Systemy Agentowe

Laboratorium 2

Celem zadania jest zestawienie rozproszonego środowiska wielo-agentowego oraz wprowadzenie w nim zabezpieczeń na poziomie infrastruktury. Następnie należy przetestować komunikację między agentami rezydującymi na różnych kontenerach.

W podstawowej konfiguracji środowiska, zawsze jeden kontener pełni rolę głównego kontenera a pozostałe kontenery przyłączają się do niego jako sfederowane. Taka konfiguracja tworzy scentralizowaną infrastrukturę o topologii gwiazda, gdzie wszystkie kontenery sfederowane są zależne od głównego. Awaria dowolnego ze sfederowanych kontenerów skutkuje jedynie usunięciem z platformy agentów na nim rezydujących, a cała platforma kontynuuje swoje działanie dopóki sprawny jest główny kontener. Aby temu zapobiec można wprowadzić zapasowe główne kontenery, które wraz z głównym kontenerem będą tworzyć topologię pierścień. W przypadku gdy kontener pełniący aktualnie funkcję głównego ulegnie awarii, pozostałe to wykryją i wybiorą nowy główny kontener, który będzie odpowiedzialny za zarządzanie platformą. Kontenery sfederowane, aby przyłączyć się do platformy, mogą połączyć się z głównym kontenerem lub dowolnym zapasowym.

Domyślnie komunikacja pomiędzy agentami nie jest w żaden sposób zabezpieczona i może być łatwo podsłuchana. Aby zapobiec odczytaniu wiadomości przez elementy spoza środowiska, można prowadzić szyfrowaną komunikację na poziomie kontenerów. W tym celu, każdy z kontenerów można wyposażyć w zbiór kluczy, których będzie używał do szyfrowania komunikacji. Należy zwrócić uwagę, na to że za szyfrowanie nie odpowiada agent (którego kod i zawartość mogłaby być przechwycona podczas migracji przez sieć) a kontener, w którym aktualnie się znajduje.

Zabezpieczenie komunikacji chroni jedynie wymieniane wiadomości. Nadal dowolny kontener może przyłączyć się do platformy. Aby temu zapobiec można wyposażyć w zbiór zaufanych kluczy publicznych. Pozwoli to na identyfikację nowych kontenerów na podstawie ich kluczy prywatnych. W sytuacji gdy nowy kontener przedstawi się niezaufanym kluczem, to nie zostanie on dopuszczony do platformy.

Aktualną wersję środowiska JADE można pobrać z http://jade.tilab.com/. Aktualnie, całe środowisko składa się z jednego pliku (jade.jar), zawierającego wszystkie niezbędne do uruchomienia klasy.

Uruchomienie pojedynczego kontenera realizuje się za pomocą następującego polecenia:

```
java -cp lib/jade.jar jade.Boot
```


Klasa `jade.Boot` jest główną klasą środowiska JADE zawierającą metodę `main()` uruchamiającą pojedynczy kontener. Uruchomienie kontenera może być skonfigurowane następującymi opcjonalnymi argumentami:

- `-container` – stworzenie instancji kontenera sfederowanego,
- `-host` – nazwa hosta, na którym uruchomiony został kontener główny.
- `-local-host` – nazwa hosta, na którym zostanie uruchomiony kontener,
- `-port` – numer portu, na którym został uruchomiony kontener główny,
- `-local-port` – numer portu, na którym zostanie uruchomiony kontener,
- `-gui` – uruchomienie agenta *RMA* (Remote Management Agent) pozwalającego na graficzne zarządzanie platformą,
- `-name` – ustawienie nazwy platformy, wykorzystywane podczas uruchomienia głównego kontenera,
- `-container-name` – ustawienie nazwy uruchamianego kontenera,
- `-nomtp` – wyłączenie domyślnej metody komunikacji pomiędzy kontenerami,
- `-icps` - zdefiniowanie metody komunikacji pomiędzy kontenerami,
- `-services` – uruchomienie dodatkowych usług na platformie,
- `-backupmain` – uruchomienie kontenera jako zapasowy główny kontener.

Skonfigurowanie topologii gwiazda wymaga uruchomienia dwóch dodatkowych usług: usługi replikacji (`MainReplicationService`) na głównym kontenerze i wszystkich zapasowych oraz usługi powiadamiania o zmianie adresu (`AddressNotificationService`) na wszystkich kontenerach w platformie.

Poniżej zamieszczono przykład uruchomienia kontenera głównego, kontenera zapasowego i kontenera sfederowanego.

Kontener główny:

```
java -cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des01.eti.pg.gda.pl -local-port 5656 \
-services \
jade.core.replication.MainReplicationService\;\
jade.core.replication.AddressNotificationService
```

Kontener zapasowy:

```
java -cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des02.eti.pg.gda.pl -local-port 4646 -backupmain \
-services \
jade.core.replication.MainReplicationService\;\
```



```
jade.core.replication.AddressNotificationService
```

Kontener sfederowany:

```
java -cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des03.eti.pg.gda.pl -local-port 4646 -container \
-services \
jade.core.replication.AddressNotificationService
```

Uwaga: w przypadku uruchamiania środowiska w systemie Linux, wywołanie `ctrl+c` na głównym kontenerze jest traktowane jako poprawne zakończenie programu i wyłącza całą platformę. Sugeruje się skorzystanie z polecenia `kill` z parametrem `-9`.

Wykorzystanie szyfrowania pomiędzy kontenerami (bez uwierzytelniania) nie wymaga generowania specjalnych kluczy, a jedynie odpowiedniego uruchomienia platformy. Należy wyłączyć standardową metodę komunikacji i zastąpić ją taką, która wykorzystuje szyfrowanie (`JICPSPeer`). Poniżej przedstawiono przykładowe uruchomienie.

Kontener główny:

```
java -cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des01.eti.pg.gda.pl -local-port 5656 \
-nomtp -icps jade.imtp.leap.JICP.JICPSPeer
```

Kontener sfederowany:

```
java -cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des02.eti.pg.gda.pl -local-port 4646 -container \
-nomtp -icps jade.imtp.leap.JICP.JICPSPeer
```

W przypadku poprawnego uruchomienia w logach głównego kontenera powinna znajdować się taka informacja:

```
INFO: JICP Secure Peer activated. (auth=false,
ta=jicp://des01.eti.pg.gda.pl:5656)
```

Wykorzystanie szyfrowania i uwierzytelniania kontenerów wymaga wykorzystania wygenerowanych kluczy oraz kontenerów kluczy. Każdy z kontenerów platformy musi być wyposażony w zestaw kluczy, które będzie wykorzystywał w procesie szyfrowania wiadomości. Dodatkowo aby zapewnić uwierzytelnianie innych kontenerów, kontener musi posiadać kontener kluczy zaufanych innych kontenerów tworzących platformę.

Poniżej zaprezentowano kilka poleceń wykorzystujących narzędzie keytool wchodzące w skład JRE.

Generowanie pary kluczy i kontenera dla nich:

```
keytool -genkeypair -keystore des01.jks -alias des01
```

Eksportowanie klucza publicznego:

```
keytool -export -keystore des01.jks -alias des01 -file des01.cer
```

Import klucza publicznego do kontenera zaufanych kluczy:

```
keytool -import -file des01.cer -alias des01 -keystore des02-ca.jks
```

Poniżej przedstawiono przykładowe uruchomienie środowiska z szyfrowaniem i uwierzytelnianiem.

Kontener główny:

```
java -Djavax.net.ssl.keyStore=des01.jks \
-Djavax.net.ssl.keyStorePassword=changeit \
-Djavax.net.ssl.trustStore=des01-ca.jks \
-cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des01.eti.pg.gda.pl -local-port 5656 \
-nomtp -icps jade.imtp.leap.JICP.JICPSPeer
```

Kontener sfederowany:

```
java -Djavax.net.ssl.keyStore=des02.jks \
-Djavax.net.ssl.keyStorePassword=changeit \
-Djavax.net.ssl.trustStore=des02-ca.jks \
-cp lib/jade.jar jade.Boot \
-host des01.eti.pg.gda.pl -port 5656 \
-local-host des02.eti.pg.gda.pl -local-port 4646 -container \
-nomtp -icps jade.imtp.leap.JICP.JICPSPeer
```

W przypadku poprawnego uruchomienia w logach głównego kontenera powinna znajdować się taka informacja:

```
INFO: JICP Secure Peer activated. (auth=true, ta=jicp://des01.eti.pg.gda.pl:5656)
```

Zadania:

- zaprezentowanie poprawnie skonfigurowanej i uruchomionej platformy z szyfrowaniem pomiędzy dwoma/trzema kontenerami (logi, komunikacja między agentami),
- zaprezentowanie poprawnie skonfigurowanej i uruchomionej platformy z szyfrowaniem i autoryzacją pomiędzy dwoma/trzema kontenerami (logi, komunikacja między agentami),
- zaprezentowanie poprawnie skonfigurowanej i uruchomionej platformy w topologii gwiazda zbudowanej z jednego głównego kontenera,

dwóch/trzech zapasowych i jednego/dwóch sfederowanego (ubijanie głównych kontenerów + logi),

- Zaprezentowanie poprawnie skonfigurowanej i uruchomionej platformy w topologii gwiazda zbudowanej z jednego głównego kontenera, dwóch/trzech zapasowych i jednego/dwóch sfederowanego z włączonymi szyfrowaniem i autoryzacją pomiędzy kontenerami (logi, komunikacja między agentami , ubijanie głównych kontenerów + logi).

Zadania realizowane w parach/trójkach.

