CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA - CEFET/RJ

POLÍTICAS DE MIGRAÇÃO PARA SOCIEDADE DE AGENTES

Nicolas da Silva Jatoba

Vinicius Machado Pinto

Monografia apresentada ao Curso de Graduação em Sistemas de Informação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ Campus Nova Friburgo, como parte dos requisitos necessários à obtenção do certificado de Bacharel em Sistemas de Informação.

Orientador: Nilson Mori Lazarin

Nova Friburgo

Janeiro 2024

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA - CEFET/RJ

POLÍTICAS DE MIGRAÇÃO PARA SOCIEDADE DE AGENTES

Monografia apresentada ao Curso de Graduação em Sistemas de Informação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ Campus Nova Friburgo, como parte dos requisitos necessários à obtenção do certificado de Bacharel em Sistemas de Informação.

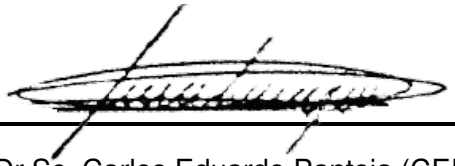
Nicolas da Silva Jatoba

Vinicius Machado Pinto

Banca Examinadora:

gov.br Documento assinado digitalmente
Nilson Mori Lazarin
Data: 25/01/2024 14:58:38-0300
Verifique em https://validar.iti.gov.br

Presidente, Professor M.Sc. Nilson Mori Lazarin(CEFET/RJ) (Orientador)



Professor Dr.Sc. Carlos Eduardo Pantoja (CEFET/RJ)

gov.br Documento assinado digitalmente
BRUNO POLICARPO TOLEDO FREITAS
Data: 25/01/2024 14:54:45-0300
Verifique em https://validar.iti.gov.br

Professor M.Sc. Bruno Policarpo Toledo Freitas (CEFET/RJ)

Nova Friburgo

Janeiro 2024

CEFET/RJ – Sistema de Bibliotecas / Biblioteca Uned Nova Friburgo

J39p Jatoba, Nicolas da Silva.
Políticas de migração para sociedade de agentes. / Nicolas da Silva Jatoba ; Vinicius Machado Pinto . — 2024.
30f.; fig. (color.) : em PDF.

Trabalho de Conclusão de Curso (Sistemas de Informação) - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia: f. 28-30.
Orientador: Nilson Mori Lazarin.

1. Sistemas de Informação. 2. Agentes Inteligentes (software). 3. Controle de acesso - computação. 4. Sistema multiagentes. I. Pinto, Vinicius Machado (co-autor). II. Lazarin, Nilson Mori (orientador) II. Título.

CDD 658.4038

Elaborada pela bibliotecária Cristina Rodrigues Alves CRB7/5932

AGRADECIMENTOS

Gostaríamos de expressar as nossos sinceros agradecimentos a todas as pessoas e instituições que contribuíram e apoiaram durante a realização desta pesquisa e elaboração desta monografia.

Primeiramente, agradecemos ao nosso orientador e professor, Nilson Mori Lazarin, pela orientação dedicada, conselhos valiosos e incentivo ao longo deste processo. Suas orientações foram fundamentais para o desenvolvimento e aprimoramento deste trabalho.

Nossa gratidão também se estende à instituição CEFET-RJ, pelo suporte acadêmico e pelos recursos fornecidos.

À nossa família e amigos, que nos ofereceram apoio incondicional, compreensão e incentivo ao longo dessa jornada acadêmica, meu profundo agradecimento.

Por fim, expressamos a nossa gratidão a todos os que, direta ou indiretamente, contribuíram para este trabalho, mesmo que não tenham sido mencionados aqui. Cada pessoa, cada conversa e cada experiência foram peças essenciais no processo de realização deste estudo.

RESUMO

POLÍTICAS DE MIGRAÇÃO PARA SOCIEDADE DE AGENTES

Um Sistema Multiagentes (SMA) é uma sociedade de softwares cognitivos e sociáveis que compartilham um mesmo ambiente e agem de forma autônoma para atingir seus próprios objetivos, podendo cooperar ou competir. Algumas dessas sociedades podem ser abertas e seus agentes podem entrar e sair livremente. Entretanto, a livre migração de agentes pode apresentar riscos à segurança do SMA, visto que um agente pode ser mal-intencionado. Este trabalho apresenta uma abordagem que contribui para o controle migratório, visando impedir a comunicação ou acesso de agentes não autorizados. Para tal, foi desenvolvido uma extensão da arquitetura dos agentes comunicadores, implementando um modelo firewall para SMA. Além disso foram realizados experimentos para avaliar o desempenho do sistema, confirmando a eficácia das políticas e regras em garantir a segurança do SMA, conforme esperado.

Palavras-chave: Controle de acesso, Políticas de migração, Sistema Multiagentes

ABSTRACT

A Multi-Agent System (MAS) is a society of cognitive and sociable software entities that share a common environment and act autonomously to achieve their own goals, either through cooperation or competition. Some of these societies may be open, allowing their agents to enter and exit freely. However, the free migration of agents can pose security risks to the MAS, as an agent may be malicious. This work presents an approach that contributes to migratory control, aiming to prevent communication or access by unauthorized agents. To achieve this, an extension of the architecture of communicating agents was developed, implementing a firewall model for the MAS. In addition, experiments were conducted to evaluate the system's performance, confirming the effectiveness of policies and rules in ensuring the security of the MAS, as expected.

Keywords: Access control, Migration policies, Multi-agent system

SUMÁRIO

1	**Introdução**	**4**
1.1	Definição do Problema	5
1.2	Contribuição	5
1.3	Estrutura do trabalho	6
2	**Fundamentação Teórica**	**7**
2.1	Firewall	7
2.2	KQML	8
2.3	ContextNet	8
2.4	Agentes comunicadores	9
3	**Trabalhos Relacionados**	**11**
4	**Metodologia**	**13**
4.1	Ações Internas propostas	14
4.2	Implementação	15
5	**Estudo de Caso**	**18**
6	**Análise de Desempenho**	**21**
6.1	Descrição dos cenários de teste	21
6.2	Resultados	24
7	**Conclusões**	**26**
7.1	Trabalhos Futuros	27
Referências		**27**

1- Introdução

Os sistemas multiagentes (SMA) são estruturas computacionais avançadas capazes de lidar com a interação complexa e adaptativa entre entidades autônomas conhecidas como agentes. Cada agente, por sua vez, atua como uma entidade computacional, capaz de tomar decisões e agir de forma independente, considerando suas percepções do ambiente, conhecimentos individuais e objetivos específicos. Esses agentes podem interagir em um ambiente compartilhado, utilizando diversos mecanismos, como comunicação direta, troca de informações e coordenação de ações, para resolver problemas, atingir objetivos comuns e adaptar-se a mudanças no ambiente. Em uma arquitetura descentralizada, onde múltiplos agentes colaboram em vez de dependerem de uma autoridade central, oferece uma abordagem altamente adaptativa para lidar com ambientes dinâmicos e complexos (ALVARES; SICHMAN, 1997).

A aplicação dos sistemas multiagentes se estende por uma variedade de campos, abrangendo desde sistemas de automação industrial até jogos, simulação, robótica e sistemas de informação. Por exemplo, os SMAs são usados para modelar a inteligência artificial (IA) de unidades e adversários em jogos RTS (Estratégia em tempo real), como o jogo StarCraft, também são utilizados para modelar o comportamento de veículos autônomos, motoristas e pedestres em simulações de tráfego urbano, são usados para modelar a interação entre aeronaves, controladores de tráfego e outros elementos no gerenciamento do tráfego aéreo, entre outros. Esses sistemas são particularmente eficazes em cenários que demandam cooperação e coordenação entre múltiplos agentes para resolver problemas complexos e dinâmicos, onde a resposta rápida e a adaptabilidade são notáveis. Essa abordagem descentralizada permite a manifestação de comportamentos coletivos e soluções distribuídas que não seriam facilmente alcançadas por um único agente ou por um sistema centralizado (FERBER, 1999; WOOLDRIDGE, 2009).

Este trabalho explora e aprofunda os desafios e as oportunidades oferecidas pelos sistemas multiagentes abertos, onde a entrada e saída contínua de agentes contribuem para a dinâmica e a evolução constante desses sistemas. A pesquisa se dedica a analisar e enfrentar os desafios associados à segurança decorrente da livre migração de agentes. Ao longo da investigação, foram desenvolvidos mecanismos que possibilitaram a

implementação de políticas e regras específicas para os SMAs, estabelecendo diretrizes que visam controlar e regular a entrada e saída de agentes de maneira a preservar a integridade e eficácia do sistema. Essas políticas oferecem uma abordagem proativa para mitigar potenciais riscos e promover um ambiente seguro e confiável para o funcionamento contínuo dos SMAs abertos (CAMILO JUNIOR; NOGUEIRA; VINHAL, 2009).

1.1- Definição do Problema

Esses sistemas podem lidar com ambientes distribuídos complexos, permitindo a entrada de novos agentes, contribuindo com habilidades específicas, enquanto outros podem sair sem afetar o funcionamento geral. A migração de agentes entre sociedades distintas é uma característica desses sistemas, proporcionando adaptabilidade e flexibilidade para formar coalizões dinâmicas, resolver problemas complexos e otimizar recursos de maneira colaborativa. No entanto, essa mobilidade cria desafios significativos de segurança, como a invasão de agentes maliciosos ou ocorrências de acessos não autorizados, impactando a confiabilidade e integridade das interações entre os agentes (HUBNER, 1995; VILA; SCHUSTER; RIERA, 2007).

1.2- Contribuição

O propósito deste trabalho é fortalecer a segurança dos SMAs, para isso foi proposta uma abordagem para prevenir acessos indesejados e agentes maliciosos com base nas políticas de migração da sociedade dos agentes (PINTO; JATOBA; LAZARIN, 2023). Para isso, são fornecidas extensões para a arquitetura de agentes comunicadores, responsáveis pela entrada e saída de agentes no sistema (SOUZA DE JESUS et al., 2018), implementando um modelo de firewall que pode controlar a migração e a comunicação do agente para outro SMA. Por meio de regras e políticas de segurança definidas, o firewall examina a mensagem recebida do agente externo e através do cabeçalho verifica a identificação da origem, os privilégios e o protocolo, antes de permitir que eles entrem ou

se comuniquem. Esse modelo de firewall foi implementado novas tecnologias e algoritmos de segurança.

Para validar a eficácia do modelo proposto, serão conduzidos testes em um ambiente controlado utilizando uma máquina virtual. Durante esses testes, simular-se-á a entrada de agentes no sistema, representando cenários diversos de interação e comunicação. Avaliar-se-ão casos de comunicação legítima entre agentes autorizados, assim como tentativas de acessos não autorizados e comunicação por parte de agentes maliciosos.

Os resultados obtidos durante os testes indicarão a capacidade do modelo de firewall em controlar adequadamente a migração e comunicação de agentes, reforçando a segurança do SMA. Também será analisado o desempenho do sistema para assegurar que a implementação do modelo de firewall não comprometa significativamente a eficiência e a adaptabilidade do sistema multiagente. Essa abordagem de teste proporcionará insights valiosos sobre a robustez do sistema diante de cenários diversos, validando a eficácia das extensões propostas para a arquitetura de agentes comunicadores.

1.3- Estrutura do trabalho

Este trabalho está estruturado da seguinte maneira: no Capítulo 2, o referencial teórico é apresentado. No Capítulo 3, os trabalhos relacionados são discutidos. No Capítulo 4, as abordagens das políticas de migração propostas são exploradas e apresentadas; no Capítulo 5, o estudo de caso, no Capítulo 6, a avaliação experimental é discutida; por fim no Capítulo 7, a conclusão é apresentada.

2- Fundamentação Teórica

Nesta seção, abordaremos os elementos relevantes da construção da solução: Firewall, KQML, ContextNet e Agentes Comunicadores. O Firewall garante a segurança contra ameaças externas, enquanto o KQML facilita a comunicação entre os agentes. O ContextNet gerencia eficientemente o contexto para uma compreensão refinada do ambiente. Os Agentes Comunicadores desempenham um papel central na interação coordenada entre os SMAs, formando uma solução integrada e eficaz.

2.1- Firewall

Um firewall pode ser baseado em hardware ou em software. Os que são baseados em hardware são dispositivos externos que atuam normalmente no ponto de conexão com a internet, podendo dar suporte a uma pequena rede local ou até mesmo em uma rede corporativa agindo como um concentrador/comutador (hub/switch) de rede. Permitindo assim bloquear/compartilhar uma conexão com determinado membro daquela rede através das políticas e regras configuradas (FORD, 2002). Já um firewall baseado em software é normalmente projetado para trabalhar com sistemas operacionais específicos. Estes firewalls após a instalação vem normalmente com seu próprio conjunto de políticas predefinidas, políticas que permitem especificar qual nível de segurança se deseja obter, podendo permitir todo o tráfego na rede, basicamente o firewall fica desabilitado e até bloquear todo e qualquer tipo de tráfego na rede (FORD, 2002).

A distinção principal entre estes dois tipos de firewalls é que o baseado em hardware protege uma rede, já o baseado em software por ser instalado em um sistema operacional tenta defendê-lo de ataques que já estão em sua rede, e eles também consomem recursos do computador, sendo eles espaço em disco, memória e processamento (FORD, 2002).

2.2- KQML

Outro ponto a ser tratado é a Knowledge Query and Manipulation Language (*KQML*), a qual é um tipo de linguagem de alto nível utilizada na interação entre agentes inteligentes. As mensagens KQML são chamadas *performativas* e estão baseadas na teoria de Atos de Fala, que contém três aspectos importantes, a locução que é maneira de falar, a ilocução que é o significado da intenção por trás da mensagem e a perlocução que é a ação resultante da locução (FININ et al., 1994).

A KQML é uma linguagem de comunicação entre agentes inteligentes. Suas três camadas-camada de conteúdo, camada de comunicação e camada de mensagem-destacam os diferentes aspectos que a KQML abrange na comunicação entre agentes. A camada de conteúdo contém o significado de uma mensagem, podendo incluir expressões linguísticas arbitrárias em seu conteúdo. A camada de comunicação codifica um conjunto de características para aceitar parâmetros de baixo nível, como a identidade do emissor e receptor da mensagem, e também um identificador único associado à mensagem. Já a camada de mensagem determina o tipo de interação que os agentes pretendem ter, para transmitir a mensagem, incluindo a performativa que o transmissor anexa ao conteúdo. A referência à identidade do emissor e receptor, bem como a determinação do tipo de interação, são elementos cruciais na teoria de agentes, que se concentra na interação autônoma de entidades inteligentes para alcançar objetivos específicos (FININ; MCKAY; FRITZSON, 1992).

2.3- ContextNet

O ContextNet é um *middleware* desenvolvido para facilitar a colaboração e a interação social em larga escala em sistemas pervasivos, como redes sociais e aplicativos colaborativos. Essa plataforma trabalha fortemente em tempo real, em cima dessas informações fornecidas através das interações sociais, permitindo que os usuários compartilhem informações e se beneficiem do contexto coletivo gerado (ENDLER et al., 2011).

Esse contexto se refere a informações adicionais fornecidas por esses usuários

que podem influenciar na interpretação e compreensão dos dados, como localização, histórico de atividades e preferências pessoais. Através do uso dessas informações, a plataforma gerencia e compartilha esse contexto, permitindo que os dispositivos e aplicativos conectados troquem informações entre si, isso permite que os usuários recebam recomendações personalizadas e melhorem sua interação com outros usuários e aplicativos (ENDLER et al., 2011). Entretanto, este middleware não apresenta mecanismos de controle ou segurança, sendo necessária a adição destes diretamente pela aplicação (FONSECA; LAZARIN, 2023).

2.4- Agentes comunicadores

A troca de informações é importante na interação entre sistemas que envolvem múltiplos agentes, sendo a comunicação entre esses agentes um elemento significativo nesse processo. Ela se dá por meio de mensagens, essas mensagens são enviadas e recebidas por agentes comunicadores, que possibilitam o compartilhamento de dados, crenças e intenções, geralmente apoiados em linguagens comuns ou protocolos definidos para garantir a compreensão mútua das mensagens transmitidas. A interação por meio da comunicação entre agentes é importante para a execução de ações coordenadas, resolução de problemas e tomada de decisões nos sistemas multiagentes (BORDINI; HBNER; WOOLDRIDGE, 2007).

Na Figura 1, há uma ilustração do comportamento dos agentes comunicadores, os quais demonstram uma função de facilitação na troca de informações entre agentes sistemas multiagente. Esses agentes desempenham um papel importante na interação autônoma entre entidades no ambiente. Eles gerenciam a entrada e saída de mensagens e agentes, assegurando a eficiência na comunicação por meio de protocolos e regras definidas (JESUS et al., 2023; LAZARIN; PANTOJA; SOUZA DE JESUS, 2021; DE SOUZA, 2023).

Agentes comunicadores são importantes para facilitar a comunicação e a mobilidade entre entidades autônomas em SMAs. Eles desempenham um papel pertinente na execução de ações coordenadas, resolução de problemas e tomada de decisões. Utilizando protocolos de transferência, esses agentes não apenas extraem o código-

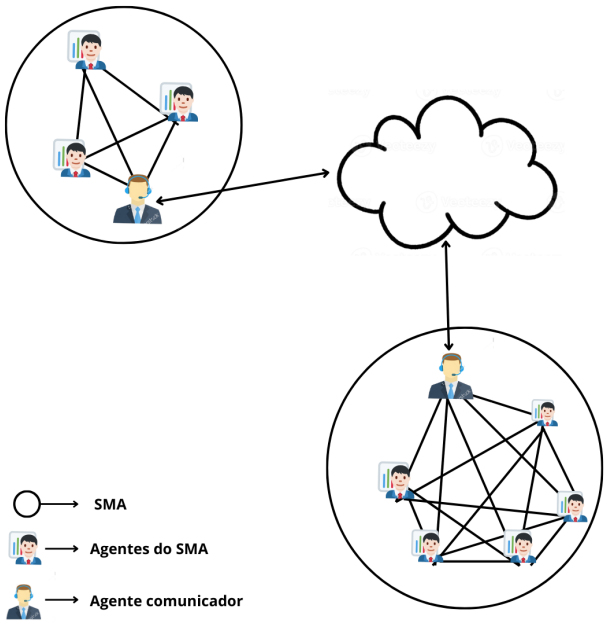


Figura 1 – Comportamento dos Agentes Comunicadores.

fonte inicial dos agentes, mas também transferem conhecimento adquirido durante sua execução. Essa integração permite a transferência eficaz de conhecimento, a manutenção de estados internos, a adaptação dinâmica e a consistência na execução, contribuindo para a eficiência, coordenação e interação no ambiente multiagente (SOUZA DE JESUS et al., 2018).

Esses protocolos, inspirados em dinâmicas ecológicas como *Predação*, *Inquilinismo* e *Mutualismo*, orientam a movimentação entre SMAs e o impacto dessa transição nos sistemas envolvidos. No protocolo de Predação, os agentes transferidos substituem os agentes nativos do receptor, dominando completamente o ambiente. Já no Inquilinismo, residem no SMA receptor como ”hóspedes”, mantendo sua integridade e conhecimento, sem controle direto sobre o ambiente. Enquanto isso, no Mutualismo, a transferência temporária de agentes ocorre para adquirir novas habilidades e conhecimentos, retornando posteriormente ao ambiente de origem (SOUZA DE JESUS, 2020).

Para ativar esses protocolos, uma ação interna é acionada, desencadeando algoritmos nos SMAs de origem e destino. Durante esse processo, a sincronização de conhecimento assegura que os agentes em trânsito mantenham uma transição fluida de informações. Em resumo, os protocolos bioinspirados regem a movimentação de agentes entre SMAs, com efeitos diferenciados na dinâmica e no controle dos ambientes receptores (SOUZA DE JESUS. et al., 2021).

3- Trabalhos Relacionados

Alguns trabalhos (CHEBOUT. et al., 2016), (GUPTA; NAIK; SENGUPTA, 2017), (VILA; SCHUSTER; RIERA, 2007) têm sido realizados no campo de sistemas multiagentes visando desenvolver abordagens e técnicas específicas para fortalecer a segurança e mitigar os riscos associados à livre migração de agentes. Esta seção irá abordar trabalhos relacionados nesse sentido.

Em *Towards preventive control for open MAS - an aspect-based approach*, embora não mencione explicitamente firewalls, é proposta uma abordagem baseada em aspectos para o controle preventivo em sistemas multiagentes abertos, por meio de uma observação dos movimentos dos agentes, interceptando todas as solicitações externa e em seguida é realizada uma análise para verificar se o agente possui ou não determinado recurso para poder prosseguir com sua solicitação, tendo um certo custo de processamento, que leva mais tempo dado que é feito toda uma análise antes de liberar o acesso ou bloqueá-lo. Diferentemente, em nossa abordagem, desenvolvemos um modelo onde são criadas regras e políticas para verificação se o agente pode ou não se comunicar, ou realizar transferências (CHEBOUT. et al., 2016).

Em *A firewall for Internet of Things*, é abordado o paradigma de Internet das Coisas (IoT) e a quantidade de dados em nuvem que são sensíveis a ataques, podendo ser comprometidos. O trabalho propõe uma solução baseada em firewall, embarcado em um Raspberry Pi que protege a comunicação com o banco de dados localizado na nuvem. Através da instalação desse firewall em determinada rede, o trabalho pôde analisar e proteger cada pacote entrando e saindo da mesma. Diferentemente o nosso trabalho, propõe a criação de um modelo de firewall baseado em software para controlar o acesso e a comunicação extra-SMA que podem estar hospedados na mesma rede ou computador (GUPTA; NAIK; SENGUPTA, 2017).

Em *Security for a Multi-Agent System based on JADE*, são apresentados os desafios existentes na busca de soluções dos problemas de segurança em sistemas multiagentes baseados no framework JADE. Dentre das soluções apresentadas destaca-se a autenticação de usuários por criptografia e a autorização do acesso a serviços por determinado grupo de usuários. O gerenciamento apresentado foi capaz de garantir

que os dados não fossem acessados por usuários sem permissão. Diferentemente, este trabalho desenvolveu uma extensão da arquitetura de agentes comunicadores capaz de gerenciar a movimentação e a comunicação entre diferentes SMA baseados em Jason (VILA; SCHUSTER; RIERA, 2007).

Em "*Transporte de Agentes Cognitivos Baseado nos Conceitos de Relações Ecológicas*", propõe protocolos de transferência de agentes em SMAs baseados em conceitos de relações ecológicas (predação, inquilinismo, mutualismo). A arquitetura desenvolvida permite que agentes se movam entre sistemas preservando conhecimento, vivendo como inquilinos temporários ou adquirindo/transmitindo conhecimento. Já o nosso trabalho, desenvolveu um firewall capaz de gerenciar e proteger esses sistemas durante suas interações, seja de comunicação ou movimentação (SOUZA DE JESUS, 2020).

A abordagem baseada em aspectos para controle preventivo (CHEBOUT. et al., 2016) revela um alto custo de processamento devido à extensa análise antes de permitir ou bloquear solicitações. Por outro lado, o firewall para Internet das Coisas (GUPTA; NAIK; SENGUPTA, 2017) possui uma limitação em sua proteção, focando apenas na comunicação com um banco de dados específico na nuvem, além de depender de hardware dedicado para sua implementação. Já a segurança em um Sistema Multiagente baseado em JADE (VILA; SCHUSTER; RIERA, 2007) concentra-se principalmente na autenticação e autorização de usuários, sem um controle minucioso sobre a comunicação e movimentação entre SMAs. Por outro lado, o trabalho sobre o transporte de agentes cognitivos (SOUZA DE JESUS, 2020) enfatiza a movimentação de agentes e transferência de conhecimento, mas não prioriza explicitamente a segurança durante essas interações.

No âmbito deste trabalho, há um gerenciamento abrangente e ativo das interações entre SMAs, proporcionando controle preciso sobre a comunicação e movimentação de agentes. Isso é possível graças à implementação de um modelo de firewall baseado em software, dispensando a dependência de hardware dedicado e garantindo maior flexibilidade na aplicação e manutenção. Essa abordagem é específica e focada na segurança durante as interações entre SMAs, assegurando a proteção dos sistemas durante suas comunicações e transferências. Com um controle adaptativo e flexível sobre a comunicação e movimentação, conseguimos garantir a segurança sem comprometer a eficiência das interações entre os agentes.

4- Metodologia

Buscando adicionar uma camada de segurança e possibilitar o controle de comunicação e acesso ao SMA, foi desenvolvido um modelo de firewall baseado em políticas e regras. Para isso, foi estendida a arquitetura do JasonEmbedded (PANTOJA et al., 2023), adicionando ações internas a serem utilizadas diretamente no código AgentSpeak(L) (RAO, 1996) do agente, capazes de analisar os cabeçalhos da comunicação KQML (FININ et al., 1994) e da migração bio-inspirada (SOUZA DE JESUS. et al., 2021). No contexto deste trabalho foram definidos regras e políticas como:

Regra: *Critérios tais como origem, tipo de interação, força ilocucionária ou protocolo que definem se uma comunicação e/ou migração deve ser permitida ou negada.*

Política: *Define os critérios para permitir ou negar uma comunicação e/ou uma migração de agentes, caso não exista uma regra que se aplique à comunicação*

Esta abordagem visa garantir que apenas agentes confiáveis ingressem na sociedade de agentes. Para isso, foi desenvolvido um mecanismo para restringir ou permitir a comunicação entre agentes com base em políticas de segurança estabelecidas. Seu objetivo é garantir que apenas comunicações KQML autorizadas sejam processadas pelo Agente Comunicador, e que o acesso não autorizado seja impedido. Na Figura 2 é apresentado o fluxograma de análise da comunicação KQML.

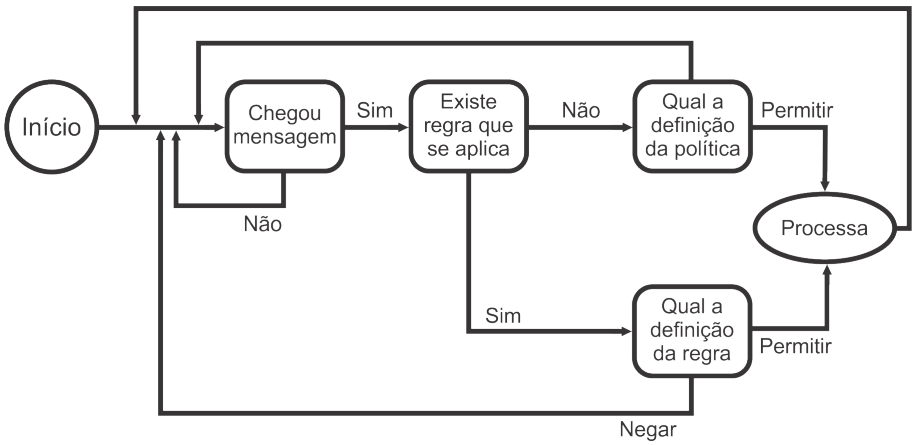


Figura 2 – Fluxograma de análise de comunicação KQML de entrada ou saída do SMA.

4.1- Ações Internas propostas

A abordagem proposta neste trabalho inclui estender a arquitetura dos agentes comunicadores, adicionando ações (*.rule* e *.policy*) para gerenciar políticas e regras de encaminhamento de mensagens e agentes móveis, contribuindo com o aumento da segurança de SMA Abertos. Esse agente irá analisar o cabeçalho que chegará na tentativa de comunicação, definindo qual será a ação a ser realizada, sem considerar a mente do agente. Abaixo é apresentada a Figura 3, sobre o modelo de plano de ação de um agente Comunicador (em AgentSpeak), definindo a política e as regras para comunicação KQML externa ao SMA. Essas regras e políticas devem ser definidas antes da conexão com o servidor e caso sejam feitas alterações em tempo real, é necessário que essa conexão seja reiniciada .

```
+!start <-
    .policy( TIPO, ABRANGENCIA, FORÇA|PROTOCOLO, DETERMINAÇÃO );
    .rule( TIPO, ABRANGECIA, ENDEREÇO, FORÇA|PROTOCOLO, DETERMINAÇÃO );
    .connectCN("skynet.chon.group", 5500, "07ba9e4a-0d539-4a0e-8c14-4ac336476858").
```

Figura 3 – O plano para gerenciamento do *firewall*

Nas políticas, é necessário definir os parâmetros Tipo, Abrangência, Força ou Protocolo e Determinação. Nas regras, é preciso definir os parâmetros tipo, abrangência, origem, destino, força ou protocolo e determinação. Abaixo estão as definições de cada parâmetro e seus possíveis valores.

- **Tipo:** define se a regra ou política se aplica à entrada ou saída $(input|output)$;
- **Abrangência:** define se a regra ou política se aplica a comunicação, transferência ou ambos $(communication|migration|all)$;
- **Endereço:** define a qual SMA a regra se aplica $(source)$;
- **Força/protocolo:** define qual é a força da mensagem ou o protocolo de transferência $(all|illocutionary_{force}|BioInsp_{protocol})$;
- **Determinação:** determina se a regra ou política irá liberar, ou bloquear todos os acessos $(accept|drop)$.

4.2- Implementação

Para a elaboração e funcionamento do firewall, foram criadas duas novas ações internas: *.policy* e *.rule*. Estas funcionam de forma que, quando novas regras ou políticas são inseridas através do código do agente comunicador, são armazenadas em listas separadas e utilizadas na validação das ações. Nas funções de transferência e comunicação do agente comunicador, foram adicionadas verificações que percorrem as listas criadas para validar quais ações serão concluídas ou bloqueadas.

O firewall atua na classe CommMiddleware antes da efetiva conexão entre dois SMAs. Ele desempenha um papel crucial durante o processo de comunicação, estabelecendo verificações e validações antes mesmo que a conexão entre os sistemas multiagentes ocorra. Essa intervenção na CommMiddleware permite que o firewall aplique regras e políticas, analisando e autorizando se a conexão entre os SMAs deve ou não ser estabelecida. Dessa forma, ele controla proativamente a comunicação entre os agentes, garantindo que apenas interações autorizadas e seguras sejam estabelecidas entre os sistemas. Para mais informações e acesso ao código-fonte, o repositório do projeto está disponível no GitHub[1]. A seguir, na Figura 4, é apresentado o diagrama de classe, onde mostra as implementações realizadas no projeto.

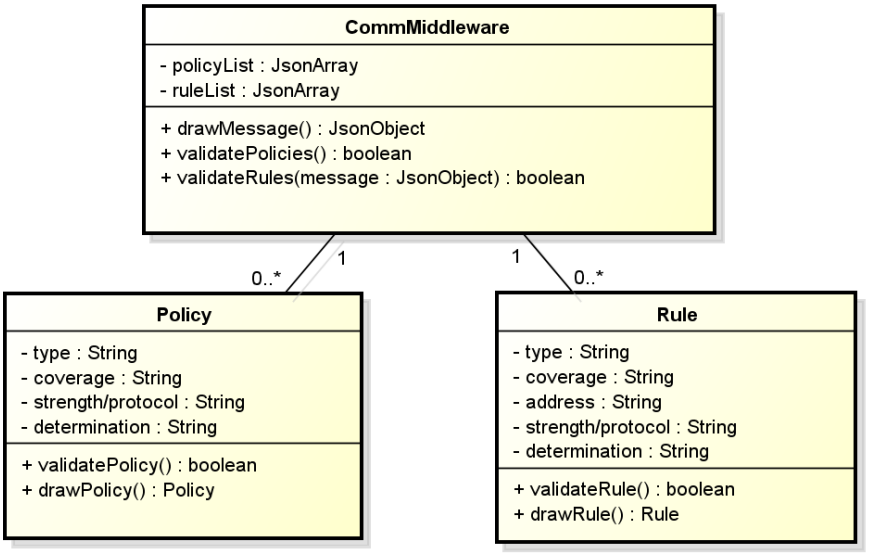


Figura 4 – Diagrama de classe da extensão do sistema.

[1] https://github.com/LabRedesCefetNF/PintoJatoba_2023

O Código 1 apresenta a função que encaminha a mensagem recebida para o ContextNet. Para a aplicação da extensão das políticas e regras, são atribuídos valores como 'output' para o type e 'communication' para a coverage. Esses dois atributos, juntamente com os valores recebidos no cabeçalho da função, são necessários para a construção da mensagem. A mensagem, do tipo JsonObject, é então passada como parâmetro para a função validateRules, que determina se a ação será concluída ou bloqueada, dependendo do valor retornado. Em seguida, esse valor é analisado e, caso a ação seja bloqueada, uma mensagem é exibida na tela informando que a ação não pode ser executada.

```
1  public void sendMsgToContextNet(String sender, String receiver,
       Term force, Term msg) {
2      String type = "output";
3      String coverage = "communication";
4      JsonObject messageJsonObject = drawMessage(type, coverage,
           receiver, force.toString(), msg);
5      boolean result = validateRules(messageJsonObject);
6      if (!result) {
7          System.out.println("The agent does not have permission to
               execute the sendOUT action.");
8      } else {
9          ApplicationMessage message = new ApplicationMessage();
10         message.setContentObject(prepareToSend(sender, force.
               toString(), msg.toString()));
11         message.setRecipientID(UUID.fromString(receiver.substring
               (1, receiver.length() - 1)));
12         try {
13             this.connection.sendMessage(message);
14         } catch (IOException e) {
15             e.printStackTrace();
16         }
17     }
18 }
```

Código 1 – Código de validação da comunicação.

O Código 2 apresenta a função responsável por facilitar a transferência de agentes para o ContextNet. Essa função opera de maneira semelhante ao código anterior, mas

está envolvida na transferência de agentes. Ao contrário do código anterior, o coverage é definida como 'migration'. Uma mensagem do tipo JsonObject é criada e, em seguida, é realizada uma análise para determinar se a migração do agente será executada. Caso a ação seja bloqueada, é exibida uma mensagem na tela informando que a operação não foi concluída.

```
1  public void sendAgentToContextNet(String receiver, Term protocol,
       Term agent) {
2      String type = "output";
3      String coverage = "migration";
4      JsonObject messageJsonObject = createMessage(type, coverage,
           receiver, protocol.toString());
5      boolean result = validateRules(messageJsonObject);
6      if (!result) {
7          System.out.println("The agent does not have permission to
               execute the moveOUT action.");
8      } else {
9          ApplicationMessage message = new ApplicationMessage();
10         this.nameAgents = new ArrayList<String>();
11         this.nameAgents.add(agent.toString());
12         message.setContentObject(prepareToSend(protocol.toString().
               toUpperCase().trim(), agent.toString()));
13         message.setRecipientID(UUID.fromString(receiver.substring
               (1, receiver.length() - 1)));
14         try {
15             this.connection.sendMessage(message);
16         } catch (IOException e) {
17             e.printStackTrace();
18         }
19     }
20 }
```

Código 2 – Código de validação da transferência.

5- Estudo de Caso

No exemplo de aplicação abaixo realizado utilizando a ChonIDE (SOUZA DE JESUS et al., 2023), foi considerado a presença de três SMAs: *Andoria*, *BirdOfPrey* e *Enterprise*. Cada um desses SMAs desempenha um papel específico na demonstração do uso do Agente Comunicador com políticas de migração.

O SMA *Andoria*, representado no Código 3, atua como receptor de transporte. Ele aguarda a chegada de uma mensagem solicitando transporte e, quando essa mensagem é recebida, realiza uma validação para determinar se o agente remetente possui acesso autorizado. Essa validação é realizada com base nas políticas e regras estabelecidas pelo Firewall do Agente Comunicador.

```
/* Initial beliefs and rules */
scott("07ba9e4a-d539-4a0e-8c14-4ac336476858").
skyNetAddress("skynet.chon.group",5500).
myUUID("41ff1712-b2f0-416d-8232-fef834651e77").

/* Initial goals */
!start.

/* Plans */
+energizing[source(X)]: scott(ScottUUID) & X=ScottUUID <- !move.
+!move: scott(UUID) <- .moveOut(UUID,inquilinism).
+!start: scott(UUID) & myUUID(ID) & skyNetAddress(Server,Port) <-
    .connectCN(Server,Port,ID);
    .print("Kirk to Scotty...");
    .sendOut(UUID,tell,beam_us_up_scotty)
.
```

Código 3 – Código AgentSpeak(L) do agente Kirk.

Por outro lado, o SMA *BirdOfPrey*, apresentado no Código 4, representa um agente que não possui acesso autorizado. Quando tenta se comunicar ou solicitar transporte, o Firewall do Agente Comunicador detecta a tentativa e bloqueia a ação, impedindo a interação não autorizada.

```
/* Initial beliefs and rules */
scott("07ba9e4a-d539-4a0e-8c14-4ac336476858").
skyNetAddress("skynet.chon.group",5500).
myUUID("8268e208-3341-49ae-82eb-b13487708bc1").

/* Initial goals */
!move.

/* Plans */
+!move: scott(UUID) & myUUID(ID) & skyNetAddress(Server,Port) <-
    .connectCN(Server,Port,ID);
    .print("Klingon to Scotty...");
    .moveOut(UUID,inquilinism).
```

Código 4 – Código AgentSpeak(L) do agente Klingon.

Por fim, o SMA *Enterprise*, mostrado no Código 5, representa um agente com acesso autorizado. Esse agente pode interagir livremente com o *Andoria*, solicitando transporte e tendo suas mensagens processadas conforme as políticas e regras estabelecidas.

```
/* Initial beliefs and rules */
skyNetAddress("skynet.chon.group",5500).
kirkUUID("41ff1712-b2f0-416d-8232-fef834651e77").

/* Initial goals */
!start.

/* Plans */
+!start: skyNetAddress(Server,Port) & kirkUUID(Kirk) <-
 .policy(all, all, all, drop);
 .rule(all, all, Kirk , all, accept);
 .conectCN(Server, Port, "07ba9e4a-0d539-4a0e-8c14-4ac336476858").

+beam_us_up_scotty[source(X)] <-
 .print("Transporter ready for ", X);
 .sendOut(X, tell, energizing).
```

Código 5 – Código AgentSpeak(L) do agente Scott.

Dessa forma, esse cenário de teste demonstra como o Agente Comunicador com Firewall contribui para a segurança em um Sistema Multiagente Aberto, controlando o acesso e garantindo que apenas interações autorizadas ocorram entre os agentes.

O agente Kirk (no SMA Andoria) solicita a transferência utilizando o Protocolo de Inquilinismo. O agente Scotty (em Enterprise), responsável pela política e regra de acesso configuradas, concede permissão para a transferência. Logo em seguida, o agente mal-intencionado Klingon (em BirdOfPrey), tenta, sem consentimento, migrar para Enterprise e tem sua solicitação negada, como ilustrado nas Figuras 5a e 5b.

(a) Kirk solicitando transporte e Scotty liberando o acesso.

(b) Klingon tentando acessar sem permissão.

Figura 5 – Execução do estudo de caso.

6- Análise de Desempenho

Neste capítulo 6 é apresentada a avaliação experimental dos protocolos do firewall para o transporte e comunicação de agentes. Além disso, é descrito todos os experimentos executados para cada um dos protocolos desenvolvidos, bem como uma análise de seus resultados. Na seção 6.1 é descrito o ambiente de software e hardware, além do dos cenários de teste realizados em nosso trabalho. Na seção 6.2 são exibidos os resultados dos experimentos executados.

6.1- Descrição dos cenários de teste

Para a realização dos testes e comparações foi utilizado uma máquina virtual com as seguintes configurações: um processador com 2 cores, 2 GB de RAM e 20 GB de HD. Em relação ao sistema operacional, foi utilizado o Ubuntu 22.04.3. Os testes foram realizados em um cenário onde dois agentes interagem entre si através da comunicação, sendo trocadas mensagens até atingir certo número de interações pré-definidas.

Em cada teste foram adicionadas diferentes politicas e regras. Inicialmente esse teste com as interações foi executado no modelo inicial sem a implementação do firewall. Em seguida, o modelo de firewall implementado foi utilizado, mas sem a adição de políticas e regras. Após essa etapa, diferentes políticas e regras foram gradualmente incorporadas, em quantidades variadas (16, 32, 64, 128, 256, 512, 1024), para efeitos de comparação de desempenho. Os Códigos 6 e 7 de cada agente, utilizados para a realização dos testes, são apresentados abaixo.

```
1 /* Initial beliefs and rules */
2 agent2("80d9c5b3-5327-4836-b722-7481061affef").
3 myUUID("af467a22-eafc-4e87-9f57-882740ab0710".
4
5 /* Initial goals */
6 !start.
```

```
/* Plans */
+!start: agente2(AgUUID) & myUUID(MyID)<-
    .policy(all,migration,all,drop);
    .policy(input,communication,all,drop);
    .policy(output,communication,all,accept);
    .rule(input,communication,AgUUID,untell,accept);
    .rule(input,communication,AgUUID,tell,accept);
    .rule(input,communication,AgUUID,achieve,accept);
    .rule(input,communication,AgUUID,unachieve,accept);
    .connectCN("127.0.0.1",5500,MyID);
    .send(agent2,achieve,start).
+nextRound[source(X)] <-
  -nextRound[source(X)];
  .abolish(ping(_));
  .send(agent2,achieve,start).
+ping(Nr)[source(Who)]<-
    .sendOut(Who, tell, ping(Nr+1)).
```

Código 6 – Código AgentSpeak(L) do agente 1.

```
/* Initial beliefs and rules */
tests(10). // The number of tests here
messages(1024). // The number of messages here
agent1("af467a22-eafc-4e87-9f57-882740ab0710").
uuid("80d9c5b3-5327-4836-b722-7481061affef").
/* Initial goals */
!connect.

/* Plans */
+!connect: agent1(AgUUID) & uuid(MyID) <-
    .policy(all,migration,all,drop);
    .policy(input,communication,all,accept);
    .policy(output,communication,all,accept);
    .rule(input,communication,AgUUID,untell,accept);
    .rule(input,communication,AgUUID,tell,accept);
    .rule(input,communication,AgUUID,achieve,accept);
    .rule(input,communication,AgUUID,unachieve,accept);
    .connectCN("localhost",5500,MyID).
```

```
19
20 +!start[source(X)]: messages(N) & tests(T) & T > 0 <-
21   .print("Starting test ",T," with: ",N);
22   .time(H,M,S);
23   .print("Inicio  ", H,":", M,":", S);
24   !send(0).
25
26 -!start <-
27     .print("End of test. Stopping SMA");
28     .stopMAS.
29
30 +!send(NR): agent1(AgUUID) <-
31     .sendOut(AgUUID, tell, ping(NR));
32     .wait(1000);
33     !arrived(NR).
34
35 +ping(Nr)[source(Who)]: messages(T) & Nr < T <-
36   !send(Nr+1).
37
38 +ping(Nr)[source(Who)]: messages(N) & (Nr >= N) & tests(T)<-
39   .time(H,M,S);
40   .print("End  ", H,":", M,":", S);
41   .wait(5000);
42   .abolish(ping(_));
43   .abolish(tests(_));
44   +tests(T-1);
45   .send(agent1,tell,nextRound).
46
47 +!arrived(NR): ping(P) & (P = NR+1).
48
49 -!arrived(NR) <- .print("ERROR:",NR); !send(NR).
```

Código 7 – Código AgentSpeak(L) do agente 2.

6.2- Resultados

A Tabela 1 abaixo fornece detalhes abrangentes sobre os resultados obtidos nos testes conduzidos, apresentando a média de execução e o desvio padrão para cada experimento. Esta tabela oferece uma visão detalhada das métricas resultantes da análise, proporcionando uma compreensão mais aprofundada do desempenho médio e da variabilidade associada aos testes realizados.

	16 Msgs	32 Msgs	64 Msgs	128 Msgs	256 Msgs	512 Msgs	1024 Msgs
1	Média: 8s Desvio: 1s	Média: 15s Desvio: 1s	Média: 29s Desvio: 1s	Média: 56s Desvio: 4s	Média: 110s Desvio: 11s	Média: 160s Desvio: 42s	Média: 282s Desvio: 29s
2	Média: 8s Desvio: 1s	Média: 16s Desvio: 1s	Média: 29s Desvio: 2s	Média: 58s Desvio: 2s	Média: 107s Desvio: 12s	Média: 170s Desvio: 42s	Média: 275s Desvio: 18s
3	Média: 8s Desvio: 1s	Média: 15s Desvio: 1s	Média: 30s Desvio: 1s	Média: 57s Desvio: 4s	Média: 105s Desvio: 16s	Média: 163s Desvio: 40s	Média: 278s Desvio: 26s
4	Média: 8s Desvio: 1s	Média: 15s Desvio: 1s	Média: 30s Desvio: 2s	Média: 56s Desvio: 5s	Média: 104s Desvio: 177s	Média: 167s Desvio: 38s	Média: 270s Desvio: 15s
5	Média: 8s Desvio: 1s	Média: 15s Desvio: 1s	Média: 29s Desvio: 2s	Média: 56s Desvio: 5s	Média: 104s Desvio: 17s	Média: 164s Desvio: 35s	Média: 283s Desvio: 29s

Tabela 1 – (1) Sem firewall; (2) Jason modificado sem regras; (3) Jason modificado com política; (4) Jason modificado com política e regra; (5) Jason modificado com 3 políticas e 4 regras.

Na Figura 6, o gráfico ilustra os resultados dos testes, revelando a ausência de queda no desempenho ao comparar o teste sem a implementação do modelo de firewall com os testes conduzidos com políticas e regras definidas. Observa-se que não foram registradas reduções significativas no desempenho, evidenciando a eficácia das políticas e regras implementadas. A figura proporciona uma representação visual clara dessas observações, contribuindo para uma compreensão mais aprofundada do impacto do modelo de firewall nos resultados dos testes realizados.

A aplicação do firewall como modelo de controle de acesso no SMA trouxe melhorias notáveis na gestão e controle das comunicações entre os agentes. Estabeleceram-se políticas claras para determinar as interações permitidas entre agentes, considerando identidades, tipos de mensagem e níveis de permissão. O firewall demonstrou eficiência ao gerenciar o fluxo de comunicações, permitindo interações necessárias e bloqueando acessos não autorizados, conforme evidenciado na Figura 6, sem afetar o desempenho do sistema. Sua escalabilidade e flexibilidade permitiram adaptações conforme o sistema cresce e as condições de segurança mudam, possibilitando a inclusão de novas políticas

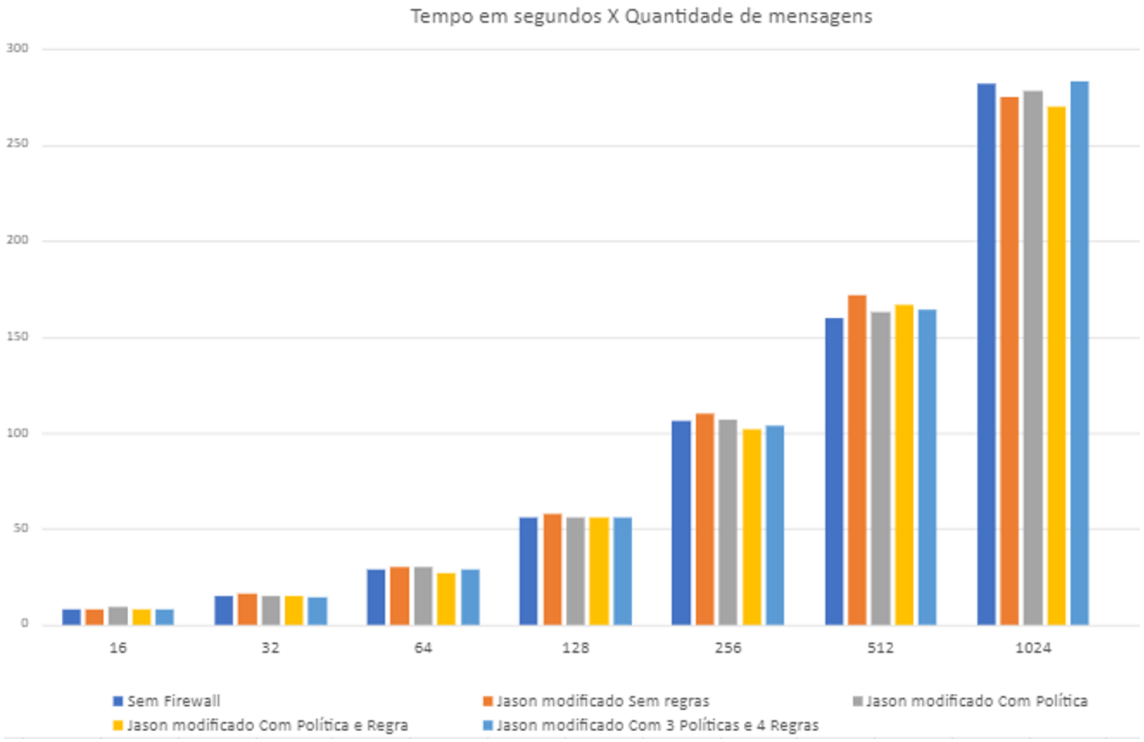


Figura 6 – Gráfico com os resultado dos testes

sem comprometer a eficiência.

Os resultados dos testes realizados corroboram a eficácia do firewall, proporcionando uma camada adicional de proteção para os SMAs. Essa implementação bem-sucedida alcançou o equilíbrio ideal entre segurança e desempenho, garantindo a proteção contra acessos não autorizados sem comprometer a eficiência de execução dos sistemas. Esse modelo de controle de acesso impactou positivamente a segurança do SMA, reduzindo significativamente o risco de vulnerabilidades e fortalecendo a confiabilidade das interações entre os agentes.

7- Conclusões

A implementação de políticas e regras nos SMAs desempenham um papel importante na gestão, controle e segurança das interações e transferências entre os agentes. Ao estabelecer políticas meticulosas e regras de acesso bem definidas, esse modelo oferece um controle preciso sobre como os agentes se comunicam e transferem dados, fundamentando-se em identidades, tipos de mensagem e níveis de permissão.

A eficácia desse firewall em administrar o fluxo de comunicação e transferência, permitindo interações e transferências de dados para o funcionamento do sistema, enquanto impede acessos e transferências não autorizados. A análise dos testes confirma sua eficiência sem comprometer o desempenho do sistema, mantendo sua integridade operacional.

A flexibilidade e adaptabilidade desse modelo são, permitindo a definição contínua de novas políticas de acesso e regras de transferência, adaptando-se às mudanças do ambiente eficientemente. Esse aspecto se traduz em um impacto positivo na segurança global dos Sistemas Multiagentes, reduzindo significativamente os riscos de acessos não autorizados e garantindo a confiabilidade e integridade das interações e transferências de dados entre os agentes.

Em resumo, a presença desse modelo de firewall não apenas se mostra como uma barreira robusta contra ameaças, mas também como um sistema adaptável e confiável, essencial para a proteção e integridade das comunicações e transferências de dados em ambientes onde a segurança da informação é de extrema importância. Sua viabilidade e relevância são inegáveis, especialmente em contextos onde a comunicação e a transferência segura de dados entre os agentes são fundamentais para o funcionamento adequado do sistema.

7.1- Trabalhos Futuros

Como trabalho futuro, pode ser feita uma validação de conflito de regras, como a possibilidade de uma regra permitir um UUID enquanto outra o bloqueia, que se apresenta como um aspecto intrigante para investigações futuras. A abordagem e resolução de conflitos em políticas de firewall são desafios significativos, e a exploração dessa problemática pode resultar em contribuições valiosas para o aprimoramento de sistemas de controle de acesso em ambientes multiagentes. Considerando a complexidade e importância dessa questão, a investigação de estratégias eficazes de validação de conflitos de regras pode ser direcionada como uma área promissora para futuros trabalhos nesta linha de pesquisa.

Outra linha de pesquisa interessante seria a garantia da confidencialidade e integridade das informações trocadas é um desafio importante em ambientes de comunicação, especialmente em sistemas multiagentes. Explorar estratégias avançadas para assegurar esses princípios, como criptografia e verificações de integridade robustas, pode representar uma área promissora para pesquisas futuras. A investigação aprofundada de técnicas e protocolos que fortaleçam a proteção de informações sensíveis, considerando as peculiaridades de ambientes multiagentes, pode ser delineada como um trabalho futuro. O desenvolvimento de abordagens inovadoras para a confidencialidade e integridade das comunicações não apenas contribuiria para a evolução desses sistemas, mas também poderia ter implicações significativas em diversos domínios onde a segurança da informação é importante.

Por fim, a inclusão de técnicas como autenticação, criptografia e verificação de integridade de mensagens trocadas entre agentes emerge como uma proposta relevante para fortalecer a segurança em sistemas multiagentes. Explorar a implementação prática dessas técnicas e avaliar seu impacto na proteção das comunicações representa uma vertente valiosa para pesquisas futuras. A análise aprofundada do desempenho e da eficácia dessas medidas de segurança, considerando as particularidades de ambientes multiagentes, pode constituir um trabalho futuro. O desenvolvimento de estratégias inovadoras que combinem de maneira otimizada autenticação, criptografia e verificação de integridade poderia contribuir para a criação de sistemas mais robustos e resilientes à medida que buscam salvaguardar a integridade e confidencialidade das informações.

Referências

ALVARES, Luis Otavio; SICHMAN, Jaime Simão. Introdução aos Sistemas Multiagentes. In: ANAIS da Jornada de Atualização em Informática - XVI JAI. Brasília: SBC, 1997.

BORDINI, Rafael H.; HBNER, Jomi Fred; WOOLDRIDGE, Michael. **Programming Multi-Agent Systems in AgentSpeak usingJason**. Chichester, UK: John Wiley & Sons, Ltd, 2007. (Wiley Series in Agent Technology). DOI: 10.1002/9780470061848.

CAMILO JUNIOR, Celso Gonçalves; NOGUEIRA, Reinaldo Gonçalves; VINHAL, Cássio Dener Noronha. Inteligência Artificial Distribuida: conhecendo para aplicar. **Revista Estudos - Vida e Saúde (Ciências Ambientais e Saúde)**, v. 35, n. 2, p. 247–256, mar. 2009. DOI: 10.18224/est.v35i2.644.

CHEBOUT., Mohamed Sedik et al. Towards Preventive Control for Open MAS - An Aspect-based Approach. In: INSTICC. PROCEEDINGS of the 13th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO. Lisbon: SciTePress, 2016. P. 269–274. DOI: 10.5220/0006005602690274.

DE SOUZA, João Pedro Bernardo. **Comunicação entre SMA Embarcados: Uma Arquitetura Baseada em Protocolos da Camada de Aplicação**. 2023. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Rio de Janeiro.

ENDLER, M. et al. ContextNet: Context Reasoning and Sharing Middleware for Large-Scale Pervasive Collaboration and Social Networking. In: PROCEEDINGS of the Workshop on Posters and Demos Track. Lisbon, Portugal: ACM, 2011. (PDT '11). ISBN 9781450310734. DOI: 10.1145/2088960.2088962.

FERBER, Jacques. **Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence**. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

FININ, Tim; MCKAY, Don; FRITZSON, Rich. An overview of KQML: A knowledge query and manipulation language. Technical report, Department of Computer Science, University of Maryland, Baltmore, 1992.

FININ, Tim et al. KQML as an agent communication language. In: PROCEEDINGS of the Third International Conference on Information and Knowledge Management. Gaithersburg, Maryland, USA: Association for Computing Machinery, 1994. (CIKM '94), p. 456–463. DOI: 10.1145/191246.191322.

FONSECA, Vitor; LAZARIN, Nilson Mori. Uma análise da segurança nas comunicações entre agentes inteligentes. In: ANAIS do XVII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESAAC 2023). Pelotas: UFPel, 2023. P. 14–19.

FORD, Jerry Lee. **Manual completo de firewalls pessoais: tudo o que você precisa saber para proteger o seu computador**. São Paulo: Pearson Education do Brasil, 2002.

GUPTA, Naman; NAIK, Vinayak; SENGUPTA, Srishti. A firewall for Internet of Things. In: 2017 9th International Conference on Communication Systems and Networks (COMSNETS). Bengaluru, India: IEEE, 2017. P. 411–412. DOI: 10.1109/COMSNETS.2017.7945418.

HUBNER, Jomi Fred. **Migração de agentes em sistemas multi-agentes abertos**. 1995. Dissertação (Mestrado) - Curso de Pós-Graduação em Ciência da Computação – Universidade Federal do Rio Grande do Sul, Porto Alegre.

JESUS, Vinicius Souza De et al. A middleware for providing communicability to Embedded MAS based on the lack of connectivity. en. **Artificial Intelligence Review**, v. 56, S3, p. 2971–3001, dez. 2023. ISSN 0269-2821, 1573-7462. DOI: 10.1007/s10462-023-10596-z.

LAZARIN, Nilson Mori; PANTOJA, Carlos; SOUZA DE JESUS, Vinicius. Um Protocolo para Comunicação entre Sistemas Multi-Agentes Embarcados. In: ANAIS do XV Workshop-Escola de Sistemas de Agentes, seus Ambientes e apliCações (WESAAC 2021). Rio de Janeiro: Cefet/RJ, ago. 2021. P. 157–168.

PANTOJA, Carlos Eduardo et al. A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In_____. **Intelligent Systems**. Cham: Springer Nature Switzerland, 2023. P. 382–396. DOI: 10.1007/978-3-031-45368-7_25.

PINTO, Vinicius Machado; JATOBA, Nicolas; LAZARIN, Nilson Mori. Uma proposta de políticas de migração para sociedade de agentes. In: ANAIS do XVII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESAAC 2023). Pelotas: UFPel, 2023. P. 8–13.

RAO, Anand S. AgentSpeak(L): BDI agents speak out in a logical computable language. In: VAN DE VELDE, Walter; PERRAM, John W. (Ed.). **Agents Breaking Away**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. P. 42–55. ISBN 978-3-540-49621-2. DOI: 10.1007/BFb0031845.

SOUZA DE JESUS, Vinicius. **Transporte de agentes cognitivos baseado nos conceitos de relações ecológicas**. 2020. Projeto Final (Graduação) – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ), Rio de Janeiro.

SOUZA DE JESUS, Vinicius et al. An IDE to Support the Development of Embedded Multi-Agent Systems. In_____. **Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection**. Cham: Springer Nature Switzerland, 2023. P. 346–358. DOI: 10.1007/978-3-031-37616-0_29.

SOUZA DE JESUS, Vinicius et al. Transporte de Agentes Cognitivos entre SMA Distintos Inspirado nos Princípios das Relações Ecológicas. In: ANAIS do XII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações. Fortaleza: UFSC, 2018. P. 179–187.

SOUZA DE JESUS., Vinicius et al. Bio-Inspired Protocols for Embodied Multi-Agent Systems. In: PROCEEDINGS of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART. Online: SciTePress, 2021. P. 312–320. ISBN 978-989-758-484-8. DOI: 10.5220/0010257803120320.

VILA, X.; SCHUSTER, A.; RIERA, A. Security for a Multi-Agent System based on JADE. **Computers Security**, v. 26, n. 5, p. 391–400, 2007. DOI: 10.1016/j.cose.2006.12.003.

WOOLDRIDGE, Michael J. **An introduction to multiagent systems**. 2nd ed. Chichester, U.K: John Wiley & Sons, 2009.