



Politechnika Śląska

Dokumentacja projektowa

BSKiST 2022/2023

Bezpieczeństwo sieci komputerowych i systemów teleinformatycznych

Kierunek: Informatyka

Członkowie zespołu:
Grzegorz Koperwas

Gliwice, 2022/2023

Spis treści

1	Wprowadzenie	2
1.1	Cel projektu	2
1.2	Zespół projektowy	2
2	Założenia projektowe	3
3	Implementacja	5
4	Badania	6
5	Podsumowanie i wnioski	7
6	Spis literatury	8

1 Wprowadzenie

1.1 Cel projektu

W trakcie realizowania projektu na przedmiot Inżynieria Oprogramowania, została zauważona potrzeba wystawienia na zewnątrz usługi ssh, w celu umożliwienia automatycznego aktualizowania środowiska produkcyjnego przez platformę *Github Actions*.

Jednak, mając wcześniejsze doświadczenia, gdzie nawet usługa ssh skonfigurowana na niestandardowym porcie 2137 oraz wymuszająca użycie kluczy nadal otrzymuje dużo uwagi z całego świata.

Celem projektu jest wdrożenie systemu wykrywania intruzów dla serwera ssh w systemie Arch Linux.

1.2 Zespół projektowy

Imię i nazwisko :: Rola w projekcie :: Zadania projektowe

2 Założenia projektowe

Usługa wdrażana w ramach projektu musiała spełniać następujące wymagania:

- Niski koszt - By usługa mogła się zmieścić w budżecie projektu, jej koszt musiał być zerowy.
- Łatwość zintegrowania z platformą istniejącą - Serwer, na którym rozwiązanie miało być wdrożone, był Dell PowerEdge R720, zainstalowanym systemem Arch Linux¹.

Środowisko funkcjonuje głównie jako host wielu rozwiązań skonteneryzowanych, serwer plików oraz platforma do transkodowania wideo za pomocą NVENC.



Rysunek 1: Środowisko, do którego zostało wdrożone rozwiązanie

¹Do dnia 2 lutego 2023, wystąpił tylko jeden incydent związany z charakterem tej dystrybucji.

- Niskie wymagania systemowe - Mimo dużej mocy obliczeniowej i dużej ilości dostępnej pamięci na serwerze, jako `root` jest używany dysk ssd 60gb podpięty przez złącze **USB 2.0**.

Dodatkowo, na serwerze nie są zainstalowane żadne środowiska graficzne i jest kładziony duży nacisk na jak największe możliwości zarządzania zdalnego.

Biorąc pod uwagę powyższe założenia, został wybrany program `fail2ban`, głównie ze względu na dobrą dokumentację skupiającą się na zabezpieczeniu usługi `ssh` właśnie.

3 Implementacja

Program `fail2ban` działa przez obserwowanie logów systemowych. Jeżeli zostanie przekroczona liczba nieudanych prób logowania z danego adresu ip w danym czasie, to ten adres jest dodawany do zapory `iptables` i połączenia nie będą odbierane z tego adresu.

1. **Instalacja** - Program został zainstalowany z repozytorium komendą `pacman -S fail2ban`.
2. **Konfiguracja** - Program konfigurujemy za pomocą dokumentacji na wiki dystrybucji:

W pliku `/etc/fail2ban/jail.d/sshd.local` konfigurujemy następująco *więzieniem*.

```
1 [sshd]
2 enabled      = true
3 filter       = sshd
4 banaction   = iptables
5 backend     = systemd
6 maxretry    = 5
7 findtime    = 1d
8 bantime    = 2w
9 ignoreip   = 10.217.1.1/24
10
```

Powyższa konfiguracja wykrywa nieudane próby logowania w przeciągu jednego dnia (`findtime`), jeżeli będzie ich więcej niż 5 (`maxretry`), to ip zostanie dodane do zapory kernela i połączenia z tego adresu nie będą odbierane przez usługę `ssh`.

Dodatkowo, lokalne adresy ip z sieci `10.217.1.1/24` są ignorowane.

3. **Uruchamianie** - Za pomocą systemu init `systemd`, sprawiamy by `fail2ban` działał przy starcie komputera:

Uruchamiamy polecenie `systemctl enable --now fail2ban`.

4. **Weryfikacja** - Za pomocą polecenia `fail2ban-client status`, weryfikujemy działanie usługi:

```
1 # fail2ban-client status
2 Status
3 |- Number of jail:  1
4 `-- Jail list: sshd
5
```

Widzimy że nasze *więzieniem* dla usługi `ssh` jest aktywne.

4 Badania

W trakcie trwającego trochę powyżej miesiąca, `fail2ban` zbanował 1705 różnych adresów ip.

5 Podsumowanie i wnioski

- *Podsumowanie* - Skonfigurowanie usługi *fail2ban* pozwoliło znaczaco zmniejszyć wolumen przychodzących prób zalogowania się na server.
- *Wnioski* - Usługi tego typu mogą być używane do ochrony przed atakami typu brute-force

6 Spis literatury