**Análise da teoria de linguagens formais e autômatos finitos determinísticos no reconhecimento de ataques SQL Injection e Cross-site Scripting (XSS) em tráfego HTTP**

Andrea Ono Sakai, Graduação em Engenharia Elétrica[[1]](#footnote-1)

Murillo Rodrigues Araujo do Prado[[2]](#footnote-2)

Victor Hugo Rodrigues Araujo[[3]](#footnote-3)

Mariana Rodrigues de Carvalho Martinelli[[4]](#footnote-4)

**RESUMO**

Com o passar dos anos, o crescimento e a facilidade de pesquisa e descoberta de assuntos diversos na Internet, abriu portas para que mais pessoas buscassem entender tópicos antes complexos, ou pouco acessíveis. Consequentemente, a segurança na internet veio a se destacar, pois as vulnerabilidades presentes em mecanismos de segurança foram rapidamente descobertas e exploradas, tais como ataques *SQL Injection* e *Cross-site Scripting (XSS)*, técnicas que visam contornar a segurança e acessar dados em bancos de dados *Structured Query Language (SQL)* e *Not Only Structured Query Language (NoSQL)*. De acordo com Barbosa e Castro (2015), 96% das aplicações atuais contêm vulnerabilidades relacionadas à segurança e, dentre elas, estão *SQL Injection* e *Cross-site Scripting (XSS)*, por esse motivo, é altamente recomendável a utilização de mecanismos, como Autômatos Finitos Determinísticos (AFD), que sejam capazes de identificar tentativas de ataque e neutralizá-las, evitando o vazamento de informações sensíveis. O artigo tem como objetivo revisar estudos sobre ataques *SQL Injection* e *XSS* junto com o uso de AFDs para reforçar a segurança de bancos de dados *SQL* e *NoSQL*. A pesquisa, realizada em bases como ACM Transactions e Google Scholar (2014–2024), analisou qualitativamente trabalhos sobre linguagens formais e segurança de aplicações, excluindo os que não tratavam diretamente do tema, buscando compreender como os Autômatos podem auxiliar na detecção e prevenção de ataques em sistemas web. Awad, Cardoso e Bussador (2024), apontam que “as análises de segurança mostram que cerca de 23% das vulnerabilidades críticas encontradas em 2023 estavam relacionadas a injeções, o que inclui tanto SQL quanto NoSQL” (p. 4), sendo o protocolo HTTP o principal meio onde são implementados os ataques. Khalaf et al. (2021) propuseram um método de validação baseado em modelos formais, capaz de detectar padrões maliciosos antes do processamento, mostrando que o uso de AFDs nesse contexto é uma estratégia eficaz e sólida para reforçar a segurança, pois permite a identificação preventiva de padrões de ataque ainda na camada de comunicação. Estes ataques utilizam comandos SQL com o objetivo de validar ou manipular entradas de usuários não autorizados, podendo ser identificados por padrões maliciosos que violam o formato esperado de entrada. Esses padrões são geralmente introduzidos por meio de *payloads*, ou seja, cadeias de dados especialmente construídas para explorar vulnerabilidades conhecidas. Um exemplo é o uso de injeções do tipo UNION SELECT, nas qual o invasor tenta concatenar resultados controlados à consulta original, a fim de extrair colunas ou linhas adicionais, sendo comumente utilizado quando a consulta legítima retorna dados, como resultados de pesquisa, e o atacante busca obter informações complementares do banco de dados. Outro tipo de ataque ocorre nas chamadas injeções de tautologia, em que o invasor insere condições sempre verdadeiras nas cláusulas de filtragem, um exemplo clássico usando este tipo de injeção, é o uso da expressão ' OR '1'='1', que faz com que a consulta retorne todos os registros de uma tabela, ignorando completamente as restrições originais do sistema, ele é especialmente perigoso em campos de login e autenticação, permitindo o acesso não autorizado a contas sem necessidade de senha válida. Segundo Hopcroft, Motwani e Ullman (2006, p. 34), “um autômato finito determinístico consiste em um conjunto finito de estados, incluindo um estado inicial e um conjunto de estados de aceitação, e uma função de transição que, dado um estado e um símbolo de entrada, retorna exatamente um próximo estado”. Diversos tipos de aplicações podem utilizá-los, tais como compiladores que utilizam seu algoritmo para reconhecer padrões de tokens, ou protocolos de comunicação que podem validar sequências de entrada e detectar comportamentos anômalos ou maliciosos. Um exemplo prático pode ser observado em sistemas que permitem somente combinações de letras e números no campo de usuário, como [a-zA-Z0-9]. Nesse caso o AFD só aceitaria esses símbolos, rejeitando, por exemplo, um *SQL Injection* como ‘ OR ‘1’ = ‘1. Considerando ataques *XSS*, o AFD age rejeita sequências com símbolos ou padrões suspeitos, como a *tag* <*script*> ou atributos *onerror* e *onclick*, frequentemente explorados nesse tipo de ataque. Assim, pode ser entendido que o autômato age como um filtro, que libera somente a passagem de cadeias seguras e bloqueia entradas maliciosas. Estudos como o de Shachi et al. (2021) destacam que, embora existam diversas técnicas de mitigação para ataques de injeção, ainda há lacunas na detecção baseada em padrões comportamentais e linguagens formais, reforçando a importância de pesquisas que explorem o uso de AFDs no reconhecimento de assinaturas maliciosas em tráfego HTTP. Portanto, conclui-se que o uso de linguagens formais e autômatos implementados para melhorar a segurança das aplicações é uma estratégia eficaz para validar entradas e evitar ataques *SQL Injection e XSS*. Contudo, ainda é necessário pesquisas futuras, na qual a integração de AFDs são aplicadas em cenários reais, verificando não só sua eficácia na detecção de ataques, mas também o impacto sobre o desempenho, escalabilidade e experiência do usuário, principalmente considerando sua capacidade em lidar com diferentes volumes de requisições HTTP.  Em síntese, a pesquisa reforça a importância de integrar técnicas baseadas em linguagens formais e autômatos finitos no desenvolvimento seguro, demonstrando que abordagens teóricas podem gerar soluções concretas e robustas para a proteção de sistemas, consolidando a relevância de estratégias formais na segurança cibernética.

**Palavra-chave: :** Autômatos Finitos Determinísticos. *SQL Injection*. *XSS*.

**REFERÊNCIAS**

AWAD, Kassem Ubinski; CARDOSO, Luciano Santos; BUSSADOR, Alessandra. NoSQL e Segurança: Um estudo de análise para prevenção de injeção em bancos de dados NoSQL. *In*: CONGRESSO LATINO-AMERICANO DE SOFTWARE LIVRE E TECNOLOGIAS ABERTAS (LATINOWARE), 21. , 2024, Foz do Iguaçu/PR. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 464-467. DOI: https://doi.org/10.5753/latinoware.2024.245334.

HOPCROFT, John E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. Introduction to Automata Theory, Languages, and Computation. 3. ed. Boston: Addison-Wesley, 2006.

KHALAF, Osamah Ibrahim; SOKIYNA, Maan; et al. *Web Attack Detection Using the Input Validation Method: DPDA Theory. Computers, Materials & Continua*, v. 68, n. 3, p. 4247–4261, 2021. DOI: 10.32604/cmc.2021.015671.

NEERA, Jeyamohan; CHEN, Xiaomin; ASLAM, Nauman; ISSAC, Biju. *A Trustworthy and Untraceable Centralised Payment Protocol for Mobile Payment*. ACM Transactions on Privacy and Security, v. 28, n. 2, Art. 22, p. 1–29, fev. 2025. DOI: https://doi.org/10.1145/3706421.

SHACHI, D.; et al. A Survey on Detection and Prevention of SQL and NoSQL Injection Attack on Server-side Applications. International Journal of Computer Applications, v. 183, n. 10, p. 25–32, 2021. DOI: 10.5120/ijca2021921396.

1. asakai@brazcubas.edu.br - ID Lattes: 0519974724105439- Graduação em Engenharia Elétrica - Mestrado e Doutorado em Engenharia Biomédica. [↑](#footnote-ref-1)
2. murilloprado1804@gmail.com - 34594086 - Centro Universitário Braz Cubas - Graduando em Ciência da Computação (6º semestre) [↑](#footnote-ref-2)
3. victor.araujo04@cs.brazcubas.edu.br - 34147021 - Centro Universitário Braz Cubas - Graduando em Ciência da Computação (6º semestre) [↑](#footnote-ref-3)
4. tijopemamari@gmail.com - 34672265 - Centro Universitário Braz Cubas - Graduando em Ciência da Computação (6º semestre) [↑](#footnote-ref-4)