Bezpieczeństwo Sieci Komputerowych

Laboratorium 4: Zapory ogniowe, filtrowanie ruchu
Instrukcja do instrukcji

Kilka informacji wstępnych:
W opisie powinny znajdować się wszystkie komendy do wykonania na zaporze potrzebne w czasie zajęć i wyjaśnienia do najistotniejszych z nich, polecam pracę z tym tekstem jako uzupełnienie instrukcji. Dodatkowo podałem parę informacji praktycznych dotyczących specyfiki pracy w środowisku laboratoryjnym i omawianych zagadnień (przede wszystkim VPN).
Do ćwiczenia będą potrzebne 2 komputery na stanowiskach, jeden z Windowsem, drugi najlepiej z Ubuntu – będziemy go wykorzystywać w dalszej części zajęć jako serwer, a na Linuksie dużo łatwiej będzie zainstalować i uruchomić potrzebne usługi sieciowe.

Wszystkie podane komendy na ASA trzeba wykonywać w trybie uprzywilejowanym, część z nich w różnych trybach konfiguracji. Według konwencji, której trzymałem się w opisie, przed komendą podany jest pełny ciąg znaków zachęty z wyjątkiem nazwy hosta, przykładowo:

– taką komendę należy wykonać bezpośrednio w trybie uprzywilejowanym (*enable*)
`(config)#` – taką komendę trzeba wykonać w trybie konfiguracji globalnej (*config t*)
`(config-if)#` – taką komendę trzeba wykonać w trybie konfiguracji interfejsu (np. *int e0/7*)

Niektóre komendy na zaporze ASA różnią się trochę od komend na routerach i przełącznikach, ponieważ zapora wykorzystuje nieco inny system operacyjny (bazujący na jądrze Linuksa, a nie Cisco IOS) – ale ogólnie zasady pracy z zaporą są takie same jak z pozostałym sprzętem Cisco. Opis zawiera tylko jedną komendę do wykonania na komputerze z Linuksem i potrzebne są do niej uprawnienia roota – oznaczona jest według standardowej konwencji.

Punkt 1.
W razie potrzeby – usuwanie konfiguracji startowej (trochę inaczej niż na urządzeniach Cisco, z którymi dotąd pracowaliśmy):

```
# write erase
# reload
```

Ładowanie domyślnych ustawień fabrycznych – w trybie konfiguracji:

```
(config)# configure factory-default
```

Domyślnie urządzenie będzie pytać o hasło do trybu *enable*, ale jest ono puste – wystarczy nacisnąć *Enter*.

Punkt 2.
Tutaj powinniśmy sprawdzić konfigurację urządzenia, szczególnie zwrócić uwagę na adresy IP i VLANy (przyda się później). Komenda do wyświetlania całej konfiguracji to standardowo:

```
# show run
```

Natomiast przypisanie portów do VLANów wyświetla się ciut inaczej niż na switchach i routerach:

```
# show switch vlan
```

Zmiana nazwy urządzenia – standardowo:

```
(config)# hostname nasze_nazwiska
```

Punkt 3.
W tym punkcie mamy podłączyć do zapory nasze komputery (dowolne porty po stronie wewnętrznej) i sieć laboratoryjną (port zewnętrzny). Komputery dostaną adresy IP z serwera DHCP uruchomionego na zaporze, ale nie otrzymają adresów serwerów DNS (domyślnie nie są ustawione na ASA) – skonfigurujcie je więc na obu komputerach ręcznie. Wystarczy jeden adres, możecie wykorzystać dowolny serwer DNS jaki znacie, np. jeden z serwerów Politechniki Wrocławskiej (156.17.8.1) albo dobrze znany serwer Google (8.8.8.8).

UWAGA – ponieważ systemy na stanowiskach są odpalane w wirtualkach, z jakiegoś powodu nie zareagują prawidłowo na podłączenie do nowej sieci i zamiast zgłosić się po nowe adresy z DHCP, cały czas będą używać starych. Żeby wymusić nowe adresy, trzeba wyłączyć i włączyć kartę sieciową, dotyczy to oczywiście i Windowsa, i Linuksa.

Z LANu powinien być dostęp do internetu i sieci w laboratorium (tego drugiego w praktyce nie musimy sprawdzać), będzie można komunikować się z serwerami DNS i WWW, ale nie będzie działał ping i tracert – zapora domyślnie blokuje wszystkie pakiety ICMP z zewnątrz.

Tworzenie kont administracyjnych:

```
(config)# username nazwa_konta password haslo privilege 15
```

Punkt 4.
Skonfigurować regułę umożliwiającą pingowanie można na kilka sposobów, najprościej chyba będzie dopisać jedną akcję w domyślnym obiekcie *policy-map*:

```
(config)# policy-map global_policy
(config-pmap)# class inspection_default
(config-pmap-c)# inspect icmp
```

To ustawienie zadziała dokładnie tak, jak potrzebujemy – każdy pakiet ICMP wysyłany z naszej sieci będzie zapamiętywany i odpowiedź na niego zostanie dynamicznie dopuszczona do sieci wewnętrznej, natomiast pakiet ICMP przysłany z zewnątrz zostanie odrzucony.

Do zablokowania FTP w sieci lokalnej wykorzystamy listę ACL:

```
(config)# access-list ftp_ban deny tcp any any eq ftp
(config)# access-list ftp_ban permit ip any any
(config)# access-group ftp_ban in interface inside
```

Utworzyliśmy tutaj nową listę o nazwie *ftp_ban*, wpisaliśmy na nią dwie reguły i przypisaliśmy ją do interfejsu (VLANu) *inside* w kierunku wejściowym (filtrowany jest ruch trafiający do zapory). Do każdego interfejsu w danym kierunku możemy przypisać jedną listę.

Zasada tworzenia reguł na liście jest następująca:

- określamy, czy na ruch w danym wpisie zezwalamy (*permit*), czy go blokujemy (*deny*)
- określamy protokół warstwy transportowej, który będziemy filtrować (np. *tcp*)

- określamy źródło i cel połączenia – *any any* oznacza oczywiście ruch skądkolwiek dokądkolwiek, inne opcje to np. określony host *(host adres_ip*) lub określona sieć (adres i maska, np. *192.168.1.0 255.255.255.0*)
- gdy reguła dotyczy całej sieci, na ASA w przeciwieństwie do routerów Cisco nie stosujemy masek wieloznacznych (wildcard mask), tylko zwykłe – tak jak w przykładzie wyżej
- jeżeli wpis dotyczy protokołu TCP lub UDP, na końcu określamy filtrowany port (np. *eq 80* lub *eq www*) – ponieważ urządzenia Cisco posiadają listę portów używanych przez typowe usługi, zamiast numeru portu możemy użyć nazwy usługi
- lista jest zawsze przeglądana od początku do znalezienia dopasowania – ważna jest więc kolejność reguł, jeżeli np. chcemy zezwolić na każdy możliwy ruch do jakiegoś urządzenia, ale z wyjątkiem pingów – najpierw powinna być reguła blokująca pakiety ICMP, dopiero następna zezwalająca na wszystko
- z poprzedniego punktu wynika jeszcze jedno, jeżeli na liście jest dużo reguł – żeby zminimalizować obciążenie procesora, reguły które dotyczą największej części ruchu powinny znaleźć się możliwie najwyżej na liście

Trzeba pamiętać o jeszcze jednej, ważnej rzeczy. Każda lista ACL zawiera na końcu niewidoczną regułę domyślną i na urządzeniach Cisco jest to *deny any*. Z tego powodu na naszej liście nie wystarczy sama reguła blokująca FTP, wtedy oprócz FTP odcięlibyśmy w ogóle LAN od łączności z internetem. Dlatego dopisaliśmy jeszcze drugą regułę, która zezwoli na cały ruch niedopasowany do pierwszej – dzięki temu rzeczywiście zablokujemy tylko protokół FTP, a nie wszystko. Połączenie FTP możemy sprawdzić nawet za pomocą przeglądarki, łącząc się np. z serwerem warszawskiego ICM* (ważne, żeby przed adresem podać protokół *ftp://*).

Dostęp do ePortalu zablokujemy za pomocą drugiej listy ACL – myślę, że teraz już rozumiecie jej działanie, więc nie będę wyjaśniał:

```
(config)# access-list eportal_ban deny ip any host 156.17.70.219
(config)# access-list eportal_ban permit ip any any
(config)# access-group eportal_ban out interface outside
```

Listy ACL możemy wyświetlić za pomocą komendy (domyślnie wszystkie listy, można też dopisać na koniec nazwę konkretnej i doprecyzować):

```
# show access-list
```

Polecenie wyświetli reguły na liście i liczbę dopasowań każdej z nich do ruchu sieciowego.

Punkt 5.
Tutaj mamy skonfigurować DMZ (strefę zdemilitaryzowaną), czyli sieć serwerową, do której ma być możliwy dostęp z zewnątrz – w związku z tym zasady filtrowania ruchu do takiej sieci są mniej rygorystyczne, ale też ta sieć jest w mniejszym stopniu zaufana i nie powinno być z niej bezpośredniego dostępu do naszej sieci lokalnej.
Do DMZ podłączymy komputer z jednego stanowiska (z Ubuntu). Mamy wykorzystać adresację wewnętrzną z zakresu 172.16.x.0/24 (*x* – nasz nr grupy) i statyczne mapowanie NAT (1:1) na adres z sieci laboratoryjnej podany przez doktora Markowskiego.

* ftp://ftp.icm.edu.pl/

Zaczniemy od stworzenia nowego VLANu i przypisania do niego zgodnie z instrukcją portu e0/7:

```
(config)# interface vlan 3
(config-if)# ip address 172.16.x.1 255.255.255.0
(config-if)# no forward interface vlan 1
(config-if)# nameif dmz
(config-if)# security-level 50
(config-if)# no shutdown
(config-if)# interface e0/7
(config-if)# switchport access vlan 3
```

Następnie skonfigurujemy NAT dla naszego serwera – załóżmy, że jego adres to 172.16.x.5:

```
(config)# object network dmz_net
(config-network-object)# host 172.16.x.5
(config-network-object)# nat (dmz,outside) static 192.168.255.xxx
```

Stworzyliśmy tutaj nowy obiekt nazwany *dmz_net* i ustawiliśmy w nim statyczną translację adresów między VLANem *dmz* (nazwanym tak przez nas przed momentem) i domyślnym VLANem *outside*. W oryginalnej instrukcji w ostatniej komendzie jest błąd – VLAN *inside* zamiast *dmz*) – i taka konfiguracja oczywiście nie działa, zgłosiłem to już doktorowi.
Po podłączeniu do odpowiedniego portu na zaporze komputera, który ma być naszym serwerem i po skonfigurowaniu na nim statycznego adresu (wiadomo, adres – ten sam, który NATujemy, brama – adres routera – nie muszę chyba wyjaśniać), powinna być możliwa łączność z internetem.

Wykonane na zaporze translacje NAT możemy sprawdzić za pomocą komendy:

```
# show xlate
```

Na serwerze instalujemy potrzebne pakiety – serwer WWW (*apache2*), FTP (*vsftpd*) i SSH (*openssh-server*):

```
# apt install apache2 vsftpd openssh-server
```

Po instalacji wszystkie usługi od razu zostaną uruchomione, możemy to potwierdzić łącząc się lokalnie (np. z serwerem WWW). Nie będzie oczywiście do nich dostępu z sieci laboratoryjnej mimo NATu 1:1, ponieważ na zaporze nie ma dodanych odpowiednich reguł.

Na zaporze usuwamy listy ACL skonfigurowane w punkcie 4:

```
(config)# clear configure access-list ftp_ban
(config)# clear configure access-list eportal_ban
```

W zasadzie powinniśmy uruchomić też serwer FTP na drugim komputerze (w LANie, z Windowsem), ale możemy to raczej spokojnie pominąć, nie będzie do niczego teraz potrzebne. Gdyby później okazało się (np. przy testowaniu połączenia VPN), że jednak FTP do czego się przyda, polecam ściągnąć i odpalić *Xlight*[*] – jest dostępna wersja działająca bez instalacji, wystarczy utworzyć nowy serwer (*New Virtual Server*, zostawiamy domyślne ustawienia), ustawić katalog (*Modify Virtual Server Configuration* > *Public Path* > *Add*), dodać użytkownika (*User List* > *Add*) i uruchomić (*Start Server*). W taki sposób chyba najszybciej można to zrobić, bez instalowania IIS.

---

* https://www.xlightftpd.com/

Punkt 6.
W tym punkcie mamy skonfigurować regułę umożliwiającą z zewnątrz ping do DMZ i dostęp do jednej z usług uruchomionych w poprzednim punkcie. Tworzymy nową listę ACL i przypisujemy ją do VLANu *outside*:

```
(config)# access-list dmz_acl permit icmp any host 172.16.1.5
(config)# access-list dmz_acl permit tcp any host 172.16.1.5 eq 80
(config)# access-group dmz_acl in interface outside
```

W tym momencie powinien być już możliwy dostęp do komputera w DMZ i wybranej przez nas pracującej na nim usługi (w tym przykładzie WWW) z sieci laboratoryjnej – możemy to sprawdzić ze stanowiska prowadzącego albo np. ze swojego laptopa po podłączeniu przewodem sieciowym. Reguły zezwalającej na ping w sieci lokalnej nie musimy osobno ustawiać, już zrobiliśmy to w punkcie 4 i nie wykorzystaliśmy do tego listy ACL – więc nie usunęliśmy jej chwilę wcześniej.

Punkt 7.
Ostatnia rzecz, którą mamy na zajęciach skonfigurować, to połączenie VPN AnyConnect. Potrzebny do tego będzie program do konfiguracji zapory za pomocą graficznego interfejsu, który można zainstalować z na komputerze z Windowsem z pliku *dm-launcher.msi* (z linii komend powinno też dać się to zrobić, ale tak będzie łatwiej). Cała konfiguracja jest dobrze opisana na końcu instrukcji, więc nie będę tutaj wszystkiego powtarzał, zwrócę tylko uwagę na kilka kwestii.

Pula adresów przypisywanych klientom VPN powinna znajdować się w sieci lokalnej (VLAN *inside*, standardowo sieć 192.168.1.0/24) i nie powinna kolidować z pulą DHCP – wszystko to możemy oczywiście sprawdzić komendą *show run*, pula DHCP konkretnie będzie zdefiniowana w linijce *dhcpd address* i domyślnie kończy się na 192.168.1.36. Jako maskę podsieci podajemy oczywiście 255.255.255.0, taką samą jak ustawiona w VLANie *inside*.
Jako adres serwera DNS możemy podać cokolwiek (standardowo, najprościej 8.8.8.8), jako domenę tak samo (może być np. bsk.pwr.edu.pl).
Po skonfigurowaniu trzeba połączyć się z jakiegoś innego komputera w sieci laboratoryjnej (stanowisko prowadzącego/laptop) z naszą zaporą po adresie zewnętrznym (na VLANie *outside*), możemy go sprawdzić komendą:

```
# show ip
```

Jeżeli jest zainstalowany klient AnyConnect, możemy od razu za jego pomocą zestawić połączenie VPN wykorzystując któregoś z utworzonych na zaporze użytkowników, jeżeli nie – najpierw połączyć się z zaporą z poziomu przeglądarki i ściągnąć instalator klienta. Po nawiązaniu połączenia VPN komputer, z którego się łączymy, będzie zachowywał się tak, jakby fizycznie znajdował się w sieci lokalnej. Dostanie w niej swój adres IP (z puli ustawionej przy konfiguracji VPN na zaporze) i będzie miał łączność z komputerem podłączonym do wewnętrznego portu zapory po jego adresie IP.

W praktyce jeżeli kiedyś będziecie konfigurować VPN do zdalnego dostępu, o ile z jakiegoś powodu nie będzie potrzebny koniecznie AnyConnect – poleciłbym trochę inne rozwiązanie. Takie połączenie, jak tutaj konfigurowaliście, wymaga instalacji dodatkowego klienta na komputerze. Nie jest to na szczęście aż tak problematyczne, jak w przypadku pierwszej implementacji IPSec w wykonaniu Cisco*, ale mimo wszystko mało wygodne.

* Która na macOS jest wpierana natywnie, na Linuksie bardzo dobrze zintegrowana z systemowymi narzędziami sieciowymi (społeczność się postarała), ale za to na Windowsie zawsze wymagała własnego klienta i – ponieważ zostało zakończone już dla niej wsparcie – nie został on wydany dla Windowsa 10… Jest na tym postawiony VPN dla pracowników PWr umożliwiający dostęp do wewnętrznej części PWr-NETu – i bardzo Was proszę, nigdy czegoś takiego nie konfigurujcie ;-)

Proponuję wykorzystać VPN oparty na IPSec z IKEv2, natywnie wspierany bez instalacji czegokolwiek na wszystkich najpopularniejszych systemach operacyjnych: Windows, Linux, macOS, Android, iOS (i pewnie różnych innych), używający certyfikatów i kryptografii klucza publicznego do uwierzytelnienia serwera. Można go uruchomić na nowszych urządzeniach Cisco (również na ASA 5505 z aktualną wersją systemu), ale też na wielu innych urządzeniach – od specjalnego sprzętu sieciowego różnych firm, po zwykły linuksowy serwer, sam konfigurowałem w ten sposób VPN do swojej sieci domowej na 1. generacji Raspberry Pi z użyciem demona strongSwan. Na początku trochę zabawy jest z wygenerowaniem własnego root CA i podpisaniem za jego pomocą certyfikatu dla serwera VPN (chyba, że mamy wystawiony zaufany powszechnie certyfikat, może być nawet Let's Encrypt), ale później możemy nawiązać połączenie z poziomu samego systemu operacyjnego.
A postawienie VPNu na zwykłym Linuksie w porównaniu z takim urządzeniem jak ASA ma jeszcze pewną istotną zaletę. Cisco z tego co widzę sprzedaje dla ASA oddzielne licencje w zależności od maksymalnej liczby użytkowników VPN + wiele z nich to licencje czasowe. Jak skonfigurujemy VPN po prostu na serwerze linuksowym, moc open source jest z nami i ogranicza nas tylko sprzęt ;-).

I to już wszystko, powodzenia i skonfigurowania wszystkiego – aż do VPNu!

Jan Potocki
Wrocław, 22.11.2018