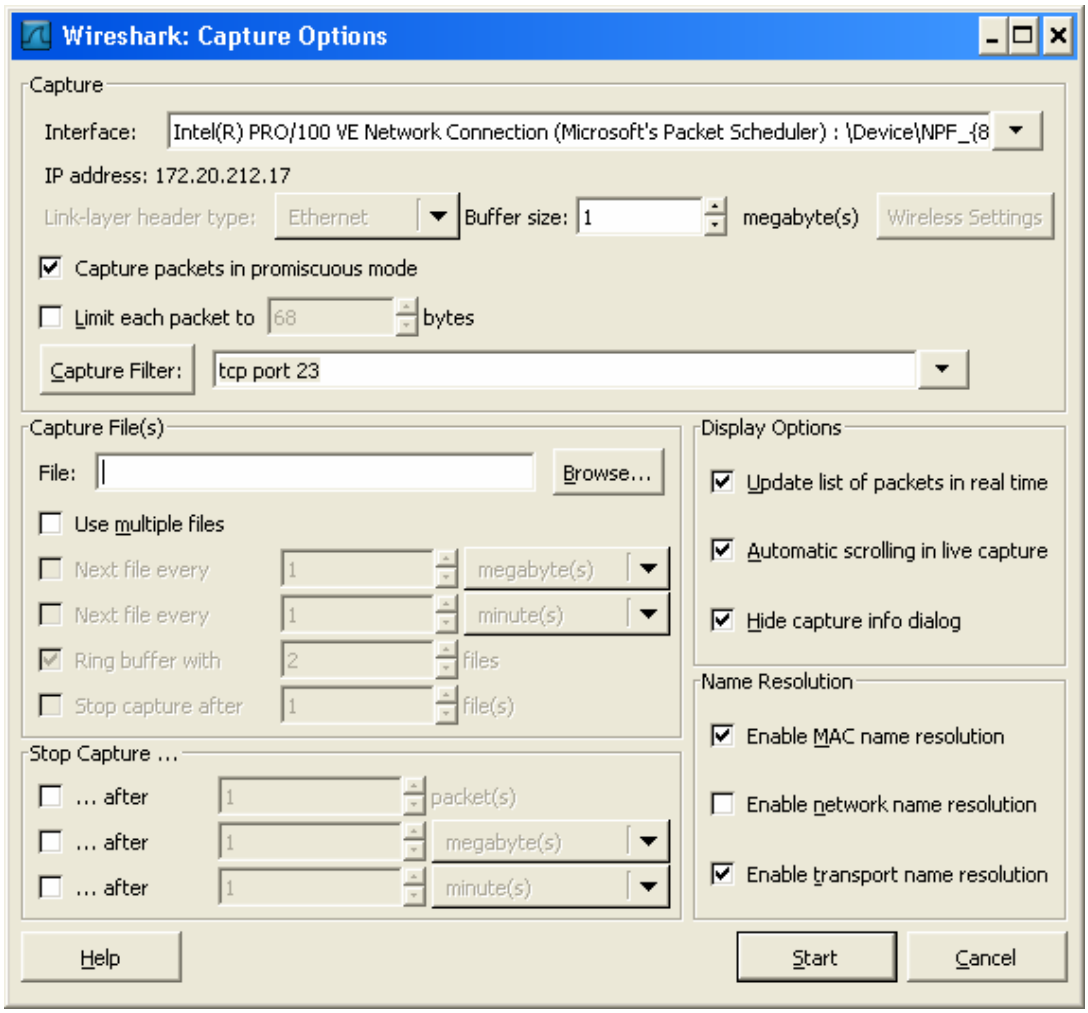
5. Analiza ruchu sieciowego w usługach publicznych bez mechanizmów zabezpieczeń

Ostatni rozdział niniejszej pracy zwiera praktyczne przykłady wykorzystania sniffera Wireshark do analizy ruchu sieciowego w usługach publicznych bez mechanizmów zabezpieczeń na przykładzie Telnetu i FTP. Analiza uwidoczniona na załączonych rysunkach obrazuje wyraźnie, że z powodu uwierzytelniania posługującego się otwartym tekstem są to usługi zapewniające wprawdzie minimalną kontrolę dostępu, ale bardzo łatwe do skompromitowania. Stąd też mogą one być używane tylko w przypadku niskich wymagań bezpieczeństwa, bądź w środowisku, które zapewnia żądany poziom bezpieczeństwa.

5.1. Telnet

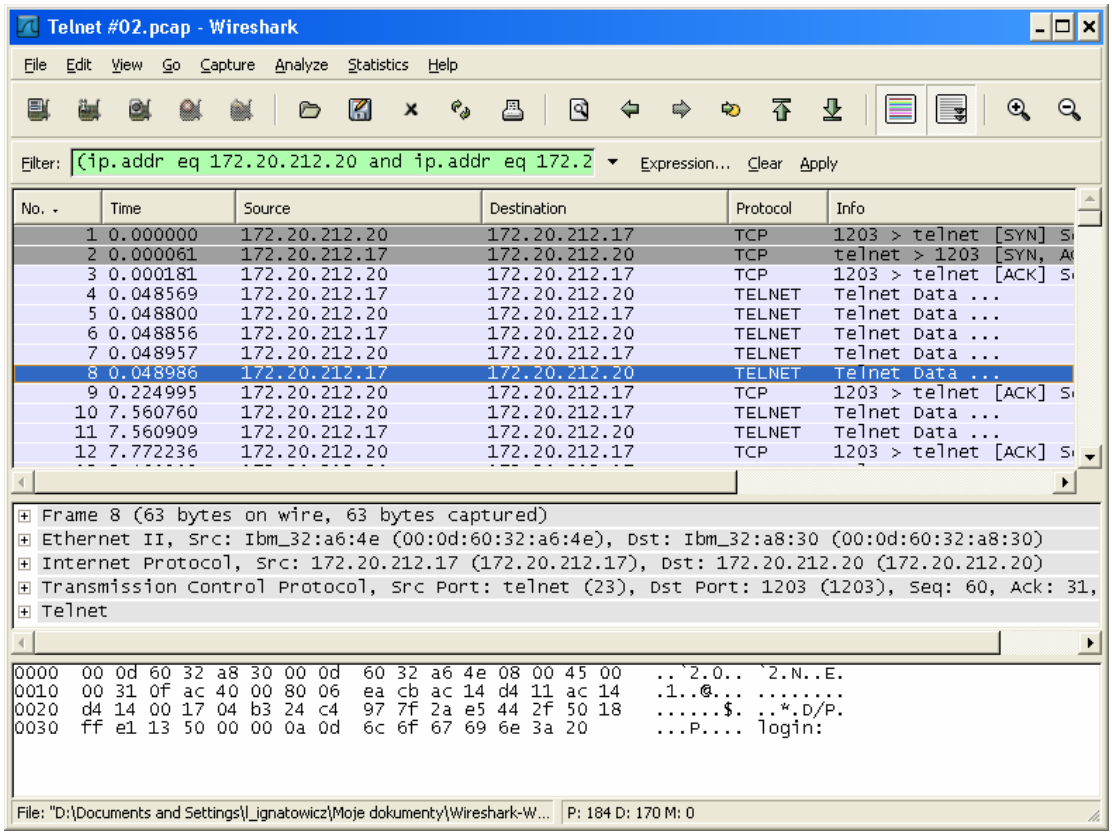
Telnet jest jednym z pierwszych protokołów TCP/IP (obecna specyfikacja jest zawarta w RFC 854). Jest to standardowy protokół aplikacyjny Internetu obsługujący logowanie do zdalnych hostów. Połączenia usługi Telnet muszą być stabilne i niezawodne, dlatego posługuje się ona protokołem transportowym TCP, wykorzystując port 23.

Przechwycenie sesji usługi Telnet zrealizowano w pracowni WWSI wykorzystując do tego celu dwie stacje robocze działające w sieci uczelni. Na jednej z nich (IP: 172.17.212.17) zainstalowano analizator Wireshark oraz uruchomiono usługę Telnet na koncie administratora. Następnie uruchomiono Wiresharka z ustawionym filtrem przechwytywania jak na rysunku 19. Na drugiej stacji roboczej (IP: 172.20.212.20) uruchomiono klienta Telnetu (HyperTerminal) i przeprowadzono sesję połączenia z pierwsza stacją.

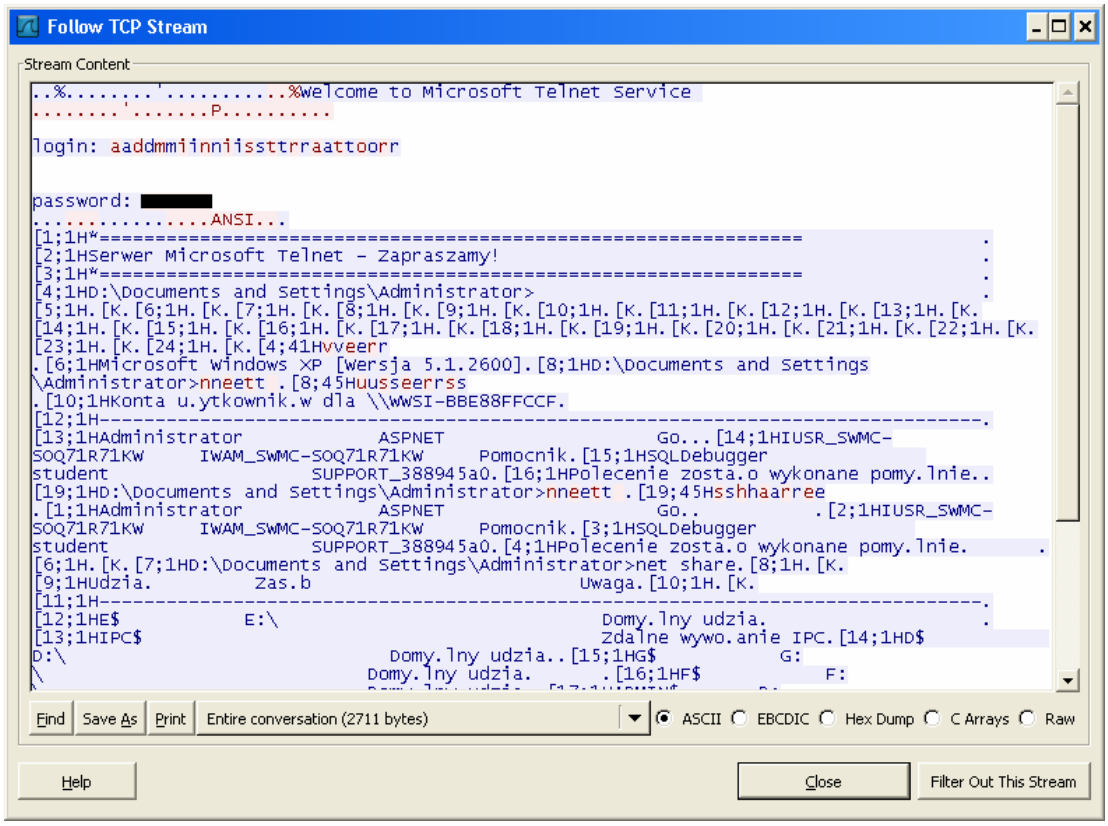


Rys. 19. Okno dialogowe Opcji Przechwytywania z ustawiony filtrem dla Telnetu.

Na rysunku 20 przedstawiono przechwyconą sesję Telnetu. W oknie głównym sniffera Wireshark widać pakiety inicjujące standardowe nawiązanie połączenia TCP oraz pakiety protokołu Telnet – Src Port: telnet (23). Na kolejnym rysunku 21 pokazano okno Śledzenia Strumienia TCP. Widać w nim login (administrator) oraz hasło (na rysunku zaczernione ze względów bezpieczeństwa). Można również odczytać wynik przykładowych poleceń: **net users** oraz **net share**.



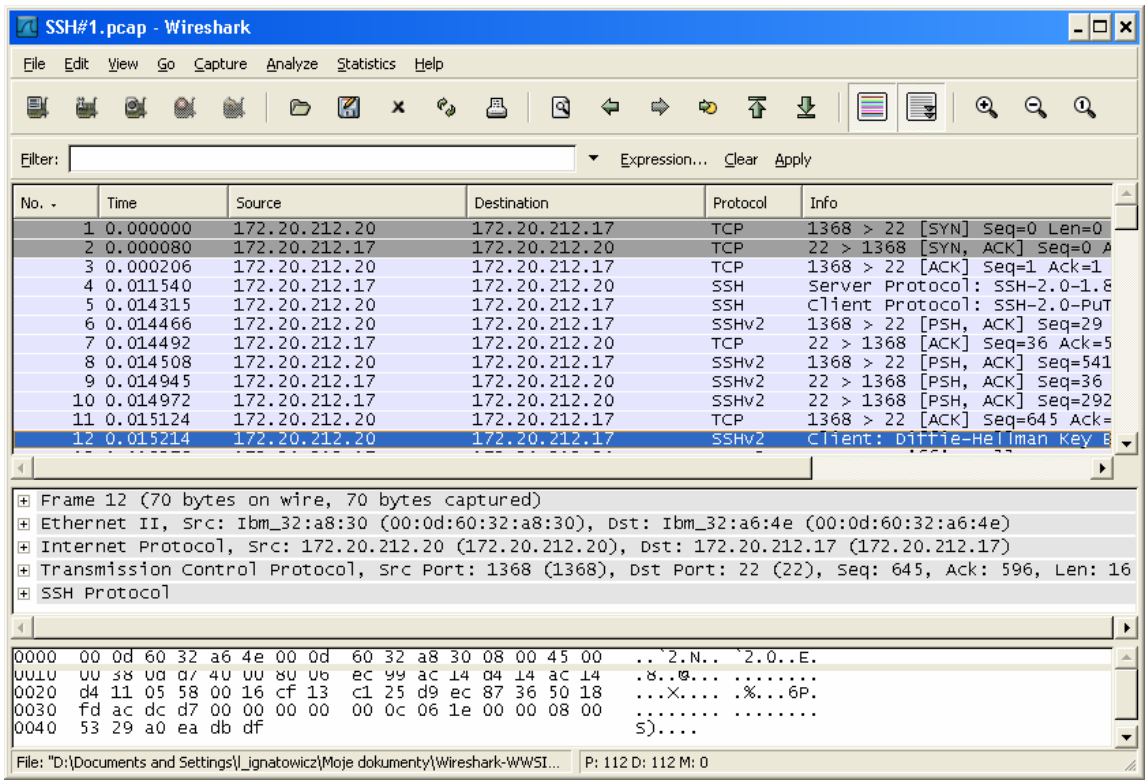
Rys. 20. Okno główne Wiresharka z przechwyconą sesją Telnetu.



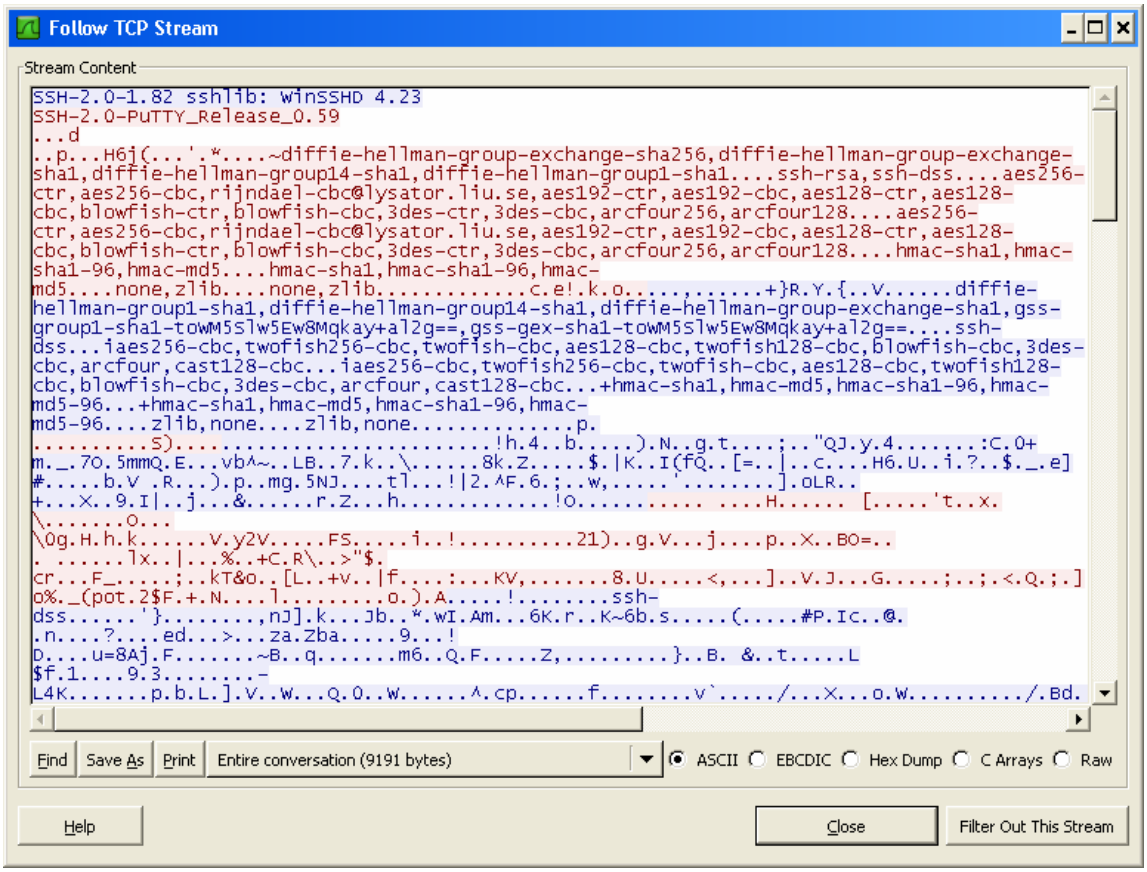
Rys. 21. Okno Śledzenia Strumienia TCP z przechwyconą sesją Telnetu.

Protokół Telnet, jak to widać na rysunku 20, podobnie jak Rlogin w systemach Unixowych jest tak łatwy do podsłuchania, że jego używanie jest ryzykowne. Jak już wspomniano poprzednio najskuteczniejszą metoda ochrony przed skutkami podsłuchu jest zastosowanie szyfrowania ruchu sieciowego. Dla wykazania skuteczności takiej ochrony w zamieszczonym poniżej przykładzie wykorzystano protokół SSH (ang. secure shell). Jako serwer SSH wykorzystano WinSSHD (http://www.bitvise.com/download-area) zainstalowany na tej samej stacji roboczej, na której uruchomiono usługę Telnet. Klientem był program PuTTy (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html).

Na rysunku 22 pokazano przechwyconą sesję SSH. W oknie głównym sniffera Wireshark widać pakiety inicjujące standardowe nawiązanie połączenia TCP oraz pakiety protokołu SSH. Na kolejnym rysunku 23 pokazano okno Śledzenia Strumienia TCP. Oczywiście widać zawartość przechwyconych pakietów, ale jedyne co można odczytać to dane dotyczące serwera i klienta SSH. Wszystkie inne dane, a zwłaszcza login użytkownika i hasło są niedostępne. Tak więc, pomimo, iż szyfrowanie nie zabezpiecza przed sniffingiem, to zdecydowanie zabezpiecza przed jego negatywnymi skutkami, bowiem napastnik nic nie jest w stanie odczytać z przechwyconych danych.



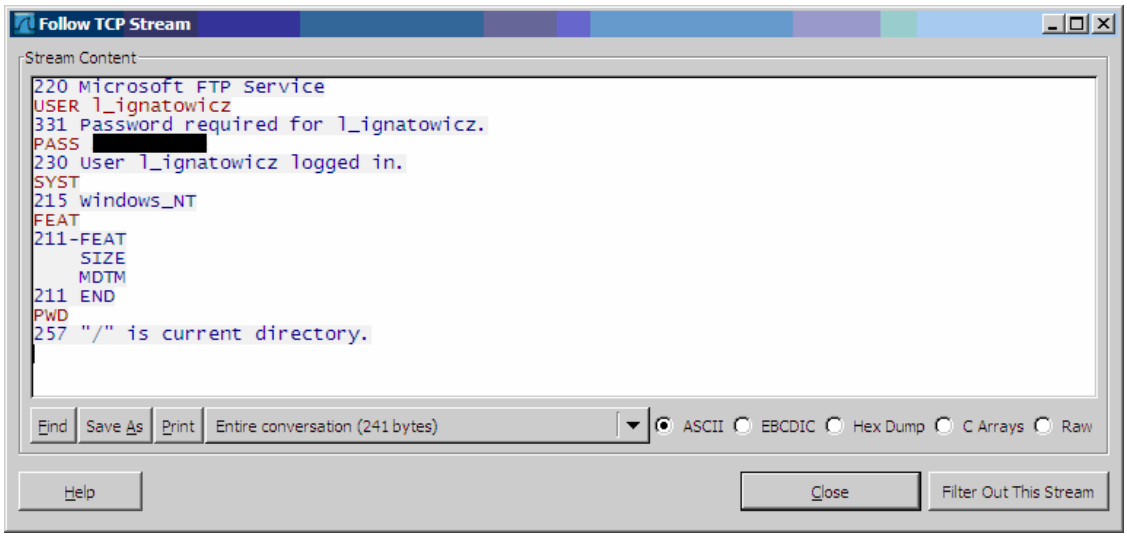
Rys. 22. Okno główne Wiresharka z przechwyconą sesją SSH.



Rys. 23. Okno Śledzenia Strumienia TCP z przechwyconą sesją SSH.

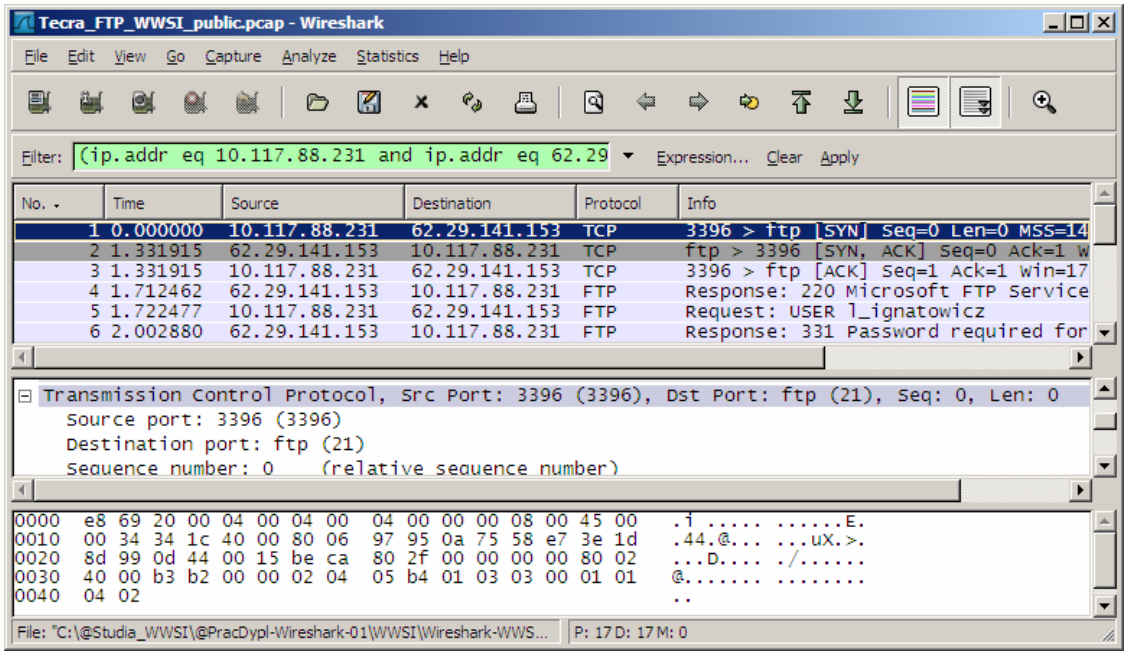
5.2. FTP

Protokół FTP jest standardem przesyłania plików w sieci TCP/IP zdefiniowanym w RFC 959. Wykorzystuje protokół transportowy TCP oraz dwa porty – 20 dla danych i 21 do sterowania.



Rys. 24. Okno Śledzenia Strumienia TCP z przechwyconą sesją FTP.

Na rysunku 25 pokazano przechwyconą przy pomocy Wiresharka (ustawiony filtr przechwytywania: tcp port 21) sesję FTP. Natomiast na rysunku 24 pokazano okno Śledzenia Strumienia TCP. Widać w nim login (l_ignatowicz) oraz hasło (na rysunku zaczernione ze względów bezpieczeństwa). Podobnie jak Telnet, FTP nie jest bezpiecznym protokołem i podobne jest również rozwiązanie tego problemu. Tak jak zamiast Telnetu warto używać szyfrowanego SSH, tak zamiast FTP można użyć szyfrowanego SFTP (ang. SSH File Transfer Protocol) w przypadku, kiedy wymagają tego względy bezpieczeństwa.



Rys. 25. Okno główne Wiresharka z przechwyconą sesją FTP.

6. Podsumowanie

Poruszony w niniejszej pracy temat monitorowania bezpieczeństwa sieci został zawężony, ze względu na zaplanowaną objętość pracy, do tematyki wykrywania zagrożeń bezpieczeństwa przy pomocy Open Source'owego sniffera Wireshark. Trzeba podkreślić, że nie zastępuje on systemów typu IDS (ang. Intrusion Detection System) bądź IPS (ang. Intrusion Prevention System), lecz może być ich wartościowym uzupełnieniem.

Zaprezentowane w pracy cechy i właściwości analizatora Wireshark, poparte praktycznymi przykładami jego działania, pozwalają na stwierdzenie, że jest to narzędzie przydatne dla administratora sieci, zarówno do celów pogłębiania wiedzy na temat ciągle ewoluujących zagrożeń bezpieczeństwa sieci teleinformatycznych, nabywania umiejętności w dziedzinie ich analizy, a przede wszystkim do skutecznego ich wykrywania.

7. Wykaz literatury

1. C. Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems,* San Francisco 2007, No Starch Press, Inc.
2. B. R. Jones, *Network Security: An Open-Source Approach*, Internet 2005 www.infosecwriters.com/text_resources/pdf/Open-Source-Approach.pdf
3. M. Szmit, M. Gusta, M. Tomaszewski, *101 zabezpieczeń przed atakami w sieci komputerowej*, Gliwice 2005, Wydawnictwo HELION
4. A. Orebaugh, *Ethereal Packet Sniffing,* Rockland 2004, Syngress Publishing, Inc.
5. C. McNab, *Network Security Assessment*, Sebastopol, CA 2004, O'Reilly Media, Inc.
6. R. Bejtlich, The Tao of Network Security Monitoring Beyond Intrusion Detection, Indianapolis, Indiana 2004, Addison Wesley