

---

## PROJEKT – DOKUMENTACJA KOŃCOWA

---

Artur M. Brodzki  
Damian Chiliński  
Krzysztof Sznejder

6 maja 2019

## 1 WSTĘP

Zgodnie z przyjętym przez nas tematem, w ramach projektu napisaliśmy moduł systemu IDS/IPS Snort, wykrywający steganograficzną transmisję danych metodą tunelowania DNS. Przetestowaliśmy również jego działanie w wirtualnym środowisku, symulującym sieć korporacyjną.

## 2 TUNELOWANIE DNS

### 2.1 Zasada działania

Tunelowanie DNS należy do metod steganograficznych i pozwala na prowadzenie niewidocznej dla administratorów transmisji danych, np. przesłanie na maszynę atakującego wykradzionych plików. Zasada działania opiera się na redundancji protokołu DNS. Typowy przebieg ataku z wykorzystaniem tunelowania DNS jest następujący:

1. Atakujący rejestruje domenę, np. `best-malware.com` i tworzy dla niej własny serwer autorytatywny.
2. Atakujący instaluje złośliwe oprogramowanie na systemach ofiary, które wykrada z nich wrażliwe dane, np. hasło bankowe.

3. Oprogramowanie atakującego koduje uzyskane informacje jako subdomenę, np: `76756C6E657261626C65.bestmalware.win` i wysyła odpowiadające jej zapytanie DNS. Trafia ono na serwer atakującego, który może odebrać i zdekodować ukryte informacje. Dane można kodować zasadniczo za pomocą kodowania szesnastkowego lub base32. Wynika to ze specyfikacji protokołu DNS, który dopuszcza w zapytaniach ograniczony zestaw znaków.
4. Aby przeprowadzić transmisję w drugą stronę, wystarczy zakodować informację w pakiecie będącym odpowiedzią na zapytanie. Najczęściej będzie to więc pakiet typu A, choć możliwe jest kodowanie danych z użyciem każdego typu rekodu DNS.

Skuteczność tego ataku wynika z kluczowej roli, jaką odgrywa DNS w komunikacji internetowej. Jego zablokowanie może uniemożliwić działanie sieci, dlatego zapytania DNS bardzo rzadko są ograniczane przez administratorów, i to nawet w sieciach o restrykcyjnych wymogach bezpieczeństwa.

## 2.2 Metody wykrywania

Aby skutecznie zablokować transmisję wykorzystującą tunelowanie DNS, konieczne jest wiarygodne odróżnienie złośliwych zapytań DNS od tych prawidłowych. Stosuje się w tym celu zbiór heurystyk, opartych m.in. o:

- analizę długości i entropii zapytania – większość „normalnych” domen nie przekracza kilkunastu znaków i charakteryzuje się niewielką entropią, typową dla słów pochodzących z języka naturalnego;
- analizę częstości – znaczący wzrost liczby wykonywanych zapytań DNS może sugerować, że część z nich jest niepożądana;
- metody słownikowe – „prawdziwe” domeny zazwyczaj składają się ze słów pochodzących z języka naturalnego lub zbudowanych na ich podstawie neologizmów;
- uczenie maszynowe.

W naszym projekcie zdecydowaliśmy się wykorzystać autorski algorytm uczenia maszynowego oparty o naiwny klasyfikator Bayesa. Klasyfikator został nauczony na zbiorze zapytań DNS wygenerowanych i przechwyconych przez nas podczas kilku tygodni codziennego użytkowania naszych domowych komputerów. Szczegółowy opis algorytmu znajduje się w sekcji

### 3 SCENARIUSZ PROJEKTOWY

Zaplanowaliśmy następujący scenariusz projektowy: firma B-Cyb S.A. dysponuje siecią korporacyjną, działającą w domenie `bcyb.com`. W ramach świadczonych usług, posiada dwa publicznie dostępne serwery HTTP, kolejno: `s1.bcyb.com` oraz `s1.bcyb.com`. Dostęp do firmowej sieci chroniony jest za pomocą firewalla działającego na firmowym routerze, jak również z użyciem systemu IDS/IPS Snort, filtrującego ruch przepływający przez sieć. Ponieważ firma B-Cyb obawia się ataku metodą tunelowania DNS, zakupiła od młodego zespołu specjalistów (Kant Security sp. z o.o.) dodatkowy moduł systemu Snort, mający wykrywać i blokować tego rodzaju ataki. W ramach sieci firmowej została wydzielona podsieć, chroniona – w celach testowych – przy użyciu nowego modułu.

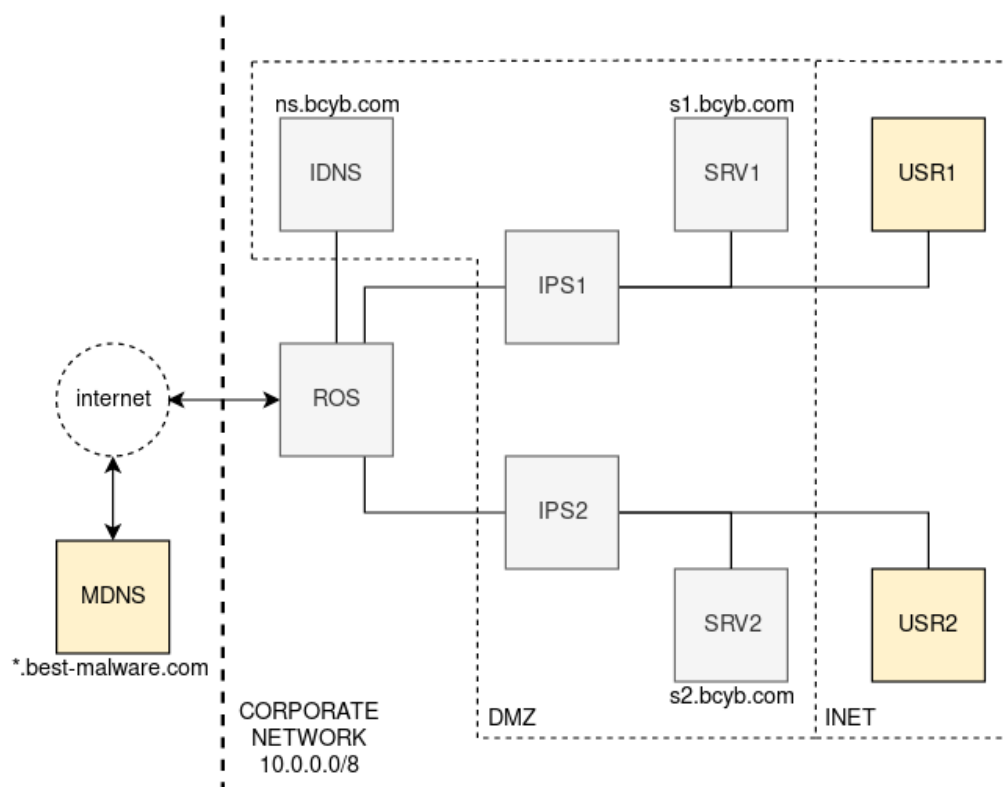
Na firmowych serwerach znajduje się jednak złośliwe oprogramowanie, którego zadaniem jest przetransmitować na serwery atakującego wykradzione pliki `/etc/shadow`. Ponieważ wyprowadzenie danych poza sieć za pomocą protokołu TCP jest niemożliwe, atakujący zdecydowali się zastosować właśnie tunelowanie DNS. Podczas próby przesłania wykradzonych haseł, nowy moduł Snorta od zespołu Kant Security wykryje i zablokuje złośliwą transmisję. Natomiast podsieć chroniona za pomocą dotychczasowej wersji systemu Snort nie wykryje ataku i pozwoli na wyprowadzenie haseł na zewnątrz.

Całość infrastruktury projektowej została zainstalowana przez nas na maszynach wirtualnych VMware i obejmuje łącznie 9 hostów, znajdujących się pod kontrolą systemu Fedora 30. Szczegółowa topologia sieci znajduje się na rys. 1. Przygotowana infrastruktura składa się z następujących urządzeń:

- router ROS: domyślna brama sieciowa dla sieci firmowej, działa pod kontrolą systemu MikroTik RouterOS i obsługuje zaporę sieciową;
- IDNS: firmowy serwer DNS;
- SRV1 i SRV2: publicznie dostępne serwery HTTP;
- IPS1 i IPS2: hosty systemu Snort, działające w trybie inline i filtrujące ruch przechodzący przez sieć;
- USR1 i USR2: komputery użytkowe, służące pracownikom firmy do wykonywania obowiązków służbowych, w tym zarządzania pozostałymi hostami;
- MDNS: komputer atakującego, stanowiący punkt odbiorczy ukrytej komunikacji.

Zapora sieciowa posiada następującą konfigurację:

1. sieć firmowa (corporate network, 10.0.0.0/8) podzielona jest na strefę zdemilitaryzowaną (DMZ) oraz sieć wewnętrzną (INET);



**Rysunek 1:** Schemat ideowy sieci, realizującej przyjęty scenariusz projektowy.

2. strefa zdemilitaryzowana: obejmuje publicznie dostępne serwery DNS i HTTP (IDNS, SRV1, SRV2), jak również chroniące je hosty systemu Snort.

- ruch wchodzący: dopuszczony jest jedynie ruch realizujący świadczone usługi, oraz ruch SSH pochodzący z sieci INET;
- ruch wychodzący: zabronione jest inicjowanie ruchu w obrębie DMZ, z wyjątkiem zapytań DNS oraz ruchu wychodzącego do serwerów zawierających repozytoria Fedory. Dopuszczony jest zatem jedynie ruch wymagany do pobierania aktualizacji oprogramowania systemowego;

3. sieć wewnętrzna: obejmuje komputery służbowe USR1 i USR2:

- ruch wchodzący: jest zasadniczo zabroniony w każdym przypadku i na wszystkich portach, dopuszczony jest jednak rzecz jasna ruch powrotny w ramach połączeń zainicjowanych przez hosty sieci INET (connection-state=established,related).
- ruch wychodzący: co do zasady nie jest filtrowany, z wyjątkiem ograniczeń wymuszonych przez zaporę dla ruchu wchodzącego do DMZ.

Z uwagi na nietrywialną złożoność wirtualizowanej infrastruktury, istotne jest określenie wymagań sprzętowych. Szczególnie ważna jest ilość pamięci operacyjnej, wymagana do działania poszczególnych hostów:

- każdy z hostów posiadających graficzny interfejs użytkownika (USR1, USR2, MDNS) wymaga 2 GB pamięci RAM;
- hosty systemu Snort (IPS1, IPS2) wymagają do sprawnego działania po 768 MB pamięci RAM każdy;
- serwery HTTP (SRV1, SRV2) zajmują po 512 MB pamięci;
- do działania serwera DNS (IDNS) konieczne jest 512 MB, a dla routera ROS – 256 MB pamięci RAM.

Doliczając do tego pamięć systemu gospodarza okazuje się, że dla uruchomienia całej wymaganej infrastruktury niezbędne jest co najmniej 12 GB pamięci operacyjnej. W praktyce będzie to zazwyczaj 16 GB.

## 4 IMPLEMENTACJA

### 4.1 Aplikacja nadawcza i odbiorcza

Aplikacje nadawcza i odbiorcza, realizujące transmisję metodą tunelowania DNS, znajdują się w repozytorium projektu w katalogu `./dns_tools`:

- skrypt `dnsenc.sh` stanowi aplikację nadawczą i przyjmuje dwa argumenty, kolejno: plik do wysłania i domenę docelową. Jego działanie polega na zakodowaniu wysyłanego pliku w formacie base32 i podzieleniu go na 8-znakowe fragmenty. Każdy fragment wysyłany jest następnie jako subdomena zapytania DNS, zgodnie z metodą opisaną w rozdziale 2. Zapytania DNS generowane są za pomocą dostępnego w systemie Linux narzędzia `nslookup`.
- skrypt `dnsdec.js` jest napisany jako skrypt NodeJS i stanowi aplikację odbiorczą. Przyjmuje jeden argument: domenę, dla której nasłuchujemy przychodzących zapytań. Zdekodowane dane wypisywane są domyślnie na standardowe wejście, skąd mogą zostać przekierowane do pliku.

### 4.2 Wtyczka systemu Snort

#### 4.2.1 Struktura plików źródłowych

Narzędzia służące do wykrywania tunelowania DNS znajdują się w katalogu `./ips_dns_tunnel`. W podkatalogach `bayes`, `snort_plugin` oraz `snortd` znajdują się kolejno:

- `bayes`:
- `snort_plugin`:

- snortd:

#### 4.2.2 Algorytm klasyfikacji

## 5 TESTY

## 6 PODSUMOWANIE

Tak przygotowany scenariusz projektowy pozwoli nam na przetestowanie możliwości wykrywania tunelowania DNS z wykorzystaniem uczenia maszynowego oraz zaprezentowanie działania systemów typu IDS/IPS w praktyce.