Jak stworzyć UFW (Uncomplicated Firewall) na systemie Linux?

W tym materiale skupimy się na tym jak stworzyć i skonfigurować Firewalla na systemie Linux

Czym jest UFW?

UFW (Uncomplicated Firewall) to prosty w użyciu interfejs do zarządzania zaporą sieciową w systemach operacyjnych opartych na Linuxie, zwłaszcza w dystrybucjach takich jak Ubuntu. UFW ma na celu uproszczenie procesu konfigurowania reguł zapory, co czyni go bardziej przystępnym dla użytkowników.

Kluczowe cechy UFW

Łatwość obsługi – UFW umożliwia konfigurowanie reguł firewall za pomocą prostych komend, bez konieczności zagłębiania się w skomplikowane struktury iptables.

Nakładka na iptables – oferuje uproszczony interfejs dla iptables, dzięki czemu można zarządzać ruchem sieciowym, zachowując pełną moc tego narzędzia, ale w bardziej dostępnej formie.

Domyślne ustawienia bezpieczeństwa – domyślnie blokuje ruch przychodzący i pozwala na ruch wychodzący, co zapewnia podstawowy poziom ochrony od momentu instalacji.

Wsparcie dla IPv4 i IPv6 – kompatybilne z nowoczesnymi standardami sieciowymi.

Profile aplikacji – umożliwia szybkie przypisywanie reguł do popularnych usług (np. SSH, HTTP), co przyspiesza konfigurację firewalla w zależności od potrzeb projektu.

Funkcje logowania – obsługuje logowanie ruchu sieciowego, co pozwala na monitorowanie aktywności i szybką reakcję na incydenty bezpieczeństwa.

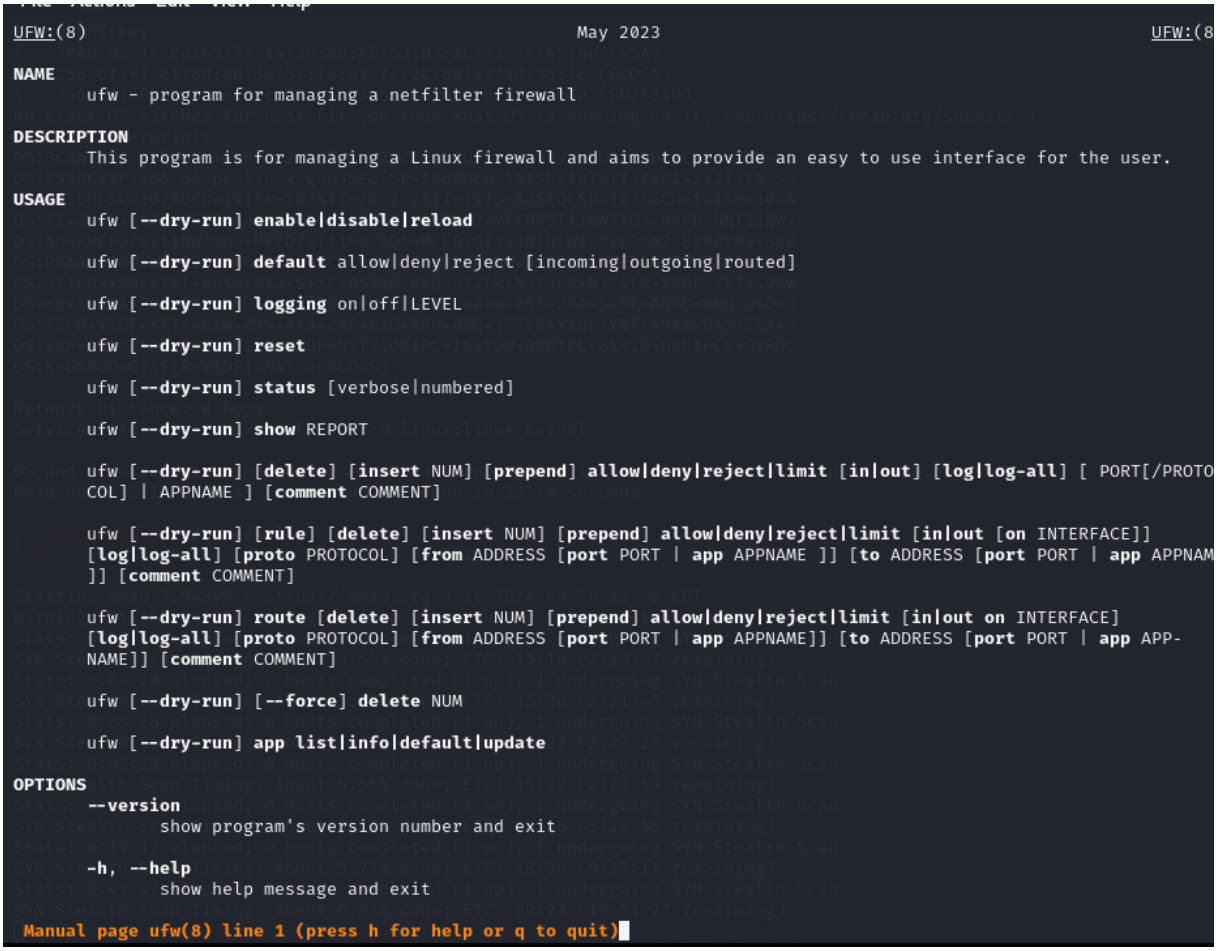
Instalacja UFW

Używamy komendy: *sudo apt install ufw*

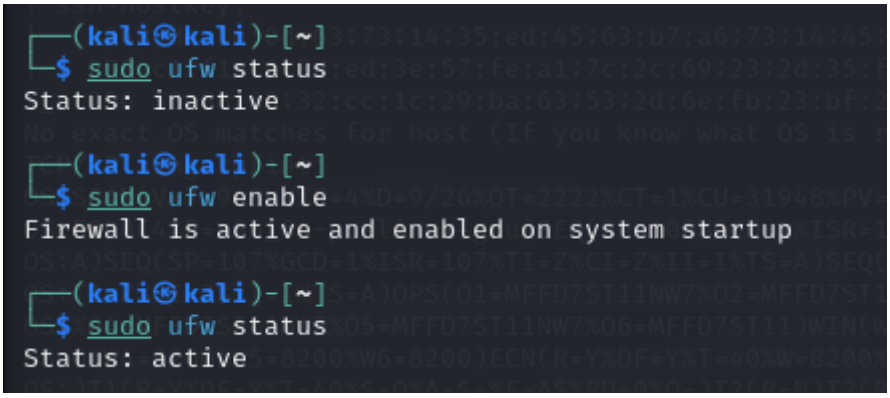
```
┌──(kali㉿kali)-[~]
└─$ sudo apt install ufw
Installing:
  ufw

Suggested packages:
  rsyslog

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 905
  Download size: 0 B / 168 kB
  Space needed: 880 kB / 64.6 GB available

Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 396606 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-6_all.deb ...
Unpacking ufw (0.36.2-6) ...
Setting up ufw (0.36.2-6) ...
Processing triggers for kali-menu (2024.3.1) ...
Processing triggers for man-db (2.12.1-2) ...
```

Po pobraniu uruchamiamy komende „*man ufw*” która bedzię zawierać wszystkie szczegółowe informacji na temat ufw, takie jak opis, nazwa, metodyka używania, opcje itd.

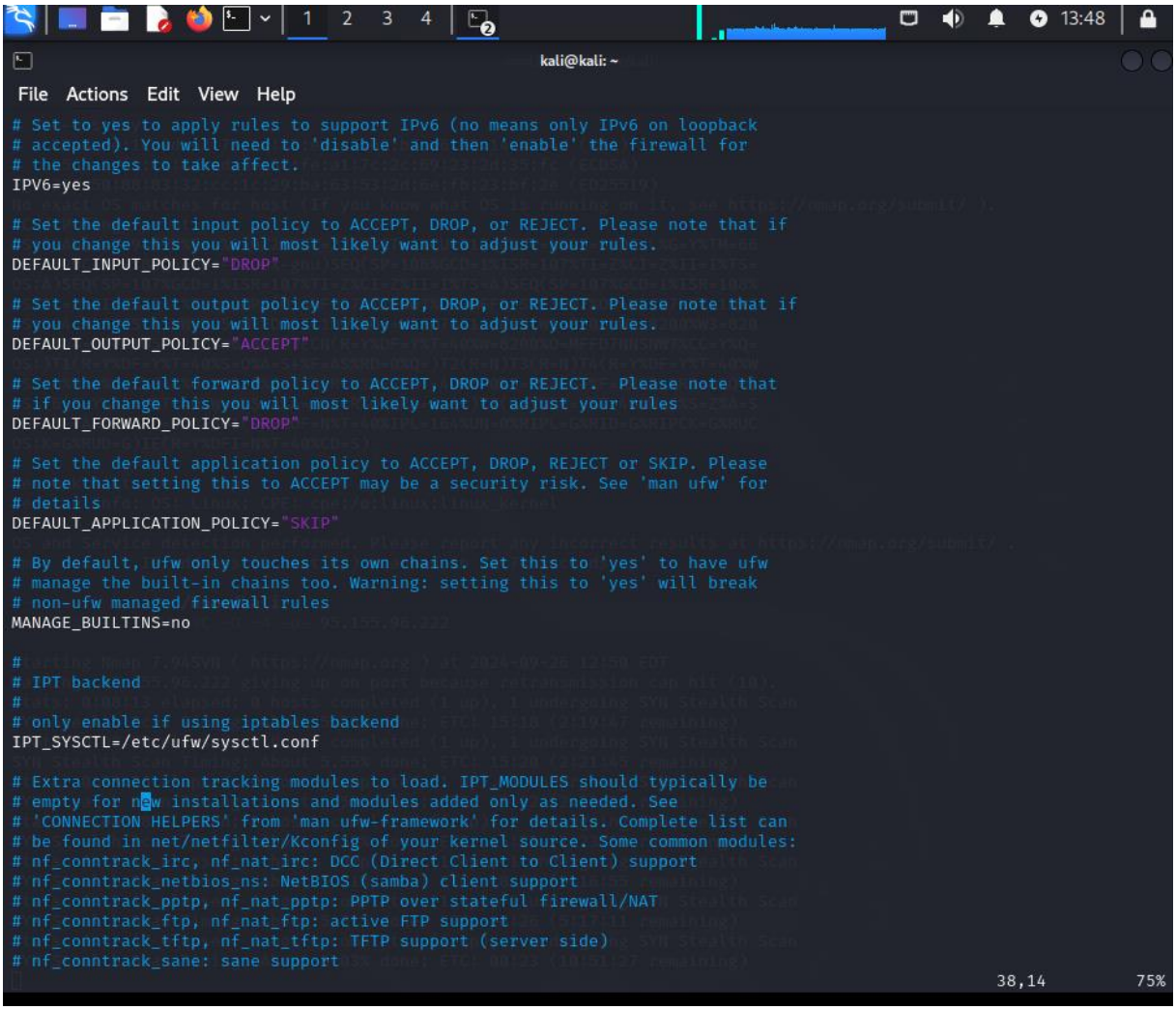
Man ufw



Pierwszą rzeczą którą chcemy zrobić przed ustawianiem reguł jest sprawdzenie czy ufw jest poprawnie zainstalowany i aktywny w tym celu żywamy poniższych komend



Następnie otwieramy plik konfiguracyjny za pomocą komendy: *vim/etc/default/ufw*



Terminologia:

Zawartość pliku /etc/default/ufw

1. **DEFAULT_INPUT_POLICY**:

Opis: Domyślna polityka dla ruchu przychodzącego.

Wartości:

ACCEPT: Pozwala na cały ruch przychodzący.

DROP: Blokuje cały ruch przychodzący, chyba że istnieją reguły pozwalające.

REJECT: Odrzuca ruch przychodzący i informuje nadawcę.

Domyślna wartość: DROP

2. **DEFAULT_OUTPUT_POLICY**:

Opis: Domyślna polityka dla ruchu wychodzącego.

Wartości:

ACCEPT: Pozwala na cały ruch wychodzący.

DROP: Blokuje cały ruch wychodzący.

REJECT: Odrzuca ruch wychodzący.

Domyślna wartość: ACCEPT

3. **DEFAULT_FORWARD_POLICY**:

Opis: Domyślna polityka dla ruchu przekazywanego (np. w przypadku routingu).

Wartości:

ACCEPT: Pozwala na cały ruch przekazywany.

DROP: Blokuje cały ruch przekazywany.

REJECT: Odrzuca ruch przekazywany.

Domyślna wartość: DROP

4. **IPV6**:

Opis: Wsparcie dla IPv6 w UFW.

Wartości:

yes: Włącza wsparcie dla IPv6.

no: Wyłącza wsparcie dla IPv6.

Domyślna wartość: yes (w nowszych wersjach)

5. **ENABLED**:

Opis: Włączenie UFW przy starcie systemu.

Wartości:

yes: UFW włączony przy starcie.

no: UFW wyłączony przy starcie.

Domyślna wartość: yes

Działanie domyślnych ustawień

Polityka przychodząca (DEFAULT_INPUT_POLICY): Z ustawieniem DROP, ruch przychodzący jest domyślnie blokowany, co zwiększa bezpieczeństwo, ale wymaga definiowania reguł pozwalających na niektóre połączenia, jak SSH (port 22).

Polityka wychodząca (DEFAULT_OUTPUT_POLICY): Z ustawieniem ACCEPT, wszystkie połączenia wychodzące są dozwolone, co jest typowe dla większości zastosowań, ponieważ pozwala na normalne funkcjonowanie aplikacji.

Polityka przekazywania (DEFAULT_FORWARD_POLICY): Z ustawieniem DROP, ruch przekazywany przez system (np. w przypadku użycia jako router) jest blokowany, co chroni przed nieautoryzowanym dostępem.

Wsparcie dla IPv6: Umożliwienie wsparcia dla IPv6 jest ważne, zwłaszcza w nowoczesnych sieciach.

Włączenie UFW przy starcie: Umożliwia automatyczne włączenie firewalla przy każdym uruchomieniu systemu, co zapewnia stałą ochronę.

Ważne uwagi

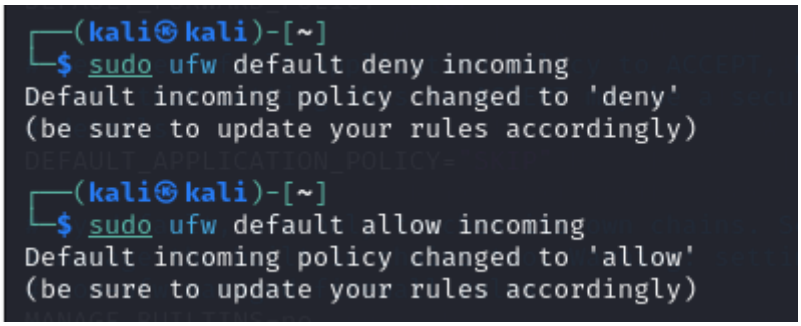
Edytowanie pliku: Zmiany wprowadzone w tym pliku będą miały wpływ na zachowanie UFW, więc należy je dokonywać ostrożnie.

Zastosowanie zmian: Po edytowaniu pliku, upewnij się, że zastosujesz zmiany, uruchamiając:

Użyj komendy: *sudo ufw reload*

Tworzenie zestawu reguł - poradnik

Przykłady:



Podczas konfiguracji twojego zestawu reguł ważne jest żebyś wiedział jakie reguły albo jakie usługi chcesz zachować aktywne w odniesieniu do ruchu wychodzącego i przychodzącego oraz jakie usługi lub protokoły chcesz aktualnie wyłączyć lub zabezpieczyć przed dostępem. Najważniejsze dwa protokoły które chcemy utrzymać aktywne to http na porcie 80 i 443 (jego zaszyfrowana wersja)oraz port 22 dla ssh

Oczywiście tworząc swój własny zestaw reguł możemy zezwolić dowolnemu protokołowi na dostęp.

Jeśli chce zezwolic konkretnemu adresowi IP na dostęp na każdym porcie używamy

Sudo ufw allow from adress ip

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 10.0.0.0
Rule added
```

Jeśli chcemy zezwolić konkretnemu adresowi na konkretnym porcie uzywamy

Sudo ufw allow from 10.0.0.0 to any port 22

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 10.0.0.0 to any port 22
Rule added
```

Podczas tworzenia zestawu reguł dla danego adresu kluczowe jest dodanie parametru /24, ponieważ definiuje on zakres adresów IP objętych tą regułą. W przypadku braku /24, zezwalasz na połączenie jedynie konkretnemu urządzeniu w sieci lokalnej, na przykład komputerowi stacjonarnemu, co sprawia, że laptop z innym adresem IP nie będzie miał dostępu. Dlatego ważne jest, aby zawsze dodawać parametr /24 do komendy.

Dodatkowo, stosując ten parametr, rozwiązujesz problem związany z dynamiczną adresacją IP. W przypadku urządzeń, których adresy IP mogą się zmieniać, dodanie /24 zapewnia, że po ponownym uruchomieniu i zmianie adresu IP maszyna wciąż będzie mogła się połączyć z siecią.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 10.0.0.0/24 to any port 22
Rule added
```

Po stworzeniu pierwszego zestawu reguł możemy go sprawdzić jak wygląda, używamy do tego komendy: *sudo ufw status numbered*

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 443                        ALLOW IN    Anywhere
[ 4] Anywhere                   ALLOW IN    10.0.0.0
[ 5] 22                         ALLOW IN    10.0.0.0
[ 6] 22                         ALLOW IN    10.0.0.0/24
[ 7] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 9] 443 (v6)                   ALLOW IN    Anywhere (v6)
```

Przykład usuwania reguł: *sudo ufw delete 5*

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw delete 5
Deleting:
 allow from 10.0.0.0 to any port 22
Proceed with operation (y|n)? y
Rule deleted
```

Reguła zostałą usunięta ponieważ jej nie potrzebujemy, zadeklarowaliśmy ją w regule 6.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 443                        ALLOW IN    Anywhere
[ 4] Anywhere                   ALLOW IN    10.0.0.0
[ 5] 22                         ALLOW IN    10.0.0.0/24
[ 6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 7] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] 443 (v6)                   ALLOW IN    Anywhere (v6)
```

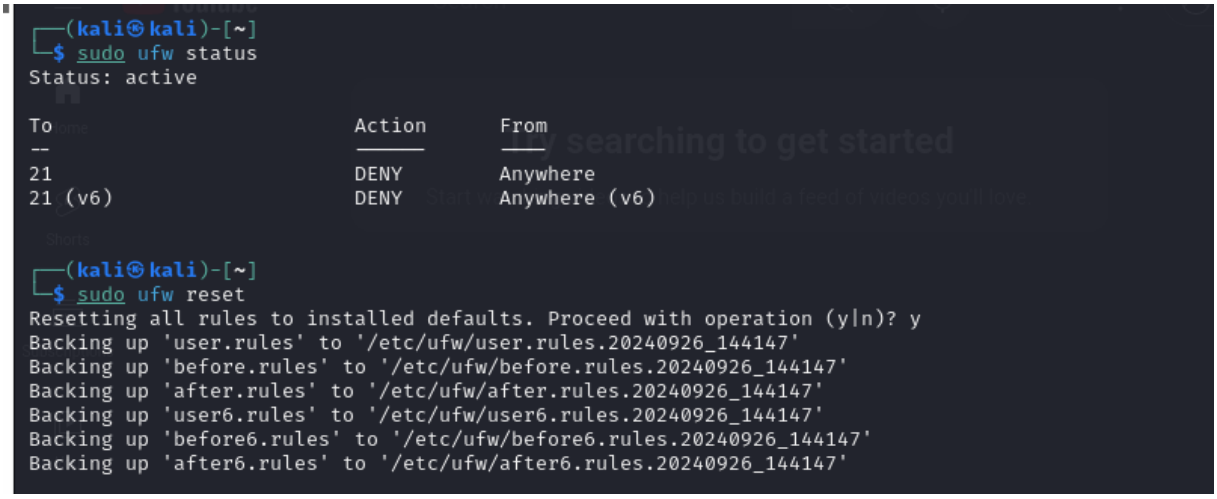
Jeśli chcemy zablokować ruch na danym porcie używamy komendy

Sudo ufw deny ftp

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw deny 21
Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
21                         DENY        Anywhere
21 (v6)                    DENY        Anywhere (v6)
```

Jeśli popełnimy błąd i chcemy np zrestartować firewalla używamy komendy

Sudo ufw reset



Po resecie firewall będzie niekatywny i bedzię trzeba go uruchomić ponownie i ponownie skonfigurować.

Najlepsze praktyki w konfiguracji UFW

Domyślna polityka zapory:

Ustaw domyślną politykę na **deny incoming** (blokuj ruch przychodzący) oraz **allow outgoing** (zezwól na ruch wychodzący). To zapewni, że żadne nieautoryzowane połączenia nie będą mogły nawiązać kontaktu z Twoim systemem.

Zezwalaj tylko na niezbędne porty:

Otwieraj tylko te porty, które są niezbędne do działania Twoich aplikacji. Przykładowo, jeśli korzystasz z SSH, otwórz tylko port 22; dla serwera WWW otwórz porty 80 i 443.

Zezwalaj na ruch z zaufanych adresów IP:

Jeśli to możliwe, ogranicz dostęp do usług (np. SSH) tylko do zaufanych adresów IP. To znacznie zmniejsza ryzyko ataków brute-force

Używaj reguł dla aplikacji:

Używaj profili aplikacji, jeśli są dostępne, aby szybko przypisać reguły do popularnych usług. Na przykład:

Monitoruj logi:

Włącz logowanie, aby monitorować ruch i identyfikować potencjalne zagrożenia. Logi mogą pomóc w analizie ataków i nieautoryzowanych prób dostępu.

Regularnie przeglądaj i aktualizuj reguły:

Regularnie sprawdzaj aktywne reguły i aktualizuj je w miarę zmieniających się potrzeb. Użyj polecenia:

Testuj po każdej zmianie:

Po każdej modyfikacji reguł przetestuj ich działanie, aby upewnić się, że wszystko działa zgodnie z oczekiwaniami. Możesz użyć narzędzi takich jak **nmap** do skanowania portów.

Zabezpiecz usługi sieciowe:

Upewnij się, że wszystkie usługi działające na otwartych portach są odpowiednio skonfigurowane i zabezpieczone. Na przykład, używaj silnych haseł dla SSH i rozważ użycie kluczy publiczny

Zarządzaj regułami z numerami:

Przy usuwaniu reguł korzystaj z numerów, aby uniknąć pomyłek. Możesz sprawdzić numery reguł poleceniem

Zabezpiecz dostęp do interfejsu zarządzającego:

Jeśli używasz interfejsu graficznego do zarządzania zaporą, upewnij się, że jest on zabezpieczony, a dostęp do niego mają tylko zaufani użytkownicy.

Zainstaluj i skonfiguruj dodatkowe narzędzia zabezpieczające:

Rozważ użycie dodatkowych narzędzi, takich jak Fail2Ban, które mogą automatycznie blokować adresy IP, które wykazują podejrzane zachowanie, takie jak wielokrotne nieudane próby logowania.

Dokumentuj zmiany:

Prowadź dokumentację zmian w regułach zapory, aby mieć pełną kontrolę nad konfiguracją i móc łatwo przywrócić wcześniejsze ustawienia w razie potrzeby.

Konfigurujemy własny „rule set” dla Firewalla

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status
[sudo] password for kali:
\\Status: inactive

┌──(kali㉿kali)-[~]
└─$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.168.1.1/24 to any port 22
WARN: Rule changed after normalization
Rules updated

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw limit ssh
Rules updated
Rules updated (v6)
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw limit ssh
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.1.1.1 to any port 25 tcp
ERROR: Wrong number of arguments

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.1.1.1 to any port 25
Rules updated

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.1.1.1 to any port 587
Rules updated

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.1.1.1 to any port 110
Rules updated

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.1.1.1 to any port 993
Rules updated

┌──(kali㉿kali)-[~]
└─$ sudo ufw logging on
Logging enabled
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw logging on
Logging enabled

┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
suFirewall is active and enabled on system startup

┌──(kali㉿kali)-[~]
└─$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       192.168.1.0/24
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
22/tcp                     LIMIT       Anywhere
25                         ALLOW       192.1.1.1
587                        ALLOW       192.1.1.1
110                        ALLOW       192.1.1.1
993                        ALLOW       192.1.1.1
80/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
22/tcp (v6)                LIMIT       Anywhere (v6)


┌──(kali㉿kali)-[~]
└─$ sudo apt install fail2ban
Installing:
```

```
└─$ nmap -A  -p- localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-26 16:14 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000053s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65534 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
| ssh-hostkey:
|   2048 de:1c:0d:63:73:14:35:ed:45:63:b7:a6:73:14:45:4d (RSA)
|   256 c1:f1:c1:8d:ed:3e:57:fe:a1:7c:2c:69:23:2d:35:fc (ECDSA)
|_  256 50:88:83:32:cc:1c:29:ba:63:53:2d:6e:fb:23:bf:2e (ED25519)
```

```
┌──(root㉿kali)-[/home/kali]
└─# sudo ufw status verbose

Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (route
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    192.168.1.0/24
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22/tcp                     LIMIT IN    Anywhere
25                         ALLOW IN    192.1.1.1
587                        ALLOW IN    192.1.1.1
110                        ALLOW IN    192.1.1.1
993                        ALLOW IN    192.1.1.1
2222                       DENY IN     Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
22/tcp (v6)                LIMIT IN    Anywhere (v6)
2222 (v6)                  DENY IN     Anywhere (v6)
```

Krótki opis mojej konfiguracji.

Zasady zezwalające i blokujące: Umożliwiam dostęp do portó 22(SSH), 80(HTTP), 443(HTTPS) oraz kilku portów dla określonego adresu IP. Dzięki temu ograniczam ryzyko ataków na niepotrzebne usługi.

Limitowanie połączeń SSH: Używam reguły LIMIT na porcie 22, co umożliwia ochrone przed atakami brute-force, ograniczając liczbę połączeń z danego adresu IP.

Zamknięcie portu 2222: Dodałem zasadę deny 2222 aby wyeliminować dodatkowy wektor ataku, co minimalizuje ryzyko nieautoryzowanego dostępu do SSH.

Szczegółowe adresowanie: Porty 25, 587, 110, 993 są dostępne tylko dla jednego adresu IP(pocztowego tutaj dajmy przykład) co zwiększa bezpieczeństwo moich usług e-maillowych.

Obsługa IPv6: Moje reguły obejmują zarówno IPv4, jak i IPv6, co jest niezbędne w kotekście nowoczesnych sieci.

Domyślna polityka deny: Blokuje wszystkie nieautoryzowane połączenia.

Monitoring i logowanie: Ustawiłem możliwość monitorowania logów w UFW co jest kluczowe do wykrywania i analizy potencjalnych zagrożeń.

Minimalizacja powierzchni ataku: Skonfigurowane reguły skutecznie redukują liczbę otwartych portów, co jest kluczowe w mojej strategii bezpieczeństwa.

Zgodność z zasadami bezpieczeństwa: Moja konfiguracja odzwierciedla podstawowe najlepsze praktyki w zakresie bezpieczeństwa, co zwiększa ogólną odporność systemu na ataki.

Pamiętaj, że ten zestaw reguł powinien być dostosowany do Twoich konkretnych potrzeb i może wymagać modyfikacji