# Plan struktury systemu – model komponentów

# Zastosowane szablony architektoniczne

Architektura jest zrealizowana na bazie MVC. Każdy z dwóch kluczowych modułów, tj. FIRE oraz CONF jest realizowany zgodnie z poniższą tabelą:

|  |  |  |
| --- | --- | --- |
|  | FIRE | CONF |
| Model | Obsługuje i ściąga pakiety z kolejki systemowej i szereguje. | Obsługuje serwer internetowy, przechwytuje komendy użytkownika. |
| View | Interpretuje otrzymane z modelu pakiety, układa je w pełne wiadomości. | Interpretuje komendę na konfigurację, przeszukuje na obecność błędów. |
| Controller | Na podstawie wiadomości, podejmuje decyzję o przesłaniu pakietów dalej lub ich odrzuceniu | Pisze/Czyta zadaną konfigurację do pliku konfiguracyjnego |

# Kluczowe elementy struktury i ich interfejsy

## Moduł FIRE – Python3

Moduł FIRE odpowiada za funkcjonalną część implementowanego firewall’a. Przechwytuje on pakiety z kolejki systemowej za pomocą pakietu Netfilter. Następnie analizuje on pakiety pod kątem reguł zadanych w pliku konfiguracyjnym i podejmuje decyzję o bądź przesłaniu dalej, bądź o opuszczeniu pakietu.

Poniższa tabela przedstawia interfejs modułu FIRE:

|  |  |  |
| --- | --- | --- |
| Metoda | Zwraca | Działanie |
| ReadMessage() | Powodzenie – Strukturę reprezentującą pełną wiadomość, wraz z oryginalnymi pakietami na nią się składającymi Porażka – kod błędu | Czyta pakiety z kolejki systemowej, póki nie poskłada z nich wiadomości. |
| AnalyseMessage(mes) | TRUE -> ACCEPT FALSE -> REJECT | Analizuje wiadomość pod kątem obecnego zbioru reguł. |
| AcceptMessage(mes) | Powodzenie – 0  Porażka – Kod Błędu | Przepuszcza pakiety składające się na wiadomość dalej. |
| RejectMessage(mes) | Powodzenie – 0  Porażka – Kod Błędu | Odrzuca pakiety, wysyłając nadawcy pakiet ICMP. |
| UpdateConfig(dir) | Powodzenie – 0  Porażka – Kod Błędu | Aktualizuje zbiór reguł na podstawie obecnej zawartości pliku konfiguracyjnego ze ścieżki |

## MODUŁ CONF – PYTHON3

Moduł CONF odpowiada za możliwość konfiguracji implementowanego firewall’a. Wystawia on interfejs sieciowy umożliwiający użytkownikowi definicję nowych bądź modyfikację/usunięcie starych reguł. Po przeparsowaniu legalności działań użytkownika, moduł modyfikuje plik konfiguracyjny.

Poniższa tabela przedstawia interfejsy modułu CONF:

|  |  |  |
| --- | --- | --- |
| Metoda | Zwraca | Działanie |
| OPENWEB() | Powodzenie – 0 Porażka – kod błędu | Wystawia interfejs WWW |
| CLOSEWEB() | Powodzenie – 0 Porażka – kod błędu | Zamyka interfejs WWW |
| ANALYSERULE(mes) | Powodzenie – struktura Rule  Porażka – Kod Błędu | Interpretuje komunikat ze strony interfejsu WWW w nową zasadę i analizuje jej legalność. |
| WRITERULE(RULE, DIR) | Powodzenie – 0  Porażka – Kod Błędu | Modyfikuje plik konfiguracyjny pisząc do niego legalną regułę zadaną przez użytkownika |
| READConfig(dir) | Powodzenie – [Rule]  Porażka – Kod Błędu | Czyta z zadanego pliku konfiguracyjnego zbiór reguł i parsuje je do tablicy struktur Rule |

## PLIK KONFIGURACYJNY – FIREWALL.CONF

Plik konfiguracyjny odpowiada za zestaw reguł stosowanych przez moduł FIRE. Działa on w trybie White Listy, a więc zdefiniowane przez owe reguły pakiety są przepuszczane a reszta blokowana. W osobnych wierszach trzymane są definicje reguł w postaci przedstawionej w poniższej tabeli:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RuleID | Name | Protocol | Profile | Direction | Analysed param | Expected Val |
| SHORT | VARCHAR(50) | [MODBUS/SLPM] | SHORT | [IN/OUT/BOTH] | VARCHAR(10) | VARCHAR(50) |

Powyższa reprezentacja może się zmienić podczas implementacji.

# Interakcje pomiędzy elementami

## CONF – FIRE

Celem zastosowanej architektury była jak największa separacja modułu FIRE od CONF, aby w razie niesprawności narażonej zewnętrznie usługi sieciowej zapewnianej przez CONF, nie wyłączyć modułu FIRE odpowiadającego za bezpieczeństwo.

Mając powyższe na względzie, jedyną spodziewaną interakcją, jest wysłanie sygnału PING przez moduł CONF przy zmianie pliku konfiguracyjnego.

## PLIK KONFIGURACYJNY – CONF

Plik konfiguracyjny jest czytany i pisany przez moduł CONF.

## FIRE – PLIK KONFIGURACYJNY

Plik konfiguracyjny jest czytany przez moduł FIRE.

# Wyjaśnienie istoty przyjętych rozwiązań

# Określenie podstawowych mechanizmów technicznych

## Sprzęt

## Systemy operacyjne

## Serwer aplikacyjny

## Inne

## System raportowania

## System analityczny

## Mechanizmy zarządzania

## Mechanizmy bezpieczeństwa