

# Wykrywanie zagrożeń i reakcja na incydenty

Laboratorium 4

Tomasz Jarząbek 272279  
Wiktoria Migasiewicz 272177

01.04.2025

## Spis treści

|          |                                     |          |
|----------|-------------------------------------|----------|
| <b>1</b> | <b>Opis laboratorium</b>            | <b>3</b> |
| <b>2</b> | <b>Rozwiązania</b>                  | <b>3</b> |
| 2.1      | Przygotowanie . . . . .             | 3        |
| 2.1.1    | Przygotowanie teoretyczne . . . . . | 3        |
| 2.2      | Domyślne reguły . . . . .           | 4        |
| 2.3      | IPtables . . . . .                  | 5        |
| 2.3.1    | Zadanie 1: . . . . .                | 7        |
| 2.3.2    | Zadanie 2: . . . . .                | 8        |
| 2.3.3    | Zadanie 3: . . . . .                | 9        |
| 2.3.4    | Zadanie 4: . . . . .                | 10       |
| 2.3.5    | Zadanie 5: . . . . .                | 11       |
| 2.3.6    | Zadanie 6: . . . . .                | 12       |
| 2.3.7    | Zadanie 7: . . . . .                | 13       |
| 2.3.8    | Zadanie 8 (Dla chętnych): . . . . . | 14       |
| 2.4      | Fail2Ban . . . . .                  | 14       |

## 1 Opis laboratorium

Laboratorium polegało na analizie i użyciu narzędzi iptables oraz Fail2Ban w środowisku dwóch maszyn wirtualnych: Kali Linux (Kali VM) oraz Kali Live. Narzędzia zostały najpierw zbadane i wypisano ich pewne właściwości, a używano ich do odrzucania ruchu między maszynami wirtualnymi i rejestracji wszelkich czynności w postaci logów systemowych.

## 2 Rozwiązania

### 2.1 Przygotowanie

W celu rozpoczęcia laboratorium, najpierw przygotowano odpowiednie środowisko pracy, które składało się z dwóch maszyn wirtualnych: Kali Linux oraz Kali Live. Obie z nich działają na systemie operacyjnym Kali Linux. Obydwie maszyny zostały umieszczone w tej samej sieci w celu zapewnienia komunikacji między nimi.

W ramach maszyny Kali Linux zainstalowano narzędzie rsyslog (*sudo apt-get -y install rsyslog*) służące do sporządzania logów systemowych, co niezbędne będzie do późniejszej analizy filtrowanych pakietów.

#### 2.1.1 Przygotowanie teoretyczne

Merytoryczne przygotowanie do zajęć polegało na odpowiedzeniu na kilka konkretnych pytań dotyczących teorii wykorzystywanych narzędzi.

**Proszę krótko opisać czym różnią się firewalles typu stateless i stateful oraz wskazać jakiego typu firewallem jest narzędzie iptables.**

Różnica polega na tym, że **stateful** mogą analizować całą otrzymany pakiet pod kątem wykrycia wszelkich nieprawidłowości. Stateful firewall śledzi stan połączenia, ale nie analizuje pakietów pod kątem np. malware (to robią IDS/IPS). W przypadku **stateless** pakiety są jedynie analizowane pod kątem wprowadzonych przez administratora zasad i reguł co do filtracji, na podstawie m.in. źródła i destynacji. **Iptables to firewall typu stateful** ze względu na fakt, że monitorowane są takie aspekty ramki jak np. stan oraz sprawdzają aktywne połączenia za pomocą conntrack. Natomiast ustawienie reguły jedynie na podstawie źródła i celu pakietu sprawiłoby, że rzeczywiście Iptables funkcjonowałby w trybie stateless.

**Proszę zapoznać się z dokumentacją narzędzia iptables i opisać:**

- Czym są „chains”, jakie są dostępne w systemie KALI Linux i do czego służą
- Czym różnią się od siebie dyrektywy DROP oraz REJECT

Według dokumentacji iptables, **chainem** nazywany zbiór reguł w wykonywanej komendzie według syntaxu:

```
1 iptables [-t table] -[AD] chain rule-specification [options]
```

Każdy pakiet jest sprawdzany kolejno według wpisanych reguł w łańcuchu. Jeśli nie pasuje do żadnej, stosowana jest domyślna polityka (policy) danego łańcucha. Jeśli reguła pasuje, może zostać podjęta akcja (np. ACCEPT, DROP, REJECT).

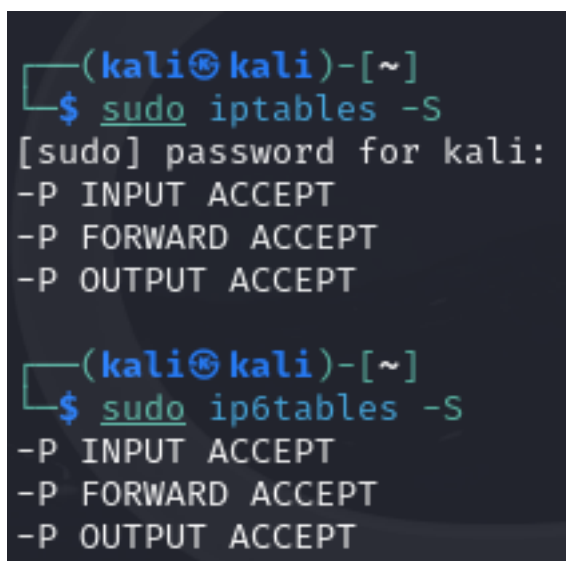
Domyślnie w Kali Linux dostępne są trzy typy chain-ów:

- INPUT – kontroluje pakiety przychodzące do systemu
- OUTPUT – kontroluje pakiety wychodzące z systemu
- FORWARD – używany w przypadku przekazywania pakietów między interfejsami (np. gdy maszyna działa jako router).

**DROP a REJECT** są bardzo do siebie podobne, bo obie kończą się tym że pakiet jest odrzucony i zostaje odfiltrowany. Natomiast różnica polega na tym, iż **DROP**, jak nazwa wskazuje, "upuszcza" pakiet. W przeciwieństwie do **REJECT**, nie zwraca do wysyłającego żadnej informacji zwrotnej. Reject, jak nazwa insynuuje, odrzuca pakiet, odsyłając go z powrotem do nadawcy. To już daje atakującemu znać, iż istnieje jakiś firewall lub urządzenie końcowe w ogóle na danym adresie.

## 2.2 Domyślne reguły

W celu sprawdzenia, jakie reguły istnieją domyślnie na maszynie Kali Linux, wykonano komendę `sudo iptables -S`, która wylistowała je.



```
(kali㉿kali)-[~]  
$ sudo iptables -S  
[sudo] password for kali:  
-P INPUT ACCEPT  
-P FORWARD ACCEPT  
-P OUTPUT ACCEPT  
  
(kali㉿kali)-[~]  
$ sudo ip6tables -S  
-P INPUT ACCEPT  
-P FORWARD ACCEPT  
-P OUTPUT ACCEPT
```

Rysunek 1: Domyślne reguły ruchu na Kali Linux.

## 2.3 IPtables

Konfiguracja maszyn wirtualnych:

Adres maszyny kali live: 192.168.100.17

```
(kali@kali)-[~]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:97:d2:d0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.17/24 brd 192.168.100.255 scope global dynamic noprefixroute eth0
        valid_lft 84678sec preferred_lft 84678sec
    inet6 fe80::5f6f:acb9:e615:ccc6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Rysunek 2: Adres Kali Live.

Adres maszyny kali linux: 192.168.100.16

```
(kali@kali)-[~]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:72:1f:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.16/24 brd 192.168.100.255 scope global dynamic noprefixroute eth0
        valid_lft 85744sec preferred_lft 85744sec
    inet6 fe80::a00:27ff:fe72:1f70/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Rysunek 3: Adres Kali Linux.

Wykaz łączności między maszynami:

```
(kali@kali)-[~]
$ ping 192.168.100.16
PING 192.168.100.16 (192.168.100.16) 56(84) bytes of data.
64 bytes from 192.168.100.16: icmp_seq=1 ttl=64 time=4.70 ms
64 bytes from 192.168.100.16: icmp_seq=2 ttl=64 time=1.25 ms
^C
— 192.168.100.16 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.251/2.973/4.695/1.722 ms
```

Rysunek 4: Łączność między maszynami.

## Sekcja 6

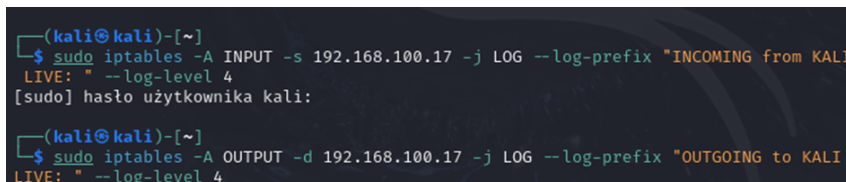
Polecenie:

Proszę skonfigurować regułę pozwalającą na logowanie połączeń z i do adresu IP maszyny Kali

Live.

**Wykorzystana komenda:**

sudo iptables -A OUTPUT -d 192.168.100.17 -j LOG --log-prefix "OUTGOING to KALI LIVE:  
-log-level 4



```
(kali@kali)-[~]
$ sudo iptables -A INPUT -s 192.168.100.17 -j LOG --log-prefix "INCOMING from KALI
LIVE: " --log-level 4
[sudo] hasło użytkownika kali:

(kali@kali)-[~]
$ sudo iptables -A OUTPUT -d 192.168.100.17 -j LOG --log-prefix "OUTGOING to KALI
LIVE: " --log-level 4
```

Rysunek 5: Logowanie połączeń - komenda.

**Komentarz:**

- -A INPUT: dodaje regułę do łańcucha INPUT (dla połączeń przychodzących).
- -s 192.168.100.17: ruch ze źródła – Kali Live.
- -d 192.168.100.17: ruch do celu – Kali Live.
- -j LOG: akcja – zapisanie do logu, a nie blokowanie.
- --log-prefix: dodaje własny tekst, żeby łatwiej znaleźć wpisy w logach.
- --log-level 4: poziom logowania (4 to warning – widać w /var/log/syslog lub /var/log/messages).

## Sekcja 7

**Polecenie:**

Proszę zapisać konfigurację iptables do pliku. **Wykorzystana komenda:**

sudo iptables-save > iptables-backup.rules

**Komentarz:** Domyślnie iptables nie zapisuje reguł po restarcie – trzeba je eksportować do pliku. Komenda zapisuje aktualne reguły do pliku iptables-backup.rules.

## Sekcja 8

**Polecenie:** Proszę ustawić, przetestować oraz zaprezentować logi dla następujących reguł w ramach maszyny Kali Linux. Po każdym z podpunktów proszę przywrócić zapisaną konfigurację narzędzia iptables. Testów nie trzeba wykonywać dla podpunktów z gwiazdką.

**Obecne reguły:**

```
(kali@kali)-[~]
$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 6 packets, 432 bytes)
 pkts bytes target    prot opt in     out     source            destination
  0      0 LOG      all  --  *      *        192.168.100.17    0.0.0.0/0         LOG flags 0 level 4
prefix "INCOMING from KALI LIVE: "

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
  0      0 LOG      all  --  *      *        0.0.0.0/0         192.168.100.17    LOG flags 0 level 4
prefix "OUTGOING to KALI LIVE: "

(kali@kali)-[~]
$
```

Rysunek 6: Obecne reguły.

Logi sprawdzane były poprzez użycie komendy: `sudo tail -f /var/log/syslog`

```
(kali@kali)-[~]
$ sudo tail -f /var/log/syslog
```

Rysunek 7: Sprawdzanie logów - komenda.

Pierwotne reguły przywracane były za pomocą komendy: `sudo iptables-restore < iptables-backup.rules`

```
(kali@kali)-[~]
$ sudo iptables-restore < iptables-backup.rules
```

Rysunek 8: Przywracanie komend.

### 2.3.1 Zadanie 1:

Blokada połączeń do usługi SSH. Test można wykonać w oparciu o polecenie `ssh`.

Wykorzystane komendy: `sudo iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "BLOCK`

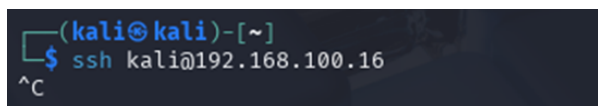
SSH: `"sudo iptables -A INPUT -p tcp --dport 22 -j DROP`

```
(kali@kali)-[~]
$ sudo iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "BLOCK SSH: "
sudo iptables -A INPUT -p tcp --dport 22 -j DROP
```

Rysunek 9: Komenda do blokady SSH.

- -A INPUT: ruch przychodzący.
- -p tcp: dotyczy protokołu TCP.
- --dport 22: port docelowy 22 (SSH).
- -j LOG: loguj próbę połączenia.
- -j DROP: odrzuć pakiet bez odpowiedzi.

Komenda wykorzystana na maszynie kali live w celu testów:



Rysunek 10: Test SSH z Kali Live.

Logi z maszyny kali linux:

```
2025-04-06T18:30:52.915079+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=49337 DF PROTO=TCP SPT=54730 DPT=22 WINDOW=64240 RES=0x00 SYN URGP=0
2025-04-06T18:30:52.915096+02:00 kali kernel: BLOCK SSH: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=49337 DF PROTO=TCP SPT=54730 DPT=22 WINDOW=64240 RES=0x00 SYN URGP=0
2025-04-06T18:31:01.106019+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:
```

Rysunek 11: Logi na Kali Linux z SSH.

Próba połączenia się z maszyny kali live na maszynę kali inux zakończyła się niepowodzeniem, tym samym reguła została pozytywnie przetestowana. Po długim okresie oczekiwania proces łączenia został przerwany kombinacją klawiszy Ctrl + C.

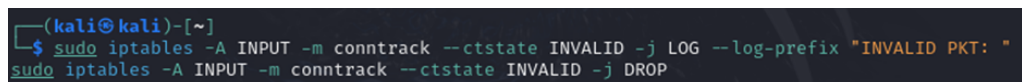
### 2.3.2 Zadanie 2:

Blokada połączeń opartych o błędne (INVALID) pakiety TCP. Test można wykonać w oparciu o polecenie nmap i skan typu „sS” z użyciem przykładowo flag SYN oraz FIN.

#### Wykorzystane komendy:

```
sudo iptables -A INPUT -m conntrack --ctstate INVALID -j LOG --log-prefix "INVALID PKT: "
```

```
sudo iptables -A INPUT -m conntrack --ctstate INVALID -j DROP
```

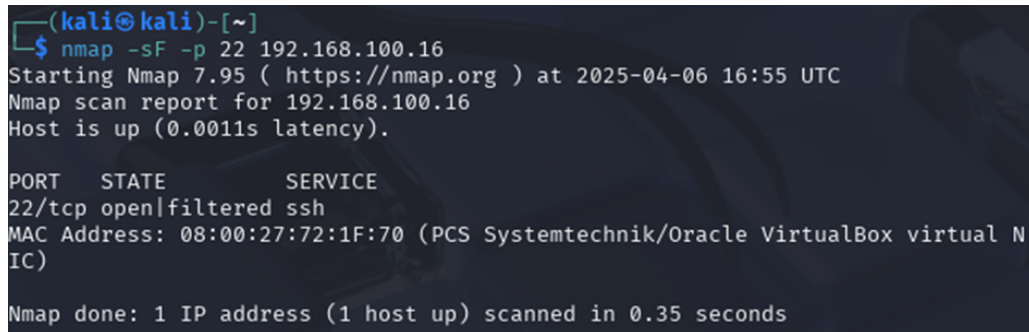


Rysunek 12: Komendy dla INVALID.

- -m conntrack: używa modułu śledzenia połączeń.
- --ctstate INVALID: pakiety nie należące do żadnego aktywnego połączenia.
- -j LOG: loguj pakiety.
- -j DROP: odrzucić pakiet.

**Komenda wykorzystana na maszynie kali live w celu testów:** nmap -sF -p 22 192.168.100.16





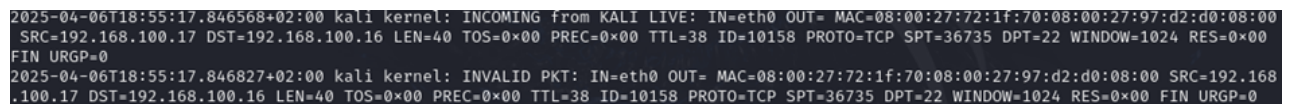
```
(kali㉿kali)-[~]
$ nmap -sF -p 22 192.168.100.16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-06 16:55 UTC
Nmap scan report for 192.168.100.16
Host is up (0.0011s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: 08:00:27:72:1F:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Rysunek 13: Test z Kali Live.

Logi z maszyny kali linux:



```
2025-04-06T18:55:17.846568+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00
SRC=192.168.100.17 DST=192.168.100.16 LEN=40 TOS=0x00 PREC=0x00 TTL=38 ID=10158 PROTO=TCP SPT=36735 DPT=22 WINDOW=1024 RES=0x00
FIN URG=0
2025-04-06T18:55:17.846827+02:00 kali kernel: INVALID PKT: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168
.100.17 DST=192.168.100.16 LEN=40 TOS=0x00 PREC=0x00 TTL=38 ID=10158 PROTO=TCP SPT=36735 DPT=22 WINDOW=1024 RES=0x00 FIN URG=0
```

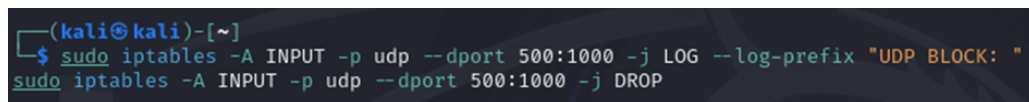
Rysunek 14: Zarejestrowane logi na Kali Linux.

Skan typu FIN został odrzucony, logi wskazują próbę użycia niepoprawnych pakietów.

### 2.3.3 Zadanie 3:

Blokada wszelkiej komunikacji UDP w zakresie portów 500 – 1000. Test można wykonać w oparciu o polecenie nmap.

Wykorzystane komendy:



```
(kali㉿kali)-[~]
$ sudo iptables -A INPUT -p udp --dport 500:1000 -j LOG --log-prefix "UDP BLOCK: "
sudo iptables -A INPUT -p udp --dport 500:1000 -j DROP
```

Rysunek 15: Wykorzystane komendy dla portów 500-1000.

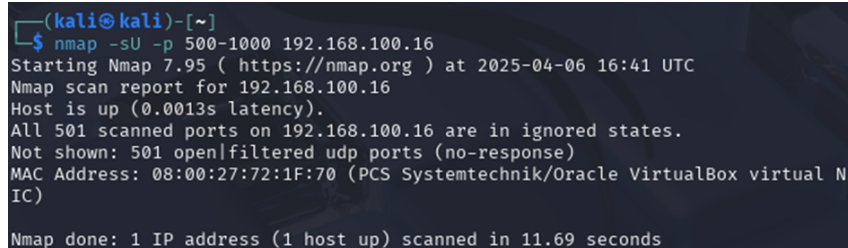
```
sudo iptables -A INPUT -p udp --dport 500:1000 -j LOG --log-prefix "UDP BLOCK: "
```

```
sudo iptables -A INPUT -p udp --dport 500:1000 -j DROP
```

- -p udp: reguła dotyczy UDP.
- --dport 500:1000: zakres portów docelowych.
- -j LOG: loguj próbę.
- -j DROP: odrzuć pakiety.

**Komenda wykorzystana na maszynie kali live w celu testów:**

`nmap -sU -p 500-1000 192.168.100.16`

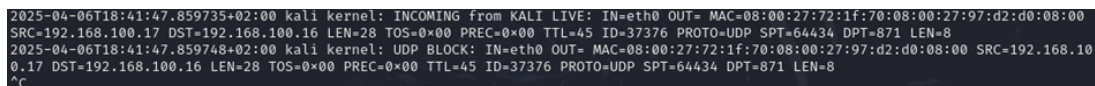


```
(kali㉿kali)-[~]
$ nmap -sU -p 500-1000 192.168.100.16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-06 16:41 UTC
Nmap scan report for 192.168.100.16
Host is up (0.0013s latency).
All 501 scanned ports on 192.168.100.16 are in ignored states.
Not shown: 501 open|filtered udp ports (no-response)
MAC Address: 08:00:27:72:1F:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 11.69 seconds
```

Rysunek 16: Test z Kali Live.

Logi z maszyny kali linux:



```
2025-04-06T18:41:47.859735+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00
SRC=192.168.100.17 DST=192.168.100.16 LEN=28 TOS=0x00 PREC=0x00 TTL=45 ID=37376 PROTO=UDP SPT=64434 DPT=871 LEN=8
2025-04-06T18:41:47.859748+02:00 kali kernel: UDP BLOCK: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.10
0.17 DST=192.168.100.16 LEN=28 TOS=0x00 PREC=0x00 TTL=45 ID=37376 PROTO=UDP SPT=64434 DPT=871 LEN=8
^C
```

Rysunek 17: Zarejestrowane logi na Kali Linux.

Skan UDP został zablokowany. Logi wskazują próby dostępu do zablokowanego zakresu portów.

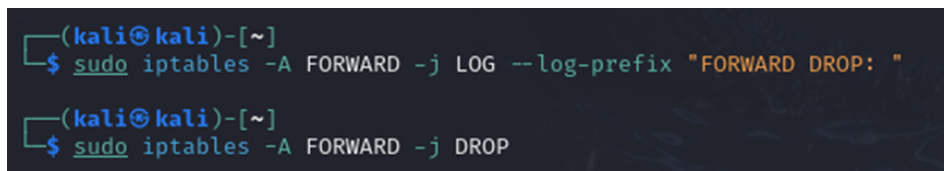
#### 2.3.4 Zadanie 4:

Blokada przekazywania pakietów.\*

**Wykorzystane komendy:**

`sudo iptables -A FORWARD -j LOG --log-prefix "FORWARD DROP: "`

`sudo iptables -A FORWARD -j DROP`



```
(kali㉿kali)-[~]
$ sudo iptables -A FORWARD -j LOG --log-prefix "FORWARD DROP: "

(kali㉿kali)-[~]
$ sudo iptables -A FORWARD -j DROP
```

Rysunek 18: Użyte komendy iptables FORWARD.

- -A FORWARD: dotyczy pakietów przekazywanych przez host.
- -j LOG: loguj przekazywane pakiety.
- -j DROP: odrzuć je.

Nie wykonano testu, ponieważ środowisko nie zakładało routingu między sieciami. Reguła poprawnie zapobiega przekazywaniu pakietów.

### 2.3.5 Zadanie 5:

Blokada połączeń ICMP typu 0 i 8 (nie wszystkich) w oparciu o dyrektywę DROP. Test można wykonać w oparciu o polecenie ping.

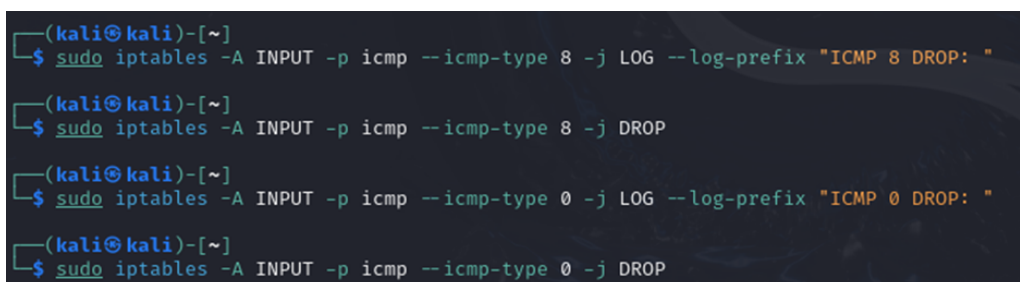
#### Wykorzystane komendy:

```
sudo iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "ICMP 8 DROP: "
```

```
sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
```

```
sudo iptables -A INPUT -p icmp --icmp-type 0 -j LOG --log-prefix "ICMP 0 DROP: "
```

```
sudo iptables -A INPUT -p icmp --icmp-type 0 -j DROP
```

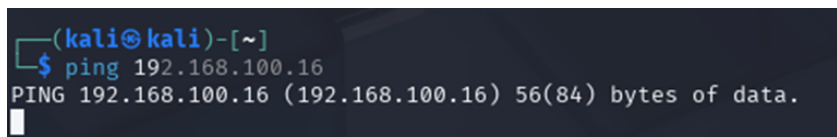


```
(kali@kali)-[~]  
$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "ICMP 8 DROP: "  
  
(kali@kali)-[~]  
$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP  
  
(kali@kali)-[~]  
$ sudo iptables -A INPUT -p icmp --icmp-type 0 -j LOG --log-prefix "ICMP 0 DROP: "  
  
(kali@kali)-[~]  
$ sudo iptables -A INPUT -p icmp --icmp-type 0 -j DROP
```

Rysunek 19: Użyte komendy iptables INPUT.

- --icmp-type 8: żądanie echo (ping).
- --icmp-type 0: odpowiedź echo.
- -j LOG i -j DROP: loguj i odrzuć pakiety.

#### Komenda wykorzystana na maszynie kali live w celu testów:



```
(kali@kali)-[~]  
$ ping 192.168.100.16  
PING 192.168.100.16 (192.168.100.16) 56(84) bytes of data.  
_
```

Rysunek 20: Test z Kali Live.

Logi z maszyny kali linux:

```
2025-04-06T18:45:26.066891+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00
SRC=192.168.100.17 DST=192.168.100.16 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=6546 DF PROTO=ICMP TYPE=8 CODE=0 ID=2 SEQ=25
2025-04-06T18:45:26.066917+02:00 kali kernel: ICMP 8 DROP: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.
100.17 DST=192.168.100.16 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=6546 DF PROTO=ICMP TYPE=8 CODE=0 ID=2 SEQ=25
^C
```

Rysunek 21: Zarejestrowane logi na Kali Linux.

Ping nie powiódł się — pakiety ICMP zostały odrzucone, co widać w logach.

### 2.3.6 Zadanie 6:

Blokada połączeń ICMP typu 0 i 8 (nie wszystkich) w oparciu o dyrektywę REJECT. Test można wykonać w oparciu o polecenie ping.

#### Wykorzystane komendy:

```
sudo iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "ICMP 8 REJECT: "
sudo iptables -A INPUT -p icmp --icmp-type 8 -j REJECT
sudo iptables -A INPUT -p icmp --icmp-type 0 -j LOG --log-prefix "ICMP 0 REJECT: "
sudo iptables -A INPUT -p icmp --icmp-type 0 -j REJECT
```

```
(kali@kali)-[~]
$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j LOG --log-prefix "ICMP 8 REJECT: "
sudo iptables -A INPUT -p icmp --icmp-type 8 -j REJECT
sudo iptables -A INPUT -p icmp --icmp-type 0 -j LOG --log-prefix "ICMP 0 REJECT: "
sudo iptables -A INPUT -p icmp --icmp-type 0 -j REJECT
```

Rysunek 22: Użyte komendy na Kali Linux REJECT.

Zamiast DROP, użyto REJECT, który aktywnie odrzuca pakiety, zwracając odpowiedź ICMP z informacją o niedostępności.

#### Komenda wykorzystana na maszynie kali live w celu testów:

ping 192.168.100.16

```
(kali@kali)-[~]
$ ping 192.168.100.16
PING 192.168.100.16 (192.168.100.16) 56(84) bytes of data.
From 192.168.100.16 icmp_seq=1 Destination Port Unreachable
From 192.168.100.16 icmp_seq=2 Destination Port Unreachable
^C
— 192.168.100.16 ping statistics —
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1002ms
```

Rysunek 23: Ping z Kali Live na Kali Linux.

Logi z maszyny kali linux:

```
2025-04-06T18:47:52.286149+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=8947 DF PROTO=ICMP TYPE=8 CODE=0 ID=3 SEQ=2
2025-04-06T18:47:52.286527+02:00 kali kernel: ICMP 8 REJECT: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=8947 DF PROTO=ICMP TYPE=8 CODE=0 ID=3 SEQ=2
2025-04-06T18:47:52.286548+02:00 kali kernel: OUTGOING to KALI LIVE: IN= OUT=eth0 SRC=192.168.100.16 DST=192.168.100.17 LEN=112 TOS=0x00 PREC=0x00 TTL=64 ID=17461 PROTO=ICMP TYPE=3 CODE=3 [SRC=192.168.100.17 DST=192.168.100.16 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=8947 DF PROTO=ICMP TYPE=8 CODE=0 ID=3 SEQ=2 ]
```

Rysunek 24: Zarejestrowane logi REJECT na Kali Linux.

Reguła skutecznie wykryła i zablokowała pakiety typu XMAS, co zostało potwierdzone logami.

### 2.3.7 Zadanie 7:

Blokada połączeń typu Christmas Tree (Pakiety TCP z flagami URG, PSH, FIN). Test można wykonać w oparciu o polecenie nmap i skan typu „sX”

#### Wykorzystane komendy:

```
sudo iptables -A INPUT -p tcp --tcp-flags ALL URG,PSH,FIN -j LOG --log-prefix "XMAS BLOCK: "
```

```
sudo iptables -A INPUT -p tcp --tcp-flags ALL URG,PSH,FIN -j DROP
```

```
(kali㉿kali)-[~]
$ sudo iptables -A INPUT -p tcp --tcp-flags ALL URG,PSH,FIN -j LOG --log-prefix "XMAS BLOCK: "
$ sudo iptables -A INPUT -p tcp --tcp-flags ALL URG,PSH,FIN -j DROP
```

Rysunek 25: Komendy INPUT na Kali Linux.

- `--tcp-flags ALL URG,PSH,FIN`: filtruje pakiety z ustawionymi wszystkimi trzema flagami — charakterystyczne dla ataków.
- `-j LOG`, `-j DROP`: logowanie i odrzucenie.

#### Komenda wykorzystana na maszynie kali live w celu testów:

```
nmap -sX 192.168.100.16
```

```
(kali㉿kali)-[~]
$ nmap -sX 192.168.100.16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-06 16:49 UTC
```

Rysunek 26: Test nmap z Kali Live.

Logi z maszyny kali linux:

```
2025-04-06T18:49:39.369861+02:00 kali kernel: INCOMING from KALI LIVE: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=40 TOS=0x00 PREC=0x00 TTL=43 ID=9303 PROTO=TCP SPT=56108 DPT=1864 WINDOW=1024 RES=0x00 URG PSH FIN URG=0
2025-04-06T18:49:39.369874+02:00 kali kernel: XMAS BLOCK: IN=eth0 OUT= MAC=08:00:27:72:1f:70:08:00:27:97:d2:d0:08:00 SRC=192.168.100.17 DST=192.168.100.16 LEN=40 TOS=0x00 PREC=0x00 TTL=43 ID=9303 PROTO=TCP SPT=56108 DPT=1864 WINDOW=1024 RES=0x00 URG PSH FIN URG=0
```

Rysunek 27: Zarejestrowane logi dla nmap na Kali Linux.

Reguła skutecznie wykryła i zablokowała pakiety typu XMAS, co zostało potwierdzone logami.

### 2.3.8 Zadanie 8 (Dla chętnych):

Limit pakietów ICMP echo - max 2 na sekundę. Test można wykonać w oparciu o polecenie ping.

**Wykorzystane komendy:** `sudo iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 2/second -j ACCEPT`

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

```
(kali@kali)~$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 2/second -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Rysunek 28: Zarejestrowane logi dla nmap na Kali Linux.

- `-m limit --limit 2/second`: pozwala na maksymalnie 2 pakiety echo-request (ping) na sekundę.
- `-j ACCEPT`: przepuszcza zgodne pakiety.
- `-j DROP`: odrzuca nadmiarowe pakiety.

**Komenda wykorzystana na maszynie kali live w celu testów:** `ping -i 0.1 192.168.100.16`

```
(kali@kali)~$ ping -i 0.1 192.168.100.16
PING 192.168.100.16 (192.168.100.16) 56(84) bytes of data:
64 bytes from 192.168.100.16: icmp_seq=1 ttl=64 time=1.54 ms
64 bytes from 192.168.100.16: icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from 192.168.100.16: icmp_seq=3 ttl=64 time=2.13 ms
64 bytes from 192.168.100.16: icmp_seq=4 ttl=64 time=1.58 ms
64 bytes from 192.168.100.16: icmp_seq=5 ttl=64 time=2.52 ms
64 bytes from 192.168.100.16: icmp_seq=6 ttl=64 time=2.30 ms
64 bytes from 192.168.100.16: icmp_seq=11 ttl=64 time=1.69 ms
```

Rysunek 29: Ping z Kali Live na Kali Linux.

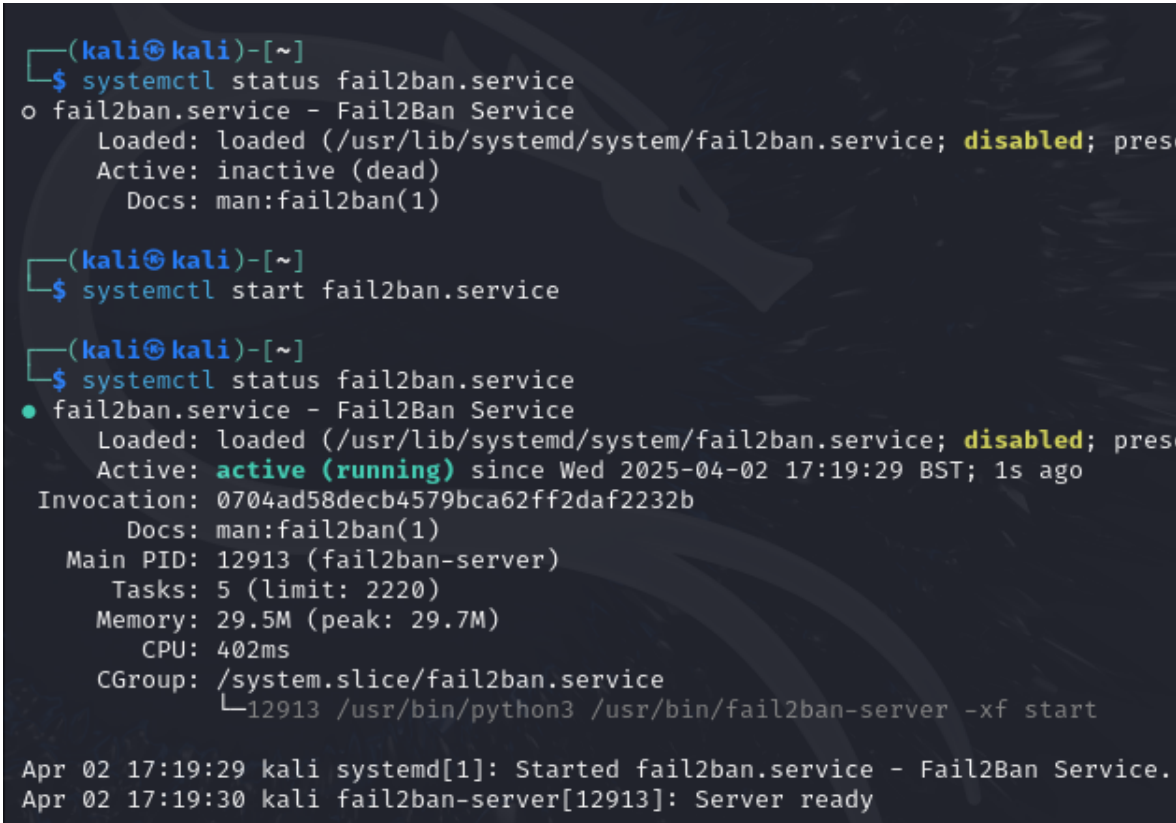
Reguła ograniczyła liczbę odpowiedzi ICMP — nadmiarowe pakiety zostały odrzucone.

## 2.4 Fail2Ban

Fail2ban jest aplikacją IPS służącą do obrony przed atakami typu brute force. Zainstalowano ją na systemie operacyjnym komendami:

- 1 `sudo apt-get update`
- 2 `sudo apt-get install fail2ban`

Usługa automatycznie zacznie nasłuchiwanie na porcie SSH, ale domyślnie jest wyłączona, dlatego należy ją ręcznie włączyć.



```
(kali㉿kali)-[~]
$ systemctl status fail2ban.service
o fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled; pres
   Active: inactive (dead)
   Docs: man:fail2ban(1)

(kali㉿kali)-[~]
$ systemctl start fail2ban.service

(kali㉿kali)-[~]
$ systemctl status fail2ban.service
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled; pres
   Active: active (running) since Wed 2025-04-02 17:19:29 BST; 1s ago
   Invocation: 0704ad58decb4579bca62ff2daf2232b
   Docs: man:fail2ban(1)
   Main PID: 12913 (fail2ban-server)
   Tasks: 5 (limit: 2220)
   Memory: 29.5M (peak: 29.7M)
   CPU: 402ms
   CGroup: /system.slice/fail2ban.service
           └─12913 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Apr 02 17:19:29 kali systemd[1]: Started fail2ban.service - Fail2Ban Service.
Apr 02 17:19:30 kali fail2ban-server[12913]: Server ready
```

Rysunek 30: Włączenie usługi Fail2ban.

Usługa ta domyślnie blokuje port SSH (domyślnie 22) w momencie wykrycia ataku brute force, więc w celu przetestowania tego, z komputera Kali Live (192.168.100.32) wysłano pojedynczą próbę połączenia SSH do Kali Linux (192.168.100.33).



```
(kali㉿kali)-[~]  
$ ssh kali@192.168.100.33  
The authenticity of host '192.168.100.33 (192.168.100.33)' can't be  
established.  
ED25519 key fingerprint is SHA256:TaZF1WjzQDggn1HY2VH  
.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no)?  
Warning: Permanently added '192.168.100.33' (ED25519)  
key to the list of known hosts.  
kali@192.168.100.33's password:  
Linux kali 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali Linux (6.11.2-  
kali) x86_64  
  
The programs included with the Kali GNU/Linux system  
are free software; the exact distribution terms for each program are  
described in individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Mar 13 11:02:33 2025 from 10.0.2.15  
zsh: corrupt history file /home/kali/.zsh_history  
(kali㉿kali)-[~]  
$ exit
```

Rysunek 31: Próba SSH z Kali Live do Kali Linux.

Widać, iż próba jest zakończona sukcesem, ponieważ była pojedyncza i z poprawnym hasłem. Teraz wykonano z Kali Live atak brute force do SSH za pomocą narzędzia nmap.

```
(kali㉿kali)-[~]  
$ exit  
Connection to 192.168.100.33 closed.  
  
(kali㉿kali)-[~]  
$ nmap -p 22 --script ssh-brute -T4 192.168.100.33  
  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-03 09:12 UTC  
NSE: [ssh-brute] Trying username/password pair: root:root  
NSE: [ssh-brute] Trying username/password pair: admin:admin  
NSE: [ssh-brute] Trying username/password pair: administrator:administrator  
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin  
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin  
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin  
NSE: [ssh-brute] Trying username/password pair: guest:guest  
■
```

Rysunek 32: Bruteforce SSH nmap z Kali Live do Kali Linux.

Narzędzie stara się zgadnąć hasło do maszyny o adresie 192.168.100.33, ale po siedmiu próbach komunikacja zostaje zablokowana.



```
(kali㉿kali)-[~]
$ sudo fail2ban-client status sshd
Status for the jail: sshd
├─ Filter
│   ├─ Currently failed: 1
│   ├─ Total failed:    9
│   └─ Journal matches: _SYSTEMD_UNIT=ssh.service + _COMM=sshd
└─ Actions
    ├─ Currently banned: 1
    ├─ Total banned:    1
    └─ Banned IP list:   192.168.100.32

(kali㉿kali)-[~]
```

Rysunek 33: Zablokowany adres Kali Live na Kali Linux.

Wykonując konkretną komendę widać, że narzędzie Fail2ban odpowiada za zablokowanie ruchu SSH z atakującej maszyny.

```
(kali㉿kali)-[~]
$ sudo fail2ban-client banned
[{'sshd': ['192.168.100.32']}]
```

Rysunek 34: Zablokowany klient Kali Live na Kali Linux.

Kolejna komenda pokazuje, że konkretny adres został przez Fail2ban zablokowany.

```
(kali㉿kali)-[~]
$ ssh kali@192.168.100.33
ssh: connect to host 192.168.100.33 port 22: Connection refused

(kali㉿kali)-[~]
```

Rysunek 35: Brak połączenia SSH z Kali Live na Kali Linux.

Można to dodatkowo potwierdzić próbując ustanowić normalne połączenie SSH z Kali Live na Kali Linux, które poprzednio było akceptowane i zakończyło się sukcesem. Teraz, jak widać na Rysunku powyżej, zostało ono zablokowane i nie jest dozwolony żaden ruch SSH przez określony czas.