Desenvolvimento de uma Aplicação Web para Análise e Correlação de Logs do Apache e ModSecurity

Dimitry Carrier Nunes[1], **Emanuel de Franceschi Vieira**[1], **Tiago Antônio Rizzetti**[1]

[1]Colégio Técnico Industrial de Santa Maria (CTISM)
Universidade Federal de Santa Maria (UFSM)

dimitry.nunes@redes.ufsm.br, emanuel.franceschi@acad.ufsm.br,
tiago.rizzetti@ufsm.br

Abstract. *This work presents the development of a Web application for analyzing and correlating logs from the Apache server and ModSecurity. The solution was motivated by the lack of lightweight and accessible tools capable of visualizing, relating, and interpreting these records efficiently, especially in environments with limited resources. The system, implemented in PHP, interprets the different log formats, reconstructs the blocks of the ModSec Audit Log, and standardizes timestamps to enable consistent comparisons across distinct sources. A Web interface provides interactive filters, parallel visualization of the logs, and a timeline that highlights the sequence of events associated with each IP address. The partial results show that the tool is capable of displaying raw records, structuring ModSecurity events, and correlating accesses, errors, and WAF actions, thus assisting in incident investigation and in understanding request behavior. The application proves to be an efficient and easily deployable alternative.*

Resumo. *Este trabalho apresenta o desenvolvimento de uma aplicação Web para análise e correlação de logs do servidor Apache e do ModSecurity. A solução foi motivada pela ausência de ferramentas leves e acessíveis que permitam visualizar, relacionar e interpretar esses registros de forma ágil, especialmente em ambientes com recursos limitados. O sistema, implementado em PHP, interpreta os diferentes formatos de log, reconstrói os blocos do ModSec Audit Log e padroniza timestamps para permitir comparações consistentes entre fontes distintas. Uma interface Web apresenta filtros interativos, visualização paralela dos logs e uma linha do tempo que evidencia a sequência de eventos associados a cada endereço IP. Os resultados parciais demonstram que a ferramenta é capaz de exibir registros brutos, estruturar eventos do ModSecurity e correlacionar acessos, erros e ações do WAF, auxiliando na investigação de incidentes e na compreensão do comportamento das requisições. A aplicação mostra-se uma alternativa eficiente e de fácil implantação.*

1. Introdução

A análise de *logs* constitui um dos elementos fundamentais para o monitoramento, diagnóstico e segurança de sistemas de informação [Schmidt et al. 2012]. No contexto *Web*, esses registros são gerados principalmente por servidores HTTP, entre os quais o *Apache* se destaca por sua ampla adoção em diferentes ambientes. De acordo com a

Pesquisa de Servidores *Web* de Maio de 2025 [Netcraft 2025], o *Apache* representa aproximadamente 15% dos *sites* ativos monitorados pela Netcraft. Embora esses registros contenham informações essenciais, sua interpretação manual costuma ser trabalhosa devido ao formato heterogêneo e, em muitos casos, não estruturado. Conforme apontam [He et al. 2017], as mensagens de *log* frequentemente apresentam texto livre e alta variabilidade, o que torna métodos manuais de análise exaustivos, suscetíveis a erros e impraticáveis em sistemas modernos.

Nesse contexto, o módulo *ModSecurity*, empregado como *Web Application Firewall* (WAF), acrescenta uma camada adicional de proteção, mas também introduz maior complexidade ao processo de análise [Ristic 2010]. Seu principal registro, o *ModSec Audit Log*, é composto por múltiplas seções e blocos de informação, o que dificulta a visualização direta e a correlação com outros arquivos, como o *Access Log* e o *Error Log*. Como consequência, processos de investigação, auditoria ou identificação de ataques podem se tornar lentos, propensos a erros e dependentes de ferramentas complexas ou proprietárias.

Diante desse cenário, este artigo apresenta uma aplicação Web voltada para facilitar a visualização e a análise de *logs* do Apache com *ModSecurity*, realizando a correlação entre três fontes de dados distintas: *Access Log*, *Error Log* e *ModSec Audit Log*. A aplicação organiza os blocos de auditoria do *ModSecurity*, extrai os dados relevantes, padroniza os *timestamps* e apresenta tudo de forma integrada em uma interface *Web* intuitiva. O objetivo principal deste estudo é fornecer uma ferramenta simples, eficiente e de fácil implementação, capaz de auxiliar analistas, pesquisadores e administradores de sistemas na análise de eventos de segurança e na compreensão do comportamento de requisições HTTP filtradas pelo *ModSecurity*.

A justificativa para a realização desta pesquisa se baseia no fato de que, apesar da grande relevância dos registros gerados pelo *Apache* e pelo *ModSecurity*, ainda existe uma carência de ferramentas simples, acessíveis e integradas que permitam visualizar, correlacionar e interpretar esses dados de forma ágil. Soluções existentes, como *ELK Stack*, *Wazuh* e *Splunk*, costumam exigir infraestrutura adicional, configuração avançada e, em alguns casos, custos elevados, o que dificulta sua adoção em ambientes com recursos limitados [Manzoor et al. 2024, Tonge et al. 2025]. Dessa forma, uma aplicação leve e de fácil implementação contribui significativamente para a melhoria dos processos de investigação e resposta a incidentes, ampliando a compreensão sobre o comportamento das requisições e a eficácia das regras de segurança aplicadas.

Este artigo está estruturado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados, discutindo ferramentas e abordagens existentes para análise de *logs*. Já a Seção 3 descreve a metodologia de funcionamento da aplicação desenvolvida. Na Seção 4 são apresentados os resultados parciais obtidos ao longo da realização deste estudo. E, por fim, a Seção 5 traz as considerações finais e as sugestões de trabalhos futuros.

2. Trabalhos Relacionados

Existem diversas soluções voltadas à análise de *logs* e ao monitoramento de servidores *Web*. Entre as ferramentas mais adotadas destacam-se sistemas de *Security Information and Event Management* (SIEM), como *Splunk* e *Elastic Stack* (ELK), que oferecem recursos avançados de coleta, processamento e indexação de grandes volumes de registros,

além de recursos de visualização e correlação em tempo real [Al-Mahbashi et al. 2017]. Esses sistemas são projetados para ambientes complexos, onde múltiplas fontes de eventos precisam ser integradas e processadas em tempo real.

Também existem abordagens mais enxutas, como o *GoAccess*, que realiza análise em tempo real dos registros de acesso do servidor HTTP e disponibiliza estatísticas detalhadas por meio de interface em terminal ou painel *Web* [Orellana 2010]. Embora útil para inspeção rápida de métricas de tráfego, o *GoAccess* é voltado principalmente ao *Access Log*, não oferecendo suporte à integração com *Error Log* ou *ModSecurity Audit Log*, nem mecanismos nativos de correlação entre eventos. No contexto específico do *ModSecurity*, soluções como o *ModSecurity Console*, o *WAF-FLE* e *dashboards* construídos sobre *Kibana* ou *Grafana* permitem visualizar as diferentes seções do *audit log* e acompanhar métricas relacionadas ao WAF [Herr 2011]. Contudo, essas ferramentas geralmente operam sobre *pipelines* já estruturados de ingestão e normalização, assumindo a existência de infraestrutura prévia ou integrações com sistemas maiores de monitoramento. Além disso, tendem a concentrar-se exclusivamente nos registros do WAF, sem oferecer uma visão integrada que incorpore também acessos e erros do servidor *Apache*.

Além dessas soluções, a literatura apresenta diversos estudos voltados à correlação de *logs* com o objetivo de identificar anomalias, padrões de ataque ou comportamentos suspeitos. Tais trabalhos exploram técnicas de agregação temporal [Liu et al. 2024], mineração de padrões [Scarabeo et al. 2015] e processamento de grandes volumes de dados [Micheal 2024]. No entanto, ainda há uma lacuna no que diz respeito a soluções simples, de fácil implantação e que permitam realizar correlação temporal direta entre *Access Log*, *Error Log* e *ModSecurity Audit Log* em ambientes com restrições de infraestrutura.

Diferentemente dessas abordagens, a aplicação apresentada neste artigo foca em simplicidade e acessibilidade, evitando dependências externas e dispensando estruturas complexas de ingestão ou armazenamento. A solução proposta organiza os blocos do *ModSecurity*, facilitando a leitura e extração de informações relevantes, além de correlacionar os dados provenientes do *Access Log*, *Error Log* e *ModSec Audit Log*. Além disso, permite realiazar filtragem, proporcionando visualização integrada diretamente na interface Web, sem necessidade de banco de dados ou serviços adicionais.

3. Funcionamento da Aplicação

O fluxo de interação entre os módulos ocorre no lado do servidor: ao receber uma requisição GET ou POST, o controlador ativa o módulo de leitura, aplica o *parsing* e, caso filtros tenham sido definidos, aciona os módulos de filtragem e correlação. O resultado é então entregue à camada de apresentação, que organiza o conteúdo em três partes principais: filtros interativos, visualização paralela dos *logs* e linha do tempo de eventos correlacionados. Essa interface permite ao usuário navegar pelos registros, expandir blocos de auditoria, destacar ocorrências e comparar diferentes tipos de eventos. A Figura 1 ilustra o fluxo de processamento.

O processamento inicia com a configuração do ambiente, etapa em que são definidos os caminhos dos arquivos de *log*, parâmetros operacionais, fuso horário e variáveis internas da aplicação. Em seguida, ocorre a coleta dos registros, momento em que são verificadas permissões, existência dos arquivos e realizada a leitura otimizada das linhas mais recentes.

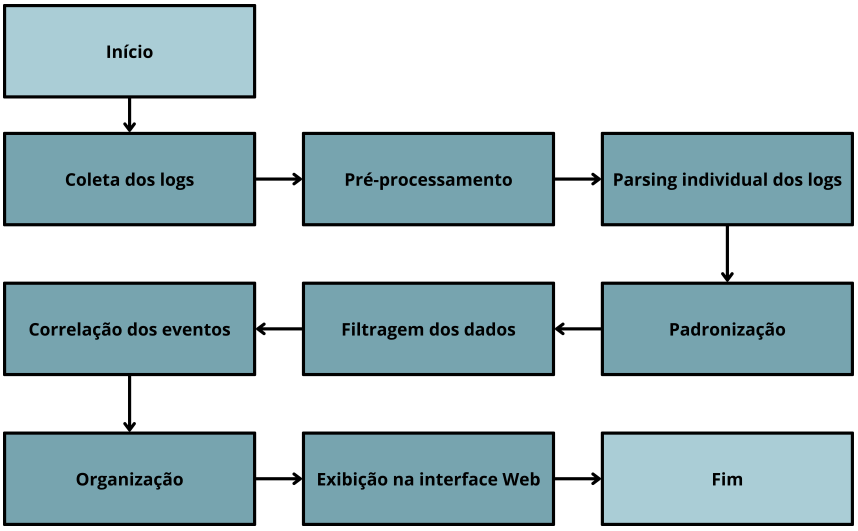

Figura 1. Fluxo de funcionamento da aplicação.

Na fase de pré-processamento, os dados brutos são organizados e validados para garantir que apenas entradas consistentes avancem para as próximas etapas. O *parsing* é então aplicado conforme o formato de cada *log*. Os arquivos do *Apache* são processados com expressões regulares (REGEX), enquanto o *log* do *ModSecurity* é segmentado em seções e reconstruído como um evento estruturado, permitindo extrair detalhes da requisição, metadados e *timestamps*.

A etapa de padronização converte campos heterogêneos para formatos uniformes, incluindo a normalização de *timestamps*, o ajuste de fuso horário e a unificação das estruturas de dados. Em seguida, a filtragem possibilita selecionar eventos por endereço IP, intervalo de tempo ou tipo de *log*. Na fase de correlação, os eventos provenientes das diferentes fontes são associados com base em características comuns e proximidade temporal, permitindo identificar sequências entre acessos, erros e ações do *ModSecurity*.

Por fim, a fase de organização ordena os eventos em ordem cronológica, agrupa-os por tipo e prepara a estrutura final para exibição. A aplicação apresenta então os resultados em uma interface Web, que oferece filtros interativos, visualização comparativa dos diferentes *logs* e acesso detalhado a cada evento correlacionado.

4. Resultados

Os resultados obtidos demonstram a capacidade da aplicação em processar, estruturar e apresentar de forma integrada os diferentes *logs* do Apache e do *ModSecurity*. Inicialmente, a ferramenta exibe os registros brutos, permitindo ao usuário verificar o conteúdo sem qualquer tratamento prévio. A Figura 2 ilustra essa visualização, na qual cada linha é apresentada tal como registrada pelo servidor. Essa visão é útil para auditorias detalhadas e para a validação do *parsing* realizado pela aplicação.

Em seguida, a aplicação disponibiliza uma visualização estruturada dos eventos do *ModSecurity*, organizada por identificador único e segmentada conforme as seções do *ModSec Audit Log*. Como mostrado na Figura 3, cada bloco contém abas colapsáveis que representam as seções. Esse formato facilita o entendimento de cada evento interceptado pelo WAF, permitindo expandir apenas as partes de interesse.

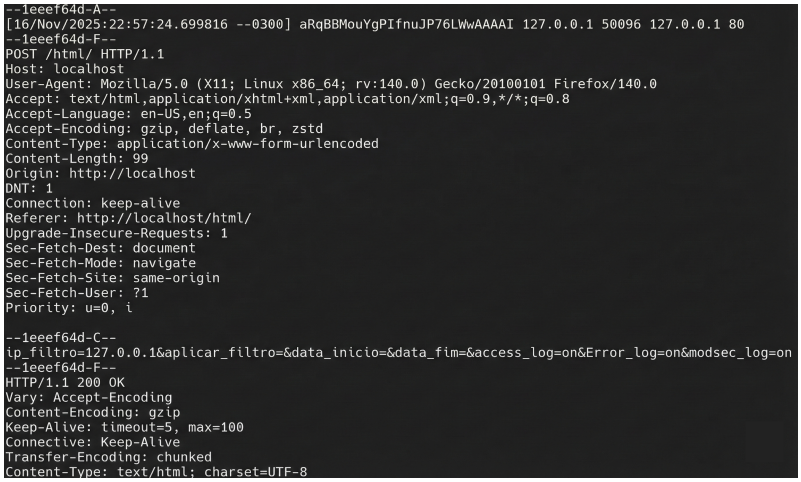

Figura 2. Log ModSec sem passar pela aplicação

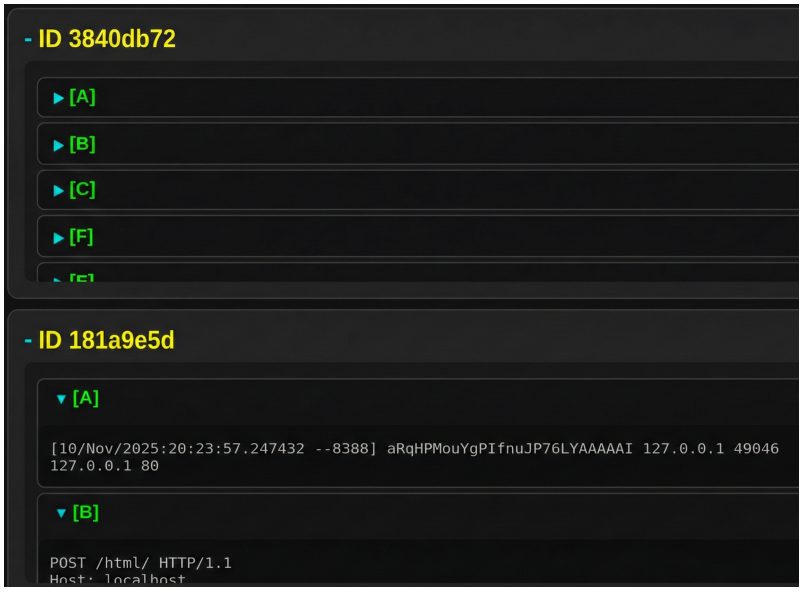

Figura 3. Resultado organização do ModSec.

Além da visualização isolada dos arquivos, um dos principais resultados intermediários está na funcionalidade de correlação temporal entre as diferentes fontes de *log*. Ao aplicar um filtro por endereço IP, a aplicação gera uma linha do tempo consolidada que agrupa eventos por minuto e destaca sua origem (*Access*, *Error* ou *ModSec*), permitindo observar a sequência completa de uma interação, desde a requisição HTTP até possíveis erros do servidor e ações tomadas pelo *ModSecurity*. Esses resultados evidenciam que a aplicação já é capaz de ler e exibir *logs* de diferentes formatos, interpretar e estruturar o *ModSec Audit Log*, correlacionar eventos de múltiplas fontes, e apresentar essas informações em uma interface Web. Tais funcionalidades evidenciam o funcionamento dos módulos de leitura, *parsing*, padronização e correlação.

5. Considerações Finais

Os resultados apresentados demonstram que a aplicação desenvolvida cumpre seu objetivo de fornecer uma ferramenta leve, integrada e de fácil utilização para análise e

correlação de *logs* do *Apache* e do *ModSecurity*. A solução foi capaz de estruturar registros brutos, organizar eventos do WAF em blocos e correlacionar diferentes fontes por meio da linha do tempo unificada, facilitando a investigação de incidentes e a compreensão do comportamento das requisições. Como trabalhos futuros, pretende-se implementar um o mecanismo de cache, incorporar processamento em tempo real por meio de *Server-Sent Events* (SSE), adicionar novos filtros, como CVEs, User-Agent e outros, além de permitir configuração dinâmica de parâmetros diretamente pela interface *Web*, incluindo limites de leitura e caminhos dos arquivos de *log*. Essas extensões visam tornar a ferramenta ainda mais flexível e adequada a cenários de monitoramento contínuo.

Agradecimentos

O presente trabalho foi realizado com apoio da Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES.

Referências

Al-Mahbashi, I. Y. M., Potdar, M. B., and Chauhan, P. (2017). Network security enhancement through effective log analysis using elk. In *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pages 566–570.

He, P., Zhu, J., Zheng, Z., and Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 33–40.

Herr, K. (2011). Web application firewall: Fast log and event console. https://github.com/klaubert/waf-fle (acesso em 12/11/2025).

Liu, Y., Ren, S., Wang, X., and Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. *Sensors*, 24(24).

Manzoor, J., Waleed, A., Jamali, A. F., and Masood, A. (2024). Cybersecurity on a budget: Evaluating security and performance of open-source siem solutions for smes. *PLOS ONE*, 19(3):1–24.

Micheal, L. (2024). Scalable data ingestion strategies: Comparative performance of kafka with spark structured streaming and flink.

Netcraft (2025). May 2025 web server survey. Disponível em: https://www.netcraft.com/blog/may-2025-web-server-survey (acesso em 10/11/2025).

Orellana, G. (2010). Goaccess - visual web log analyzer. Disponível em: https://goaccess.io/man (acesso em 12/11/2025).

Ristic, I. (2010). *ModSecurity Handbook*. Feisty Duck, London, GBR.

Scarabeo, N., Fung, B., and Khokhar, R. (2015). Mining known attack patterns from security-related events. *PeerJ Computer Science*, 1:e25.

Schmidt, K., Phillips, C., and Chuvakin, A. (2012). *Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management*. Newnes.

Tonge, A. S., Baniya, B. K., and GC, D. (2025). Efficient, scalable, and secure network monitoring platform: Self-contained solution for future smes. *Network*, 5(3).