Laboratorium 5

Linux – Zabezpieczanie i utwardzanie konfiguracji cz. 2

W sprawozdaniu zamieść zrzuty ekranu z istotnych etapów oraz odpowiedzi na pytania.

Taki symbol oznacza, że trzeba w sprawozdaniu dodać zrzut ekranu (najczęściej 1) z wyniku działania polecenie.

Taki symbol oznacza, że należy dodać opis (najczęściej 1 zdanie) z wyniku działania polecenia.

I. Bezpieczniejsze SSH

SSH jest kluczowym narzędziem umożliwiającym zdalną administrację serwerami Linux. Jego bezpieczeństwo ma kluczowe znaczenie, dlatego należy zwrócić szczególną uwagę na odpowiednią konfigurację. Poniżej opisano krok po kroku, jak zainstalować i skonfigurować serwer SSH, korzystać z uwierzytelniania kluczem, zmienić domyślny port oraz zwiększyć bezpieczeństwo.

1. Uruchom maszynę wirtualną z systemem Kali Linux.

2. Zainstaluj serwer SSH:

```
sudo apt install openssh-server
```

3. Uruchom usługę:

```
sudo systemctl start ssh
```

4. Sprawdź status usługi:

```
sudo systemctl status ssh
```

5. Zaloguj się na zdalną maszynę Kali Linux z poziomu CMD lub PowerShell na komputerze hosta, używając SSH.

```
ssh nazwa_użytkownika@adres_IP
```

6. Otwórz terminal w Kali Linux.

Wbrew powszechnej opinii zmiana portu, na którym działa serwer SSH, z domyślnego 22 na inny, nie zwiększa poziomu bezpieczeństwa. Jeśli hasło do konta root jest słabe, to jego złamanie metodą *brute-force* będzie równie łatwe, gdy atakujący znajdzie nowy port, np. za pomocą skanera takiego jak Nmap. Jednak zmiana portu ma pewną zaletę: zmniejsza ilość niepotrzebnych wpisów w logach. Urządzenia z otwartym portem TCP 22 są stale skanowane w Internecie. Zmiana portu pozwala ograniczyć aktywność najprostszych i najczęściej używanych skanerów.

7. Otwórz plik konfiguracyjny `sshd_config` w edytorze tekstowym, np. `nano`.

```
sudo nano /etc/ssh/sshd_config
```

8. Znajdź linijkę:

```
#Port 22
```

9. Usuń znak `#`, a następnie zmień numer portu, np. na `2222`.

Upewnij się, że wybrany port nie jest już używany przez inne usługi i nie jest blokowany przez zaporę sieciową.

10. Zapisz zmiany i zrestartuj usługę SSH:

```
sudo systemctl restart ssh
```


11. Zaloguj się ponownie, podając nowy port:

```
ssh -p nowy_port nazwa_użytkownika@adres_IP
```

Istotnym elementem zabezpieczenia serwera jest zablokowanie możliwości logowania się na konto root przez SSH. W przypadku Kali Linux w nowszych wersjach systemu jest to domyślnie niemożliwe, ponieważ usunięto domyślne hasło użytkownika root. W innych dystrybucjach jednak nie zawsze jest to standardem i warto to skonfigurować ręcznie.

12. Ponownie otwórz plik konfiguracyjny serwera SSH.

13. Znajdź linijkę:

```
#PermitRootLogin prohibit-password
```

14. Usuń znak `#`, aby odkomentować tę linię, i zmień wartość `prohibit-password` na `no`.

Wartość `no` całkowicie uniemożliwia zalogowanie jako `root` przez SSH. Mamy jednak do dyspozycji również inne warianty:

- `yes` – zezwala na logowanie się hasłem, co umożliwia przeprowadzenie ataków siłowych.
- `prohibit-password` – umożliwia logowanie jako root wyłącznie przy użyciu kluczy SSH.

15. Zapisz zmiany i zrestartuj usługę SSH.

Klucze SSH zapewniają łatwy i bezpieczny sposób logowania się do serwera i są zalecane dla wszystkich użytkowników. Pierwszym krokiem jest utworzenie pary kluczy na komputerze klienta, czyli tym, który będzie używany do logowania (w naszym przypadku na systemie gospodarzu).

16. Wygeneruj parę kluczy w standardzie Ed25519 na komputerze klienta: w CMD lub PowerShell:

```
ssh-keygen -t ed25519
```

17. Powinieneś otrzymać podobne wyjście jak poniżej:

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\Michał Ćmil/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Michał Ćmil/.ssh/id_ed25519
Your public key has been saved in C:\Users\Michał Ćmil/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:u3I7RHmetgSTGkyoLlDucEclcxZjVER1VApaEhfyJAM michał Ćmil@DESKTOP-7ISDA9
The key's randomart image is:
+--[ED25519 256]--+
|    o+E*XoBoo..  |
|  . o*.. @ o .   |
| o o o  .o. .    |
|o + . o * .      |
|.= .   +S= .     |
|. o   . ..=      |
| .     ..o .     |
|      . o..      |
|       ooo       |
+----[SHA256]-----+
```

W wyniku działania generatora kluczy powstaną dwa pliki: klucz publiczny (z sufiksem `.pub`) oraz klucz prywatny (opisany jako *identification*). Pamiętaj, że plik klucza prywatnego jest sekretem uwierzytelniania i należy go bezwzględnie chronić przed ujawnieniem. Dodatkowo, generator kluczy tworzy obrazek semigrafizyczny (ang. Randomart image), który służy do wizualnego porównania kluczy – dla wielu użytkowników jest to łatwiejsza metoda porównania niż analizowanie długich ciągów tekstowych.

18. Aby ponownie wyświetlić obrazek semigrafizyczny, użyj komendy:

```
ssh-keygen -lv -f twój_klucz.pub
```

19. Inną formą wyświetlania klucza jest tzw. bubblebabble, która umożliwia łatwe przedyktowanie klucza przez telefon:

```
ssh-keygen -B -f twój_klucz.pub
```

20. Wynik powinien wyglądać mniej więcej tak:

```
256 xozam-hutop-[…]-nexex michał@DESKTOP-7dsda9 (ED25519)
```

21. Aby wyświetlić zawartość swojego klucza publicznego, otwórz go w edytorze tekstu (np. Notatniku). Powinieneś zobaczyć coś podobnego do poniższego:

```
ssh-ed25519   AAAAC3NzaC1lZDI1NT[…]ZTIgHqBp/fflJEt6T/S   michał@DESKTOP-7ISDA9
typ klucza          materiał klucza publicznego              etykieta
```

Klucz publiczny składa się z trzech elementów: pierwsza część określa typ klucza (np. ssh-rsa, ssh-ed25519), druga to właściwy klucz, a trzecia to etykieta. Etykieta może być dowolnym ciągiem znaków lub w ogóle nie występować – jej edycja nie wpływa na działanie klucza. Należy jednak pamiętać, że nie należy traktować etykiety jako dowodu pochodzenia klucza – atakujący, który podłoży klucz zapewne dostosuje etykietę, by ta nie wzbudzała podejrzeń.

Przejdźmy teraz do implementacji klucza na serwerze. Domyślnie w katalogu domowym użytkownika, do którego się logujemy, klucz publiczny powinien znaleźć się w pliku `~/.ssh/authorized_keys`. Wpis dla danego klucza to jedna linia, taka sama jak ta w pliku .pub na komputerze klienta. W przypadku Kali Linux, katalog `.ssh` oraz plik `authorized_keys` domyślnie nie istnieją. Ponadto, plik `authorized_keys` musi

mieć uprawnienia `600`, a katalog `.ssh` – `700`. Ważne jest, aby te pliki były powiązane z odpowiednim użytkownikiem, a nie z użytkownikiem root.

22. Utwórz strukturę zgodnie z powyższym opisem.

23. Wklej zawartość pliku `twój_klucz.pub` do `authorized_keys`.

24. Przejdź do konsoli na systemie gospodarza i spróbuj połączyć się z wykorzystaniem klucza:

```
ssh -p nowy_port -i klucz_prywatny nazwa_użytkownika@adres_IP
```

Udało nam się zalogować przy użyciu klucza. Jednak mechanizm uwierzytelniania oparty na haśle jest nadal aktywny, co oznacza, że serwer wciąż jest narażony na ataki siłowe. Czas to zmienić.

25. Otwórz plik konfiguracyjny SSH.

26. Znajdź linijkę:

```
#PasswordAuthentication yes
```

27. Usuń znak `#` na początku linii i ustaw wartość na `no`. Spowoduje to wyłączenie możliwości logowania się do konta przez SSH przy użyciu hasła.

28. Zapisz zmiany i zrestartuj SSH.

29. Spróbuj zalogować się przez SSH bez użycia klucza, korzystając z PuTTy – nie powinno być to możliwe.

II. Firewall i ochrona warstwy sieciowej

Jak skonfigurować zaporę sieciową w systemie Linux? Konfiguracja zapory sieciowej w systemie Linux zależy od jej przeznaczenia. Inaczej będzie skonfigurowana zapora na serwerze WWW, inaczej w klastrze Kubernetes, a jeszcze inaczej na laptopie z wirtualnymi maszynami podłączonymi do sieci Wi-Fi.

Podstawowy mechanizm zarządzania ruchem sieciowym w Linuksie to *netfilter*, który jest wbudowany w jądro systemu. Natomiast narzędzie iptables jest to tylko interfejsem służącym do konfiguracji tego mechanizmu.

W iptables istnieją wbudowane tablice, które zawierają łańcuchy (np. INPUT, OUTPUT). W łańcuchach definiuje się reguły, które mogą być związane ze specyficznymi adresami IP, usługami czy portami. Na tej podstawie system podejmuje decyzje, czy dany ruch sieciowy jest dozwolony..

Wiele działających obecnie systemów linuksowych jest zabezpieczonych przez firewalle zbudowane z wykorzystaniem tych właśnie narzędzi (bardzo często schowanych pod jedną z nakładek ułatwiających zarządzenie np. ufw.)

Zapora sieciowa pełni kilka (tutaj wymienione 3) kluczowych funkcji, takich jak:

- **Filtrowanie ruchu przychodzącego** – blokowanie lub ograniczanie niepożądanego ruchu. Wpuszczane są jedynie pakiety zgodne z ustalonymi regułami, np. odpowiedzi na wysłane zapytania.
- **Filtrowanie ruchu wychodzącego** – Choć rzadziej stosowane, może być używane do zapobiegania wyciekom danych oraz utrudniania działań atakującym.
- **Monitorowanie ruchu** – Zapisywanie informacji o ruchu sieciowym, takich jak liczba przesłanych pakietów czy logowanie nawiązanych i odrzuconych połączeń.

Aby upewnić się, że zapora działa poprawnie, zaleca się regularne skanowanie systemu z zewnątrz, np. za

pomocą narzędzia `Nmap`, w tym przynajmniej raz z pełnym zakresem portów TCP/UDP.

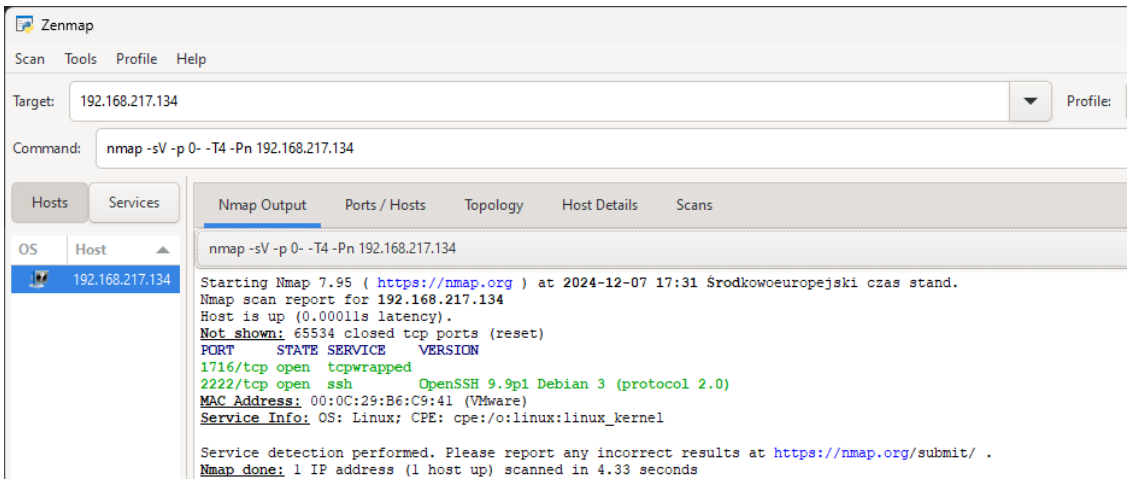
1. Sprawdź czy na komputerze gospodarza zainstalowany jest Nmap. Jeśli Nmap nie jest zainstalowany, możesz go pobrać z oficjalnej strony i zainstalować..

2. Uruchom konsolę cmd lub Zenmap (nakładka graficzna na Nmap).

3. Uruchom komendę:

```
nmap -Pn -p0- -sV -T4 adres_IP_maszyny
```

- **`-Pn`**: Oznacza pominięcie etapu sprawdzania, czy host jest dostępny (tzw. host discovery). Nmap zakłada, że host jest online, i przechodzi bezpośrednio do skanowania portów, niezależnie od tego, czy host odpowiada na ping (ICMP echo request) lub inne metody weryfikacji.
- **`-p0-`**: Określa zakres portów do przeskanowania. 0- oznacza, że skanowane będą wszystkie porty, od 0 do 65535.
- **`-sV`**: Aktywuje detekcję wersji usług działających na otwartych portach. Po zidentyfikowaniu otwartego portu Nmap wysyła specjalne zapytania, aby zidentyfikować wersję usługi, np. wersję serwera SSH, HTTP itp.
- **`-T4`**: Określa poziom agresywności skanowania (od T0 do T5). T4 oznacza szybkie skanowanie, przeznaczone dla stosunkowo stabilnych sieci i systemów.
- **`adres_IP_maszyny`**: Adres IP hosta, który jest celem skanowania.



Jak widać, w tym przypadku wykryto tylko dwa otwarte porty, jednak warto zaznaczyć, że było to szybkie skanowanie. Dokładniejsze skanowanie (choć znacznie wolniejsze), uwzględniające protokół UDP, mogłoby ujawnić więcej otwartych portów.

4. Przejdź do terminala Kali Linux.

5. Wyświetla listę aktualnych reguł zapory sieciowej (iptables) w trybie szczegółowym (verbose):

```
sudo iptables -L -v
```

- `iptables`: Narzędzie do zarządzania zaporą sieciową.
- `-L`: Listuje (wyświetla) reguły w każdej łańcuchu w domyślnej tabeli (filter).
- `-v`: Tryb szczegółowy, który wyświetla dodatkowe informacje o regułach, takie jak liczba pakietów oraz ilość danych (w bajtach) przetworzonych przez każdą z nich.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Obecnie nie masz zdefiniowanych żadnych reguł. Czas to zmienić.

Do tej pory mogłeś bez problemu pingować maszynę wirtualną z komputera gospodarza. W dostępnych w Internecie poradnikach można jednak czasem trafić na sugestię całkowitego blokowania ruchu ICMP, aby stać się „niewidzialnym" dla przestępców.

ICMP to protokół używany przez polecenie `ping`. Aby zablokować pingowanie, należy odrzucić pakiety ICMP typu echo-request (czyli potocznie pingi).

6. Dodaj regułę blokującą ICMP echo-request:

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

- `-A INPUT`: Dotyczy pakietów przychodzących.
- `-p icmp`: Dotyczy protokołu ICMP.
- `--icmp-type echo-request`: Dotyczy tylko żądań ping (echo request).
- `-j DROP`: Ignoruje takie pakiety.

7. Aby upewnić się, że reguła została zastosowana, sprawdź aktualne reguły w zaporze:

```
sudo iptables -L -v
```

8. Spróbuj teraz spingować maszynę wirtualną z maszyny gospodarza – powinno to zakończyć się niepowodzeniem.

```
PS C:\Users\Michał Ćmil> ping 192.168.217.134

Pinging 192.168.217.134 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Należy jednak rozwiać pewne wątpliwości dotyczące całkowitego blokowania ICMP. Nawet jeśli odrzucimy pakiety typu *echo-request*, maszyna wirtualna wciąż będzie widoczna w sieci dla odpowiednio zaawansowanego skanera portów, który i tak znajdzie wszystkie otwarte usługi.

Blokowanie ICMP może również pozbawić maszynę ważnych funkcji diagnostycznych, ponieważ wiele takich komunikatów opiera się właśnie na pakietach ICMP. Na przykład polecenie traceroute stanie się bezużyteczne.

Lepszym rozwiązaniem niż całkowite odcięcie ICMP może być limitowanie ruchu ICMP, co skutecznie chroni przed atakami typu flood ping (czyli atakami DDoS przy użyciu pingów) bez negatywnego wpływu na diagnostykę sieci.

9. Usuń starą regułę blokującą ICMP:

```
sudo iptables -D INPUT -p icmp --icmp-type echo-request -j DROP
```

10. Upewnij się, że reguła została usunięta, sprawdzając aktualne reguły.

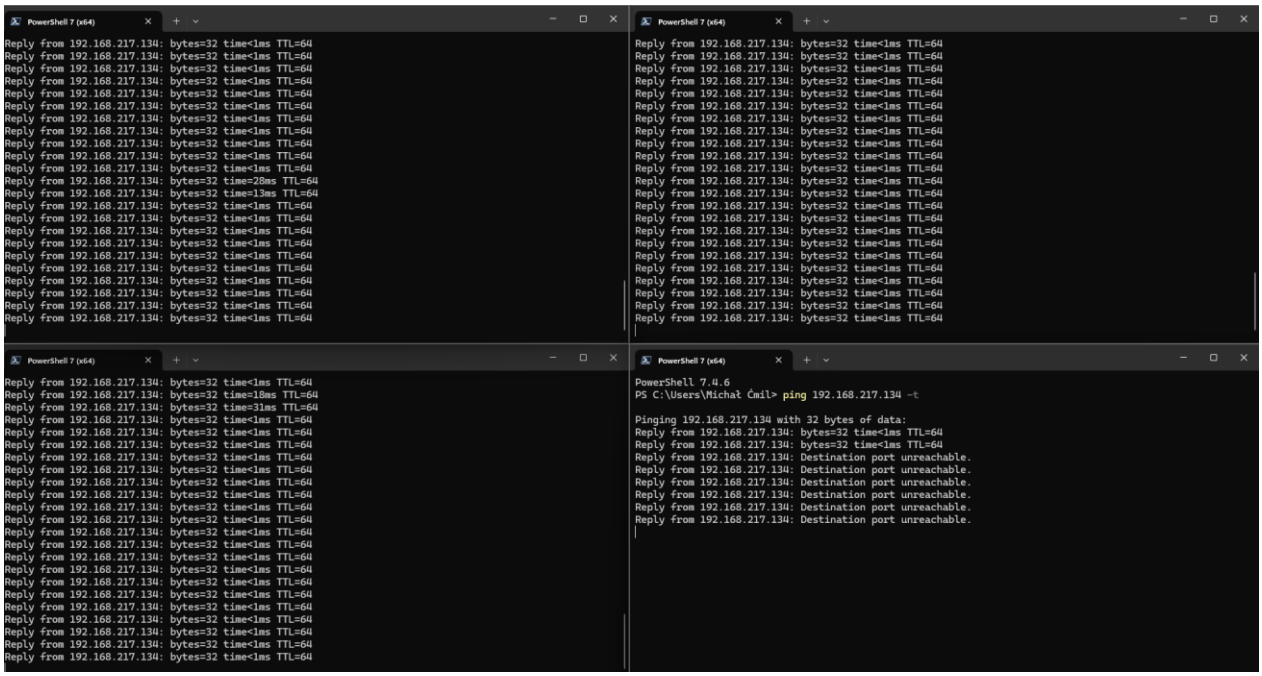
```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 655 packets, 49469 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

11. Aby ograniczyć liczbę odbieranych pakietów ICMP, wykonaj następujące polecenia:

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 3/s -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT
```

- Pierwsza reguła pozwala na odbieranie maksymalnie 3 pingów na sekundę.
- Druga reguła odrzuca wszystkie pakiety ICMP przekraczające ten limit.

12. Uruchom trzy instancje wiersza poleceń (CMD) na maszynie gospodarza i wykonaj polecenie `ping` do maszyny wirtualnej. Następnie uruchom kolejne instancje, aby sprawdzić, jak działa limitowanie ruchu ICMP.



13. Usuń wszystkie reguły utworzone do tej pory:

```
sudo iptables -F
```

14. Zainstaluj Apache na systemie Kali Linux:

```
sudo apt install apache2
```

15. Uruchom serwer Apache:

```
sudo systemctl start apache2
```

16. Sprawdź jego status:

```
sudo systemctl status apache2
```

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Sat 2024-12-07 13:38:23 EST; 12min ago
 Invocation: 074ddc677ba644feb84c42e760907a6b
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 164694 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 164710 (apache2)
      Tasks: 8 (limit: 2258)
     Memory: 21.7M (peak: 22.1M)
        CPU: 227ms
     CGroup: /system.slice/apache2.service
             ├─164710 /usr/sbin/apache2 -k start
             ├─164713 /usr/sbin/apache2 -k start
             ├─164714 /usr/sbin/apache2 -k start
             ├─164715 /usr/sbin/apache2 -k start
             ├─164716 /usr/sbin/apache2 -k start
             ├─164717 /usr/sbin/apache2 -k start
             ├─164771 /usr/sbin/apache2 -k start
             └─168144 /usr/sbin/apache2 -k start
```

17. Sprawdź działanie Apache w przeglądarce: Otwórz przeglądarkę internetową na komputerze gospodarza i wpisz adres IP maszyny wirtualnej. Powinieneś zobaczyć domyślną stronę Apache.

```
http://<adres_IP_twojej_maszyny>
```

Do tej pory blokowaliśmy ruch sieciowy, który uznawaliśmy za zbędny (np. ICMP). Nie jest to jednak najlepsze podejście. Zgodnie z zasadą minimalnych uprawnień (principle of least privilege), zapora powinna być skonfigurowana według reguły: „Zablokuj wszystko, a potem zezwalaj na to, co potrzebne."

Oznacza to, że należy najpierw zablokować cały ruch (ograniczamy się tutaj do ruchu przychodzącego), a następnie definiować wyjątki dla konkretnych usług, protokołów i portów.

Ważnym nawykiem jest dokumentowanie reguł zapory. To, co dziś wydaje się oczywiste, jutro może być zapomniane. Jeśli ruch lub usługa wzbudza wątpliwości lub jest niezrozumiała – najlepiej ją zablokować. Jeśli pojawią się problemy, użytkownicy sami zgłoszą potrzebę odblokowania.

18. Zablokuj cały ruch przychodzący:

```
sudo iptables -P INPUT DROP
```

Po wykonaniu tej komendy cały ruch przychodzący zostanie zablokowany. Jeśli byłeś połączony przez SSH, połączenie zostanie zerwane.

19. Sprawdź, czy możesz połączyć się przez SSH oraz otworzyć stronę Apache w przeglądarce. Oba działania powinny zakończyć się niepowodzeniem.

20. Dodaj reguły zezwalające na ruch na portach SSH i HTTP:

```
sudo iptables -A INPUT -p tcp --dport xxx -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

21. Sprawdź dostęp do SSH oraz HTTP.

22. Sprawdź, czy można pingować maszynę wirtualną.

23. Usuń wszystkie reguły dodane do zapory:

```
sudo iptables -F
```

24. Przywróć domyślne polityki dla łańcuchów:

```
sudo iptables -P INPUT ACCEPT
```

25. Sprawdź aktualne reguły:

```
sudo iptables -L -v
```

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 6532 packets, 5492K bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Jak mogłeś zauważyć, tworzenie reguł w iptables może być czasochłonne i skomplikowane. Istnieją jednak narzędzia, takie jak ufw, które upraszczają ten proces.

26. Zaktualizuj listę pakietów, a następnie zainstaluj UFW na Kali Linux:

```
sudo apt install ufw
```

27. Ustaw domyślną politykę łańcucha OUTPUT na blokowanie ruchu wychodzącego:

```
sudo ufw default deny outgoing
```

28. Uruchom UFW:

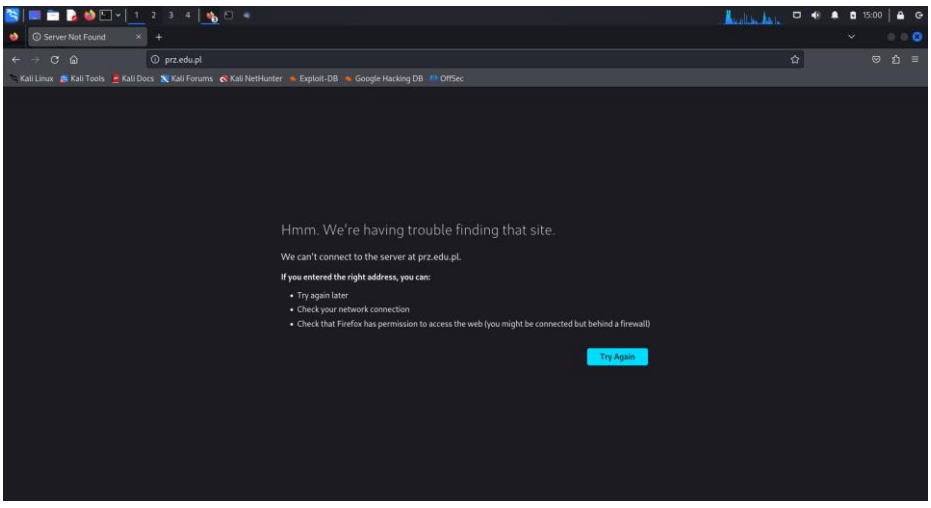
```
sudo ufw enable
```

29. Sprawdź status UFW:

```
sudo ufw status verbose
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip
```

30. Otwórz przeglądarkę w Kali Linux i spróbuj wejść na `prz.edu.pl`.



31. Spróbuj spingować adres IP 62.93.32.24 (prz.edu.pl).

```
┌──(kali㉿kali)-[~]
└─$ ping 62.93.32.24
PING 62.93.32.24 (62.93.32.24) 56(84) bytes of data.
^C
── 62.93.32.24 ping statistics ──
66 packets transmitted, 0 received, 100% packet loss, time 66547ms
```

32. Jak widać, oba działania nie są możliwe z powodu blokady. Przywróć domyślną politykę OUTPUT na zezwalanie ruchu wychodzącego:

```
sudo ufw default allow outgoing
```

33. Sprawdź, czy polityka została przywrócona:

```
sudo ufw status verbose
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

Widać, że domyślna polityka dla ruchu wychodzącego (OUTPUT) jest ustawiona na allow, co oznacza, że cały ruch wychodzący jest teraz dozwolony.

34. Po przywróceniu polityki spróbuj ponownie pingować adres IP 62.93.32.24 oraz wejść na stronę prz.edu.pl.

35. Spróbuj połączyć się z serwerem FTP ftp.helion.pl (nie loguj się).

```
┌──(kali㉿kali)-[~]
└─$ ftp ftp.helion.pl
Connected to ftp.helion.pl.
220 ProFTPD Server ready.
Name (ftp.helion.pl:kali):
```

36. Zablokuj ruch wychodzący na porcie 21 (FTP):

```
sudo ufw deny out 21/tcp
```

37. Spróbuj ponownie połączyć się z serwerem FTP. Połączenie powinno być niemożliwe.

38. Wyświetl wszystkie reguły wraz z ich numerami:

```
sudo ufw status numbered
```

39. Usuńmy daną regułę. Aby usunąć regułę, użyj poniższego polecenia, zamieniając [NUMER_REGUŁY] na odpowiedni numer:

```
sudo ufw delete [NUMER_REGUŁY]
```