Ewelina Jasińska
Wydział Informatyki i Zarządzania
Wyższej Szkoły Informatyki w Łodzi

Promotor: dr hab. Marek Rudnicki, prof. WSInf

ATAKI I WŁAMANIA DO SIECI BEZPRZEWODOWYCH

Streszczenie – Artykuł przybliża problematykę ataków i włamań do sieci bezprzewodowych. Dziś, gdy z dobrodziejstw sieci bezprzewodowych może korzystać każdy, warto spojrzeć na problem ich zabezpieczenia z punktu widzenia zwykłego użytkownika. Jest to bowiem nie tylko kwestia wyboru, ale umiejętności, podstawowej wiedzy i narzędzi do sprawnego wykorzystania, także dla powszedniego „zjadacza chleba".

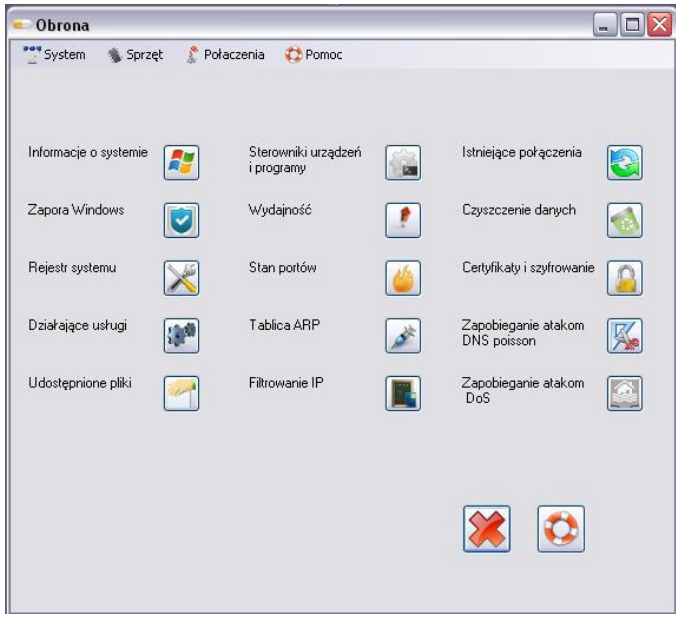
1 Wprowadzenie

Od lat 50 – tych XX – ego wieku, kiedy po raz pierwszy wykorzystano sygnał radiowy do przesłania wiadomości, sieci bezprzewodowe fascynowały swoją niezależnością i mobilnością. Możliwość wymiany informacji, bez oplątanych kablami i przykutymi do gniazdka w ścianie urządzeń, zapowiadała nową jakość życia. Rzeczywistość przesyłu bezprzewodowego jest dużo mniej ciekawa, niż obietnice swobody i wolności; korzystając z fal radiowych jako nośnika informacji musimy się pogodzić z ich dostępnością, bo czyż można zamknąć powietrze? Tematyka włamań do sieci bezprzewodowych jest nieodmiennie fascynująca; dotyka problemów z jakimi dziś boryka się Internet – swobody korzystania z ogólnodostępnych źródeł. Czy korzystanie z fali radiowej jest przestępstwem? Czy odczyt wiadomości unoszącej się w powietrzu jest naruszeniem wolności osobistej? Gdzie postawić granicę między powszechnością a poufnością?

2 Założenia projektowe działania programu OBRONA

Program OBRONA został napisany w języku Visual Basic .NET, przy wykorzystaniu platformy programistycznej .NET Framework w wersji Microsoft Visual Basic 2005 Express Edition. Zapobieganie ataków

sieciowych opiera się o analizę systemu operacyjnego Windows XP, jednak w przypadku starszych wersji systemów firmy Microsoft, większość funkcji pozostaje niezmieniona co pozwala użytkować program także na innych platformach. Aplikację uruchamiano na komputerach z następującymi wersjami systemu: Windows 2000 SP1, Windows XP samodzielnym jak i z dodatkami w postaci SP2 i SP3. W przypadku systemu Windows Vista, aplikacja także działa stabilnie, platforma ta odziedziczyła większość funkcji po starszych wersjach systemu. Różnice mogą się zawierać w nazwach apletów systemowych wywoływanych przez program, jednak wszystkie polecenia pozostały niezmienione. Choć przeznaczeniem aplikacji jest działanie na stacjach roboczych, to podobnie w przypadku systemów serwerowych jej funkcjonowanie przebiegało bez zakłóceń. Funkcje aplikacji działały bez zarzutu wykonując zlecone działania w Windows 2000 Serwer, Windows Serwer 2003.



Rys. 1. Menu główne programu OBRONA

Główną częścią kodu programu, pozwalającą na zarządzanie zasobami systemowymi, jest klasa „*comDial*”, która za pomocą zmiennej „*psi*” wprowadza komendę do konsoli:

```
Public Class cmdDial
Dim command As String
    Public Sub SetCommand(ByVal command2 As String)
```

```
'przypisanie zmiennej command2 do zmiennej string
        command = command2
    End Sub
    Private Sub cmdDial_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
'deklarowanie zmiennej psi jako elementu obszaru nazw
System Diagnostic zawierającej metodę składni komend
w konsoli Windows:
        Dim psi As
System.Diagnostics.ProcessStartInfo = New
System.Diagnostics.ProcessStartInfo("cmd.exe", "/C "
+ command + " > c:\test.txt")
'przekierowanie ze standardowego wyjścia konsoli na
ukryte okno systemu
        psi.RedirectStandardOutput = False
        psi.WindowStyle =
System.Diagnostics.ProcessWindowStyle.Hidden
        psi.UseShellExecute = False
'przekazanie informacji o stanie procesu do zmiennej
proces, zadeklarowanej jako część System Diagnostics
o metodzie zmiennej psi
        Dim proces As System.Diagnostics.Process
        proces =
System.Diagnostics.Process.Start(psi)
' czas oczekiwania na przekazanie informacji
określony na 3 minuty
        proces.WaitForExit(30000)
'deklaracja zmiennej wynik jako ciągu tekstowego
        Dim wynik As String
'wyczyszczenie okna
        wynik = String.Empty
'warunek zakładający działanie zmiennej proces
        If proces.HasExited Then
'deklaracja zmiennej czytacz, służącej do
przechowywania informacji zebranych przez konsolę i
tworzącą plik tekstowy w lokalizacji c:\ pozwalający
na przekierowanie informacji do TextBoxu
umieszczonego w formacie comDiag
            Dim czytacz As System.IO.StreamReader
            czytacz = New
System.IO.StreamReader("c:\test.txt")
            wynik = czytacz.ReadToEnd()
            TextBox1.Text = wynik
            czytacz.Close()
```

```
        End If
    End Sub
```

Uruchomienie okna systemowego z grupy control, zarządzającego np. *Panelem sterowania* w Windows, zostało obsłużone za pośrednictwem klasy o nazwie „*Uruchamiacz*”:

```
Public Class Uruchamiacz
'deklaracja zeminnych tekstowych
    Private command As String
    Private parametry As String
'wykorzystanie zmiennych w nowym formacie
    Public Sub New(ByVal command As String, ByVal
parametry As String)
        Me.command = command
        Me.parametry = parametry
    End Sub
'wykorzystanie zmiennej psi z klasy comDiag do
wywołania linii poleceń o argumentach zmiennych typu
string:
    Public Sub Uruchom()
        Dim psi As
System.Diagnostics.ProcessStartInfo = New
System.Diagnostics.ProcessStartInfo(command,
parametry)
'przekazanie informacji o stanie procesu
        Dim proces As System.Diagnostics.Process
        proces =
System.Diagnostics.Process.Start(psi)
    End Sub
End Class
```

Za pomocą podobnych mechanizmów, program pozwala na modyfikację rejestru, tablic routingu, wypisów w tablicy ARP, testowania wydajności czy wyświetlenia informacji o systemie.

3 Typy włamań i metody ich zapobiegania wykorzystane w programie OBRONA

Można wyróżnić cztery podstawowe fazy każdego ataku:

- *sniffing* - zbieranie informacji (rodzaj systemu operacyjnego, typy zabezpieczeń, ilość użytkowników, numery IP, struktura sieci);

- *przejęcie sesji* – (*session hijacking*) - uzyskanie dostępu do sieci za pomocą luki w systemie zabezpieczeń;

- *spoofing* - nadanie uprawnień istniejącego klienta sieci;

- *odmowa dostępu* (*denial of service*) – ta kategoria jest odmienna, ze względu na cel, jakim jest zablokowanie wszystkich użytkowników sieci i całkowite uniemożliwienie dalszego działania

Sniffing - monitorowanie komunikacji i zbieranie informacji (rodzaj systemu operacyjnego, typy zabezpieczeń, ilość użytkowników, numery IP, struktura sieci) np. poprzez skanowanie portów. Zwykle *sniffing* jest określany jako metoda bierna lub *„wstęp”* do włamania. Najpopularniejszymi programami do przechwytywania ruchu w sieci są: *dSniff,* oraz *Ethereal*, *Kismet* czy *NetStumber.*

Elementem systemu pozwalającym na przeglądanie zasobów sieciowych i ustawień użytkownika jest protokół NetBIOS, działający w ten sam sposób od czasu Windows`a NT. NetBIOS to protokół sieciowy stworzony przez Microsoft dla małych sieci, pozwalający komunikować się komputerom w jednym jej segmencie (typu *non-routable*). W przypadku sieci WiFi, także ma to znaczenie, ponieważ w momencie utworzenia połączenia z siecią, zasoby są dostępne dla każdego użytkownika bez kontroli. Oprócz tego, podczas instalacji systemu Windows, tworzone są automatyczne udziały dla wszystkich dysków logicznych np. C$ dla dysku c:\, oraz dodatkowo dla katalogu %SystemRoot%, tworzony jest udział %Admin%. Ponieważ korzystanie z zasobów dyskowych umożliwia automatyczne logowanie, trudno w tym wypadku mówić o włamaniu, zwłaszcza gdy administrator nie określi reguł hasła. Dodatkowym udziałem jest IP$ - podział połączenia, wykorzystywany w ataku nazywanym „zerową sesją”. Aby zalogować się bez podawania hasła na konkretny komputer, wystarczy wpisać polecenie:

```
net use \\ numer_IP_komputera\IPC$ “” /u: “”
```

gdzie “ ” oznacza puste hasło. Za pośrednictwem programu można zapobiegać pustym sesjom poprzez filtrowanie ruchu sieciowego, modyfikacje rejestru oraz śledzenie udostępnionych udziałów. W aplecie *Udostępnione pliki, z*apobieganie enumeracji sieci, oraz zmiana stanu udostępniania plików, zostały zaprogramowane przy pomocy następującego kodu:

```
'dodanie klucza rejesrtu blokującego dostęp
nieznanych użytkowników:
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SYS
TEM\CurrentCOntrolSet\Control\LSA",
True).SetValue("restrictanonymous", 1)
'Zakończenie udostępniania wszystkich plików:
        Dim okno1 As New cmdDial()
        okno1.SetCommand("openfiles /disconnect /id
*")
'Wyświetlenie zawartości klucza informacji NetBIOS:
```

```
        Dim klucz As Array
        klucz =
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SYS
TEM\CurrentControlSet\Control\SecurePipeServers\winre
g\AllowedPaths").GetValue("Machine")
        Dim klucza As String
        klucza = String.Empty
        For Each o As Object In klucz
            klucza += o + ControlChars.NewLine
        Next
        MsgBox("Lista udostępnionych informacji za
pomocą: " & klucza, , "Informacja")
'Zablokowanie udostępniania:
        Dim potwier As Integer
        potwier = MsgBox("Ten przycisk wprowadzi
zmiany do Twojego rejestru. Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If potwier = 1 Then
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SYS
TEM\CurrentControlSet\Services\lanmanserver\parameter
s", True).SetValue("AutoShareServer", 0)
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SYS
TEM\CurrentControlSet\Services\lanmanserver\parameter
s", True).SetValue("AutoSharewks", 0)
        End If
```

Dzięki zastosowaniu programu, komputer nie odpowiada na zapytania programów skanujących sieć, blokuje możliwość zalogowania się na systemowo utworzone konta, a także zrywa połączenia do otwartych udziałów.

Przejęcie sesji – (*session hijacking*) uzyskanie dostępu do sieci za pomocą luki w systemie zabezpieczeń i przejęciu części komunikacji między serwerem a klientem, symulujące jej uczestnika. Wykorzystywane techniki to:

- ***route poisoning*** (zatrucie tablic trasowania) - przejęcie tożsamości jest możliwe dzięki kradzieży pakietów po uwierzytelnieniu, czyli uznaniu klienta przez serwer. Aby tego dokonać hacker musi należeć do tej samej warstwy sieci co klient albo serwer, oraz przeprowadzić modyfikacje trasy pakietów. W tablicach trasowania zawarte są informacje dotyczące węzłów i sieci lokalnych zawierające najlepszą trasę przesyłu pakietu do konkretnego źródła. Atakujący musi zmienić zawartość tablic, wskazując swój adres jako miejsce, przez które będzie przebiegało połączenie. W tym celu używane są do trasowanie zmienione pakiety ICMP (*Internet Control Message Protocol*); co

umożliwia nawet wysyłanie zmodyfikowanych aktualizacji programom, czy podszycie się pod dowolną witrynę.

Protokół ICMP (*Internet Control Message Protocol*) pełni rolę kontrolera transmisji sieciowej, zgłasza błędy w połączeniach sieciowych, informuje o zbyt dużym obciążeniu buforów routera, zmianach w tablicy routingu (wykorzystywanych przy zatruciu tablic trasujących routera), czy niedostępności adresata. Atak ICMP jest typowym atakiem typu DoS, polegającym na zalaniu hosta komunikatami o braku dostępu do punktu docelowego; aby temu zapobiec najlepiej w przypadku klientów wyłączyć działanie kontrolera za pomocą kodu:

```
'Zapobieganie atakom na ICMP
     Dim pierw As Integer
        pierw = MsgBox("Ten przycisk wprowadzi zmiany
do Twojego rejestru.Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If pierw = 1 Then
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\TcpIp\Parameters",
True).SetValue("EnableICMPRedirect", 0)
        End If
```

Luka w protokole SNMP została odkryta przez hakerów z grupy GNUCitizen; wysyłając zapytania SNMP zebrali listę nazw i modeli sprzętu, typów systemów operacyjnych oraz jego poprawek. Słabością protokołu SNMP korzystającego z datagramów UDP jest możliwość sfałszowania adresu źródłowego pakietu, co nie przeszkadza uzyskaniu odpowiedzi od komputera za pomocą ustawień protokołu NetBIOS, udostępniających informacje. Zapobiec atakom tego typu może kod:

```
'Zapobieganie atakom na SNMP
        Dim drugi As Integer
        drugi = MsgBox("Ten przycisk wprowadzi zmiany
do Twojego rejestru.Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If drugi = 1 Then
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("EnableDeadGWDetect", 0)
        End If
```

- zatrucie tablicy ARP – ARP spoofing - wykorzystywane z sieciach z włączonym filtrowaniem adresów MAC; atak polega na sfałszowaniu

tablicy ARP użytkownika bądź serwera, przez wysyłanie fałszywych pakietów zawierających adres MAC włamywacza, co pozwala przechwycić pakiety (zawierające hasła, dane osobiste), kierowane do pośredniczącego w komunikacji serwera.

By przyspieszyć wykrywanie komputerów, ich dane są przechowywane w lokalnej tablicy adresowej i używane do komunikacji w sieci; do przechwycenia ruchu sieciowego między klientem a serwerem wystarczy zmieniona odpowiedź ARP zawierająca numer IP serwera i numer MAC komputera hakera, co zmieni zapisane dane w tablicy. Odtąd cały ruch do serwera będzie przekazywany przez komputer hakera, który może zatruć tablice ARP także na serwerze, działając jednocześnie wewnątrz i „na zewnątrz" jako klient i serwer. Technika ta działa głównie w przypadku hostów przewodowych, a więc dotyczy hostów poza punktem dostępowym (ale w przypadku masowego ataku, możliwe jest przejęcie kontroli nad ruchem w sieci i przewodowej i bezprzewodowej). W obu przypadkach, włamywacz ma dostęp fizyczny do sieci. Aplet *Tablice ARP* pozwalają prześledzić trasy pakietów, obejrzeć tablicę ARP i wypisy *routingu*, za pomocą następującego kodu:

```
'Wypisy tablicy ARP:
        Dim okno1 As New cmdDial()
        okno1.SetCommand("arp -a")
        okno1.Text = "Tablica ARP"
        okno1.Show()
'Wykazanie tras routingu
        Dim okno1 As New cmdDial()
        okno1.SetCommand("netstat -r")
        okno1.Text = "Trasy routingu"
        okno1.Show()
'Śledzenie trasy dla adresu docelowego www.wp.pl
        Dim okno1 As New cmdDial()
        okno1.SetCommand("tracert www.wp.pl")
        okno1.Text = "Śledzenie pakietów"
        okno1.Show()
'Śledzenie pakietów i start na punktach
pośredniczących dla adresu docelowego www.wp.pl
        Dim okno1 As New cmdDial()
        okno1.SetCommand("pathping www.wp.pl")
        okno1.Text = "Śledzenie trasy"
        okno1.Show()
```

- Man In the Middle – technika stosowana w połączeniach szyfrowanych, umożliwia przechwycenie komunikacji za pomocą

serwera odpowiadającego na zapytania użytkownika; np. wybierając adres swojej skrzynki pocztowej, użytkownik jest przekonany że nawiązane połączenie jest szyfrowane, podczas gdy połączył się z komputerem hakera. Na tym poziomie, możliwe jest przechwycenie certyfikatów służących do uwierzytelnienia klienta np. kluczy szyfrujących dane.

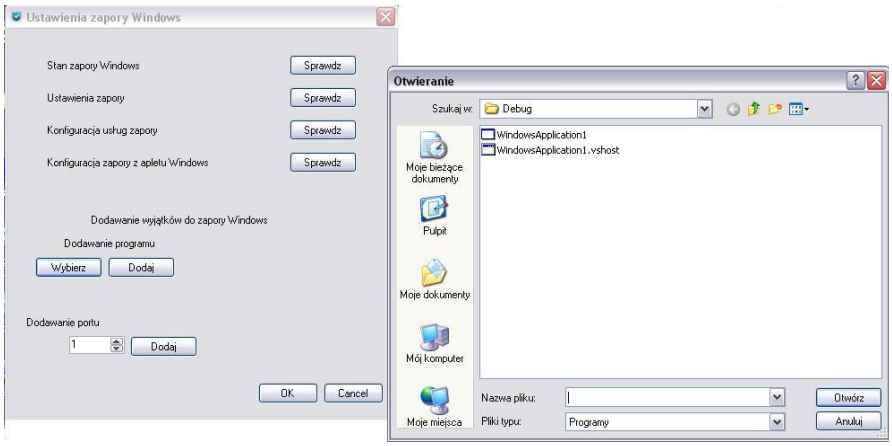
Za pośrednictwem programu można dodać adresy do tablic routingu i zaufanych połączeń systemu:

```
Dim ipAdres As System.Net.IPAddress =
System.Net.IPAddress.None
' Warunek wpisania adresu
    If
System.Net.IPAddress.TryParse(MaskedTextBox1.Text,
ipAdres) Then
            If TextBox1.Text.Length > 0 Then
                Dim strumien As
System.IO.StreamWriter = New
System.IO.StreamWriter("c:\windows\system32\drivers\e
tc\hosts", True)
                strumien.Write(ControlChars.NewLine +
MaskedTextBox1.Text + " " + TextBox1.Text)
                strumien.Close()
            Else
                MessageBox.Show("Nie podałeś nazwy
hosta", "Błąd", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End If
'Warunek podania nazwy hosta
            Console.WriteLine(MaskedTextBox1.Text)
        Else
            MessageBox.Show("Podałeś zły adres",
"Błąd", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
```

Najczęściej wykorzystywane przez włamywaczy oraz malware adresy, które mogą znaleźć się wśród połączeń istniejących to tzw. adresy nieroutowalne, takie jak: 10.*.*.*, od 172.16.0.0 – 172.31.255.255, czy 192.168.*.*; często też pojawiają odwołania do adresów zarezerwowanych przez IANA takich jak: 0.0.0.0/8, 1.0.0.0/8, 2.0.0.0/8, 5.0.0.0/8, 7.0.0.0/8, 23.0.0.0/8, 31.0.0.0/8, 36.0.0.0/8, 37.0.0.0/8, 39.0.0.0/8, od 41 do 42.0.0.0/8, od 58 do 60.0.0.0/8, 67.0.0.0/8 do 127.0.0.0/8, 219.0.0.0/8 do 223.0.0.0/8, 240 do 255.0.0.0/8. obudzić czujność użytkownika powinno istnienie połączeń

nasłuchujących na niestandardowych portach, np. Trojan o nazwie *backdoor* otwierał połączenia na portach 1234 oraz 27374.

Istotnym elementem ochrony systemu Windows XP, instalowanym wraz z dodatkiem SP2, jest systemowa zapora, umożliwiająca dodawanie wyjątków w postaci programów i portów.



Rys. 2. Okno dodawania programów do wyjątków zapory Windows.

Formant *Ustawienia zapory Windows* (Rysunek 2), sprawdza stan usługi systemowej, pozwala na konfigurację oraz dodawanie wyjątków do firewalla. W celu dodania do zapory wyjątku w postaci programu lub portu należy kliknąć przycisk wyboru, uruchamiający okno zasobów komputera, a następnie wskazać dany plik z rozszerzeniem .exe; w przypadku portów wystarczy wpisać, bądź wybrać za pomocą pola numerycznego, określoną liczbę i potwierdzić wybór klikając na przycisk *Dodaj.* Wprowadzone zmiany można sprawdzić ponownie wybierając przycisk stanu i konfiguracji. Formant zarządza usługą zapory systemowej, pozwalając na zautomatyzowanie dodawania wyjątków, dzięki implementacji:

```
'Kod pozwala wybrać jedynie  pliki uruchamialne:
        Dim ofd As OpenFileDialog = New
OpenFileDialog()
        ofd.Filter = "Programy|*.exe"
        If ofd.ShowDialog() =
System.Windows.Forms.DialogResult.OK Then
            Label8.Text = ofd.FileName
        End If
'Kod dodający dany proogram wybrany przez użytkownika
do wyjątków zapory:
```

```
        If Label8.Text.Contains("\") Then
            Dim okno1 As New cmdDial()
            Dim path As String = Label8.Text
            Dim poczatekNazwy = path.LastIndexOf("\")
+ 1
            Dim dlugoscNazwy = path.Length -
poczatekNazwy
      Dim name As String =
path.Substring(poczatekNazwy, dlugoscNazwy)
            Console.WriteLine(path)
            Console.WriteLine(name)
            okno1.SetCommand("netsh firewall add
allowedprogram " & path & " " & name & " Enable")
            okno1.Show()
            Label8.Text = String.Empty
        End If
'Kod dodający wybrany numer portu do wyjątków zapory:
        Dim okno1 As New cmdDial()
        okno1.SetCommand("netsh firewall add
portopening all " & NumericUpDown1.Value & " Obrona-"
& NumericUpDown1.Value & " Enable")
        okno1.Show()
```

Dbając o bezpieczeństwo należy pamiętać o śledzeniu tras pakietów, co umożliwia orientacyjne rozpoznanie kluczowych elementów docelowych. Trasa pakietów za każdym badaniem może być inna, co fałszuje zliczanie TTL(*Time To Live*) trasy. Jednak nawet szkicowe zaznaczenie trasy pozwoli wykluczyć atak typu MITM, gdy narastająca liczba punktów drogi jest taka sama. By prześledzić trasy oraz obejrzeć trwałe wpisy w tablicy należy wybrać ustawienie *Śledzenie trasy*, bądź *Tablica ARP.*

Najczęściej zdarzającym się typem ataku, zarówno na pojedynczego hosta jak i AP, czy router są zatrucia DNS, wykorzystujące automatyczne podawanie adresów domenowych; np. *DNS spoofing* i *DNS poisoning* (fałszowanie odpowiedzi lub zalanie nieprawdziwymi adresami pytającego zanim DNS, do którego zostało skierowane zapytanie, odpowie). Oba typy zasadzają się na dynamicznie zmienianych tablicach serwerów, które bezwiednie rozsyłają dalej szkodliwe informacje. Szczyt inwazji na serwery DNS, zapoczątkowany został w 2008 roku, kiedy Dan Kaminsky odkrył lukę umożliwiającą zatrucie pamięci *cache* serwera, pozwalającą nie tylko na podmianę adresu, ale także na „doklejenie" dodatkowego rekordu (*Additional RR*) zawierającego fałszywy numer IP[1]. Przeszukiwanie list oraz testowanie

[1] http://iname.pl/2008/07/dns-pl-analiza-problemu-dns-cache-poisoning/

używanego serwera pod kontem wyżej wspomnianej luki, umożliwia zestaw instrukcji:

```
' Czyszczenie pamięci cache serwera DNS
        Dim okno1 As New cmdDial()
        okno1.SetCommand("ipconfig /flushdns")
        okno1.Text = "Czyszczenie pamięci DNS"
        okno1.Show()
' Wyświetlenie inforamcji o wszystkich serwerach DNS
        Dim okno1 As New cmdDial()
        okno1.SetCommand("nslookup set all")
        okno1.Text = "Informacje o DNS"
        okno1.Show()
' Zatrzymywanie i wznawianie pracy serwera
        Dim okno1 As New cmdDial()
        okno1.SetCommand("net start dns")
        okno1.Text = "Włączenie DNS"
        okno1.Show()
        Dim okno1 As New cmdDial()
        okno1.SetCommand("net stop dns")
        okno1.Text = "Wyłączenie DNS"
        okno1.Show()
'Lista skonfigurowanych serwerów
        Dim okno1 As New cmdDial()
        okno1.SetCommand("ipconfig /displaydns")
        okno1.Text = "Skonfigurowane serwery DNS"
        okno1.Show()
'Testowanie serwera na zabezpieczenia ataku zatrucia
cache DNS (luki Kaminskyego)
        Dim okno1 As New cmdDial()
        okno1.SetCommand("nslookup -type=txt -
timeout=30 porttest.dns-oarc.net ns1.your-isp.com")
        okno1.Text = "Testowanie serwera DNS"
        okno1.Show()
```

- MAC flooding - polegający na zalaniu tablicy pamięci CAM (*Content Addressable Memory*), wpisami adresowymi i doprowadzeniu do zapchania tablicy. Switch rozpocznie przekazywanie pakietów na wszystkie porty poza tym, z którego otrzymał ramkę (zmieni się w hub`a), co ułatwi *sniffing* i przechwytywanie pakietów. W celu przechwycenia sesji wykorzystuje się najczęściej dwa programy: *juggernraut* lub *Ettercap.*

Aplet programu: *Stan portów,* informuje o połączeniach ustanowionych, zamkniętych lub nasłuchujących; duża liczba tych

ostatnich powinna zaalarmować użytkownika. Półotwarte połączenia są charakterystyczne dla ataków DoS i świadczą o działaniu programu oczekującego poleceń z zewnątrz. Większość programów wymaga, co najmniej jednego portu dla poprawnego funkcjonowania; jeśli połączenie ustanowione wybierane jest przez program o nieznanej tożsamości lub jeśli program przekracza wykorzystanie dopuszczalnej liczby portów, możemy podejrzewać multiplikacje procesów, a co za tym idzie istnienie w systemie konia trojańskiego.

Jedną z technik przysyłania ruchu sieciowego za pomocą router`a jest NAT (*Network Address Translation*), zmieniający adresy źródłowe, czasem również docelowe, oraz nazwy portów na postać zamaskowaną. Zaletą korzystania z mechanizmu, oprócz oszczędności adresów IP, jest anonimowość. Często odpowiedzi NAT są przeszukiwane przez agresorów, przy wykorzystaniu źródłowego IP routera, w poszukiwaniu informacji dotyczących typologii sieci; aby uodpornić protokół IP przed próbami automatycznych odpowiedzi zaleca się wprowadzenie do rejestru następujących zmian:

```
'Ustawienia zabepzieczenia NAT
        Dim k As Integer
        k = MsgBox("Wciśnięcie klawisza spowoduje
zmiany w rejestrze, czy chcesz kontunuowac?",
MsgBoxStyle.OkCancel, "Uwaga")
        If k = 1 Then
'wyłączenie routingu IP dla wysyłania źródłówego
adresu IP:
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("DisableIPSourceRouting", 1)
'ochrona przez atakaiem typu multicast floodiing, I
automatycznymi odpowiedziami na multicast`y
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("EnableMulticastForwarding", 0)
'zablokowanie wysyłania pakietów pomiędzy sieciami,
za wyjątkiem firewalla
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("IPEnableRouter", 0)
'kontrolowanie odpowiedzi na zapytania ICMP
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("EnableAddrMaskReply", 0)
        End If
```

Dzięki zastosowanym zmianom, komputer jest odporny na ataki typu ICMP backdoor, czy programy typu meterpreter.

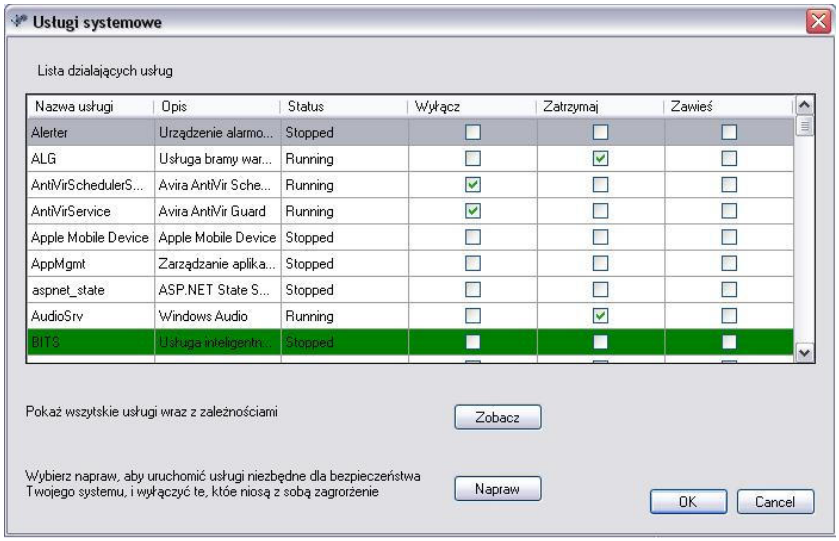
Spoofing (*masquerading*) – przyjmowanie fałszywej tożsamości w celu uzyskania nieuprawnionego dostępu do systemu i usług. Ponieważ „podrabianie" dotyczy wszystkich warstw systemu OSI, bardzo trudno się przed nim bronić. Wyróżnia się dwie główne odmiany *spoofingu*:

Blind spoofing – polega na przesyłaniu danych uwierzytelniających, przy braku dostępu do informacji, aby określić parametry sieci, śledzić zmiany ustawień lub, w szczególnym przypadku, uzyskać połączenie. Do „badania na ślepo" (*blind scanning*) wykorzystywane są komputery *zombie*, wcześniej przejęte przez hackera.

Active spoofing – monitorowanie, uszkadzanie, generowanie bądź usuwanie pakietów wysyłanych w trakcie komunikacji, między uwierzytelnionym użytkownikiem a serwerem, w celu uzyskania uprawnień.

Jedyną obroną przed *spoofingiem* są ścisłe reguły filtrowania pakietów, stosowanie zabezpieczeń kryptograficznych np. SSH, czy PGP i walka z lekkomyślnością użytkownika, który przez nieuwagę zezwala na połączenie. Często użytkownicy sami pobierają programy umożliwiające wykradzenie klucza, zwłaszcza gdy łatwowiernie korzysta się z automatycznych aktualizacji programów (nawet zaufane witryny często padają ofiarami podrabiania), lub pobiera pliki i programy z nieznanych źródeł.

Programy działają często jako ukryty aplet, podobnie jak usługi – zarządzane systemowo procesy wykonujące określone zadania systemowe. Programy te dzielą się na dwie podstawowe grupy: systemową – zawiadującą pracą systemu, i programową – związaną z działaniem sterowników i zainstalowanych programów, np. ochroną antywirusową. Każda z usług działa w powiązaniu z innymi i może mieć trzy stany: działania, zawieszenia, zatrzymania oraz dwa sposoby uruchamiania: automatyczny (wraz z każdym startem systemu), bądź ręczny. Ponieważ te podprogramy działają niezależnie od profilu użytkownika, ich zachowanie można zdefiniować osobno na każdym z kont. Z uwagi na sposób sterowania, większość z nich jest skonfigurowana automatycznie, choć sam Microsoft zaleca wyłączenie niektórych usług, jeśli nie są one używane. Aby poznać usługi działające na komputerze w programie dodano funkcję Usługi, pokazującą w tabeli, ich nazwę, opis i istniejący stan (Rysunek 3).



Rys. 3. Aplet Usługi systemowe

Aplet *Usługi* przedstawia w postaci tabeli zdefiniowane ustawienia wraz z ich stanem. Dzięki programowi można automatycznie zmienić stan około 70 usług, zgodnie z zaleceniami Microsoft`u. Pierwszym krokiem do implementacji było utworzenie funkcji ServicesDS, wymagającej zaimportowania komponentu Service Process:

```
'Importowanie komponentów systemowych:
Imports System.ComponentModel
Imports System.ServiceProcess
Public Class ServicesDS
'Implementacja kolekcji:
    Implements IListSource
    Public ReadOnly Property ContainsListCollection()
_
    As Boolean Implements
System.ComponentModel.IListSource.ContainsListCollect
ion
        Get
            Return False
        End Get
    End Property
'Lista usług:
    Public Function GetList() As
System.Collections.IList Implements
System.ComponentModel.IListSource.GetList
```

```
        Dim uslu As New BindingList(Of
ServiceController)
        Return uslu
    End Function
End Class
```

Teraz wystarczy zadeklarować procedurę zawierającą listę niebezpiecznych procesów za pomocą kodu:

```
Private Sub uslugi_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        DataGridView1.DataSource =
ServiceController.GetServices()
        listaNiebezpiecznycProcesow = New ArrayList()
        listaNiebezpiecznycProcesow.Add("ClipSrv")
```

Rys. 4. Aplet Sterowniki i programy

Dla wielu specjalistów zajmujących się zabezpieczeniem systemów operacyjnych, rozpoznanie zagrożeń rozpoczyna się od monitorowania systemu i analizowania dzienników pod kątem powtarzających się zdarzeń, liczby powtórzeń przekraczającej konkretną wartość, oraz okresu w jakim się pojawiają. Aplet *Sterowniki* urządzeń i programy, powstał w celu śledzenia działających na komputerze aplikacji. Uruchamiające się bez kontroli, wraz ze startem systemu, programy obniżają wydajność, oraz niosą z sobą zagrożenie w postaci otwartych

portów nasłuchiwania. Większość programów szkodliwych (z grupy malware), działa w tle na poziomie ukrytym, wysyłając informacje przez Internet, lub nasłuchując na określonych portach poleceń od hackera. Sterowniki urządzeń komputera, podają wersję tylko w momencie instalacji, a ponieważ większość z nich działa podobnie do usług systemowych (uzależnionych od innych procesów), w przypadku częściowej aktualizacji mogą być zagrożeniem dla stabilności systemu. Instalowanie poprawek, nie tylko systemowych, sprzyja bezpieczeństwu i wydajności systemu.

Programy takie jak konie trojańskie rezydują w systemie, uruchamiając się dzięki wpisom do kluczy rejestru, oraz pików rozruchowych, często też tworzą zapisy w plikach *boot`owalnych*, czyli zarządzających każdym startem systemu. Aby kontrolować te wpisy należy regularnie przeglądać listy działających programów, pliki rozruchowe, ustawienia kluczy autostartu oraz dzienniki zdarzeń. Badanie wydajności systemu, pozwala na wykrycie nieprawidłowości w wykorzystaniu zasobów komputera i odnalezienie programów, użytkowników i usług mogących negatywnie wpływać na system. Uwagę powinny zwracać zwłaszcza nietypowe parametry ruchu sieciowego, takie jak błędy w transmisji, czy liczba odrzuconych pakietów. Dzięki analizie wykorzystania połączenia sieciowego komputera, można odnaleźć szkodliwe programy łączące się z siecią, wiodące do niespodziewanych lokacji, lub powtarzające się w określonym czasie, próby nawiązania połączenia z usługami w sieci. Zarówno badanie wydajności pamięci i całego systemu, oraz połączenia internetowego jest dostępne przy pomocy programu OBRONA i formantowi *Wydajność.*

4 Metody socjotechniczne

Metody socjotechniczne (inżynieria społeczna, *social ingineuring*), to zestaw technik do manipulowania społeczeństwem; w informatyce rozumiany jako metody zmierzające do uzyskania informacji poufnych, albo umożliwiających zdobycie takowych, poprzez wykorzystanie zaufania lub naiwności użytkowników. Najbardziej znaną osobą wykorzystującą socjotechnikę jest Kevin Mitnick – jeden z najbardziej znanych włamywaczy komputerowych, należących do grupy *Black Hat*. W swoich dwóch książkach: *Sztuce podstępu* i *Sztuce infiltracji*, opisał stosowane przez siebie metody oraz sposoby obrony przed nimi. Do pięciu najważniejszych cech dobrego manipulanta należą: cierpliwość, uprzejmość, zrozumienie, koleżeństwo, pewność siebie i dodatkowo - pochlebstwo.

Rys. 5. Okno Certyfikaty i szyfrowanie

Narzędziem podstawowym socjotechnika jest telefon, umożliwiający zebranie informacji bez konieczności kontaktu „*twarzą – w - twarz*". Techniki inżynierii społecznej, stosowane w informatyce, można podzielić na trzy główne grupy:

wyłudzanie informacji – wykorzystujące sposoby manipulacji takie jak: perswazja, błędne skojarzenia (nazywana także metodą podtekstów - słów kluczy, słów branżowych - stosowanych by uwiarygodnić atakującego), zaangażowania - konsekwencji, wzajemności, czy prawa limitu (presja czasu), najczęściej stosowanego przez Mitnicka;

Phishing – rozsyłanie e-mail`i z treścią zgłaszającą konieczność reaktywowania konta np. w banku, połączone z podrabianiem stron internetowych przechwytujących informacje;

E-mail spoofing – modyfikowanie nagłówka listu tak by wyglądał na pochodzący z innego źródła, wykorzystywane do rozsyłania spamu.

Już system Windows NT, pozwał na dodawanie wyjątków, tworzenie własnych zabezpieczeń i przegląd zainstalowanych certyfikatów ograniczających internetowy *phishing;* każdorazowe połączenie się ze stroną o odmiennym certyfikacie bezpieczeństwa niż ten pobrany i zainstalowany w systemie, będzie powiadamiane komunikatem systemowym. Do sprawdzania poprawności certyfikatów *X.509* służy moduł kryptograficzny Microsoft *rsaenh.dll* dodawany do *Microsoft Kernel Mode Cryphtographic Module*. Przeprowadzanie uwierzytelnienia w systemie Windows, jest automatyczne i wymaga włączenia NAP

(*Network Access Protection*), dostępnej dla zainstalowanych systemów z poprawką SP3. NAP to platforma wymuszania zasad zgodności kondycji systemu, pozwalającej na sprawdzenie jego stanu przed nawiązaniem połączenia. W dodatku SP3, pojawiła się także nowa forma zabezpieczenia WPA2/WPS IE standardu WiFi zaczerpniętego z IEEE 802.11i, umożliwiającego wykorzystanie AES do szyfrowania i uwierzytelniania połączeń. Program OBRONA pozwala na dostęp do ustawień certyfikatów lokalnych, przegląd zabezpieczeń kont użytkowników oraz konfiguracji zabezpieczeń sieciowych, szyfrowania i ustawień podpisu elektronicznego. Przyciski znajdujące się w aplecie *Certyfikaty i szyfrowanie*, pozwalają przetestować ustawienia zabezpieczeń lokalnych pod kątem bezpieczeństwa grup użytkowników.

5 Ataki kryptologiczne

Tego typu atak polega na przechwytywaniu i deszyfrowaniu wiadomości w celu nabycia fałszywej tożsamości; należą do nich:

- Atak FMS - (Fluhrera, Mantina i Shamira), który deszyfruje pakiety zabezpieczonych protokołem WEP, za pomocą luk w generowaniu kluczy prywatnych;

- Brute force – atak polega na testowaniu wszystkich możliwych kombinacji kluczy w celu uzyskania dostępu do zasobów;

- Atak słownikowy – wykorzystywanie ograniczonej listy haseł w celu uzyskania dostępu;

- Atak ze znanym szyfrogramem – uzyskanie szyfrogramu polega na przechwyceniu pakietów, a następnie deszyfrowaniu wiadomości w oparciu o znajomość szyfru;

- Atak ze znanym tekstem jawnym – stosowany, gdy znany jest jedno lub więcej słów w tekście jawnym i ich szyfrogram;

- Meet in the middle – atak kryptograficzny z jawnym tekstem, działający głównie na szyfrach wykorzystujących co najmniej dwa klucze;

- Caffe Latte – poprzez zalewanie AP, dużą ilość zaszyfrowanych pakietów, atakujący uzyskuje dostęp do sieci. Metoda wymaga wygenerowania odpowiedniego ruchu sieciowego, przechwycenia kluczy WEP, wpisania się do tablicy ARP, oraz poprawnego odszyfrowania najwyżej sześciu pakietów. Ten typ ataku po raz pierwszy przedstawił na konferencji Toorcon w 2007 roku, Vivek Ramachandran. Nazwa pochodzi od konieczności szybkiego deszyfrowania pakietów WEP, wymaganego przy atakach na *hot spot`y*, który musi trwać tyle, ile przygotowanie *cafe latte*. We włamaniu decydującą rolę odgrywa podrabianie pakietów ARP, polegające na zmianie kilku bitów w przechwyconych danych, a następnie wysłaniu ich ponownie do klienta.

Przykładami narzędzi ułatwiających spoofing są: *SMAC, IP Sp00fer 5.1, AirSnort, Cain&Abel.*

W przypadku ataków kryptograficznych, odpowiedź programu działającego jedynie po stronie klienta, może dotyczyć zablokowania możliwości odczytu lub zmiany hasła użytkowników, wykradzenia loginów i poufnych haseł do kont internetowych. Zablokowanie udziałów i uważna obserwacja nawiązywanych połączeń nie gwarantuje bezpieczeństwa danych, zwłaszcza gdy przechowywane są w domyślnych lokacjach. Najbardziej narażone na przeszukiwanie są miejsca, w których przechowywane są tymczasowe pliki internetowe, pliki gromadzone w profilach użytkownika w katalogu *Documents and Settings*, a także zrzuty pamięci w postaci plików stronicowania. W systemie Windows istnieje mechanizm do zarządzania pamięcią wirtualną (VMM), umożliwiający ochronę zasobów pamięci, oraz dodanie pamięci wirtualnej wykorzystywanej przez programy działające na komputerze. VMM tworzy plik stronicowania, w którym zawarte są informacje dotyczące pamięci oraz pliku wymiany. Plik wymiany w momencie gdy zabraknie miejsca w pamięci, jest zapisywany na dysku twardym, razem z informacjami w nim przechowywanymi. Zaleca się przed wyłączeniem komputera, oczyścić plik stronicowania poprzez zapełnienie go zerami. Jedyna wadą tego procesu, jest dłuższy czas zamykania systemu. Podobne znaczenie ma czyszczenie pamięci podręcznej tablicy ARP i ustawień NetBios`u, o ile ten ostatni jest włączony. Składowane na dysku systemowym pliki tymczasowe, mogą być zagrożeniem dla integralności systemu. Implementacja zapobiegania tego typu procesom, jest następująca:

```
'czyszczenie pamięci z plików bibliotek
        Dim pierwszy As Integer
        If pierwszy = 1 Then
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SOF
TWARE\Microsoft\Windows\CurrentVersion\Explorer\Alway
sUnloadDLL", True).SetValue("@", 1)
'zerowanie pliku stronicowania
        Dim drugikl As Integer
        If drugikl = 1 Then
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SYS
TEM\CurrentControlSet\Control\Session Manager\Memory
Management",
True).SetValue("ClearPageFileAtShutdown", 1)
'usuwanie listy ostatnio używanych plików
         Dim trzeci As Integer
```

```
        trzeci = MsgBox("Ten przycisk wprowadzi
zmiany do Twojego rejestru.Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If trzeci = 1 Then
Microsoft.Win32.Registry.LocalMachine.CreateSubKey("S
oftware\Microsoft\Windows\CurrentVersion\Policies\Exp
lorer", True).SetValue("ClearRecentDocsOnExit", 1)
Microsoft.Win32.Registry.CurrentUser.CreateSubKey("So
ftware\Microsoft\Windows\CurrentVersion\Policies\Expl
orer", True).SetValue("ClearRecentDocsOnExit", 1)
        End If
'Usuwanie historii w przeglądarce Internet Explorer
        Dim klucz As Microsoft.Win32.RegistryKey =
Microsoft.Win32.Registry.CurrentUser.OpenSubKey("Soft
ware\Microsoft\Internet Explorer\TypedURLs", True)
        For Each wartoscKlucza As String In
klucz.GetValueNames
            klucz.DeleteValue(wartoscKlucza)
        Next
'Czyszczenie pamięci NetBIOS
        Dim okno1 As New cmdDial()
        okno1.SetCommand("nbtstat -RR")
        okno1.Text = "Czyszczenie NetBIOS"
        okno1.Show()
'Czyszczenie tablic ARP
        Dim okno1 As New cmdDial()
        okno1.SetCommand("arp -d")
        okno1.Text = "Czyszczenie tablic ARP"
        okno1.Show()
'usuwanie plików z katalogu TEMP
        Dim czwarty As Integer
        czwarty = MsgBox("Ten przycisk usunie
zawartość katalogu TEMP.Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If czwarty = 1 Then
            Dim okno1 As New cmdDial()
            okno1.SetCommand("del c:Windows\TEMP\
/Q")
            okno1.Text = "Usuwanie plików z katalogu
TEMP"
            okno1.Show()
        End If
'Wymuszenie usuwania tymczasowych plików
internetowych po każdym uruchomieniu przeglądarki
        Dim piaty As Integer
```

```
        piaty = MsgBox("Ten przycisk wprowadzi zmiany
do Twojego rejestru.Czy chcesz kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If piaty = 1 Then
            Dim wymusz As Integer
            wymusz =
Microsoft.Win32.Registry.CurrentUser.OpenSubKey("Soft
ware\Microsoft\Windows\CurrentVersion\Internet
Settings\Cache").GetValue("Persistent")
            If wymusz = Nothing Then
Microsoft.Win32.Registry.CurrentUser.OpenSubKey("Soft
ware\Microsoft\Windows\CurrentVersion\Internet
Settings\Cache", True).SetValue("Persistent", 0)
            End If
        End If
```

Dzięki zastosowaniu programu, niemożliwym staje się odczytanie zapisanych w zrzutach pamięci hash`y rainbow tables, i odczyt hasła użytkownika za pomocą programów takich jak Ophcrack czy pwdump.

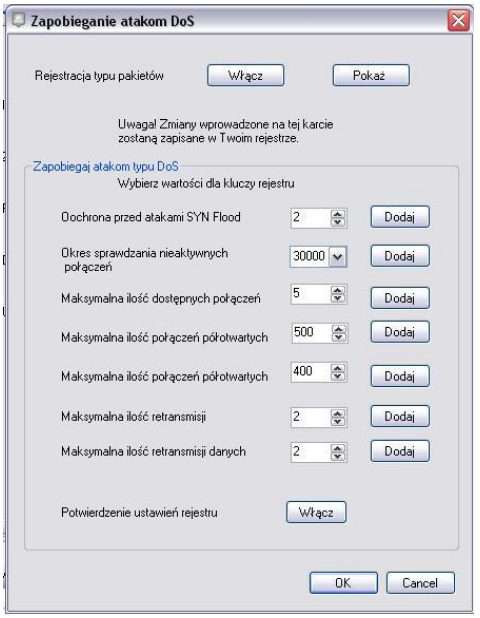
6 DoS – Denial – of – Service

DoS polega na zalaniu sieci niepożądanym ruchem, czego skutkiem jest „zapchanie" łącza i odcięcie pozostałych użytkowników od korzystania z usług sieciowych. Może być skierowany na warstwę aplikacji poprzez wysyłanie do niej dużej ilości żądań np. otwarcia strony WWW serwera, co doprowadza do jego zablokowania. Podobnie przebiega atak *SYN flood*, polegający na zasypaniu hosta pakietami typu SYN, wykorzystywanymi do ustanawiania połączeń, co powoduje przepełnienie bufora śledzenia aktywnych połączeń TCP (systemy operacyjne mają limit dopuszczalnych połączeń) i choć sieć działa poprawnie to stacja ulegnie zablokowaniu. Niestety wszystkie ataki typu DoS w sieciach bezprzewodowych są łatwe do przeprowadzenia i bardzo trudne do zapobiegania.

W sieciach o małej przepustowości wykorzystywane jest także zapełnienie sieci dużą ilością danych (np. pakietami ICMP - *ICMP flood`ing*, służących do badania właściwości hosta i świadczonych przez niego usług), co powoduje przepełnienie serwera, porzucanie pakietów, odcięcie lokalnej sieci od Internetu. W przypadku sieci bezprzewodowej, taki atak skutkuje odcięciem od sieci stacje połączonych z konkretnym punktem dostępowym, nawet jeśli brama obsłuży część pakietów. Możliwe jest przypadkowe obciążenie sieci, za pośrednictwem programów intensywnie korzystających z zasobów Internetu, np. typu P2P. Inną metodą jest także wysyłanie nieprawidłowych ramek

rozgłoszeniowych (bez zawartości) do wszystkich hostów w sieci. Dla sieci komunikującej się za pomocą dostępnego powszechnie sygnału, nawet zaszyfrowanego, możliwe jest przepełnienie łącza i odcięcie hostów od sieci (np. *land attack* wykorzystuje *loopback SYN*). W sytuacji gdy jeden z host`ów a korzysta z anteny zbiorczej A, a inny z anteny zbiorczej B, gdzie obie anteny mają wspólny punkt dostępowy, klient B może odłączyć od sieci A, jeśli zmieni swój adres MAC na adres klienta A i wyśle go do anteny B ale sygnałem o wyższym natężeniu; wówczas antena zacznie wysyłać i odbierać ramki od adresu B dla adresu A, a wszystkie pakiety z adresu A zostaną zignorowane. Nawet przy stosowaniu uwierzytelnienia WEP można użyć ataku typu DoS, poprzez podstawiony punkt dostępowy, wystarczy że klient korzystający z punktu dostępowego będzie charakteryzował się silniejszym sygnałem. Punkt dostępowy hakera używającego wzmacniacza zawsze będzie wybierany, o ile poda poprawny identyfikator ESSID. Obecnie istnieją różne narzędzia wykonujące atak typu DoS, np.: *Blast v2.0, UDPFlood v2.0.*

Ataki typu Denial of Service, są najtrudniejsze do wykrycia; aby się przed nimi obronić należało by zupełnie zrezygnować z przyjmowania pakietów, co równa się odcięciem od sieci. Istnieje wiele typów ataków DoS, i nie sposób uodpornić komputer na nie wszystkie; rozsądnym jednak wydaje się ograniczenie liczby nawiązywanych i podtrzymywanych połączeń oraz otwartych portów, czy retransmisji pakietów w razie otrzymania komunikatu o błędzie.



Rys. 6. Okno zapobiegania atakom DoS

Zapełnienie bufora urządzenia docelowego żądaniami połączenia powoduje zmniejszenie wydajności komputera a często całkowite jego odłączenie od usług sieciowych, może także wiązać się z problemami przydziału zasobów i skutkować zniszczeniem istotnych danych. Atak DoS typu *SYN flooding,* polega na nadawaniu żądań nawiązania połączenia TCP szybciej niż może na nie odpowiedzieć adres docelowy. *SYN flooding* zmierza do zmniejszenia wydajności adresata, a nawet jego całkowitego unieruchomienia, gdyż atakowany system przydziela coraz to nowe zasoby by utworzyć połączenia. Adres komputera nadającego sygnał jest zwykle zafałszowany, więc retransmitowany pakiet SYN-ACK, jest wysyłany wielokrotnie do wyczerpania określonego czasu, po którym zasoby zostają zwolnione. Często stosowany jest tzw. *atak choinki*, polegający na łączeniu specyficznych znaczników pakietów np. SIN+FIN jednocześnie się wkluczających. Do nieprawidłowych kombinacji należą: SYN+ACK, FIN, FIN+ACK, SYN+FIN. Aby sprawdzić przychodzące pakiety należy wybrać przycisk rejestracji pakietów, który poprzez systemowy firewall zapisze informacje w postaci pliku, dostępnego do oglądu po wybraniu przycisku *Pokaż*:

```
Dim okno1 As New cmdDial()
        okno1.SetCommand("netsh firewall set logging
droppedpackets = enable filelocation = c:\firewal.txt
maxfilesize = 4096")
        okno1.Text = "Włączanie rejesrtowania
pakietów"
        okno1.Show()
        Dim u As Uruchamiacz = New
Uruchamiacz("notepad", "c:\firewal.txt")
        u.Uruchom()
```

Ograniczenie połączeń w zakresie ich ilości, jest kluczem do wydajnego połączenia z siecią. Kontrola połączeń przychodzących, nasłuchujących i działających jest możliwa dzięki programowi w zakładce *Istniejące połączenia,* w tym wypadku możliwe jest ich ograniczenia za pomocą ustawień w rejestrze i zmuszenia systemu na zamykanie połączeń przekraczających określoną liczbę lub brak odpowiedzi ze strony adresu docelowego. Program OBRONA umożliwia automatyczne modyfikowanie rejestru za pomocą pól numerycznych, które użytkownik ustawia na dowolną wartość (domyślnie ustawiona jest wartość zalecana przez Microsoft) i potwierdzenia ustawień w postaci zatwierdzenia zamian systemowych np. dla następujących kluczy:

- *HKLM\System\CurrentControlSet\Services\TCPIP\Parameters\SynAttackProtect* – pozwala na ustawienie limitu retransmisji i opóźnienie tworzenia wpisu RCE (*Route Cache Entry*)
- *HKLM\System\CurrentControlSet\Services\TCPIP\Parameters\TcpMaxHalfOpen* – limit połączeń półotwartych
- *HKLM\System\CurrentControlSet\Services\TCPIP\Parameters\EnablePMTUDiscovery* – limit dla połączeń poza siecią lokalną MTU (*Maximum Transmission Unit*)
- *HKLM\System\CurrentControlSet\Services\TCPIP\Parameters\KeepAliveTime* – okres powtarzania sprawdzenia nieaktywnych połączeń;
- *HKLM\System\CurrentControlSet\Services\TCPIP\Parameters\TCPMaxConnectResponseRetransmissions* – limit prób połączenia pozostawionych bez odpowiedzi.

Przeglądanie istniejących połączeń pozwala odkryć próby typu *atack land* – łatwe do rozpoznania, ponieważ adres oraz port źródłowy są identyczne jak adres docelowy (była to technika stosowana najczęściej przez Kevina Mitnicka). Jedną z technik przysyłania ruchu sieciowego za pomocą router`a jest NAT (*Network Address Translation*), zmieniający adresy źródłowe, czasem również docelowe, oraz nazwy portów na postać zamaskowaną. Zaletą korzystania z mechanizmu, oprócz oszczędności adresów IP, jest anonimowość. Często odpowiedzi NAT są przeszukiwane przez agresorów, przy wykorzystaniu źródłowego IP routera, w poszukiwaniu informacji dotyczących typologii sieci; aby uodpornić protokół IP przed próbami automatycznych odpowiedzi zaleca się wprowadzenie do rejestru następujących zmian:

```
'Ustawienia zabepzieczenia NAT
        Dim k As Integer
        k = MsgBox("Wciśnięcie klawisza spowoduje
zmiany w rejestrze, czy chcesz kontunuowac?",
MsgBoxStyle.OkCancel, "Uwaga")
        If k = 1 Then
'wyłączenie routingu IP dla wysyłania źródłówego
adresu IP:
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("DisableIPSourceRouting", 1)
'ochrona przez atakaiem typu multicast floodiing, I
automatycznymi odpowiedziami na multicasty
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("EnableMulticastForwarding", 0)
'zablokowanie wysyłania pakietów pomiędzy sieciami,
za wyjątkiem firewalla
```

```
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("IPEnableRouter", 0)
'kontrolowanie odpowiedzi na zapytania ICMP
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("Sys
tem\CurrentControlSet\Services\Tcpip\Parameters",
True).SetValue("EnableAddrMaskReply", 0)
        End If
'Ustawienia usług kryptograficznych
        Dim okno1 As New cmdDial()
        okno1.SetCommand("netsh nap show
configuration")
        okno1.Text = "Ustawienia usług
kryptograficznych"
        okno1.Show()
'Sprawdzanie trybu uwierzytelnienia
        Dim okno1 As New cmdDial()
        okno1.SetCommand("netsh ras show authmode")
        okno1.Text = "Tryb uwierzytelnienia"
        okno1.Show()
'zablokowanie wprowadzania innych stron startowych do
przeglądarki Internet Explorer:s
        Dim a As Integer
        a = MsgBox("Naciśnięcie przycisku wprowadzi
zmiany w Twoim rejestrze.Kontynuować?",
MsgBoxStyle.OkCancel, "Uwaga")
        If a = 1 Then
Microsoft.Win32.Registry.CurrentUser.OpenSubKey("Soft
ware\Policies\Microsoft\Internet Explorer\Control
Panel", True).SetValue("HomePage", 1)
        End If
```

7 Podsumowanie

„Ponieważ więc książę obowiązany jest umieć używać bestii, powinien sobie wybrać lisa i lwa, lew bowiem nie poradzi przeciw sieciom, lis nie poradzi przeciw wilkom. Należy więc być lisem, aby się poznać na sieciach, i lwem, aby odstraszać wilków”.

Niccolo Machiavelli: *Książe*.

Osiągnięcie bezpieczeństwa komputera w sieci bezprzewodowej jest niezwykle trudne i wymaga starannej obserwacji systemu oraz działań bezpośrednich, np. planowania ustawień systemowych. Wyprzedzanie ataku, możliwe jest dzięki zbieraniu informacji dotyczących komputera,

słabości protokołów z jakich korzystamy, planowania uaktualnień i poprawek oraz gromadzenia danych na temat funkcjonowania połączeń. Naprawa istniejących problemów jest jedynie uwieńczeniem pracy, zapewniającą spokój i odporność naszego systemu.

Niniejsza praca miała za zadanie wykazać, że idee wolności i swobody sieci bezprzewodowej można osiągnąć za pomocą kompromisów, otrzymując stabilny i odporny na ataki system. Zarówno przygotowany program jak i zgromadzone informacje nie są wyczerpujące. Samo przygotowanie komputera do działania w sieci, za pomocą programu OBRONA, jest zaledwie połową sukcesu, drugą jest świadomość użytkownika. Rozumiejąc motywy włamywaczy komputerowych, znając etapy ataków, ich sposobu myślenia, uzyskujemy przewagę.

„Pseudowolność" internetowa, określona stwierdzeniem „*bo mogę*", nie bierze pod uwagę możliwych zniszczeń, konsekwencji czy kosztów. Jest to motto nowego świata, ugruntowane sukcesami: nie można jej powstrzymać, tak jak nie da się ocenzurować Internetu. Powtarzające się włamania do sieci bezprzewodowych uczą, że ignorowanie problemu go nie rozwiązuje, a jedynie naraża nas na straty. Z pewnością doświadczenie to, nie wystarczy by popaść w przesadę i przysłowiowo „wyciągnąć" wtyczkę z sieciowego gniazdka, ale uczy pokory dla nowych form komunikacji i praw globalnego dostępu. Powszechna wolność często prowadzi do nadużyć praw autorskich, poufności czy nawet inwigilacji, lecz jest to cena jaką płacimy za olbrzymią wiedzę i wygodę związaną z nowoczesnymi technologiami. Tylko od nas zależy w jaki sposób się do niej przygotujemy, czy staniemy bezbronni wobec faktów, czy też będziemy „*lisem i lwem*" nowej przyszłości.

Literatura

[1] Lockhart A., 100 sposobów na bezpieczeństwo sieci, O`Riley, Tokyo 2004

[2] Lukatsky A., Wykrywanie włamań i aktywna ochrona danych , Gliwice 2005, s. 113-115.

[3] Piasek A., Dziura w SNMP zagraża Sieci [w:] www.notebooki.idg.pl/news/33236/Dziura.w.SNMP.zagraza.Sieci.html

[4] Potter B., Fleck B, *802.11.Bezpieczeństwo*. Wydawnictwo Helion, Gliwice, 2004.

[5] Hack Proofing Your Network, Wydawnictwo Heliion, Gliwice 2002.

[6] Chusteczki J., Cisco ostrzega przed atakami opartymi na ICMP, [w:]www.idg.pl/news/77578/Cisco.ostrzega.przed.atakami.opartymi.na.ICMP.html

[7] Scambray J., McClure S., Kurtz G. Hakerzy, cała prawda. Sekrety zabezpieczeń sieci komputerowych, Wydawnictwo Translator, Warszawa 2001

[8] Edney J., Arbaugh W., Real 802.11 Security WiFi protected Access and 802.11i , Addison Wesley 2004, Boston.

[9] Beaver K., Davis P., Hacking Wireless Networks, Addison Wesley Indiana, 2005 .

[10] Wójczuk P., Usługi w Windows XP, [w:] www.centrumxp.pl/WindowsXP/643,1,Us%C5%82ugi_w_Windows_XP.aspx

Strony internetowe:

[11] www.iname.pl/2008/07/dns-pl-analiza-problemu-dns-cache-poisoning/

[12] www.msdn.microsoft.com/en-us/library/aa302363.aspx

[13] www.technet.microsoft.com/pl-pl/library/cc750356%28en-us%29.aspx

[14] www.microsoft.com/poland/protect/yourself/phishing/engineering.mspx

[15] www.microsoft.com/poland/technet/bazawiedzy/centrumrozwiazan/cr327_01.mspx

[16] www.wi-fiplanet.com/tutorials/article.php/10724_3716241_1

ATTACKS AND BREAK FOR WIRELESS NETWORKS

Summary – Article introduces the issue of attacks and intrusions into wireless networks. Today, when the benefits of wireless networks can be used by anyone, you might want to look at the problem of their security from the perspective of a user. This is not only a matter of choice, but the skills, basic knowledge and tools for efficient use, even for daily "bread eater."